RESTful Network API for Dynamic Navigation

Approved Version 1.0 – 17 Dec 2013

Open Mobile Alliance

OMA-TS-REST_NetAPI_DynNav-V1_0-20131217-A

Contents

Figures

1. Scope

This specification defines a RESTful API for Dynamic Navigation using HTTP protocol bindings, based on application requirements and architecture defined in [DynNav_ER].

In the document, in order to encode transportation related information, XML data structure defined in ISO TS 24530-2,3 [TTI LOC], [TTI RTM] are used, in accordance with OMA policy of reuse of existing standards.

The reproduction of examples extracted from ISO TS 24530-1,2,3 and 4 specifications, issued in 2006, has been granted by UNI 'Ente Nazionale Italiano di Unificazione' – Via Battistotti Sassi 11/B Milan (Italy) tel +3902700241 fax +390270105992 email diffusione@uni.com on behalf of ISO – International Organization for Standardization.

ISO TS 24530-2, 3 [TTI LOC] and [TTI RTM] provide TPEG XML data structures in DTD format, referenced in DynNav XML schema [REST_SUP_DYNNAV].

2. References

2.1 Normative References

[DynNav_ER] “OMA Dynamic Navigation Enabler”, Open Mobile Alliance™, OMA-ER-DynNav-V1_0, URL:http://www.openmobilealliance.org/

[IETF Draft Forte] Labels for Common Location-Based Services, URL:http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03

[REST_NetAPI_Common] “Common definitions for RESTful Network APIs”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Common-V1_0, URL:http://www.openmobilealliance.org/

[REST_SUP_DYNNAV] “XML schema for the RESTful Network API for DynNav”, Open Mobile Alliance™, OMA-SUP-XSD_rest_netapi_DynNav-V1_0, URL: http://www.openmobilealliance.org/

[RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC2616] “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding et. al, January 1999, URL:http://www.ietf.org/rfc/rfc2616.txt

[RFC3986] “Uniform Resource Identifier (URI): Generic Syntax”, R. Fielding et. al, January 2005, URL:http://www.ietf.org/rfc/rfc3986.txt

[RFC4627] “The application/json Media Type for JavaScript Object Notation (JSON)”, D. Crockford, July 2006, URL: http://www.ietf.org/rfc/rfc4627.txt

[RFC5139] Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO), M. Thomson, J. Winterbottom, February 2008, URL:http://www.ietf.org/rfc/rfc5139.txt

[SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[**SUPL 2.0]** “Secure User Plane Location”, Version 2.0, Open Mobile Alliance™, OMA-ERP-SUPL-V2_0, URL:http://www.openmobilealliance.org/

[TTI LOC] “Traffic and Travel Information (TTI)” ISO/TS 24530, Part 2: tpeg-locML, URL:http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=54706

[TTI RTM] “Traffic and Travel Information (TTI)” ISO/TS 24530, Part 3: tpeg-rtmML, URL:http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=54706

[W3C_POI] W3C Points of Interest Core, URL:http://www.w3.org/TR/2011/WD-poi-core-20110512/

[XMLSchema1] W3C Recommendation, XML Schema Part 1: Structures Second Edition, URL: http://www.w3.org/TR/xmlschema-1/

[XMLSchema2] W3C Recommendation, XML Schema Part 2: Datatypes Second Edition, URL: http://www.w3.org/TR/xmlschema-2/

2.2 Informative References

[OMADICT] “Dictionary for OMA Specifications”, Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_8, URL:http://www.openmobilealliance.org/

[REST_WP] “Guidelines for RESTful Network APIs”, Open Mobile Alliance™, OMA-WP-Guidelines_for_RESTful_Network_APIs, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

DynNav Application/ Client	An entity that is in charge of interacting with a DynNav Server to get route information and/or real-time and forecast traffic information and complimentary data. Throughout this document DynNav client and DynNav application can be used interchangeably.
DynNav Server	An entity that is in charge of providing the DynNav Application with route information or real-time and forecast traffic information and complimentary data.
Lightweight ND	A navigation device that accesses a server for route estimation functionalities and for retrieving roads shape representation, if not available in a local maps database.
Location URI	A URI that enables the current location of a device to be obtained from a particular location server using a particular dereferencing protocol.
Navigation Device (ND)	An entity that, using GNSS service, assists the driver showing correct route to reach the final destination. This entity may process real-time and predicted traffic information and dynamically estimates the optimal route, according to user preferences.
Network Performance Parameter/Performance parameters	Information regarding the performances (i.e. speed, delay and travel time) of road segments related to an area or a route. Throughout this document, network performance parameters and performance parameters can be used interchangeably.
Point Of Interest	POI describes information about locations such as name, category, unique identifier, or civic address.
Polyline	A continuous line used in graphic computing composed of one or more line segments, defined by specifying the endpoints of each segment
Route Information	Information which coordinates of segment end points and complimentary data from the defined origin and the destination
Smart ND	A navigation device that is able to calculate the route(s), using a roads network database available on the device itself.
Traffic Event	Information regarding events related to an area or a route that are either imposed or planned by the road network operator (i.e. road works leading to lane closures) or events that occur outside the control of the network operator (i.e. accidents)
Traffic Information	Information which consists of traffic events and network performance parameters related to an area or a route.

For the purpose of this TS, all definitions from the OMA Dictionary apply [OMADICT].

3.3 Abbreviations

API	Application Programming Interface
HTTP	HyperText Transfer Protocol
IMSI	International Mobile Subscriber Identity
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
JSON	JavaScript Object Notation
MDN	Mobile Directory Number

MIN	Mobile Identification Number
MSISDN	Mobile Subscriber Integrated Services Digital Network Number
NAI	Network Access Identifier
ND	Navigation Device
OMA	Open Mobile Alliance
POI	Point Of Interest
REST	REpresentational State Transfer
RTM	Road Traffic Message
SCR	Static Conformance Requirements
TPEG	Transport Protocol Expert Group
TS	Technical Specification
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	eXtensible Markup Language
XSD	XML Schema Definition

4. Introduction

The Technical Specification of the RESTful Network API for Dynamic Navigation (DynNav) contains HTTP protocol bindings for dynamic routing of vehicle based on real time and forecast traffic information, using the REST architectural style. The specification provides resource definitions, the HTTP verbs applicable for each of these resources, and the element data structures, as well as support material including flow diagrams and examples using the various supported message body formats (i.e. XML, JSON). Requirement and architecture for the DynNav enabler are defined in [DynNav_ER].

4.1 Version 1.0

Version 1.0 of this specification supports the following operations:

- Request and provide a set of routes based on the journey parameters defined by the user
- Provide traffic information related to the route and an area defined by the ND
- Provide complementary information(i.e. POI) related to defined routes and/or areas
- Manage subscriptions to notification services for receiving updates on traffic information and alternative route proposal

5. Dynamic Navigation API definition

This section is organized to support a comprehensive understanding of the Dynamic Navigation API design. It specifies the definition of all resources, definition of all data structures, and definitions of all operations permitted on the specified resources.

The DynNav API allows the user to access real time and forecast traffic information for dynamic routing of vehicles.

Common data types, naming conventions, fault definitions and namespaces are defined in [REST_NetAPI_Common].

The remainder of this document is structured as follows:

Section 5 starts with a table listing all the resources (and their URL) used by this API, along with the data structure and the supported HTTP verbs (section 5.1). What follows are the data structures (section 5.2). A sample of typical use cases is included in section 5.3, described as high level flow diagrams.

Section 6 contains detailed specification for each of the resources. Each such subsection defines the resource, the request URL variables that are common for all HTTP commands, and the supported HTTP verbs. For each supported HTTP verb, a description of the functionality is provided, along with an example of a request and an example of a response. For each unsupported HTTP verb, the returned HTTP error status is specified, as well as what should be returned in the Allow header.

All examples in section 6 use XML as the format for the message body. JSON examples are provided in Appendix D. Appendix B provides the Static Conformance Requirements (SCR).

Reserved characters in URL variables (parts of a URL denoted below by a name in curly brackets) MUST be percent-encoded according to [RFC3986]. Note that this always applies, no matter whether the URL is used as a Request URL or inside the representation of a resource (such as in “resourceURL” and “link” elements).

For requests and responses that have a body, the following applies: in the requests received, the server SHALL support JSON and XML encoding of the parameters in the body. The Server SHALL return either JSON or XML encoded parameters in the response body, according to the result of the content type negotiation as specified in [REST_NetAPI_Common]. In notifications to the Client, the server SHALL use either XML or JSON encoding, depending on which format the client has specified in the related subscription. The generation and handling of the JSON representations SHALL follow the rules for JSON encoding in HTTP Requests/Responses as specified in [REST_NetAPI_Common].

Note: Throughout this document client and application can be used interchangeably.

5.1 Resources Summary

This section summarizes all the resources used by the RESTful Network API for DynNav.

The "apiVersion" URL variable SHALL have the value “v1” to indicate that the API corresponds to this version of the specification. See [REST_NetAPI_Common] which specifies the semantics of this variable.

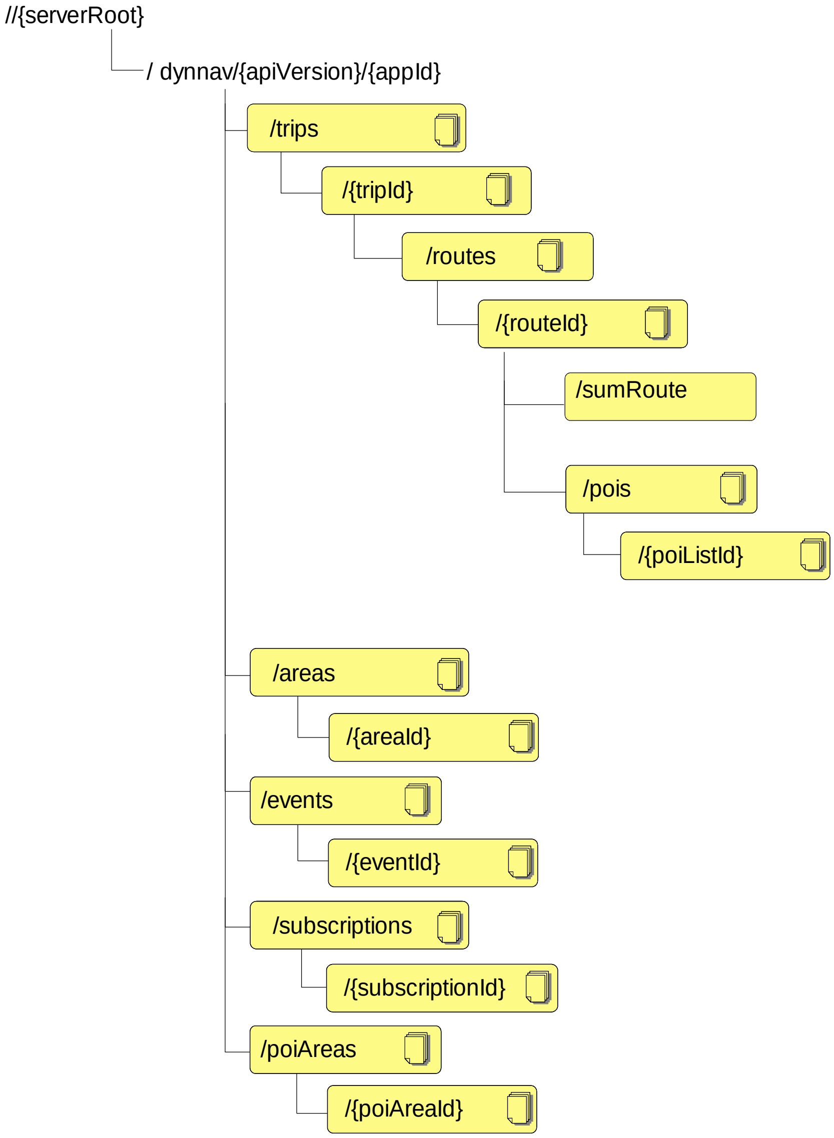

Figure 1 Resource structure defined by this specification

The following tables give a detailed overview of the resources defined in this specification, the data type of their representation and the allowed HTTP methods.

Purpose: Trips management

Resource	URL Base URL: http://{serverRoot}/ dynnav/{apiVersion}/ {appId}	Data Structures	HTTP verbs			
			GET	**PUT**	**POST**	**DELETE**
Trips created by the application	/trips	TripList (used for GET) Trip (used for POST) common:ResourceReference (optional alternative for POST response)	Read list of all trips created by the application.	No	Create new trip	No
Individual trip description	/trips/{tripId}	Trip	Read trip settings, preferences and link to the related routes	Modify parameters that describe the trip	No	Delete trip

Purpose: Management of routes defined for a trip

Resource	URL Base URL: http://{serverRoot}/dynnav/{apiVersion}/{appId}	Data Structures	HTTP verbs			
			GET	**PUT**	**POST**	**DELETE**
Routes related to a trip	/trips/{tripId}/routes	Route common:ResourceReference (optional alternative for POST response)	No **Note**: Routes Ids are available in Trip resource	No	Add a new route to the trip	No
Individual route description in full format	/trips/{tripId}/routes/{routeId}	Route	Read data about specified route	Modify a route previously uploaded	No	Delete route
Individual route description in summarized	/trips/{tripId}/ routes/{routeId}/sumRoutes	Route	Read data about specified route	No	No	No

Resource	**URL** **Base URL: http://{serverRoot}/dynnav/{apiVersion}/{appId}**	**Data Structures**	**HTTP verbs**			
			GET	**PUT**	**POST**	**DELETE**
format						

Purpose: Area management

Resource	**URL** **Base URL: http://{serverRoot}/ dynnav/{apiVersion}/ {appId}**	**Data Structures**	**HTTP verbs**			
			GET	**PUT**	**POST**	**DELETE**
Areas created by the application for traffic information	/areas	AreaList (used for GET) Area (used for POST) common:ResourceReference (optional alternative for POST response)	Read all areas created by the application	No	Create a new area	No
Individual area for traffic information	/areas/{areaId}	Area	Read area information	No	No	Delete an area

Purpose: Subscriptions management for Trip, Route, Event and Areaupdates

Resource	**URL** **Base URL: http://{serverRoot}/ dynnav/{apiVersion}/ {appId}**	**Data Structures**	**HTTP verbs**			
			GET	**PUT**	**POST**	**DELETE**
Subscriptions crated by the application	/subscriptions	SubscriptionList (used for GET) Subscription (used for POST) common:ResourceReference (optional alternative for POST response)	Read list of all subscriptions created by the application	No	Create new subscription	No

Resource	URL Base URL: http://{serverRoot}/ dynnav/{apiVersion}/ {appId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Individual subscription settings	/subscriptions/{subscriptionId}	Subscription	Read subscribed resources	Update subscription settings	No	Delete subscription

Purpose: Callback notifications for Trip, Route, Event and Areaupdates

Resource	URL <specified by the client>	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Client notification about areas and trips updates	<specified by the client when a subscription is created>	Notification	No	No	Notifies client about updates in subscribed resources (areas and trips with related routes and events).	No

Purpose: Events management

Resource	URL Base URL: http://{serverRoot}/dynnav/{apiVersion}/{appId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Events related to the application	/events	EventList	Read all available events	No	No	No
Individual event information	/events/{eventId}	Event	Read a single event	No	No	No

Purpose: POI management

Resource	URL Base URL: http://{serverRoot}/dynnav/{apiVersion}/{appId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE

Resource	URL Base URL: http://{serverRoot}/dynnav/{apiVersion}/{appId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
POIs lists related to a route	/trips/{tripId}/routes/{routeId}/pois	POIListsSet (used for GET) POIList (used for POST) common:ResourceReference (optional alternative for POST response)	Read the set of references to lists of POIs	No	Create a list of POIs	No
Individual list of POIs related to a route	/trips/{tripId}/routes/{routeId}/pois/{poiListId}	POIList	Read the set of POIs defined in the list	No	No	Delete a list of POIs
POIs lists defined for an area	/poiAreas	POIListsSet (used for GET) POIList (used for POST) common:ResourceReference (optional alternative for POST response)	Read the set of references to lists of POIs	No	Create a list of POIs	No
Individual list of POIs related to an area	/poiAreas/{poiAreaId}	POIList	Read the set of POIs defined in the list	No	No	Delete a list of POIs

5.2 Data Types

5.2.1 XML Namespaces

The XML namespace for the DynNav data types is:

urn:oma:xml:rest:netapi:dynnav:1

The 'xsd' namespace prefix is used in the present document to refer to the XML Schema data types defined in XML Schema [XMLSchema1, XMLSchema2]. The 'common' namespace prefix is used in the present document to refer to the data types defined in [REST_NetAPI_Common]. The use of namespace prefixes such as 'xsd' is not semantically significant.

The XML schema for the data structures defined in the section below is given in [REST_SUP_DYNNAV].

5.2.2 Structures

The subsections of this section define the data structures used in the DynNav API.

Some of the structures can be instantiated as so-called root elements.

5.2.2.1 Type: TripList

List of trips created by the application

Element	Type	Optional	Description
trip	Trip [0…unbounded]	Yes	It may contain an array of *Trip*
resourceURL	xsd:anyURI	Yes	Self-referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named tripList of type TripList is allowed in request and/or response bodies.

5.2.2.2 Type: Trip

Description of single trip defined by the application for which route information and/or traffic information is provided

Element	Type	Optional	Description
originWGS84	Location_Point	Choice	This field represents the origin of the trip for which route information and related traffic information are requested from the server. *Location_Point* structure is defined in tpeg-locML [TTI LOC]. One element either *originWGS84* or *originAddress* MUST be specified when *Trip* resource is created. This element is mandatory when the *Trip* resource is read by the client. This field can be used to indicate the assumed current position of the client, enabling route information updating procedure on the server.
originAddress	Civic_Address	Choice	This field represents the origin of the *Trip* and it is present when the origin is expressed according to IETF Civic Address [RFC5139]. One element either *originWGS84* or *originAddress* MUST be specified when *Trip* resource is created.
destinationWGS84	Location_Point	Choice	This field represents the destination of the trip for which route information and related traffic information are requested from the server. *Location Point* structure is defined in tpeg-locML [TTI LOC]. One element among *destinationWGS84*, *destinationAddress* or *destination3rdParty* MUST be specified when *Trip* resource is created. This structure is mandatory when the *Trip* resource is read by the client.

destinationAddress	Civic_Address	Choice	This field represents the destination of the trip and it is present when the destination is expressed according to IETF Civic Address [RFC5139]. One element among *destinationWGS84 destinationAddress* or *destionation3rdParty* MUST be specified when *Trip* resource is created. This structure may be provided by the server in case the user define a destination using *destinationWGS84* or *destionation3rdParty* structures.
destination3rdParty	xsd:string	Choice	3rd party ID is used for retrieving the 3rd party's position and the position is used as a destination of the trip. One element among *destinationWGS84*, *destinationAddress*, or *destionation3rdParty* MUST be specified when *Trip* resource is created.
thirdPartyIDType	ThirdPartyIDTypeList	Yes	Indicate which type of the 3rd party ID is used in the *destination3rdParty* element. If *destination3rdParty* is present, *thirdPartyIDType* MUST be present.
waypoints	Location_Point [0…unbounded]	Yes	The waypoints may be used to provide additional information about the trip. *Location_Point* structure is defined in tpeg-locML [TTI LOC].
startingTime	xsd:dateTime	Yes	Starting time of the planned trip. If not present, current time is used.
tollRoad	xsd:boolean	Yes	This field carries the information whether toll roads MAY be included in route estimation If true or not present, toll road are allowed.
vehicleType	xsd:string	Yes	This field describes the type of vehicle for which route information is requested. This field SHALL be encoded according to the list of values defined in table RTM01 provided in [TTI RTM]
calculateRoute	xsd:boolean	Yes	If present and set to True, the server MUST propose, for the defined Trip, a set of routes with related traffic events and performance parameters, and/or alternative routes in case of congestion. If set to False or absent, no action is required from the server.
requestedEventsCategories	xsd:string [0..unbounded]	yes	Categories of traffic information, related to the defined Trip, requested by the application. This field shall be encoded according to the list of values defined in the rtm00 table available in [TTI RTM]. If this field is not present, the server MUST provide traffic information for all defined categories (including network performance parameters).

link	common:Link [0..unbounded]	Yes	Links to routes related to the trip. Attribute “rel” must be set to “Route”.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named trip of type Trip is allowed in request and/or response bodies.

5.2.2.3 Type: Route

The route information structure describes a path that matches with trip parameters

Element	Type	Optional	Description
travellingTime	xsd:float	Yes	Total travelling time (in minutes) for the route.
distance	xsd:float	Yes	Total distance (in Km) of the route.
origin	Location_Point	No	This field represent the origin of the route expressed in WGS84 coordinates. *Location_Point* structure is defined in tpeg-locML [TTI LOC].
partialRouteInformation	xsd:boolean	Yes	If set to true, the *Route* is described with partial information: only changed segments sequence is provided with respect to a reference route. The reference route is defined in *link* field of this structure. The partial route encoding schema is described in Appendix H. The partial encoding schema MAY be used for full routes resources. If this field is absent or set to false, the route information is complete.
firstSegment	xsd:integer [0…unbounded]	Yes	This field represents one or more index of the first segment in the reference route segments sequence to be replaced by partial route segments sequence. In a partial route, a sequence of deviations MAY be provided with respect to the reference route: for each deviation it is provided the index of the first segment in the reference route that has to be replaced by partial route segments sequence. This field is present only in case of partial route encoding schema (*partialRouteInformation* set to True) (see Appendix H).
lastSegment	xsd:integer [0…unbounded]	Yes	This field represents one or more index of the last segment in the reference route segments sequence to be replaced by the segments sequence of partial route. Only used for the partial route case (see

			Appendix H). In a partial route, a sequence of deviations MAY be provided with respect to the reference route: for each deviation it is provided the index of the last segment in the reference route that has to be replaced by partial route segments sequence. This field is present only in case of partial route encoding schema (*partialRouteInformation* set to True and for more detail see Appendix H).
numSegments	xsd:integer [0…unbounded]	Yes	This field represents the number of segments that constitutes each single deviation of the partial route. Only used for the partial route information case (see Appendix H). In a partial route, a sequence of deviations MAY be provided with respect to the reference route: for each single deviation the number of describing segments is provided. The sum of the number of segment of each deviation should be equal to the number of segments provided in the partial route. This field is present only in case of partial route encoding schema (*partialRouteInformation* set to True and for more detail see Appendix H).
segment	Segment [1…unbounded]	No	Sequence of road segments that forms the route. In case of in partial route description, only the segment sequences describing the deviations are provided (see Appendix H). In case of partial route with multiple deviations, each single deviation is identified by the length of each sequence reported *in numSegment* fields of this structure.
trafficEvents	CategorizedEventListReference [0..unbounded]	Yes	List of traffic events related to the route, as defined in tpeg-rtmML [TTI RTM]. The events are grouped by the categories, defined in RTM00 table provided in [TTI RTM].
link	common:Link [0…2]	Yes	Link to reference route resource. There are two different kinds of reference route resources. 1) Reference to the route for which it is proposed as alternative. Attribute “rel” must be set to “*Route*”. 2) Reference to the route for which the partial route information is referred. Attribute “rel” must be set to “*ReferenceRoute*”.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included

			in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named route of type Route is allowed in request and/or response bodies.

5.2.2.4 Type: AreaList

Contains an array of links to all areas defined by the application

Element	Type	Optional	Description
area	Area [0…unbounded]	Yes	It may contain an array of Area structure used to access traffic events and network performance parameters.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named areaList of type AreaList is allowed in request and/or response bodies.

5.2.2.5 Type: Area

Description of a single area

Element	Type	Optional	Description
areaDesc	Location Container [1… 2]	No	It describes the area for which traffic information, traffic events and network performance parameters, are requested. It is encoded according to *Location Container* structure as defined in tpeg-locML [TTI LOC]. If the *tripAreaDesc* field is set to TRUE, this field MUST contain origin and destination points each of them encoded as *Location Coordinates* structures in two different *Location Container* structures [TTI LOC]**note**: the *Area_tree_entity* defined in the human readable area description of LocML [TTI LOC chap. 5.3.1.1] is not used in DynNav application and parameters of *Area_tree_entity* structure have no meaning.
tripAreaDesc	xsd:Boolean	Yes	If present and set to true, the server should provide traffic information related to trip information (origin and destination) available in *areaDesc* field. If set to true, *areaDesc* field should describe a Trip in terms of 2 points encoded as two *Location Container* structures each of them containing a *Location*

			Coordinates [TTI LOC], the first is the origin and the second is the destination. This element is only used by the application of smart ND scenario, see paragraph 5.3.2.
startValidityTime	xsd:dateTime	No	Starting time of the interval for which events and network performance parameters are requested.
endValidityTime	xsd:dateTime	No	Ending time of the interval for which events and network performance parameters are requested.
requestedEventsCategories	xsd:string [0..unbounded]	Yes	Categories of traffic information requested by the application in the defined Area. This field shall be encoded according to the list of values defined in the rtm00 table available in [TTI RTM]. If this field is not present, the server MUST provide traffic events of all categories (including network performance parameters)
timeResolution	xsd:float	Yes	The resolution in time domain of requested/provided network performance parameters in minutes. This element is present only in case network performance parameters are requested and provided from/by the server.
roadLinkType	xsd:string [0..unbounded]	Yes	List of types of road where traffic information is requested. The categories for roads are defined in table loc09 in tpeg-locML [TTI LOC].
events	CategorizedEventListReference [0..unbounded]	Yes	List of events related to the defined area. The information provided relates to the road network and associated infrastructure.
segmentPerformance	Segment [0..unbounded]	Yes	This field provides real-time and forecast network performance parameters for the list of road segments pertaining to the selected area.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named area of type Area is allowed in request and/or response bodies.

5.2.2.6 Type: Segment

Description of single segment that comprises the route

Element	Type	Optional	Description
originPoint	Location_Point	Yes	This field represents the origin of the segment

			encoded according to *Location_Point* structure as defined in tpeg-locML [TTI LOC]. In case *segment* structure is used for describing a route and this field is not present, the starting point of the segment should be assumed equal to the ending point of the previous segment, or the trip origin in case of the first segment of the route. In case of partial route, the origin of the first segment of each deviation is the ending point of the last valid segment in reference route.
endpoint	Location_Point	No	This field represents the end of the segment encoded according to *Location_Point* structure as defined in tpeg-locML [TTI LOC].
midwayPoint	Location_Point [0…unbounded]	Yes	This field is used to identify unambiguously the target road segment. It is encoded according Location_Point structure as defined in tpeg-locML [TTI LOC].
polyLine	xsd:string	Yes	Polyline is used to describe the shape of a segment. This field is a string that contains a sequence of geographic points expressed in WGS84 coordinates. Each single point is encoded as a sequence of • WGS84 Latitude, • Blank (character), • WGS84 Longitude, • Colon (character), • Blank (character). The shape of segments is provided by the server if explicitly requested by the application. The level of polyline resolution is defined by the DynNav Server. When used in *full route* resource, the polyline resolution has to target a correct representation of segments on turn-by-turn navigation maps. In *summarized route* resource the resolution has to target the high level representation of the route on top of roads maps. Polyline example: 45.12345 7.009876, 45.12355 7.09866, …
linkName	xsd:string	Yes	Name of the road or street, which the segment belongs to.
distance	xsd:float	Yes	Length of the segment in km.
regularTravellingTime	xsd:float	Yes	Estimated regular time to drive through the segment in low traffic conditions, expressed in minutes.

performanceParameters	PerformanceParameters [0..unbounded]	Yes	This field contains performance parameters related to each segment. When *segment* structure is used to report network performance parameters for an area, a sequence of *performanceParameters* structure is included in the *segment* structure, providing information for the requested time interval and time resolution.
positionUpdate	xsd:boolean	Yes	If present and set to True, the application is requested to upload its current position when the Navigation Device enters this segment.

5.2.2.7 Type: PerformanceParameter

This structure contains information about network performance parameter for a single road segment

Element	Type	Optional	Description
trafficInfoType	TrafficInfoType	Yes	This element is used to define whether the data is estimated in real-time or it is forecast. Possible values are: (see 5.2.2.2) • real-time • forecast
time	xsd:dateTime	Yes	This field indicates the starting time of validity interval for reported performance parameters.
delay	xsd:float	Yes	Estimated delay (real-time or forecast) along the segment expressed in minutes with respect to regular travelling time. **Note**: regular travelling time for the segment is available in *regularTravellingTime* parameter of *segment* structure.
speed	xsd:float	Yes	Estimated speed (real-time measurements or forecast) along the segment expressed in m/s.
performance	xsd:string	Yes	Description of traffic conditions (real-time or forecasted) along the segment. This field should be encoded according to RTM34 table definition [TTI RTM].

5.2.2.8 Type: SubscriptionList

List of subscriptions

Element	Type	Optional	Description
subscription	Subscription [0…unbounded]	Yes	It may contain an array of Subscription.

resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named subscriptionList of type SubscriptionList is allowed in request and/or response bodies.

5.2.2.9 Type: Subscription

Individual subscription to notifications

Element	Type	Optional	Description
callbackReference	common:CallbackReference	No	Client's Notification endpoint and parameters.
link	common:Link [1…unbounded]	No	References to resources subscribed by the application. Attribute “rel” indicates the type of resource subscribed. It may assume the following values: • “Trip”: in order to get notified about: o new traffic events and performance parameter related to the set of routes defined for the trip o new alternative route proposals • “Area”: in order to be notified of new traffic events and performance parameters updates Attribute “href” specifies the URL of subscribed resource. Subscribed resource’s type must be the same of that specified in “rel” attribute. **Note**: notified information for an existing route are: a) new traffic events provided with links included in the *route* resource itself; b) performance parameters available in updated *performanceParameter* filed of *segment* structures.
trackingProc	xsd:boolean	Yes	If present and set to True, the application communicate to the server user’s availability to provide position information through an external location application.
deviceLocationURI	xsd:anyURI	Yes	This parameter is used by the server for accessing Navigation Device position information.
tracking3rdParty	xsd:boolean	Yes	If present and set to True, the DynNav server tracks the 3rd party position and notifies the availability of

			updated information when the 3rd party position is changed.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named subscription of type Subscription is allowed in request and/or response bodies.

5.2.2.10 Type: Notification

Notification about updates in subscribed routes, areas and trips

Element	Type	Optional	Description
link	common:Link [1…unbounded]	No	Link to updated resources. Attribute “rel” attribute indicates type of resource updated and may assume “Trip”, “Route”, “Event” and “Area” values.

A root element named notification of type Notification is allowed in request and/or response bodies.

5.2.2.11 Type: EventList

Contains a list of all events available

Element	Type	Optional	Description
event	Event [0…unbounded]	Yes	Contains a list of events. Event information is defined in tpeg-rtmML [TTI RTM].
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named eventList of type EventList is allowed in request and/or response bodies.

5.2.2.12 Type: CategorizedEventList

Contains an array of links for a specific category

Element	Type	Optional	Description
category	xsd:string	No	This field shall be encoded according to the list of values defined in the rtm00 table available in tpeg-

			rtmML definition [TTI RTM].
link	common:Link [1..unbounded]	No	Contains a list of references to events belonging to the defined category. Attribute “rel” must be set to “Event”.

5.2.2.13 Type: Event

Description of single traffic event and all the possible traffic events are described in tpeg-rtmML [TTI RTM]

Element	Type	Optional	Description
rtMessage	Road_Traffic_Message	No	This field includes one or more traffic events. Event information is defined in tpeg-rtmML [TTI RTM].
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named event of type Event is allowed in request and/or response bodies.

5.2.2.14 Type: POIListsSet

Description of single POIs list and each list includes the set of POI proposed for defined parameters

Element	Type	Optional	Description
link	common:Link [0..unbounded]	Yes	It includes one or more link to a POIs list. Attribute “rel” must be set to “POIList”.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named poiListsSet of type POIListsSet is allowed in request and/or response bodies.

5.2.2.15 Type: POIList

Contains a list of Points of interest as defined by W3C POI

Element	Type	Optional	Description
category	xsd:string [1..unbounded]	No	List of categories for which POIs are requested as defined in [IETF Draft Forte].

routeRange	xsd:integer	Yes	POIs maximum distance from the route. This field is present if the POIs list is related to a route.
area	Location_Container	Yes	This field describes the area where POIs are requested. Location Container structure is defined in tpeg-locML [TTI LOC]. This field is present if the POIs list is related to an area. **Note**: the Area_tree_entity defined in the human readable area description of LocML [TTI LOC chap. 5.3.1.1] is not used in DynNav application and parameters of Area_tree_entity structure have no meaning.
pois	Pois	Yes	Pois structure is defined in [W3C_POI].
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named poiList of type POIList is allowed in request and/or response bodies.

5.2.3 Enumerations

The subsections of this section define the enumerations used in the DynNav API.

5.2.3.1 Enumeration: ThirdPartyIDTypeList

Type of 3rd Party ID enumearation. It is used to describes what kind of identifier is used. The format of each type of identifier is defined in [SUPL 2.0] section 11.7 (one string character is used for encoding each digit of the identifiers).

Enumeration	Description
MSISDN	The type of the 3rd party ID is MSISDN
MDN	The type of the 3rd party ID is MDN
IMSI	The type of the 3rd party ID is IMSI
NAI	The type of the 3rd party ID is NAI
IPv4	The type of the 3rd party ID is IPv4
IPv6	The type of the 3rd party ID is IPv6

5.2.3.2 Enumeration: TrafficInfoType

Traffic Information Type enumeration. It is use to describes how performance parameters are estimated.

Enumeration	Description

Real-time	Network performance parameters are estimated by real time traffic monitoring.
Forecast	Network performance parameters are estimated based on historical traffic data and/or planned actions on road infrastructure.

5.2.4 Values of the Link “rel” attribute

The “rel” attribute of the Link element is a free string set by the server implementation, to indicate a relationship between the current resource and an external resource. The following are possible strings (list is non-exhaustive, and can be extended):

- Trip
- Route
- ReferenceRoute
- Event
- Area
- POIList

These values indicate the kind of resource that the link points to.

5.3 Sequence Diagrams

The following sub-sections describe the resources, methods and steps involved in typical scenarios for DynNav.

5.3.1 Request of Route Information and Related Traffic Information by the Application in a Lightweight ND

This section describes a typical scenario of DynNav application where lightweight ND requests route and traffic information from the DynNav server. The main functionalities defined for this scenario are: (1) the delivery of route information in summarized format and/or full format, (2) the subscription to notification services, (3) current position reporting by the application, (4) the re-routing in case of: (a) congestion along the proposed route, (b) deviation and diversion from the route in use, (c) the destination is the 3rd party’s position and the 3rd party’s position is changed.

In this scenario the user of DynNav application defines the journey in terms of origin, destination and other preferences; these parameters are immediately sent by the ND to the DynNav server. The destination may be defined using a 3rd Party ID; in this case, the DynNav server acquires the 3rd party’s position through an external location application and considers the 3rd party’s position as the destination. The DynNav server will reply with a set of routes matching with journey parameters taking into account real-time and forecast traffic information. For bandwidth optimization, the routes are available in the DynNav server in two different formats, summarized and full. The application accesses the proposed routes in summarized format: with this information the user can select a route out of the proposed set to be used for navigation. The application requests the full format for the selected route and it may delete the routes not used. Due to limited length, complexity of the journey and network capabilities, the proposed routes may be encoded right from the beginning in full format; in this case the DynNav server does not need to encode the routes in summarized format. The application may request from the server the information about the segments shape of routes (WGS84 coordinates polyline), if this data is not available on the ND in a roads database.

The application subscribes to notification services for receiving traffic information updates (performance parameters and traffic events for selected categories) for the route in use, alternative route proposals in case of congestion along that route, and for receiving updated information (updated destination, additional segment or alternative route) in case the 3rd party ID is used as a destination and the 3rd party’s position is changed. The application will update its current position on the DynNav server after the vehicle drives a certain distance. With this information, the server will delete segments already travelled from the route in use and remove the routes not compatible anymore with current position (if not previously deleted by the application).

Afterwards, the user deviates and diverts from the route in use. Under these conditions, the application uploads its updated current position, and the DynNav server recognizes that the current position is not compatible with the route in use and proceeds to new route estimation, based on updated position information; the new route identifier is sent to the application in the current position update procedure (the notification procedure for the new route is therefore not needed). To minimize the interaction with the user for safety reason, the notification service will be automatically extended to the new proposed route(s).

Later, due to a traffic jam on the selected route, the DynNav server notifies the application of updated traffic information for the route in use and a proposal of an alternative route and the application accesses the notified resources. The DynNav server will automatically provide notification service for the new proposed route if not deleted.

In case the 3rd party ID is used as a destination, the destination may be changed because the 3rd party moves. In this case, the DynNav server notifies the application of the updated information. The types of updated information are: (1) updated destination, (2) additional segment, (3) alternative route (see Appendix I). In order to decrease the number of interactions, the DynNav server may notify the updated information only when the user of application is in the vicinity of the destination according to the type of the updated information.

The sequence describes the following operation on the resources:

- To define and modify the parameters of a trip, create and modify resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips
- To access the identifiers of the proposed routes related to the defined trip, read resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}
- To access information related to summarized route, read resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/routes/{routeId}/sumRoute
- To access information related to one or more full routes, read resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/routes/{routeId}
- To access traffic events related to the route, read resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/events/{eventId}
- To remove unnecessary routes, delete resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/routes/{routeId}
- To subscribe to notification service for a trip and related routes, create resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/subscriptions
  (The server will send notifications to the URL specified in the subscription resource; the notification will contain the URLs of the updated resources)
- To send notification to the application with the identifiers for the updated resources, create resource under the resource defined by the application
  (This resource is provided by the client)

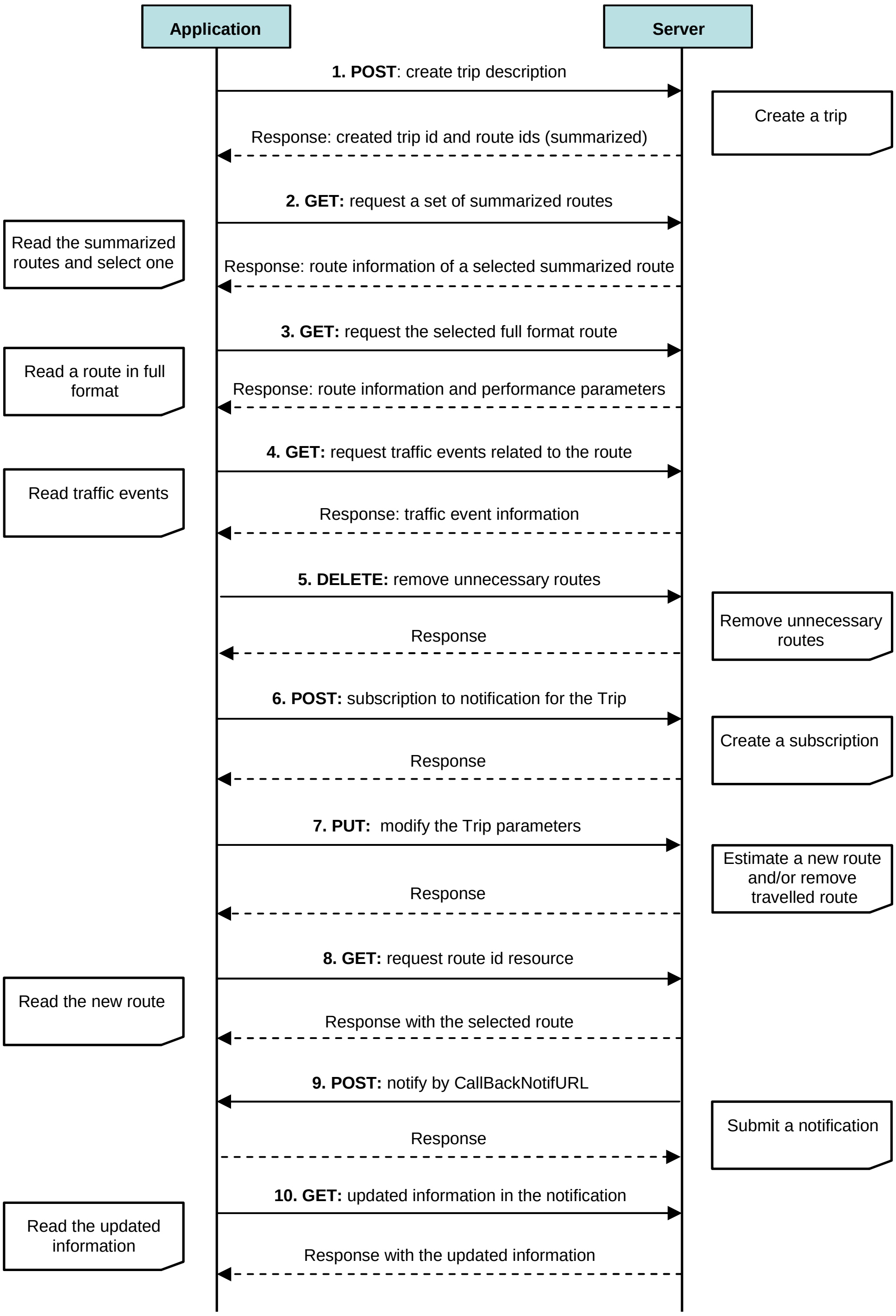
Application
Server
1. POST: create trip description
Create a trip
Response: created trip id and route ids (summarized)
2. GET: request a set of summarized routes
Read the summarized routes and select one
Response: route information of a selected summarized route
3. GET: request the selected full format route
Read a route in full format
Response: route information and performance parameters
4. GET: request traffic events related to the route
Read traffic events
Response: traffic event information
5. DELETE: remove unnecessary routes
Remove unnecessary routes
Response
6. POST: subscription to notification for the Trip
Create a subscription
Response
7. PUT: modify the Trip parameters
Estimate a new route and/or remove travelled route
Response
8. GET: request route id resource
Read the new route
Response with the selected route
9. POST: notify by CallBackNotifURL
Submit a notification
Response
10. GET: updated information in the notification
Read the updated information
Response with the updated information

Figure 2 Sequence for Lightweight ND

Outline of the flows:

1. The application creates a trip using the journey parameters defined by the user using POST: the server proposes a set of routes for the journey with related traffic information and replies with a representation of created “trip” resource, which contains route identifiers of the proposed routes.
2. The application accesses the set of routes in summarized format using GET. This step is repeated for all the routes proposed by the server. If, however, the length and complexity of the trip is limited and the network quality is adequate, full format route information can be used at this stage. The application may request shape information (WGS84 coordinates polyline) for the proposed routes, if this information is not available in the ND.
3. The user of the application selects one route among the proposed set, the application accesses full format information for the route the user has selected, using GET. The application may request shape information (WGS84 coordinates polyline) for the proposed route, if this information is not available in the ND. If, in the step 2, the full format route has been retrieved, this step is not required. The server replies with the selected route information with related traffic information.
4. The application accesses traffic events related to the route in use, using links to traffic events resources provided in route representation, using GET. The access to the traffic events may be limited to the categories selected by the user.
5. The application removes unnecessary routes previously proposed by the server and not selected by the user, using DELETE.
6. The application creates a subscription to notification services for the trip using POST. The client is notified by the server of the following events:
   a. Performance parameters update and new traffic events (for selected categories) for all the routes related to the trip.
   b. Alternative proposed routes in case of congestion on the route in use.
   c. Updated destination and/or route to the 3rd party, in case the trip destination is the 3rd party’s position and the 3rd party’s position is changed, To enable the notification of this information, the application has to request the tracking procedure of the 3rd party’s position by the DynNav server, in the subscription to notifications.
7. The vehicle deviates and diverts from the route in use; the application modifies origin parameter in Trip resource with PUT operation. The server recognizes that the current position does not belong to the route in use and it calculates a new route with the new origin. The server replies to the PUT operation with the identifier of the new route included in the Trip representation, and it removes the old one. In case the modified origin parameter used in the PUT operation belongs to the route, the DynNav server uses this information to delete segments already travelled from the route representation.

   Note: This step (PUT operation on Trip resource) occurs when the vehicle deviates and diverts and when the vehicle drives a certain distance from the previous reporting position, and/or when the vehicle enters a segment where the DynNav server has requested to upload the current position.
8. The application accesses the new proposed route with performance parameters and traffic events using GET operation. Since the application has subscribed to notification service for the Trip resource, the subscription will cover the new proposed route.
9. Traffic events, severe congestion along the proposed routes and/or change of 3rd party’s position are detected by the server, the server notifies using POST the URL of updated information.
10. The application accesses the updated information for the route in use, new related traffic events and the proposed alternative route using GET, as the subscription to notification service include all the routes related to the trip, notification will be extended to the proposed alternative route. In case the 3rd party’s position is changed, the application accesses the changed 3rd party’s position information as the destination in the Trip resource and/or to the updated route resource using GET.

5.3.2 Request of Traffic Information Related to Routes Estimated by the Application in Smart ND

This section describes a typical scenario of DynNav application where a smart ND, with route estimation functionalities, requests traffic information related to one or more estimated routes from the DynNav server. The main functionalities defined for this scenario are: (1) preliminary access to traffic information related to selected areas, (2) access to performance parameters for a set of routes estimated by smart ND for the defined trip and (3) the subscription to notification services for real time traffic information updates, (4) current position reporting by the application, and (5) access traffic information for routes described with partial information, in case of re-routing by the smart ND.

In this scenario the user of the DynNav application defines the journey parameters (e.g. origin, destination, and road preferences), these parameters are uploaded on the DynNav server by the application; the smart ND estimates one or more geographical areas related to the defined journey and it accesses traffic information (events and performances parameters) reported by the DynNav server for the selected areas; Using this traffic information, the ND can propose to the user a set of routes for the defined journey, trying to avoid congested road segments; the user selects a reference route. The application uploads the selected route on the DynNav server accessing related traffic information (real-time and forecast performance parameters). Furthermore, for real time optimal route estimation, the application subscribes to notification services for the trip, in order to receive updated traffic information related to the route in use (performance parameters and traffic events for selected categories).

At a given moment, an accident and/or severe congestion may occur along the current route: a notification message is triggered by the DynNav server toward the application. The application accesses updated traffic information available for the route: as a consequence of degraded performances, the ND estimates an alternative route and requests related traffic information from the DynNav server. If the new route is less congested than the previous one, the old one is then removed by the ND, since the ND is no longer interested in the notification service for this resource. In case the performances of the proposed alternative route are poor, before removing the previous one, the ND may look for a less congested one. The ND can repeatedly estimate a set of alternative routes uploading them on the server. The application may choose to upload partial route information for bandwidth optimization (see Appendix H).

The application periodically reports its current position to the DynNav server, based on travelled distance: with updated position information the server can remove the segments already travelled by the vehicle from the route representation. In a later stage the vehicle diverts from the planned route, the ND estimates a new route that is uploaded on the server to access related traffic information: The new route replaces the previous one and the notification service will cover the new resource.

The sequence describes the following operation on the resources:

- To define and modify the parameters of a trip, create and modify resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips
- To define areas related to the trip, create resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/areas
- To access traffic events related to the area, read resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/events/{eventId}
- To access traffic information related to a route, create or modify a full format route under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/routes
- To subscribe to notification service for an area and/or trip with the related route, create resource under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/subscriptions
  (the server will send notifications to the URL specified in the subscription resource; the notification will contain the URLs of the updated resources)
- To remove an old route, delete a route under
  http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/routes
- To send notification to the application with the identifiers for the updated resources, create resource under the resource defined by the application
  (This resource is provided by the client)

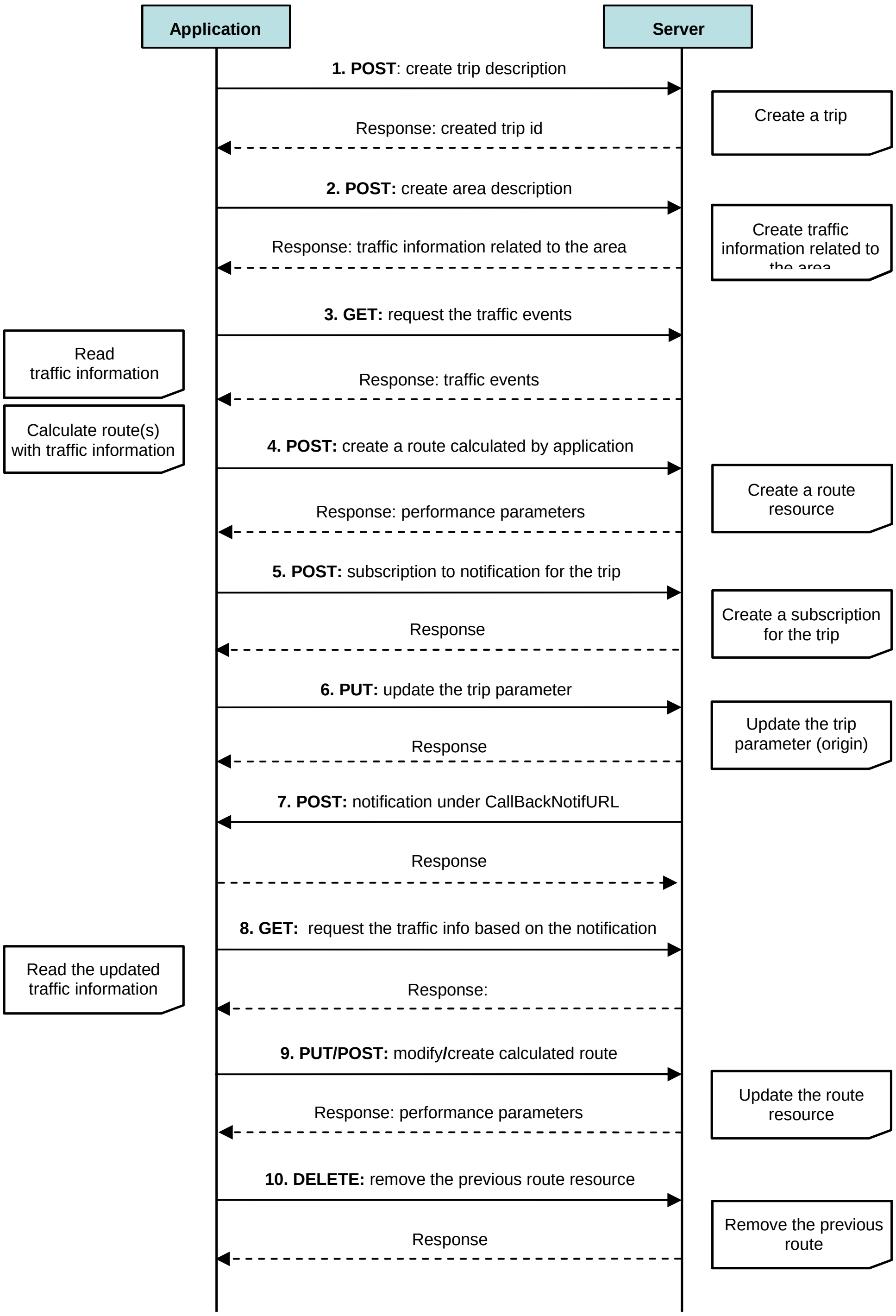
Application
Server
1. POST: create trip description
Create a trip
Response: created trip id
2. POST: create area description
Create traffic information related to the area
Response: traffic information related to the area
3. GET: request the traffic events
Read traffic information
Response: traffic events
Calculate route(s) with traffic information
4. POST: create a route calculated by application
Create a route resource
Response: performance parameters
5. POST: subscription to notification for the trip
Create a subscription for the trip
Response
6. PUT: update the trip parameter
Update the trip parameter (origin)
Response
7. POST: notification under CallBackNotifURL
Response
8. GET: request the traffic info based on the notification
Read the updated traffic information
Response:
9. PUT/POST: modify/create calculated route
Update the route resource
Response: performance parameters
10. DELETE: remove the previous route resource
Remove the previous route
Response

Figure 3 Sequence for Smart ND

Outline of the flows:

1. The application creates a Trip with the journey parameters defined by the user using POST and it receives from the server a representation of created "trip" resource, with trip identifier and defined parameters. The application specifies that routes estimation functionalities are not requested.

2. The application creates an Area description using POST to request traffic information related to the trip (in this case the Area structure will be identified with origin and destination coordinates). The server may reply with traffic information including selected categories of traffic events for the area related to the described trip, and performance parameters for the area around the origin only in case there is severe congestion.
3. The application reads the reported link(s) to traffic information using GET. This information, together with performance parameters retrieved in step 2, is used by the ND to calculate a set of routes avoiding critical road segments (affected by accidents, construction, or congestions).
4. The application uploads an estimated route (selected by the user among a set proposed by ND) on the server using POST. The server replies with a representation of the 'route' resource, which contains performance parameters and link(s) to traffic events.
5. The application subscribes to the notification service for the selected area (step 2) and for the uploaded route (step 4) using POST. The application will be notified of performance parameters and traffic events related to selected area and to the routes uploaded for the trip.
6. The application periodically updates its current position using PUT to modify the origin parameter of Trip resource. This operation is triggered when the vehicle drives a certain distance from the previous reporting position; the DynNav server utilizes this information to delete the segments already travelled from the route information.
7. When traffic events and/or severe congestion along the proposed routes are detected by the server, the server notifies the application. The server provides updated traffic information on the current route, using POST on the link specified by the application.
8. The application accesses the updated traffic information (selected traffic events and performance parameters) related to the route using GET.
9. The ND decides to re-calculate a new route under the conditions:

   a) The application receives the updated traffic information in the step 8.

   b) The ND detects that the vehicle is deviating and diverting from the defined route.

   The application uploads the new calculated route to the server with modify or create operation using PUT on a an existing route or POST on route factory resource, depending on whether or not the application wishes to keep valid the previous route. The server replies with a representation of the "route" resource which contains performance parameters.

   This step may be repeated several times until the performance of the re-calculated route is better than the previous routes. However, in order to avoid going into a loop, the application can define a new area description to acquire traffic information in the area where the repeated query occurs with operations similar to those described in the step 2 and 3.

   Note: for bandwidth optimization, the application can choose to use partial route schema (see Appendix H), uploading only the changed segments with respect to already defined reference route.

10. The application deletes the previous routes from the set of proposed routes when the previous routes are no longer in use. The application deletes the new calculated route from the set of proposed routes when the performance of the new route is worse than the route in use. The application unsubscribes the previous routes from notification service using DELETE. (If the new route has replaced the old one, with a modify operation, at the step 9, the DELETE operation is not needed).

    Note: If the delete operation is executed on a route that is referenced in resources described with partial route information, the server has to keep the resources description consistent (i.e. complete route description should be provided for route previously encoded as partial).

5.3.3 Request of Traffic Information and Points Of Interest for a Defined Area by Application in Smart ND

The figure below shows a scenario for the application in smart ND that calculates the routes and interacts with the DynNav server to retrieve information such as traffic information and point of interests. In this scenario the application requests traffic information (performance parameters and events for selected categories) related to an area from the DynNav server in order to estimate a route for given origin and destination. No further interactions with the DynNav server will be required, as the user does not want to subscribe to real time traffic updates.

Further down the journey, while still travelling, the user can see what it appears a big historical site. It then sends a new request to the DynNav server asking for information about this POI; the DynNav server will reply with a message that contains the name of the site (it is an ancient roman ruins park) and a pointer to a web site containing more specific information (e.g., entrance hours, cost, etc.).

The resources:

- To define a new area for which traffic events are requested, create resource under **http://{serverRoot}dynnav/{apiVersion}//{appId}/areas**
- To read parameters and events related to a previously defined area, read resource under **http://{serverRoot}/dynnav/{apiVersion}/{appId}/areas/{areaId}**
- To access a specific traffic event related to the area, read resource under **http://{serverRoot}/dynnav/{apiVersion}/{appId}/events/{eventId}**
- To define a set of points of interest in an area, create resource under **http://{serverRoot}/dynnav/{apiVersion}/{appId}/poiAreas**
- To read a list of points of interest in an area, read resource under **http://{serverRoot}/dynnav/{apiVersion}/{appId}/poiAreas/{poiAreaId}**

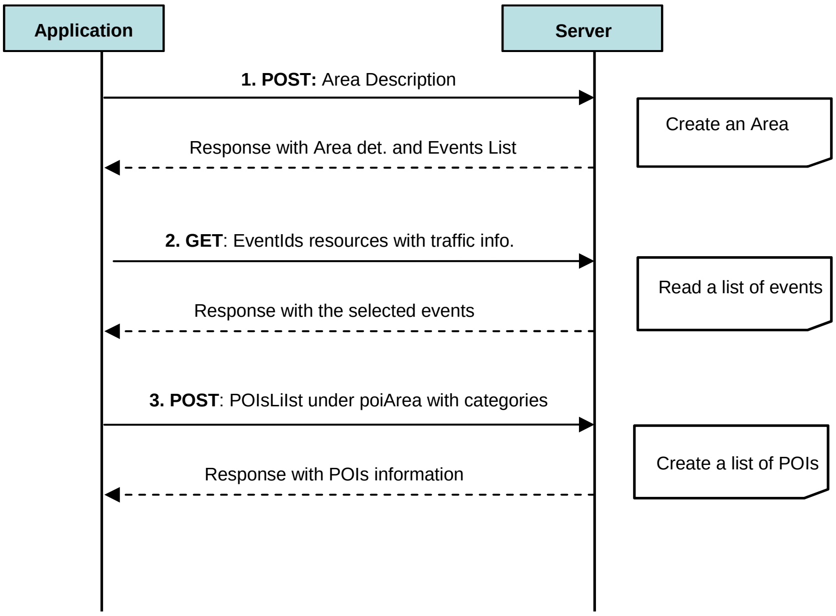
Application
Server
1. POST: Area Description
Create an Area
Response with Area det. and Events List
2. GET: EventIds resources with traffic info.
Read a list of events
Response with the selected events
3. POST: POIsLiIst under poiArea with categories
Create a list of POIs
Response with POIs information

Figure 4 Smart ND requesting traffic and Point Of Interest information

Outline of the flows:

1. The user of DynNav application selects an area where performance parameters and selected categories of traffic events are requested from the server. The application sends the description of the area to the server using POST, the server replies with a resource containing performance parameters and links to events (parted in categories) available for the selected area.

   a) Server may reply with the location of the created resource. In this case an additional get operation on the location is needed to retrieve content of resource.

2. The application reads all events of categories that it considers interesting using GET. The access to traffic events may be limited to categories selected by the user. Considering all information available at this point, application (or user) may decide to request traffic information for other areas repeating steps 1 and 2.

3. To request a list of points of interest, application defines an area of interest and one or more categories of requested POIs using POST. Application sends POI parameters to the DynNav server, and it will reply with a resource containing all points that satisfies user's request.

6. Detailed specification of the resources

6.1 Resource: Trips created by the application

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips

This resource is used to store the list of journeys for which the application request traffic information.

6.1.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.1.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.1.3 GET

This operation is used for reading all trips defined by application.

6.1.3.1 Example: regular trip list request (Informative)

6.1.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/trips HTTP/1.1
Accept: application/xml
Host: example.com
```

6.1.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 26 Oct 2011 16:30:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:tripList xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">

  <trip>
    <originWGS84>
      <WGS84 longitude="45.19074" latitude="7.63441" />
    </originWGS84>
    <destinationWGS84>
      <WGS84 longitude="45.11451" latitude="7.64410" />
    </destinationWGS84>
    <startingTime> 2011-10-26T16:10:00 </startingTime>
     <vehicleType vehicle_type= "rtm01_1" />
    <calculateRoute> true </calculateRoute>
    <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"  />
    <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt02"  />
    <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001 </resourceURL>
  </trip>
  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips </resourceURL>
</dynnav:tripList>
```

6.1.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.1.5 POST

This operation is used for defining parameters of an individual trip. If *calculateRoute* field of Trip structure is set to True, the server should propose a set of route matching trip parameters, providing related traffic information (performance parameters and traffic events). If *calculateRoute* field is set to False no action is required from the server.

6.1.5.1 Example 1: Create a new trip, returning a representation of created resource (Informative)

6.1.5.1.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 26 Oct 2011 16:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:trip xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <originWGS84>
    <WGS84 longitude="45.19074" latitude="7.63441" />
  </originWGS84>
  <destinationWGS84>
    <WGS84 longitude="45.11451" latitude="7.64410" />
  </destinationWGS84>
  <startingTime> 2011-10-26T16:10:00 </startingTime>
  <vehicleType vehicle_type="rtm01_1" />
  <calculateRoute> true </calculateRoute>
</dynnav:trip>
```

6.1.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001
```

```
Content-Length: nnnn
Date: Wed, 26 Oct 2011 16:00:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:trip xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <originWGS84>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
  </originWGS84>
  <destinationWGS84>
    <WGS84 longitude="45.11451" latitude="7.64410" />
  </destinationWGS84>
  <startingTime> 2011-10-26T16:10:00 </startingTime>
  <vehicleType vehicle_type="rtm01_1" />
  <calculateRoute> true </calculateRoute>
  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"  />
  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt02"  />

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001 </resourceURL>
</dynnav:trip>
```

6.1.5.2 Example 2: Create a new trip, returning the location of created resource (Informative)

6.1.5.2.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com
Date: Wed, 26 Oct 2011 16:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:trip xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <originWGS84>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
  </originWGS84>
  <destinationWGS84>
    <WGS84 longitude="45.11451" latitude="7.64410" />
  </destinationWGS84>
  <startingTime> 2011-10-26T16:10:00 </startingTime>
  <vehicleType vehicle_type="rtm01_1" />
  <calculateRoute> true </calculateRoute>
</dynnav:trip>
```

6.1.5.2.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001
Content-Length: nnnn
Date: Wed, 26 Oct 2011 16:00:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
```

```
  <resourceURL>  http://example.com /exampleAPI/dynnav/v1/app0001/app0001/trips/trip001 </resourceURL>
</common:resourceReference>
```

6.1.5.3 Example 3: Unsuccessful trip creation, because of unknown destination address (Informative)

6.1.5.3.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed,30 Nov 2011 17:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:trip xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1" xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
  <originWGS84>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
  </originWGS84>
  <destinationAddress>
   <ca:country>IT</ca:country>
   <ca:A3>Torino</ca:A3>
   <ca:A6>via XXXX</ca:A6>
   <ca:HNO>123</ca:HNO>
     </destinationAddress>
  <startingTime> 2011-10-26T16:10:00 </startingTime>
  <vehicleType vehicle_type="rtm01_1" />
  <calculateRoute> true </calculateRoute>
</dynnav:trip>
```

6.1.5.3.2 Response

```
HTTP/1.1 400 Bad Request
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 30 Nov 2011 17:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <serviceException>
    <messageId>SVC0002</messageId>
    <text> Invalid input value for message part %1 </text>
    <variables> A6:via XXXX </variables>
  </serviceException>
</common:requestError>
```

6.1.5.4 Example 4: Unsuccessful trip creation, because service is not supported in the target Area (Informative)

6.1.5.4.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips HTTP/1.1
Accept: application/xml
```

```
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed,30 Nov 2011 17:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:trip xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1" xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
  <originWGS84>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
  </originWGS84>
  <destinationAddress>
   <ca:country>IT</ca:country>
   <ca:A3>Milano</ca:A3>
   <ca:A6>via turati</ca:A6>
   <ca:HNO>123</ca:HNO>
     </destinationAddress>
  <startingTime> 2011-10-26T16:10:00 </startingTime>
  <vehicleType vehicle_type="rtm01_1" />
  <calculateRoute> true </calculateRoute>
</dynnav:trip>
```

6.1.5.4.2 Response

```
HTTP/1.1 503 Service unavailable
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 30 Nov 2011 17:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <serviceException>
     <messageId>POL1012</messageId>
    <text> %1 %2 </text>
    <variables> region not subscribed by the user </variables>
     <variables> A3:Milano </variables>
</serviceException>
</common:requestError>
```

6.1.5.5 Example 5: Unsucessful Trip creation because location of 3rd party is not authorized (Informative)

6.1.5.5.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 26 Oct 2011 16:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:trip xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <originWGS84>
    <WGS84 longitude="45.19074" latitude="7.63441" />
  </originWGS84>
  <destination3rdParty>+393313332222</destination3rdParty>
  <thirdPartyIDType>MSISDN</thirdPartyIDType>
  <startingTime> 2011-10-26T16:10:00 </startingTime>
  <vehicleType vehicle_type="rtm01_1" />
  <calculateRoute> true </calculateRoute>
</dynnav:trip>
```

6.1.5.5.2 Response

```
HTTP/1.1 403 Forbidden
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 30 Nov 2011 17:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <serviceException>
    <messageId>SVC1004</messageId>
    <text> Location estimation of a 3rd party not available for DynNav service: %1 </text>
     <variables> 3rd party location not authorized by the target </variables>  </serviceException>
</common:requestError>
```

6.1.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].

6.2 Resource: Individual trip description

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}

This resource is used for storing settings of a trip, user preferences and references to routes related to the trip.

6.2.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
tripId	unique trip identifier generated by server

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.2.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.2.3 GET

This operation is used for reading settings about the trip. References to routes related to the trip are returned if available.

6.2.3.1 Example: regular trip information request (Informative)

6.2.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.2.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 26 Oct 2011 18:20:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:trip xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <originWGS84>
    <WGS84 longitude="45.19074" latitude="7.63441" />
  </originWGS84>
  <destinationWGS84>
    <WGS84 longitude="45.11451" latitude="7.64410" />
  </destinationWGS84>
  <startingTime> 2011-10-26T16:10:00 </startingTime>
  <vehicleType vehicle_type="rtm01_1" />
  <calculateRoute> true </calculateRoute>
  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"  />
  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt02"  />
  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001 </resourceURL>
</dynnav:trip>
```

6.2.4 PUT

This operation is used to modify trip parameters, and particular to update origin/current position parameter in trip description. If the origin/current position parameter uploaded with this operation belongs to the currently proposed set of route, current position information is used by the server to delete travelled segments from the defined set of routes or to remove other proposed routes not followed by the user. If the *Origin* parameters do not belong to the proposed set of Route and *calculateRoute* parameters in Trip resource is set to true, the server must calculate a new set of proposed routes and send back the resources identifiers to the application.

6.2.4.1 Example 1: Modify trip parameters, returning a representation of created resource (Informative)

6.2.4.1.1 Request

```
PUT  /exampleAPI/dynnav/v1/app0001/trips/trip001 HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 26 Oct 2011 16:00:00 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<dynnav:trip xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <originWGS84>
    <WGS84 longitude="45.18035" latitude="7.64982" />
  </originWGS84>
  <destinationWGS84>
    <WGS84 longitude="45.11451" latitude="7.64410" />
  </destinationWGS84>
  <startingTime> 2011-10-26T16:10:00 </startingTime>
  <vehicleType vehicle_type="rtm01_1" />
  <calculateRoute> true </calculateRoute>
</dynnav:trip>
```

6.2.4.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001
Date: Wed, 26 Oct 2011 16:00:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:trip xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <originWGS84>
    <WGS84 longitude="45.18035" latitude="7.64982" />
  </originWGS84>
  <destinationWGS84>
    <WGS84 longitude="45.11451" latitude="7.64410" />
  </destinationWGS84>
  <startingTime> 2011-10-26T16:10:00 </startingTime>
  <vehicleType vehicle_type="rtm01_1" />
  <calculateRoute> true </calculateRoute>
  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"  />
  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt02"  />

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001 </resourceURL>
</dynnav:trip>
```

6.2.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.2.6 DELETE

This operation is used for deleting a trip and all routes contained in the rip.

6.2.6.1 Example (Informative)

6.2.6.1.1 Request

```
DELETE  /exampleAPI/dynnav/v1/app0001/trips/trip001 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.2.6.1.2 Response

```
HTTP/1.1 204 No content
Date: Wed, 19 Oct 2011 16:30:00 GMT
```

6.3 Resource: Routes related to a trip

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/routes

This resource provides access to routes related to trips defined by the application. This resource is used by the application as factory resource to upload routes estimated by the smart ND.

6.3.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
tripId	unique trip identifier generated by server

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.3.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.3.3 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: POST’ field in the response as per section 14.7 of [RFC 2616].

6.3.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: POST’ field in the response as per section 14.7 of [RFC 2616].

6.3.5 POST

This operation is used for requesting traffic information related to route proposed by DynNav application running on a ND. The route information, usually described with the whole sequence of segments, may be represented for bandwidth optimization with partial encoding schema, in this case only the sequence of segments that are different respect to the a reference route information is provided (see Appendix H).

6.3.5.1 Example 1: Create a new route, returning a representation of created resource (complete route information) (Informative)

6.3.5.1.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips/trip001/routes HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 26 Oct 2011 17:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">

  <origin>
    <WGS84 longitude="45.19074" latitude="7.63441" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>

  <segment>
    <endPoint>
      <WGS84 longitude="45.18035" latitude="7.64982" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
    </endPoint>
    <linkName> SP2 </linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12864" latitude="7.69526" />
      <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
      <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
    </endPoint>
    <linkName> RA10 </linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.69562" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
    </endPoint>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12080" latitude="7.64055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
    </endPoint>

    <linkName> A55 </linkName>
  </segment>

  <segment>
```

```
    <endPoint>
      <WGS84 longitude="45.12495" latitude="7.63992" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
    </endPoint>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.11451" latitude="7.64410" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
      <location_descriptor descriptor_type="loc03_8" descriptor="Via Paganelli" />
    </endPoint>
    <linkName> Corso Giuseppe Garibaldi </linkName>


  </segment>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>

</dynnav:route>
```

6.3.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01
Date: Wed, 26 Oct 2011 17:00:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <travellingTime> 14 </travellingTime>
  <distance> 17.2 </distance >

  <origin>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>

  <segment>
    <endPoint>
      <WGS84 longitude="45.18035" latitude="7.64982" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
    </endPoint>
    <linkName>SP2</linkName>

    <distance>1.7</distance>
    <regularTravellingTime> 2 </regularTravellingTime>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12864" latitude="7.69526" />
      <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
      <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
    </endPoint>
    <linkName>RA10</linkName>

    <distance>7.6</distance>
<regularTravellingTime> 4 </regularTravellingTime>  </segment>

  <segment>
      <endPoint>
        <WGS84 longitude="45.13028" latitude="7.69562" />
        <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
      </endPoint>

    <distance>1</distance>
<regularTravellingTime> 2 </regularTravellingTime>  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12080" latitude="7.64055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
    </endPoint>
    <linkName>A55</linkName>

    <distance>5.1</distance>
    <regularTravellingTime> 4 </regularTravellingTime>
    <performanceParameters>
      <trafficInfoType>Real-time</trafficInfoType>
      <delay> 1 </delay>
      <speed> 22 </speed>
      <performance >rtm34_4</performance>
    </performanceParameters>
 </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12495" latitude="7.63992" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
    </endPoint>

    <distance>0.7</distance>
<regularTravellingTime> 1 </regularTravellingTime>
</segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.11451" latitude="7.64410" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
      <location_descriptor descriptor_type="loc03_8" descriptor="Via Paganelli" />
    </endPoint>
    <linkName> Corso Giuseppe Garibaldi </linkName>

    <distance>1.2</distance>
    <regularTravellingTime> 2 </regularTravellingTime>

  </segment>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <trafficEvents>
    <category>rtm00_8</category>
    <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004" />
  </trafficEvents>

  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01</resourceURL>
</dynnav:route>
```

6.3.5.2 Example 2: Create a new route, returning the location of created resource (complete route information) (Informative)

6.3.5.2.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips/trip001/routes HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 26 Oct 2011 17:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">

  <origin>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>

  <segment>
    <endPoint>
      <WGS84 longitude="45.18035" latitude="7.64982" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
    </endPoint>
    <linkName> SP2 </linkName>

  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12864" latitude="7.69526" />
      <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
      <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
    </endPoint>
    <linkName> RA10 </linkName>

  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.69562" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
    </endPoint>

  </segment>
  <segment>
    <endPoint>
      <WGS84 longitude="45.12080" latitude="7.64055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
    </endPoint>

    <linkName> A55 </linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12495" latitude="7.63992" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
    </endPoint>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.11451" latitude="7.64410" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
      <location_descriptor descriptor_type="loc03_8" descriptor="Via Paganelli" />
    </endPoint>
    <linkName> Corso Giuseppe Garibaldi </linkName>

  </segment>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>

</dynnav:route>
```

6.3.5.2.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01
Date: Wed, 26 Oct 2011 17:00:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01 </resourceURL>
</common:resourceReference>
```

6.3.5.3 Example 3: Create a new partial route, returning a representation of created resource (partial route information) (Informative)

6.3.5.3.1 Request

For the detail information about partial route encoding schema, see Appendix H.

```
POST /exampleAPI/dynnav/v1/app0001/trips/trip001/routes HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
```

```
Content-Length: nnnn
Date: Wed, 26 Oct 2011 17:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <origin>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>

<partialRouteInformation>true</partialRouteInformation>

<firstSegment>4</firstSegment>
<firstSegment>13</firstSegment>

<lastSegment>10</lastSegment>
<lastSegment>18</lastSegment>

<numSegments>4</numSegments>
<numSegments>3</numSegments>


  <segment>
    <endPoint>
      <WGS84 longitude="45.18035" latitude="7.64982" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
    </endPoint>
    <linkName> SP2 </linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12864" latitude="7.69526" />
      <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
      <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
    </endPoint>
    <linkName> RA10 </linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.69562" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
    </endPoint>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12080" latitude="7.64055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
    </endPoint>


    <linkName> A55 </linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.13212" latitude="7.8326" />
      <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
      <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
    </endPoint>
    <linkName> RA10 </linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.89562" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
    </endPoint>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.14080" latitude="7.94055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
      </endPoint>
  </segment>

  <requestedEventsCategories>rtm00_8</requestedEventsCategories>

<link rel="ReferenceRoute"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"  />

</dynnav:route>
```

6.3.5.3.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt02
Date: Wed, 26 Oct 2011 17:00:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <travellingTime> 4</travellingTime>
  <distance> 3.2 </distance >

  <origin>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>

<partialRouteInformation>true</partialRouteInformation>
<firstSegment>4</firstSegment>
<firstSegment>13</firstSegment>
```

```
<lastSegment>10</lastSegment>
<lastSegment>18</lastSegment>

<numSegments>4</numSegments>
<numSegments>3</numSegments>

<segment>
    <endPoint>
      <WGS84 longitude="45.18035" latitude="7.64982" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
    </endPoint>
    <linkName> SP2 </linkName>

     <distance>1.7</distance>
    <regularTravellingTime> 2 </regularTravellingTime>
  </segment>

<segment>
    <endPoint>
      <WGS84 longitude="45.12864" latitude="7.69526" />
      <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
      <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
    </endPoint>
    <linkName> RA10 </linkName>

     <distance>7.6</distance>
      <regularTravellingTime> 4 </regularTravellingTime>
  </segment>

<segment>

    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.69562" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
    </endPoint>

     <distance>1</distance>
      <regularTravellingTime> 2 </regularTravellingTime>
  </segment>

  <segment>
    <endPoint>

      <WGS84 longitude="45.12080" latitude="7.64055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
    </endPoint>

    <linkName> A55 </linkName>

     <distance>0.7</distance>
      <regularTravellingTime> 1 </regularTravellingTime>
  </segment>
  <segment>
    <endPoint>
      <WGS84 longitude="45.13212" latitude="7.8326" />
      <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
      <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
    </endPoint>
    <linkName> RA10 </linkName>

     <distance>1.3</distance>
      <regularTravellingTime> 3.4 </regularTravellingTime>

  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.89562" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
    </endPoint>

     <distance>2.3</distance>
    <regularTravellingTime> 6.4 </regularTravellingTime>
    <performanceParameters>
      <trafficInfoType>Real-time</trafficInfoType>
      <delay> 2 </delay>
      <speed> 22 </speed>
      <performance >rtm34_4</performance>
   </performanceParameters>

  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.14080" latitude="7.94055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
      </endPoint>
     <distance>2.7</distance>
    <regularTravellingTime> 5.4 </regularTravellingTime>

  </segment>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <trafficEvents >
    <category>rtm00_8</category>
    <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004" />
  </trafficEvents>

  <link rel="ReferenceRoute"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"  />

  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01</resourceURL>
</dynnav:route>
```

6.3.5.4 Example 4: Unsuccessful route creation because of bad route description (Informative)

6.3.5.4.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips/trip001/routes HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com
Date: Wed, 30 Nov 2011 17:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">

  <origin>
    <WGS84 longitude="45.19074" latitude="7.63441" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>

  <segment>
    <endPoint>
      <WGS84 longitude="45.18035" latitude="7.64982" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
    </endPoint>
    <linkName> SP2 </linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12864" latitude="7.69526" />
      <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
      <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
    </endPoint>
    <linkName> RA10 </linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.69562" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
    </endPoint>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12080" latitude="7.64055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
    </endPoint>

    <linkName> A55 </linkName>
  </segment>

  <segment>
```

```
    <endPoint>
      <WGS84 longitude="45.12495" latitude="7.63992" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
    </endPoint>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.11451" latitude="8.64410" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
      <location_descriptor descriptor_type="loc03_8" descriptor="Via Paganelli" />
    </endPoint>
    <linkName> Corso Giuseppe Garibaldi </linkName>
  </segment>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>

</dynnav:route>
```

6.3.5.4.2 Response

```
HTTP/1.1 400 Bad Request
Date: Wed, 30 Nov 2011 17:00:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <serviceException>
    <messageId>SVC0002</messageId>
    <text> Invalid input value for message part %1 </text>
    <variables> segment : longitude="45.11451" latitude="8.64410" </variables>
  </serviceException>
</common:requestError>
```

6.3.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: POST’ field in the response as per section 14.7 of [RFC 2616].

6.4 Resource: Individual route description in full format

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/routes/{routeId}

This resource is used to describe the route in terms of road segments with related performances and traffic events.

6.4.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI

apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
tripId	unique trip identifier for which the route is proposed
routeId	unique route identifier generated by server

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.4.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.4.3 GET

Read one route from server. The route performance information may be updated by the server and the notification procedure trigs a new reading of the resource. The application may optionally request from the server graphical representation of the route, shape information is encoded as a sequence of WGS84 points in *polyline* field available in each *segment* structure. The resolution of the polyline is defined by the server in order to enable a correct representation on turn-by-turn navigation maps.

Supported parameters in the query string of the Request URL are:

Name	Type/Values	Optional	Description
shapeReq	Xsd:Boolean	Yes	This parameter specifies whether graphical representation for the route has to be provided in GET resposnse. If it is set to true, shape information, encoded in *polyline* field in each single *segment*, SHALL be provided, if set to false or absent the segments shape is not requested.

6.4.3.1 Example 1: regular route information request with graphical representation (Informative)

6.4.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01?shapeReq=true HTTP/1.1
Accept: application/xml
Host: example.com
```

6.4.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 19 Oct 2011 16:30:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <travellingTime> 14 </travellingTime>
  <distance> 17.2 </distance >

  <origin>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
```

```
</origin>
<segment>
  <endPoint>
    <WGS84 longitude="45.18035" latitude="7.64982" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
    <location_descriptor descriptor_type="loc03_8" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
  </endPoint>
  <polyLine>45.19075 7.63269, 45.190751 7.632691, 45.190752 7.632692, 45.190753 7.632693, 45.190751 7.632694</polyLine>
  <linkName>SP2</linkName>

  <distance>1.7</distance>
  <regularTravellingTime> 2 </regularTravellingTime>

</segment>

<segment>
  <endPoint>
    <WGS84 longitude="45.12864" latitude="7.69526" />
    <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
    <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
  </endPoint>
  <polyLine>45.12075 7.63269, 45.120751 7.632691, 45.120752 7.632692, 45.120753 7.632693, 45.190754 7.632694</polyLine>
  <linkName>RA10</linkName>

  <distance>7.6</distance>
  <regularTravellingTime> 4 </regularTravellingTime>

</segment>

<segment>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.69562" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
    </endPoint>
 <polyLine>45.12075 7.63269, 45.190751 7.632691</polyLine>

  <distance>1</distance>
  <regularTravellingTime> 2 </regularTravellingTime>

</segment>

<segment>
  <endPoint>
    <WGS84 longitude="45.12080" latitude="7.64055" />
    <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
    <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
  </endPoint>
 <polyLine>45.12075 7.63269, 45.190751 7.632691</polyLine>
  <linkName>A55</linkName>

  <distance>5.1</distance>
   <regularTravellingTime> 4 </regularTravellingTime>
   <performanceParameters>
    <trafficInfoType>Real-time</trafficInfoType>
    <delay> 2 </delay>
    <speed> 22 </speed>
```

```
      <performance >rtm34_4</performance>
   </performanceParameters>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12495" latitude="7.63992" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
    </endPoint>
   <polyLine>45.12075 7.63269, 45.190751 7.632691</polyLine>

    <distance>0.7</distance>
    <regularTravellingTime> 1 </regularTravellingTime>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.11451" latitude="7.64410" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
      <location_descriptor descriptor_type="loc03_8" descriptor="Via Paganelli" />
    </endPoint>
   <polyLine>45.12075 7.63269, 45.190751 7.632691</polyLine>
    <linkName> Corso Giuseppe Garibaldi </linkName>

    <distance>1.2</distance>
    <regularTravellingTime> 2 </regularTravellingTime>

  </segment>

  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <trafficEvents >
    <category>rtm00_8</category>
    <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004" />
  </trafficEvents>

  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01 </resourceURL>
</dynnav:route>
```

6.4.3.2 Example: regular route information request without graphical representation (default) (Informative)

6.4.3.2.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.4.3.2.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 19 Oct 2011 16:30:00 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <travellingTime> 14 </travellingTime>
  <distance> 17.2 </distance >

  <origin>
    <WGS84 longitude="45.19074" latitude=" 7.63441" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>

  <segment>
    <endPoint>
      <WGS84 longitude="45.18035" latitude="7.64982" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
    </endPoint>
    <linkName>SP2</linkName>

    <distance>1.7</distance>
    <regularTravellingTime> 2 </regularTravellingTime>

  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12864" latitude="7.69526" />
      <location_descriptor descriptor_type="loc03_7" descriptor="RA10: Raccordo autostradale Torino-Caselle" />
      <location_descriptor descriptor_type="loc03_8" descriptor="1 Autostrade" />
    </endPoint>
    <linkName>RA10</linkName>

    <distance>7.6</distance>
    <regularTravellingTime> 4 </regularTravellingTime>

  </segment>

  <segment>
      <endPoint>
        <WGS84 longitude="45.13028" latitude="7.69562" />
        <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
      </endPoint>

    <distance>1</distance>
    <regularTravellingTime> 2 </regularTravellingTime>

  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12080" latitude="7.64055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
    </endPoint>
    <linkName>A55</linkName>
    <distance>5.1</distance>
     <regularTravellingTime> 4 </regularTravellingTime>
     <performanceParameters>
      <trafficInfoType>Real-time</trafficInfoType>
      <delay> 2 </delay>
      <speed> 22 </speed>
      <performance >rtm34_4</performance>
   </performanceParameters>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12495" latitude="7.63992" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
    </endPoint>

    <distance>0.7</distance>
    <regularTravellingTime> 1 </regularTravellingTime>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.11451" latitude="7.64410" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
      <location_descriptor descriptor_type="loc03_8" descriptor="Via Paganelli" />
    </endPoint>
    <linkName> Corso Giuseppe Garibaldi </linkName>

    <distance>1.2</distance>
    <regularTravellingTime> 2 </regularTravellingTime>

  </segment>

  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <trafficEvents >
    <category>rtm00_8</category>
    <link rel="Event"  href="http://example.com/exampleAPI/dynnavv1//app0001/events/evt004" />
  </trafficEvents>

  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01 </resourceURL>
</dynnav:route>
```

6.4.4 PUT

This operation is used by the application for requesting traffic information related to a route that replaces a previously defined one. This operation is used when the vehicle diverts from the previously defined route: traffic information is requested for a new proposed route replacing the previous one.

6.4.4.1 Example 1: Modify route description, returning a representation of the resource with performance parameters (Informative)

6.4.4.1.1 Request

```
PUT /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01 HTTP/1.1
Accept: application/xml
```

```
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 26 Oct 2011 16:00:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <travellingTime> 14 </travellingTime>
  <distance> 17.2 </distance >

  <origin>
    <WGS84 longitude="45.14048" latitude=" 7.65575" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>
  <segment>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.65778" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="A55 Tangenziale di Torino" />
    </endPoint>
    <linkName>SP2</linkName>


  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12653" latitude="7.65952" />
      <location_descriptor descriptor_type="loc03_7" descriptor=" A55: Tangenziale di Torino " />
    </endPoint>
    <linkName>RA10</linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12080" latitude="7.64055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
    </endPoint>
    <linkName>A55</linkName>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12495" latitude="7.63992" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
    </endPoint>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.11451" latitude="7.64410" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
      <location_descriptor descriptor_type="loc03_8" descriptor="Via Paganelli" />
    </endPoint>
    <linkName> Corso Giuseppe Garibaldi </linkName>
  </segment>

  <requestedEventsCategories>rtm00_8</requestedEventsCategories>

  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01</resourceURL>
</dynnav:route>
```

6.4.4.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001//routes/rt01
Date: Wed, 26 Oct 2011 16:00:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <travellingTime> 14 </travellingTime>
  <distance> 17.2 </distance >

  <origin>
    <WGS84 longitude="45.14048" latitude=" 7.65575" />
    <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>

  <segment>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.65778" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="A55 Tangenziale di Torino" />
    </endPoint>
    <linkName>SP2</linkName>

    <distance>1.1</distance>
    <regularTravellingTime> 6 </regularTravellingTime>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12653" latitude="7.65952" />
      <location_descriptor descriptor_type="loc03_7" descriptor=" A55: Tangenziale di Torino " />
    </endPoint>
    <linkName>RA10</linkName>

    <distance>0.5</distance>
    <regularTravellingTime> 1 </regularTravellingTime>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12080" latitude="7.64055" />
      <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
      <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
    </endPoint>
    <linkName>A55</linkName>

    <distance>1.3</distance>
    <regularTravellingTime> 2 </regularTravellingTime>
     <performanceParameters>
      <trafficInfoType>Real-time</trafficInfoType>
      <delay> 2 </delay>
      <speed> 22 </speed>
      <performance >rtm34_4</performance>
   </performanceParameters>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.12495" latitude="7.63992" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
    </endPoint>

    <distance>0.7</distance>
    <regularTravellingTime> 1 </regularTravellingTime>
  </segment>

  <segment>
    <endPoint>
      <WGS84 longitude="45.11451" latitude="7.64410" />
      <location_descriptor descriptor_type="loc03_7" descriptor="Corso Giuseppe Garibaldi" />
      <location_descriptor descriptor_type="loc03_8" descriptor="Via Paganelli" />
    </endPoint>
    <linkName> Corso Giuseppe Garibaldi </linkName>

    <distance>1.2</distance>
    <regularTravellingTime> 2 </regularTravellingTime>

  </segment>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <trafficEvents >
    <category>rtm00_8</category>
    <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004" />
  </trafficEvents>

  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01</resourceURL>
</dynnav:route>
```

6.4.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.4.6 DELETE

This operation is used for removing a route from a trip.

6.4.6.1 Example (Informative)

6.4.6.1.1 Request

```
DELETE  /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.4.6.1.2 Response

```
HTTP/1.1 204 No content
Date: Wed, 19 Oct 2011 16:30:00 GMT
```

6.5 Resource: Individual route description in the summarized format

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/Routes/{routeId}/sumRoute

This resource is used to describe a route in summarized format: only main road segments are provided, with related performances and traffic events.

6.5.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
tripId	unique trip identifier for which the route is proposed
routeId	unique summarized route format identifier generated by server

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.5.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.5.3 GET

Read one summarized route from the server. Access to summarized routes resources allows bandwidth optimization in DynNav application: only the most significant segments of routes are provided to the application. This operation can be exploited in the preliminary stage of the navigation application when, for the defined trip, a set of route is proposed by the DynNav server; at this stage the user may need to access only to high level description of routes, with no need for detailed information. The application may optionally request from the server graphical representation of the summarized route, this information is encoded as a sequence of WGS84 points in *polyline* field of each *segment* structure of the *route*. The resolution of the *polyline* for summarized routes segments, defined by the server, shall target an high level representation of the route on roads maps.

Supported parameters in the query string of the Request URL are:

Name	Type/Values	Optional	Description
shapeReq	xsd:Boolean	Yes	This parameter specifies whether graphical representation for the route has to be provided in GET resposnse. If it is set to true, shape information, encoded in polyline field for each single segment, SHALL be provided, if set to false or absent the segments shape is not requested.

6.5.3.1 Example: regular summarized route information request (informative)

6.5.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/route001/sumRoutes HTTP/1.1
Accept: application/xml
Host: example.com
```

6.5.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 19 Oct 2011 16:30:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:route xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <travellingTime> 15 </travellingTime>
  <distance> 17.2 </distance >

  <origin>
   <WGS84 longitude="45.19074" latitude=" 7.63441" />
   <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
  </origin>

  <segment>
    <endPoint>
     <WGS84 longitude="45.13028" latitude="7.69562" />
     <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino" />
    </endPoint>

   <distance>10.2</distance>
   <regularTravellingTime> 8 </regularTravellingTime>

  </segment>

  <segment>
   <endPoint>
    <WGS84 longitude="45.12080" latitude="7.64055" />
    <location_descriptor descriptor_type="loc03_7" descriptor="A55: Tangenziale di Torino " />
    <location_descriptor descriptor_type="loc03_8" descriptor="Venaria" />
   </endPoint>
   <linkName>A55</linkName>

   <distance>7</distance>
   <regularTravellingTime> 7</regularTravellingTime>
```

```
     <performanceParameters>
      <trafficInfoType>Real-time</trafficInfoType>
      <delay> 4 </delay>
      <speed> 22 </speed>
      <performance >rtm34_4</performance>
   </performanceParameters>  </segment>

  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <trafficEvents >
    <category>rtm00_8</category>
    <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004" />
  </trafficEvents>

  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"  />
  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/route001/sumRoutes </resourceURL>
</dynnav:route>
```

6.5.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, ’ field in the response as per section 14.7 of [RFC 2616].

6.5.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, ’ field in the response as per section 14.7 of [RFC 2616].

6.5.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET,’ field in the response as per section 14.7 of [RFC 2616].

6.6 Resource: Areas created by the application for traffic information

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/areas

This resource is used to provide access to all areas defined by the application. An area is used to retrieve traffic information on-demand, notification procedure may be subscribed by the application.

6.6.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.6.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.6.3 GET

This operation is used for reading traffic information and parameters for all areas currently defined by the application.

6.6.3.1 Example: read traffic information related to all the defined area (Informative)

6.6.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/areas HTTP/1.1
Accept: application/xml
Host: example.com
```

6.6.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 23 Nov 2011 15:13:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:areaList xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <area>
    <areaDesc language=" loc41_30">
      <location_descriptions>
          <area_reference country="loc40_106" area_tree_version="1">
             <area_tree_entry level="1" branch="1" predecessor_branch="0">
                    <area_type area_type="loc06_8"/>
                    <area_descriptor area_name="Torino"/>
             </area_tree_entry >
          </area_reference>
      </location_descriptions>
    </areaDesc>
    <startValidityTime> 2011-11-23T16:00:00  </startValidityTime>
    <endValidityTime> 2011-11-23T20:00:00  </endValidityTime>
   <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <requestedEventsCategories>rtm00_5</requestedEventsCategories>
  <timeResolution> 60 </timeResolution>

    <events>
      <category>rtm00_8</category>
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt002"  />
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004"  />
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt005"  />
    </events>

<!-- For readability in the example performance parameters are provided only for one segment -->

<segmentPerformance>
    <originPoint>
      <WGS84 longitude="45.14048" latitude=" 7.65575" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
    </originPoint>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.65778" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="A55 Tangenziale di Torino" />
     </endPoint>
    <linkName>SP2</linkName>
    <distance>1.1</distance>
    <regularTravellingTime> 2 </regularTravellingTime>
    <performanceParameters>
       <trafficInfoType>Real-time</trafficInfoType >
       <time > 2011-11-23T15:45:00 </time >
       <delay> 1.2 </delay>
       <speed> 22 </speed>
       <performance>rtm34_4</performance>
    </performanceParameters>
         <performanceParameters>
        <trafficInfoType >Forecast</trafficInfoType >
       <time > 2011-11-23T17:00:00 </time >
       <delay> 0.8 </delay>
       <speed> 42 </speed>
       <performance>rtm34_2</performance>
         </performanceParameters>
         <performanceParameters>
         <trafficInfoType >Forecast</trafficInfoType >
         <time> 2011-11-23T18:00:00 </time>
         <delay> 0.2 </delay>
         <speed> 70 </speed>
        </performanceParameters>
        <performanceParameters>
         <trafficInfoType >Forecast</trafficInfoType >
         <time> 2011-11-23T19:00:00 </time>
         <delay> 0.3 </delay>
         <speed> 66 </speed>
       </performanceParameters>
       <performanceParameters>
        <trafficInfoType >Forecast</trafficInfoType >
        <time> 2011-11-23T20:00:00 </time>
        <delay> 0.1 </delay>
        <speed> 80</speed>
       </performanceParameters>
    </segmentPerformance>


    <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001 </resourceURL>
  </area>
  <area>
    <areaDesc language="loc41_30">
      <location_descriptions>
          <area_reference country= "loc40_106" area_tree_version="1">
             <area_tree_entry level=" 1" branch="1" predecessor_branch="0">
                    <area_type area_type="loc06_8" />
                    <area_descriptor area_name="Rivoli" />
             </area_tree_entry >
          </area_reference>
      </location_descriptions>
    </areaDesc>
    <startValidityTime> 2011-11-23T18:00:00  </startValidityTime>
    <endValidityTime> 2011-11-23T22:00:00  </endValidityTime>
    <requestedEventsCategories >rtm00_2</requestedEventsCategories >
    <requestedEventsCategories >rtm00_8</requestedEventsCategories >
    <events>
      <category>rtm00_2</category>
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt010"  />
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt015"  />
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt017"  />
    </events>
    <events>
      <category>rtm00_8</category>
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt0017"  />
    </events>
    <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/areas/area002 </resourceURL>
  </area>
  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/areas </resourceURL>
</dynnav:areaList>
```

6.6.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC 2616].

6.6.5 POST

This operation is used for requesting traffic information (events and performance parameters) in a selected area.

Note: The Area is describebd according to LocML formalism [TTI LOC], please note that the *Area_tree_entity* defined in the human readable area description of LocML [TTI LOC chap. 5.3.1.1] is not used in DynNav application and parameters of *Area_tree_entity* structure have no meaning.

6.6.5.1 Example 1: Create a new area, returning a representation of created resource (Informative)

6.6.5.1.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/areas HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 23 Nov 2011 15:00:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:area xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
    <areaDesc language=" loc41_30">
      <location_descriptions>
          <area_reference country="loc40_106" area_tree_version="1">
            <area_tree_entry level="1" branch="1" predecessor_branch="0">
                  <area_type area_type="loc06_8"/>
                  <area_descriptor area_name="Torino"/>
            </area_tree_entry >
          </area_reference>
      </location_descriptions>
    </areaDesc>
  <startValidityTime> 2011-11-23T16:00:00  </startValidityTime>
  <endValidityTime> 2011-11-23T20:00:00  </endValidityTime>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <requestedEventsCategories>rtm00_5</requestedEventsCategories>
  <timeResolution> 60 </timeResolution>

</dynnav:area>
```

6.6.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001
Content-Length: nnnn
Date: Wed, 23 Nov 2011 15:00:15 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:area xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
    <areaDesc language=" loc41_30">
      <location_descriptions>
          <area_reference country="loc40_106" area_tree_version="1">
             <area_tree_entry level="1" branch="1" predecessor_branch="0">
                    <area_type area_type="loc06_8"/>
                    <area_descriptor area_name="Torino"/>
             </area_tree_entry >
          </area_reference>
      </location_descriptions>
    </areaDesc>
  <startValidityTime> 2011-11-23T16:00:00  </startValidityTime>
  <endValidityTime> 2011-11-23T20:00:00  </endValidityTime>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <requestedEventsCategories>rtm00_5</requestedEventsCategories>
  <timeResolution> 60 </timeResolution>

    <events>
      <category>rtm00_8</category>
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt002"  />
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004"  />
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt005"  />
    </events>

<!-- For readability in the example performance parameters are provided only for one segment -->
    <segmentPerformance>
    <originPoint>
      <WGS84 longitude="45.14048" latitude=" 7.65575" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
    </originPoint>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.65778" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="A55 Tangenziale di Torino" />
     </endPoint>
    <linkName>SP2</linkName>
    <distance>1.1</distance>
    <regularTravellingTime> 2 </regularTravellingTime>
    <performanceParameters>
      <trafficInfoType>Real-time</trafficInfoType >
      <time > 2011-11-23T15:45:00 </time >
      <delay> 1.2 </delay>
      <speed> 22 </speed>
      <performance>rtm34_4</performance>
    </performanceParameters>
        <performanceParameters>
       <trafficInfoType >Forecast</trafficInfoType >
      <time > 2011-11-23T17:00:00 </time >
      <delay> 0.8 </delay>
      <speed> 42 </speed>
      <performance>rtm34_2</performance>
        </performanceParameters>
        <performanceParameters>
        <trafficInfoType >Forecast</trafficInfoType >
        <time> 2011-11-23T18:00:00 </time>
        <delay> 0.2 </delay>
        <speed> 70 </speed>
       </performanceParameters>
       <performanceParameters>
        <trafficInfoType >Forecast</trafficInfoType >
        <time> 2011-11-23T19:00:00 </time>
        <delay> 0.3 </delay>
        <speed> 66 </speed>
      </performanceParameters>
      <performanceParameters>
       <trafficInfoType >Forecast</trafficInfoType >
       <time> 2011-11-23T20:00:00 </time>
       <delay> 0.1 </delay>
       <speed> 80</speed>
      </performanceParameters>
    </segmentPerformance>

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001 </resourceURL>
</dynnav:area>
```

6.6.5.2 Example 2: Create a new area, returning the location of created resource (Informative)

6.6.5.2.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/areas HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 23 Nov 2011 15:00:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:area xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
    <areaDesc language=" loc41_30">
```

```
      <location_descriptions>
          <area_reference country="loc40_106" area_tree_version="1">
             <area_tree_entry level="1" branch="1" predecessor_branch="0">
                    <area_type area_type="loc06_8"/>
                    <area_descriptor area_name="Torino"/>
             </area_tree_entry >
          </area_reference>
      </location_descriptions>
    </areaDesc>
  <startValidityTime> 2011-11-23T16:00:00  </startValidityTime>
  <endValidityTime> 2011-11-23T20:00:00  </endValidityTime>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <requestedEventsCategories>rtm00_5</requestedEventsCategories>
  <timeResolution> 60 </timeResolution>

</dynnav:area>
```

6.6.5.2.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001
Date: Wed, 23 Nov 2011 15:00:15 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <resourceURL>  http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001 </resourceURL>
</common:resourceReference>
```

6.6.5.3 Example 3: Create a new area for preliminary access to traffic information, returning the location of created resource (Informative)

6.6.5.3.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/areas HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 23 Nov 2011 15:00:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:area xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
   <areaDesc language="loc41_30">
    <location_coordinates location_type="loc01_7">
     <location_point>
      <WGS84 longitude="45.14048"  latitude=" 7.65575" />
     </location_point>
    </location_coordinates>
   </areaDesc>
   <areaDesc language="loc41_30">
    <location_coordinates location_type="loc01_7">
     <location_point>
```

```
      <WGS84 longitude="45.13028" latitude="7.65778" />
    </location_point>
   </location_coordinates>
  </areaDesc>
  <startValidityTime> 2011-11-23T16:00:00  </startValidityTime>
  <endValidityTime> 2011-11-23T20:00:00  </endValidityTime>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <requestedEventsCategories>rtm00_5</requestedEventsCategories>
</dynnav:area>
```

6.6.5.3.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001
Date: Wed, 23 Nov 2011 15:00:15 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <resourceURL>  http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001 </resourceURL>
</common:resourceReference>
```

6.6.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].

6.7 Resource: Individual area for traffic information

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/areas/{areaId}

This resource is used to provide access to traffic information related to the defined area.

6.7.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
areaId	area identifier

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.7.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.7.3 GET

This operation is used for accessing to traffic events and network performance parameters related to a selected area.

Note: The Area is describebd according to LocML formalism [TTI LOC], please note that the Area_tree_entity defined in the human readable area description of LocML [TTI LOC chap. 5.3.1.1] is not used in DynNav application and parameters of Area_tree_entity structure have no meaning.

6.7.3.1 Example 1: Read events and performance parameters related to an area (Informative)

6.7.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/areas/area001 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.7.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 23 Nov 2011 15:33:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:area xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
    <areaDesc language=" loc41_30">
      <location_descriptions>
          <area_reference country="loc40_106" area_tree_version="1">
             <area_tree_entry level="1" branch="1" predecessor_branch="0">
                    <area_type area_type="loc06_8"/>
                    <area_descriptor area_name="Torino"/>
             </area_tree_entry >
          </area_reference>
      </location_descriptions>
    </areaDesc>
  <startValidityTime> 2011-11-23T16:00:00  </startValidityTime>
  <endValidityTime> 2011-11-23T20:00:00  </endValidityTime>
  <requestedEventsCategories>rtm00_8</requestedEventsCategories>
  <requestedEventsCategories>rtm00_5</requestedEventsCategories>
  <timeResolution> 60 </timeResolution>

    <events>
      <category>rtm00_8</category>
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt002"  />
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004"  />
      <link rel="Event"  href="http://example.com/exampleAPI/dynnav/v1/app0001/events/evt005"  />
    </events>

<!-- For readability in the example performance parameters are provided only for one segment  -->

   <segmentPerformance>
    <originPoint>
      <WGS84 longitude="45.14048" latitude=" 7.65575" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
    </originPoint>
    <endPoint>
      <WGS84 longitude="45.13028" latitude="7.65778" />
      <location_descriptor descriptor_type="loc03_7" descriptor="SP2: Strada Provinciale di Germagnano" />
      <location_descriptor descriptor_type="loc03_8" descriptor="A55 Tangenziale di Torino" />
     </endPoint>
    <linkName>SP2</linkName>
    <distance>1.1</distance>
    <regularTravellingTime> 2 </regularTravellingTime>
    <performanceParameters>
       <trafficInfoType>Real-time</trafficInfoType >
       <time > 2011-11-23T15:45:00 </time >
       <delay> 1.2 </delay>
       <speed> 22 </speed>
       <performance>rtm34_4</performance>
    </performanceParameters>
         <performanceParameters>
        <trafficInfoType >Forecast</trafficInfoType >
       <time > 2011-11-23T17:00:00 </time >
       <delay> 0.8 </delay>
       <speed> 42 </speed>
       <performance>rtm34_2</performance>
         </performanceParameters>
         <performanceParameters>
         <trafficInfoType >Forecast</trafficInfoType >
         <time> 2011-11-23T18:00:00 </time>
         <delay> 0.2 </delay>
         <speed> 70 </speed>
        </performanceParameters>
        <performanceParameters>
         <trafficInfoType >Forecast</trafficInfoType >
         <time> 2011-11-23T19:00:00 </time>
         <delay> 0.3 </delay>
         <speed> 66 </speed>
       </performanceParameters>
       <performanceParameters>
        <trafficInfoType >Forecast</trafficInfoType >
        <time> 2011-11-23T20:00:00 </time>
        <delay> 0.1 </delay>
        <speed> 80</speed>
       </performanceParameters>
    </segmentPerformance>

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001 </resourceURL>
</dynnav:area>
```

6.7.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.7.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.7.6 DELETE

This operation is used for removing an area from the application.

6.7.6.1 Example (Informative)

6.7.6.1.1 Request

```
DELETE /exampleAPI/dynnav/v1/app0001/areas/area001 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.7.6.1.2 Response

```
HTTP/1.1 204 No content
Date: Wed, 23 Nov 2011 18:30:00 GMT
```

6.8 Resource: Subscriptions created by the application

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/subscriptions

This resource is used for defining a set of subscriptions to notification service. Subscription is defined for *Area* and *Trip* resources. A subscription for *Area* resource requests notification for new traffic events and updated network performances parameters in the selected area. A subscription to a *Trip* requests notification service for: (a) alternative routes related to the Trip; (b) updates on the already proposed routes: new traffic events (with links in the route itself); (b) updated performance parameters (in *performanceParameter* field in each *segment* structure of the *route).*

The subscription operations are used also to manage the following functionalities:

- authorization the DynNav server to track, through an external location platform, the ND running the DynNav application, (a locationURI for locating the ND may optionally be provided)
- requesting 3rd party tracking procedure, through an external location platform, in case the final destination of the trip is represented by a 3rd party ID

6.8.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.8.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.8.3 GET

This operation is used for reading the list of subscriptions and related settings created by an application.

6.8.3.1 Example: regular request (Informative)

6.8.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com
```

6.8.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 23 Nov 2011 16:13:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:subscriptionList xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <subscription>
    <callbackReference>
      <notifyURL>http://app001.example.com/notifications/DynNavNotification</notifyURL>
    </callbackReference>
    <link rel="Trip"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"  />
    <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"  />
    <trackingProc> true </trackingProc>
    <deviceLocationURI> http://locationserver.example.com/hu4u43b780  </deviceLocationURI>
    <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001 </resourceURL>
  </subscription>

  <subscription>
    <callbackReference>
      <notifyURL>http://app002.example.com/notifications/DynNavNotification</notifyURL>
    </callbackReference>
    <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip002/routes/rt03"  />
    <trackingProc> true </trackingProc>
    <deviceLocationURI> http://locationserver.example.com/ten23orbc9  </deviceLocationURI>
    <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub002 </resourceURL>
  </subscription>

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions </resourceURL>
</dynnav:subscriptionList>
```

6.8.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].

6.8.5 POST

This operation is used to subscribe to notification service for receiving updated information for defined DynNav resources. The notification service can be subscribed for a list of Trips and/or Areas. In case of subscription for a trip, notification service will cover all the routes defined for the Trip (not removed by the application) and related events. A subscription for an *Area* requests notification for related new traffic events and updated network performances parameters information.

If *trackingProc* parameter is set to true and optionally a *LocationURI* is provided, the DynNav server will track the ND position through an external application. If *tracking3rdParty* is set to true and the application has defined a 3rd party ID as final destination, the DynNav server is requested to track the 3rd party through an external location platform.

6.8.5.1 Example 1: Create a new subscription, returning a representation of created resource (Informative)

6.8.5.1.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 23 Oct 2011 17:45:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:subscription xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DynNavNotification</notifyURL>
  </callbackReference>

  <link rel="Trip"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"  />

  <trackingProc> true </trackingProc>
  <deviceLocationURI> http://locationserver.example.com/hu4u43b780c </deviceLocationURI>
</dynnav:subscription>
```

6.8.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001
Date: Wed, 23 Oct 2011 17:45:05 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:subscription xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DynNavNotification</notifyURL>
  </callbackReference>

  <link rel="Trip"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"  />

  <trackingProc> true </trackingProc>
  <deviceLocationURI> http://locationserver.example.com/hu4u43b780c </deviceLocationURI>

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001 </resourceURL>
</dynnav:subscription>
```

6.8.5.2 Example 2: Create a new subscription, returning the location of created resource (Informative)

6.8.5.2.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 26 Oct 2011 17:04:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:subscription xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DynNavNotification</notifyURL>
  </callbackReference>

  <link rel="Trip"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip002"  />

  <trackingProc> true </trackingProc>
  <deviceLocationURI> http://locationserver.example.com/hyf45ty7wa </deviceLocationURI>
</dynnav:subscription>
```

6.8.5.2.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub002
Date: Wed, 26 Oct 2011 17:04:02 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <resourceURL>  http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub002 </resourceURL>
</common:resourceReference>
```

6.8.5.3 Example 3: Unsuccessful subscription creation, because of a reference to not existing resource (Informative)

6.8.5.3.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 23 Oct 2011 17:45:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:subscription xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DynNavNotification</notifyURL>
  </callbackReference>
  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/routes/tr01"  />

  <trackingProc> true </trackingProc>
  <deviceLocationURI> http://locationserver.example.com/hu4u43b780c </deviceLocationURI>
</dynnav:subscription>
```

6.8.5.3.2 Response

```
HTTP/1.1 400 Bad Request

Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 23 Oct 2011 17:45:00 GMT
<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <serviceException>
    <messageId>SVC0002</messageId>
    <text> Invalid input value for message part %1 </text>
    <variables> link: href="http://example.com/exampleAPI/dynnav/v1/app0001/routes/tr01 </variables>
  </serviceException>
</common:requestError>
```

6.8.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].

6.9 Resource: Individual subscription settings

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/subscriptions/{subscriptionId}

This resource is used to access to the list of subscribed resources for each single subscription to notification service.

6.9.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
subscriptionId	subscription identifier

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.9.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.9.3 GET

This operation is used for reading the list of resources (*Trip Area*) covered by each single subscription to notification service..

6.9.3.1 Example: regular request (Informative)

6.9.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/subscriptions/sub001 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.9.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 23 Nov 2011 16:13:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:subscription xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DynNavNotification</notifyURL>
  </callbackReference>

  <link rel="Trip"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"  />

  <trackingProc> true </trackingProc>
  <deviceLocationURI> http://locationserver.example.com/hu4u43b780c </deviceLocationURI>

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001 </resourceURL>
</dynnav:subscription>
```

6.9.4 PUT

This operation is used for editing settings of a subscription.

6.9.4.1 Example: modify subscription settings (Informative)

6.9.4.1.1 Request

```
PUT /exampleAPI/dynnav/v1/app0001/subscriptions/sub001 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 23 Nov 2011 16:13:00 GMT


<?xml version="1.0" encoding="UTF-8"?>
<dynnav:subscription xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DynNavNotification</notifyURL>
  </callbackReference>
  <link rel="Trip"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"  />
  <link rel="Area"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/area001"  />
  <trackingProc> true </trackingProc>
  <deviceLocationURI> http://locationserver.example.com/hu4u43b780c </deviceLocationURI>

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001 </resourceURL>
</dynnav:subscription>
```

6.9.4.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 23 Nov 2011 16:13:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:subscription xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DynNavNotification</notifyURL>
  </callbackReference>
  <link rel="Trip"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"  />
  <link rel="Area"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/area01"  />
  <trackingProc> true </trackingProc>
  <deviceLocationURI> http://locationserver.example.com/hu4u43b780c </deviceLocationURI>

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001 </resourceURL>
</dynnav:subscription>
```

6.9.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, PUT, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.9.6 DELETE

This operation is used for removing a subscription.

6.9.6.1 Example (Informative)

6.9.6.1.1 Request

```
DELETE /exampleAPI/dynnav/v1/app0001/subscriptions/sub001 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.9.6.1.2 Response

```
HTTP/1.1 204 No content
Date: Wed, 23 Oct 2011 16:30:00 GMT
```

6.10 Resource: Client notification about resources updates

This resource is a client provided call-back URL for notification about route, area and trip updates. This specification does not make any assumption about the structure of this URL.

6.10.1 Request URL variables

Client provided.

6.10.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.10.3 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: POST’ field in the response as per section 14.7 of [RFC 2616].

6.10.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: POST’ field in the response as per section 14.7 of [RFC 2616].

6.10.5 POST

This operation is used to notify client when new information on subscribed resource is available. Notification service is subscribed for *Area* and *Trip* resources. A subscription for *Area* resource requests notification for new traffic events and updated network performances parameters in the selected area. A subscription to a *Trip* requests notification service for: (a) alternative routes related to the Trip; (b) updates on the already proposed routes: new traffic events (with links in the route itself); (b) updated performance parameters (in *performanceParameter* field in each *segment* structure of the *route).*

6.10.5.1 Example: Notification of available updates (Informative)

6.10.5.1.1 Request

```
POST /notifications/DynNavNotification HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: application.example.com


<?xml version="1.0" encoding="UTF-8"?>
<dynnav:notification xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <link rel="Trip"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"  />
  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"  />
  <link rel="Route"  href="http://example.com/exampleAPI/dynnav/v1/app0001/ trips/trip001/routes/rt02"  />


</dynnav:notification>
```

6.10.5.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

6.10.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: POST’ field in the response as per section 14.7 of [RFC 2616].

6.11 Resource: All events related to the application

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/events

This resource gives access to all events available for the application.

6.11.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.11.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.11.3 GET

This operation is used to read all events available for the application. In individual *Route* or *Area* resources the events are grouped by category, with this information the application can filter the access to events based on categories of interest for the user.

Supported parameters in the query string of the Request URL are:

Name	Type/Values	Optional	Description
eId	xsd:string [1..unbunded]	Yes	Any number of eId=eventId pairs separated by & are allowed. GET response body SHALL include only events whose identifiers are specified in the query.

6.11.3.1 Example 1: Retrieve all events (default) (Informative)

6.11.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/events HTTP/1.1
Accept: application/xml
Host: example.com
```

6.11.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 16 Nov 2011 16:12:00 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<dynnav:eventList xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
 <event>
  <rtMessage message_id="124"  version_number="1"  message_generation_time="2002-04-03T13:40:00Z"
                  severity_factor="rtm31_2">

   <!-- Location is A811 at Drymen -->
   <location_container language="loc41_30">
    <location_coordinates location_type="loc01_5">
     <location_point>
      <WGS84 longitude="-4.45451" latitude="56.05573"/>
      <location_descriptor descriptor_type="loc03_7" descriptor="A811"/>
      <location_descriptor descriptor_type="loc03_8" descriptor="A809"/>
      <location_descriptor descriptor_type="loc03_24" descriptor="Dumbarton"/>
      <location_descriptor descriptor_type="loc03_24" descriptor="Stirling"/>
     </location_point>
    </location_coordinates>
   </location_container>

   <!-- Temporary traffic lights -->
   <facilities_performance>
    <traffic_control traffic_control_type="rtm42_11" traffic_control_status="rtm43_12">
     <position position="rtm10_37"/>
    </traffic_control>
   </facilities_performance>
  </rtMessage>
  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/events/evt001 </resourceURL>
 </event>

 <event>
  <rtMessage message_id="123"  version_number="1"  message_generation_time="2002-04-03T13:03:00Z"
                                     severity_factor= "rtm31_4" >

   <!-- Location is on A12 in Brentford, Essex -->
   <location_container language="loc41_30">
    <location_coordinates location_type="loc01_5">
     <location_point>
      <WGS84 longitude="-0.1337" latitude="51.52641"/>
      <location_descriptor descriptor_type="loc03_7" descriptor="A12"/>
      <location_descriptor descriptor_type="loc03_8" descriptor="A128"/>
      <location_descriptor descriptor_type="loc03_24" descriptor="Brentwood"/>
      <location_descriptor descriptor_type="loc03_25" descriptor="Essex"/>
     </location_point>
     <direction direction_type="loc02_2"/>
    </location_coordinates>
   </location_container>

   <!-- Accident in thick fog involving 50 vehicles -->
   <accidents number_of="1">
    <position position="rtm10_37"/>
    <vehicles number_of="50">
     <vehicle_problem vehicle_problem="rtm03_22"/>
    </vehicles>
   </accidents>
   <visibility>
```

```
        <obscurity obscurity_problem="rtm17_2" visibility_distance="20"/>
      </visibility>
      <network_conditions>
        <position position="rtm10_37"/>
        <restriction restriction="rtm49_1"/>
      </network_conditions>
    </rtMessage>
  </event>

  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/events </resourceURL>
</dynnav:eventList>
```

6.11.3.2 Example 2: Retrieve events whit selected identifiers (Informative)

6.11.3.2.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/events?eId=evt002 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.11.3.2.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 16 Nov 2011 16:13:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:eventList xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <event>
    <rtMessage message_id="123"  version_number="1"  message_generation_time="2002-04-03T13:03:00Z"
                                                                          severity_factor= "rtm31_4" >

      <!-- Location is on A12 in Brentford, Essex -->
      <location_container language="loc41_30">
        <location_coordinates location_type="loc01_5">
          <location_point>
            <WGS84 longitude="-0.1337" latitude="51.52641"/>
            <location_descriptor descriptor_type="loc03_7" descriptor="A12"/>
            <location_descriptor descriptor_type="loc03_8" descriptor="A128"/>
            <location_descriptor descriptor_type="loc03_24" descriptor="Brentwood"/>
            <location_descriptor descriptor_type="loc03_25" descriptor="Essex"/>
          </location_point>
          <direction direction_type="loc02_2"/>
        </location_coordinates>
      </location_container>

      <!-- Accident in thick fog involving 50 vehicles -->
      <accidents number_of="1">
        <position position="rtm10_37"/>
        <vehicles number_of="50">
          <vehicle_problem vehicle_problem="rtm03_22"/>
        </vehicles>
      </accidents>
      <visibility>
```

```
        <obscurity obscurity_problem="rtm17_2" visibility_distance="20"/>
      </visibility>
      <network_conditions>
        <position position="rtm10_37"/>
        <restriction restriction="rtm49_1"/>
      </network_conditions>
    </rtMessage>
  </event>

  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/events </resourceURL>
</dynnav:eventList>
```

6.11.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].

6.11.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].

6.11.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].

6.12 Resource: Individual event information

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/events/{eventId}

This resource gives access to an event available for the application.

6.12.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
eventide	event identifier

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.12.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.12.3 GET

This operation is used to read an event available for the application. In order to filter the access to the events, in *Route or Area* resources the events are grouped by category

6.12.3.1 Example: Retrieve a traffic event (Informative)

6.12.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/events/evt002 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.12.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 16 Nov 2011 16:11:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:event xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <rtMessage message_id="123"  version_number="1"  message_generation_time="2002-04-03T13:03:00Z"
                                     severity_factor= "rtm31_4" >

   <!-- Location is on A12 in Brentford, Essex -->
   <location_container language="loc41_30">
    <location_coordinates location_type="loc01_5">
     <location_point>
      <WGS84 longitude="-0.1337" latitude="51.52641"/>
      <location_descriptor descriptor_type="loc03_7" descriptor="A12"/>
      <location_descriptor descriptor_type="loc03_8" descriptor="A128"/>
      <location_descriptor descriptor_type="loc03_24" descriptor="Brentwood"/>
      <location_descriptor descriptor_type="loc03_25" descriptor="Essex"/>
     </location_point>
     <direction direction_type="loc02_2"/>
    </location_coordinates>
   </location_container>

   <!-- Accident in thick fog involving 50 vehicles -->
   <accidents number_of="1">
    <position position="rtm10_37"/>
    <vehicles number_of="50">
     <vehicle_problem vehicle_problem="rtm03_22"/>
    </vehicles>
   </accidents>
   <visibility>
    <obscurity obscurity_problem="rtm17_2" visibility_distance="20"/>
   </visibility>
   <network_conditions>
    <position position="rtm10_37"/>
    <restriction restriction="rtm49_1"/>
   </network_conditions>
  </rtMessage>
  <resourceURL> http://example.com /exampleAPI/dynnav/v1/app0001/events/evt002 </resourceURL>
</dynnav:event>
```

6.12.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].

6.12.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].

6.12.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].

6.13 Resource: POIs Lists related to a route

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/routes/{routeId}/pois

This resource provides access to the lists of POIs related to a route.

6.13.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
tripId	unique trip identifier generated by server
routeId	route identifier

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.13.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.13.3 GET

This method is used to access to the list of POIs defined for a *route*.

6.13.3.1 Example: Read the POIs of a list (Informative)

6.13.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/ HTTP/1.1
Accept: application/xml
Host: example.com
```

6.13.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 30 Nov 2011 17:00:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:poiListsSet xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <link rel="POIList"  href=" http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01"  />
  <link rel="POIList"  href=" http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL02"  />
  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois </resourceURL>
</dynnav:poiListsSet>
```

6.13.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].

6.13.5 POST

This operation is used to request a list of POIs related to a route that matches defined parameters (i.e. POI category and distance range from the route).

6.13.5.1 Example 1: Create a new list of POIs related to a route, returning a representation of created resource (Informative)

6.13.5.1.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/ HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Date: Wed, 23 Nov 2011 18:00:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:poiList xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <category> urn:service:fuel </category>
  <routeRange> 0 </routeRange >
</dynnav:poiList >
```

6.13.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01
Date: Wed, 23 Nov 2011 18:00:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:poiList xmlns:poi="http://www.w3.org/poi" xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:oma:xml:rest:netapi:dynnav:1
file:///E:/DynNav/exampleReview/OMA_SUP_XSD_rest_DynNav.xsd">
<category> urn:service:fuel</category>
<routeRange> 0 </routeRange>
```

```
 <pois>
  <poi:poi id="http://www.dynnavExamples.org/pois/45343489">
   <poi:label term="XX Petrol Sattion"/>
   <poi:description>
    <poi:value>XX Petrol Station, Food and Drinks</poi:value>
   </poi:description>
   <poi:category term="service:fuel" scheme="http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03">
   </poi:category>
   <poi:link term="related" href="http://www.geonames.org/maps/google_7.678992_45.062735.html" type="text/html"
        scheme="http://www.iana.org/assignments/link-relations/link-relations.xml"/>
   <poi:location>
    <poi:point term="centroid">
     <poi:Point srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
      <poi:posList>7.678992 45.062735</poi:posList>
     </poi:Point>
    </poi:point>
   </poi:location>
  </poi:poi>
 </pois>
 <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01</resourceURL>
</dynnav:poiList>
```

6.13.5.2 Example 2: Create a new list of POIs related to a route, returning the location of created resource (Informative)

6.13.5.2.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/ HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 23 Nov 2011 18:00:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:poiList xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <category> urn:service:fuel </category>
  <routeRange> 0 </routeRange >
</dynnav:poiList >
```

6.13.5.2.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01
Date: Wed, 23 Nov 2011 18:00:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <resourceURL>  http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01</resourceURL>
</common:resourceReference>
```

6.13.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].

6.14 Resource: Individual list of POIs related to a route

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/trips/{tripId}/routes/{routeId}/pois/{poiListId}

This resource provides access to the list of POIs related to a route previously created.

6.14.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
tripId	unique trip identifier generated by server
routeId	route identifier
poiListId	Unique identifier for a list of POIs

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.14.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.14.3 GET

This method is used to access to the list of POIs matching previously defined parameters.

6.14.3.1 Example: Read the POIs of a list related to a route (Informative)

6.14.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.14.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 30 Nov 2011 17:00:10 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<dynnav:poiList xmlns:poi="http://www.w3.org/poi" xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1" >
<category> urn:service:fuel</category>
<routeRange> 0 </routeRange>
 <pois>
  <poi:poi id="http://www.dynnavExamples.org/pois/45343489">
   <poi:label term="XX Petrol Sattion"/>
   <poi:description>
    <poi:value>XX Petrol Station, Food and Drinks</poi:value>
   </poi:description>
   <poi:category term="service:fuel" scheme="http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03">
   </poi:category>
   <poi:link term="related" href="http://www.geonames.org/maps/google_7.678992_45.062735.html" type="text/html"
        scheme="http://www.iana.org/assignments/link-relations/link-relations.xml"/>
   <poi:location>
    <poi:point term="centroid">
     <poi:Point srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
      <poi:posList>7.678992 45.062735</poi:posList>
     </poi:Point>
    </poi:point>
   </poi:location>
  </poi:poi>
 </pois>
 <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01</resourceURL>
</dynnav:poiList>
```

6.14.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.14.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.14.6 DELETE

This operation is used for removing a list of POIs from a route.

6.14.6.1 Example (Informative)

6.14.6.1.1 Request

```
DELETE /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.14.6.1.2 Response

```
HTTP/1.1 204 No content
Date: Wed, 30 Nov 2011 20:00:10 GMT
```

6.15 Resource: POIs lists defined for an area

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/poiAreas

This resource provides access to the lists of POIs related to an area.

6.15.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.15.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.15.3 GET

This method is used to access to the list of POIs that match the parameters (i.e. categories and geographic area) defined by the user.

6.15.3.1 Example: Read the POI of a list related to an area (Informative)

6.15.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/poiAreas HTTP/1.1
Accept: application/xml
Host: example.com
```

6.15.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 30 Nov 2011 17:00:10 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:poiListsSet xmlns:poi="http://www.w3.org/poi" xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
  <link rel="POIList"  href=" http://example.com/exampleAPI/dynnav/v1/app0001/poiL01" />
  <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/poiAreas </resourceURL>
</dynnav:poiListsSet>
```

6.15.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].

6.15.5 POST

This operation is used to request a list of POIs matching defined parameters (i.e. categories and geographic area).

Note: The Area is described according to LocML formalism [TTI LOC], please note that the Area_tree_entity defined in the human readable area description of LocML [TTI LOC chap. 5.3.1.1] is not used in DynNav application and parameters of Area_tree_entity structure have no meaning.

6.15.5.1 Example 1: Create a new list of POIs related to an area, returning a representation of created resource (Informative)

6.15.5.1.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/poiAreas HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 23 Nov 2011 18:00:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:poiList xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
<category> urn:service:fuel</category>
  <area  language="loc41_30">
    <location_coordinates location_type="loc01_1">
    <location_point>
       <WGS84 longitude="7.688992" latitude="45.062935">
         <expansion radius_of_circle="1000" />
       </WGS84>
        </location_point>
    </location_coordinates>
  </area>
</dynnav:poiList>
```

6.15.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/dynnav/v1/app0001/poiAreas/poiL01
Content-Length: nnnn
Date: Wed, 23 Nov 2011 18:00:12 GMT

<dynnav:poiList xmlns:poi="http://www.w3.org/poi" xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1" >
 <category> urn:service:fuel</category>
  <area  language="loc41_30">
    <location_coordinates location_type="loc01_1">
    <location_point>
       <WGS84 longitude="7.688992" latitude="45.062935">
         <expansion radius_of_circle="1000" />
       </WGS84>
        </location_point>
    </location_coordinates>
  </area>

<pois>
 <poi:poi id="http://www.dynnavExamples.org/pois/45343489">
  <poi:label term="XX Petrol Sattion"/>
   <poi:description>
    <poi:value>XX Petrol Station, Food and Drinks</poi:value>
    </poi:description>
   <poi:category term="service:fuel" scheme="http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03">
  </poi:category>

  <poi:link term="related" href="http://www.geonames.org/maps/google_7.678992_45.062735.html" type="text/html"
     scheme="http://www.iana.org/assignments/link-relations/link-relations.xml"/>
   <poi:location>
    <poi:point term="centroid">
      <poi:Point srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
        <poi:posList>7.678992 45.062735</poi:posList>
      </poi:Point>
     </poi:point>
    </poi:location>
  </poi:poi>
 </pois>
 <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01</resourceURL>
</dynnav:poiList>
```

6.15.5.2 Example 2: Create a new list of POIs related to an area, returning the location of created resource (Informative)

6.15.5.2.1 Request

```
POST /exampleAPI/dynnav/v1/app0001/poiAreas HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com
Content-Length: nnnn
Date: Wed, 23 Nov 2011 18:16:11 GMT

<?xml version="1.0" encoding="UTF-8"?>
<dynnav:poiList xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1">
<category> urn:service:fuel</category>
   <area  language="loc41_30">
     <location_coordinates location_type="loc01_1">
     <location_point>
        <WGS84 longitude="7.688992" latitude="45.062935">
          <expansion radius_of_circle="1000" />
        </WGS84>
         </location_point>
     </location_coordinates>
   </area>
</dynnav:poiList >
```

6.15.5.2.2 Response

```
HTTP/1.1 201 Created
```

```
Content-Type: application/xml
Location: http://example.com/exampleAPI/dynnav/v1/app0001/poiAreas/poiL01
Content-Length: nnnn
Date: Wed, 23 Nov 2011 18:16:12 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <resourceURL>  http://example.com/exampleAPI/dynnav/v1/app0001/poiAreas/poiL01 </resourceURL>
</common:resourceReference>
```

6.15.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].

6.16 Resource: Individual list of POIs related to an area

The resource used is:

http://{serverRoot}/dynnav/{apiVersion}/{appId}/poiAreas/{poiAreaId}

This resource provides access to the list of POIs related to an area previously defined by the user.

6.16.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
appId	application identifier
poiAreaId	unique identifier for a list of POIs

See section 5 for a statement on the escaping of reserved characters in URL variables.

6.16.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common]. For Policy Exception and Service Exception fault codes applicable to DynNav, see section 7.

6.16.3 GET

This method is used to access to the list of POIs defined by the application.

6.16.3.1 Example: Read the POIs of a list related to an area (Informative)

6.16.3.1.1 Request

```
GET /exampleAPI/dynnav/v1/app0001/poiAreas/poiL01 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.16.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Wed, 30 Nov 2011 17:56:10 GMT

<dynnav:poiList xmlns:poi="http://www.w3.org/poi" xmlns:dynnav="urn:oma:xml:rest:netapi:dynnav:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:oma:xml:rest:netapi:dynnav:1
file:///E:/DynNav/exampleReview/OMA_SUP_XSD_rest_DynNav.xsd">
 <category> urn:service:fuel</category>
   <area  language="loc41_30">
     <location_coordinates location_type="loc01_1">
     <location_point>
        <WGS84 longitude="7.688992" latitude="45.062935">
          <expansion radius_of_circle="1000" />
        </WGS84>
         </location_point>
     </location_coordinates>
   </area>

<pois>
 <poi:poi id="http://www.dynnavExamples.org/pois/45343489">
  <poi:label term="XX Petrol Sattion"/>
   <poi:description>
    <poi:value>XX Petrol Station, Food and Drinks</poi:value>
    </poi:description>
   <poi:category term="service:fuel" scheme="http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03">
  </poi:category>

  <poi:link term="related" href="http://www.geonames.org/maps/google_7.678992_45.062735.html" type="text/html"
     scheme="http://www.iana.org/assignments/link-relations/link-relations.xml"/>
   <poi:location>
    <poi:point term="centroid">
      <poi:Point srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
        <poi:posList>7.678992 45.062735</poi:posList>
      </poi:Point>
     </poi:point>
    </poi:location>
  </poi:poi>
 </pois>
 <resourceURL> http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01</resourceURL>
</dynnav:poiList>
```

6.16.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.16.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.16.6 DELETE

This operation is used for removing a list of POIs.

6.16.6.1 Example (Informative)

6.16.6.1.1 Request

```
DELETE /exampleAPI/dynnav/v1/app0001/poiAreas/poiL01 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.16.6.1.2 Response

```
HTTP/1.1 204 No content
Date: Wed, 30 Nov 2011 21:00:10 GMT
```

7. Fault definitions

7.1 Service Exceptions

For common Service Exceptions refer to [REST_NetAPI_Common].

The following additional Service Exception code is defined for the DynNav API.

7.1.1 SVC1004: 3rd party location failure

Name	Description
MessageID	SVC1004
Text	Location estimation of a 3rd party not available for DynNav service: %1
Variables	%1 textual description of failure. Value may be: “3rd party position not available” “3rd party location not authorized by the target”
HTTP status code(s)	403 Forbidden, 503 Service unavailable

7.2 Policy Exceptions

For common Policy Exceptions refer to [REST_NetAPI_Common].

The following additional Policy Exception code is defined for the DynNav API.

7.2.1 POL1021: Service not provided in the target area

Name	Description
MessageID	POL1021
Text	%1 %2
Variables	%1 reason for not supporting in the area the service. Values may be: “data not available in the region” “region not subscribed by the user” %2 message part describing the area where service is not provided
HTTP status code(s)	403 Forbidden

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-REST_NetAPI_DynNav-V1_0-20131217-A	17 Dec 2013	Status changed to Approved by TP TP Ref # OMA-TP-2013-0388-INP_DynNav_V1_0_ERP_for_final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for REST. DYNNAV Server

Item	Function	Reference	Requirement
REST-DYNNAV-SUPPORT-S-001-M	Support for the RESTful [DynNav] API	5	
REST-DYNNAV-SUPPORT-S-002-M	Support for the XML request & response format	6	
REST-DYNNAV-SUPPORT-S-003-M	Support for the JSON request & response format	Appendix C	

B.1.1 SCR for REST.DYNNAV.TRIPS Server

Item	Function	Reference	Requirement
REST-DYNNAV-TRIPS-S-001-M	Support of the definition of a trip	6.1	
REST-DYNNAV-TRIPS-S-002-M	Create a trip requesting optionally route information	6.1.5	
REST-DYNNAV-TRIPS-S-003-M	Read the list of active Trip	6.1.3	

B.1.2 SCR for REST.DYNNAV.INDIVIDUAL.TRIP Server

Item	Function	Reference	Requirement
REST-DYNNAV-IND-TRIP-S-001-M	Support the access to trip information	6.2	
REST-DYNNAV-IND-TRIP-S-002-M	Read information related to a single Trip	6.2.3	
REST-DYNNAV-IND-TRIP-S-002-M	Modify information related to a single Trip	6.2.4	
REST-DYNNAV-IND-TRIP-S-004-M	Cancel information related to an individual trip and stop sending related notifcation	6.2.6	

B.1.3 SCR for REST.DYNNAV.ROUTES Server

Item	Function	Reference	Requirement
REST-DYNNAV-ROUTES.SERVER-S-001-O	Support the management of routes related to a Trip	6.3	REST-DYNNAV-ROUTES.SERVER-S-002-O REST-DYNNAV-IND-ROUTE.SERVER-S-003-O
REST-DYNNAV-ROUTES.SERVER-S-002-O	Create of a route proposed by the client	6.3.5	

B.1.4 SCR for REST.DYNNAV.INDIVIDUAL.ROUTE Server

Item	Function	Reference	Requirement
REST-DYNNAV-IND-ROUTE.SERVER-S-001-M	Support the management of a route proposed by the server or the client	6.4	
REST-DYNNAV-IND-ROUTE.SERVER-S-002-M	Read a route	6.4.3	
REST-DYNNAV-IND-ROUTE.SERVER-S-003-M	Modify the route proposed by the client	6.4.4	
REST-DYNNAV-IND-ROUTE.SERVER-S-004-M	Cancel a Route and stop sending related notification	6.4.5	

B.1.5 SCR for REST.DYNNAV.INDIVIDUAL.SUMROUTE Server

Item	Function	Reference	Requirement
REST-DYNNAV-IND-SUMROUTE.SERVER-S-001-M	Support the management of a summarized route proposed by the server	6.5	
REST-DYNNAV-IND-SUMROUTE.SERVER-S-002-M	Read a summarized route	6.5.3	

B.1.6 SCR for REST.DYNNAV.AREAS Server

Item	Function	Reference	Requirement
REST-DYNNAV-AREAS-S-001-O	Support of management of areas for traffic information	6.6	REST-DYNNAV-AREAS-S-002-O REST-DYNNAV-AREAS-S-003-O REST-DYNNAV-IND-AREA-S-001-O
REST-DYNNAV-AREAS-S-002-O	Read the list of active areas for traffic information	6.6.3	
REST-DYNNAV-AREAS-S-003-O	Create an area for traffic information	6.6.5	

B.1.7 SCR for REST.DYNNAV.INDIVIDUAL.AREA Server

Item	Function	Reference	Requirement
REST-DYNNAV-IND-AREA-S-001-O	Support the management of an area for traffic information	6.7	REST-DYNNAV-IND-AREA-S-002-O REST-DYNNAV-IND-AREA-S-003-O REST-DYNNAV-AREAS-S-001-O
REST-DYNNAV-IND-AREA-S-002-O	Read traffic information related to a single Area	6.7.3	
REST-DYNNAV-IND-AREA-S-003-M	Cancel information related to a single area and stop sending related notification	6.7.5	

B.1.8 SCR for REST.DYNNAV.SUBSCRIPTIONS Server

Item	Function	Reference	Requirement
REST-DYNNAV-SUBSCRIPTIONS-S-001-M	Support for management of subscriptions for trips and traffic areas	6.8	
REST-DYNNAV-SUBSCRIPTIONS-S-002-M	Read the list of subscriptions of the application	6.8.3	
REST-DYNNAV-SUBSCRIPTIONS-S-003-M	Create a subscription for a trip or an area	6.8.5	

B.1.9 SCR for REST.DYNNAV.INDIVIDUAL.SUBSCRIPTION Server

Item	Function	Reference	Requirement
REST-DYNNAV-IND-SUBSCRIPTION-S-001-M	Support for manage to information of each subscription	6.9	
REST-DYNNAV-IND-SUBSCRIPTION-S-002-M	Read a subscription settings	6.9.3	
REST-DYNNAV-IND-SUBSCRIPTION-S-003-M	Modify a subscription settings	6.9.4	
REST-DYNNAV-IND-SUBSCRIPTION-S-004-M	Delete a subscription settings	6.9.6	

B.1.10 SCR for REST.DYNNAV.NOTIFICATION Server

Item	Function	Reference	Requirement
REST-DYNNAV-IND-NOTIFIC-S-001-M	Support for manage the notification procedure	6.10	
REST-DYNNAV-IND-NOTIF-S-002-M	Create a notification	6.10.5	

B.1.11 SCR for REST.DYNNAV.EVENTS Server

Item	Function	Reference	Requirement
REST-DYNNAV-EVENTS-S-001-O	Support for read events defined for the application	6.11	REST-DYNNAV-AREAS-S-001-O REST-DYNNAV-IND-AREA-S-001-O REST-DYNNAV-EVENTS-S-002-O REST-DYNNAV-IND-EVENT-S-001-O
REST-DYNNAV-EVENTS-S-002-O	Read all the events or single events	6.11.3	

B.1.12 SCR for REST.DYNNAV.INDIVIDUAL.EVENTS Server

Item	Function	Reference	Requirement
REST-DYNNAV-IND-EVENT-S-001-O	Support for read event whose link are provided in route or area resources	6.12	REST-DYNNAV-IND-EVENT-S-002-O REST-DYNNAV-AREAS-S-001-O REST-DYNNAV-IND-AREA-S-001-O
REST-DYNNAV-IND-EVENT-S-002-O	Read single event	6.12.3	

B.1.13 SCR for REST.DYNNAV.ROUTEPOISLISTS Server

Item	Function	Reference	Requirement
REST-DYNNAV-ROUTEPOISLISTS -S-001-O	Support for management to the lists of POI related to a routes	6.13	
REST-DYNNAV-ROUTEPOISLISTS -S-002-O	Read the set of POI lists related to a route	6.13.3	
REST-DYNNAV-ROUTEPOISLISTS -S-003-O	Create a list of POIs related to a route	6.13.5	

B.1.14 SCR for REST.DYNNAV.INDIVIDUAL.ROUTEPOISLIST Server

Item	Function	Reference	Requirement
REST-DYNNAV- IND-ROUTEPOISLIST -S-001-O	Support for management to the lists of POI related to a routes	6.14	
REST-DYNNAV-IND-ROUTEPOISLIST -S-002-O	Read a single list of POI related to a route	6.14.3	
REST-DYNNAV- IND-ROUTEPOISLIST-S-003-O	Cancel a list of POIs related to a route	6.14.6	

B.1.15 REST.DYNNAV.POISLISTS Server

Item	Function	Reference	Requirement
REST-DYNNAV-POISLISTS -S-001-O	Support for management to the lists of POI related to the described area	6.15	
REST-DYNNAV-POISLISTS -S-002-O	Read the set of POI lists related to the described area	6.15.3	
REST-DYNNAV-POISLISTS -S-003-O	Create a list of POIs related to the described area	6.15.5	

B.1.16 SCR for REST.DYNNAV.INDIVIDUAL.POISLIST Server

Item	Function	Reference	Requirement
REST-DYNNAV- IND-POISLIST -S-001-O	Support for management to the lists of POI related to the described area	6.16	

Item	Function	Reference	Requirement
REST-DYNNAV-IND-POISLIST -S-002-O	Read a single list of POI related to the described area	6.16.3	
REST-DYNNAV- IND-POISLIST-S-003-O	Cancel a list of POIs related to the described area	6.16.6	

Appendix C. Application/x-www-form-urlencoded Request Format for POST Operations (Normative)

In most OMA RESTful Network API specifications, Appendix C defines a format for API requests where the body of the request is encoded using the application/x-www-form-urlencoded MIME type.

In this particular specification, Appendix C has been intentionally left empty.

Appendix D. JSON examples (Informative)

JSON (JavaScript Object Notation) is a light-weight, text-based, language-independent data interchange format. It provides a simple means to represent basic name-value pairs, arrays and objects. JSON is relatively trivial to parse and evaluate using standard JavaScript libraries, and hence is suited for REST invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC 4627].

The following examples show the request and response for various operations using the JSON data format. The examples follow the XML to JSON serialization rules in [REST_NetAPI_Common]. A JSON response can be obtained by using the content type negotiation mechanism specified in [REST_NetAPI_Common].

For full details on the operations themselves please refer to the section number indicated.

D.1 Create a new trip, returning a representation of created resource (section 6.1.5.1)

Request:

```
POST /exampleAPI/dynnav/v1/app0001/trips HTTP/1.1
Accept: application/json
Content-Type: application/json
Host: example.com
Date: Wed, 26 Oct 2011 16:00:00 GMT

{
 "trip": {
  "originWGS84": {
   "WGS84": {
    "longitude": 45.19074,
    "latitude": 7.63441
   }
  },
  "destinationWGS84": {
   "WGS84": {
    "longitude": 45.11451,
    "latitude": 7.64410
   }
  },
  "startingTime": " 2011-10-26T16:10:00 ",
  "vehicleType": {
   "vehicle_type": "rtm01_1"
  },
  "calculateRoute": " true "
 }
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001
Date: Wed, 26 Oct 2011 16:00:10 GMT

{"trip": {
  "originWGS84": {
   "WGS84": {
    "longitude": 45.19074,
    "latitude":  7.63441
   }
  },
  "destinationWGS84": {
   "WGS84": {
    "longitude": 45.11451,
    "latitude": 7.64410
   }
  },
  "startingTime": " 2011-10-26T16:10:00 ",
  "vehicleType": {
   "vehicle_type": "rtm01_1"
  },
  "calculateRoute": " true ",
  "link": [
   {
    "rel": "Route",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"
   }, {
    "rel": "Route",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt02"
   }
  ],
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001 "
 }
}
```

D.2 Regular trip information request (section 6.2.3.1)

Request:

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Wed, 26 Oct 2011 18:20:00 GMT

{"trip": {
  "originWGS84": {
   "WGS84": {
    "longitude": 45.19074,
    "latitude":  7.63441
   }
  },
  "destinationWGS84": {
   "WGS84": {
    "longitude": 45.11451,
    "latitude": 7.64410
   }
  },
  "startingTime": " 2011-10-26T16:10:00 ",
  "vehicleType": {
   "vehicle_type": "rtm01_1"
  },
  "calculateRoute": " true ",
  "link": [
   {
    "rel": "Route",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"
   }, {
    "rel": "Route",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt02"
   }
  ],
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001 "
 }
}
```

D.3 Create a new route, returning a representation of created resource (section 6.3.5.1)

Request:

```
POST /exampleAPI/dynnav/v1/app0001/trips/trip001/routes HTTP/1.1
Accept: application/json
Content-Type: application/json
Host: example.com
Date: Wed, 26 Oct 2011 17:00:00 GMT

{
 "origin": {
  "WGS84": {
   "longitude": 45.19074,
   "latitude": 7.63441
  },
  "location_descriptor": {
   "descriptor_type": "loc03_7",
   "descriptor": "SP2: Strada Provinciale di Germagnano"
   }
  },
  "segment": [
   {
   "endPoint": {
     "WGS84": {
      "longitude": 45.18035,
      "latitude": 7.64982
     },
     "location_descriptor": [
      {
```

```
    "descriptor_type": "loc03_7",
    "descriptor": "SP2: Strada Provinciale di Germagnano"
   },
  {
    "descriptor_type": "loc03_8",
    "descriptor": "RA10: Raccordo autostradale Torino-Caselle"
   }
   ]
 },
 "linkName": " SP2 "
},
{
 "endPoint": {
  "WGS84": {
   "longitude": 45.12864,
   "latitude": 7.69526
  },
  "location_descriptor": [
   {
    "descriptor_type": "loc03_7",
    "descriptor": "RA10: Raccordo autostradale Torino-Caselle"
   },
    {
    "descriptor_type": "loc03_8",
    "descriptor": "1 Autostrade"
   }
  ]
 },
 "linkName": " RA10 "
}, {
 "endPoint": {
  "WGS84": {
   "longitude": 45.13028,
   "latitude": 7.69562
  },
  "location_descriptor": {
   "descriptor_type": "loc03_7",
   "descriptor": "A55: Tangenziale di Torino"
  }
 }
}, {
 "endPoint": {
  "WGS84": {
   "longitude": 45.12080,
   "latitude": 7.64055
  },
  "location_descriptor": [
   {
    "descriptor_type": "loc03_7",
    "descriptor": "A55: Tangenziale di Torino "
   }, {
    "descriptor_type": "loc03_8",
    "descriptor": "Venaria"
   }
  ]
 },
    "linkName": " A55 "
   }, {
    "endPoint": {
     "WGS84": {
      "longitude": 45.12495,
      "latitude": 7.63992
     },
     "location_descriptor": {
      "descriptor_type": "loc03_7",
      "descriptor": "Corso Giuseppe Garibaldi"
     }
    }
   }, {
    "endPoint": {
     "WGS84": {
      "longitude": 45.11451,
      "latitude": 7.64410
     },
     "location_descriptor": [
      {
       "descriptor_type": "loc03_7",
       "descriptor": "Corso Giuseppe Garibaldi"
      }, {
       "descriptor_type": "loc03_8",
       "descriptor": "Via Paganelli"
      }
     ]
    },
    "linkName": " Corso Giuseppe Garibaldi "
   }
  ]}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01
Date: Wed, 26 Oct 2011 17:00:10 GMT

{"route": {
  "xmlns:dynnav": "urn:oma:xml:rest:netapi:dynnav:1",
  "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
  "xsi:schemaLocation": "urn:oma:xml:rest:netapi:dynnav:1 file:///E:/DynNav/exampleReview/OMA_SUP_XSD_rest_DynNav.xsd",
  "travellingTime": " 14 ",
  "distance": " 17.2 ",
  "origin": {
   "WGS84": {
    "longitude": 45.19074,
    "latitude": 7.63441
   },
   "location_descriptor": {
    "descriptor_type": "loc03_7",
    "descriptor": "SP2: Strada Provinciale di Germagnano"
   }
  },
  "segment": [
{
 "endPoint": {
  "WGS84": {
   "longitude": 45.18035,
   "latitude": 7.64982
  },
  "location_descriptor": [
   {
    "descriptor_type": "loc03_7",
    "descriptor": "SP2: Strada Provinciale di Germagnano"
   }, {
    "descriptor_type": "loc03_8",
    "descriptor": "RA10: Raccordo autostradale Torino-Caselle"
   }
  ]
 },
 "linkName": "SP2",
 "distance": 1.7,
 "regularTravellingTime": " 2 "
}, {
 "endPoint": {
  "WGS84": {
   "longitude": 45.12864,
   "latitude": 7.69526
  },
  "location_descriptor": [
   {
    "descriptor_type": "loc03_7",
    "descriptor": "RA10: Raccordo autostradale Torino-Caselle"
   }, {
    "descriptor_type": "loc03_8",
    "descriptor": "1 Autostrade"
   }
  ]
 },
 "linkName": "RA10",
 "distance": 7.6,
 "regularTravellingTime": " 4 "
}, {
 "endPoint": {
  "WGS84": {
   "longitude": 45.13028,
   "latitude": 7.69562
  },
  "location_descriptor": {
   "descriptor_type": "loc03_7",
   "descriptor": "A55: Tangenziale di Torino"
  }
 },
 "distance": 1,
 "regularTravellingTime": " 2 "
}, {
 "endPoint": {
  "WGS84": {
   "longitude": 45.12080,
   "latitude": 7.64055
```

```
     },
     "location_descriptor": [
      {
       "descriptor_type": "loc03_7",
       "descriptor": "A55: Tangenziale di Torino "
      }, {
       "descriptor_type": "loc03_8",
       "descriptor": "Venaria"
      }
     ]
    },
    "linkName": "A55",
    "distance": 5.1,
    "regularTravellingTime": " 4 ",
    "performanceParameters": {
     "trafficInfoType": "Real-time",
     "time": " 2011-11-23T16:00:00  ",
     "delay": " 1 ",
     "speed": " 22 ",
     "performance": "rtm34_4"
    }
   }, {
    "endPoint": {
     "WGS84": {
      "longitude": 45.12495,
      "latitude": 7.63992
     },
     "location_descriptor": {
      "descriptor_type": "loc03_7",
      "descriptor": "Corso Giuseppe Garibaldi"
     }
    },
    "distance": 0.7,
    "regularTravellingTime": " 1 "
   }, {
    "endPoint": {
     "WGS84": {
      "longitude": 45.11451,
      "latitude": 7.64410
     },
     "location_descriptor": [
      {
       "descriptor_type": "loc03_7",
       "descriptor": "Corso Giuseppe Garibaldi"
      }, {
       "descriptor_type": "loc03_8",
       "descriptor": "Via Paganelli"
      }
     ]
    },
    "linkName": " Corso Giuseppe Garibaldi ",
    "distance": 1.2,
    "regularTravellingTime": " 2 "
   }
  ],
  "trafficEvents": {
   "category": "rtm00_8",
   "link": {
    "rel": "Event",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004"
   }
  },
  "resurceURL": " http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01"}}
```

D.4 Regular route information request (section 6.4.3.2)

Request:

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01?shapeReq=true HTTP/1.1
Accept: application/json
Host: example.com]
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Wed, 19 Oct 2011 16:30:00 GMT

{"route": {
  "xmlns:dynnav": "urn:oma:xml:rest:netapi:dynnav:1",
  "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
  "xsi:schemaLocation": "urn:oma:xml:rest:netapi:dynnav:1 file:///E:/DynNav/exampleReview/OMA_SUP_XSD_rest_DynNav.xsd",
  "travellingTime": " 14 ",
  "distance": " 17.2 ",
  "origin": {
   "WGS84": {
    "longitude": 45.19074,
    "latitude":  7.63441
   },
   "location_descriptor": {
    "descriptor_type": "loc03_7",
    "descriptor": "SP2: Strada Provinciale di Germagnano"
   }
  },
  "segment": [
   {
    "endPoint": {
     "WGS84": {
      "longitude": 45.18035,
      "latitude": 7.64982
     },
     "location_descriptor": [
      {
       "descriptor_type": "loc03_7",
       "descriptor": "SP2: Strada Provinciale di Germagnano"
      }, {
       "descriptor_type": "loc03_8",
       "descriptor": "RA10: Raccordo autostradale Torino-Caselle"
      }
     ]
    },
    "polyLine": "45.19075 7.63269, 45.190751 7.632691, 45.190752 7.632692, 45.190753 7.632693, 45.190751 7.632694",
    "linkName": "SP2",
    "distance": 1.7,
    "regularTravellingTime": " 2 "
   }, {
    "endPoint": {
     "WGS84": {
      "longitude": 45.12864,
      "latitude": 7.69526
     },
     "location_descriptor": [
      {
       "descriptor_type": "loc03_7",
       "descriptor": "RA10: Raccordo autostradale Torino-Caselle"
      }, {
       "descriptor_type": "loc03_8",
       "descriptor": "1 Autostrade"
      }
     ]
    },
    "polyLine": "45.12075 7.63269, 45.120751 7.632691, 45.120752 7.632692, 45.120753 7.632693, 45.190754 7.632694, ",
    "linkName": "RA10",
    "distance": 7.6,
    "regularTravellingTime": " 4 "
   }, {
    "endPoint": {
     "WGS84": {
      "longitude": 45.13028,
      "latitude": 7.69562
     },
     "location_descriptor": {
      "descriptor_type": "loc03_7",
      "descriptor": "A55: Tangenziale di Torino"
     }
    },
    "polyLine": "….",
    "distance": 1,
    "regularTravellingTime": " 2 "
   }, {
    "endPoint": {
     "WGS84": {
      "longitude": 45.12080,
      "latitude": 7.64055
     },
     "location_descriptor": [
      {
       "descriptor_type": "loc03_7",
       "descriptor": "A55: Tangenziale di Torino "
      }, {
       "descriptor_type": "loc03_8",
       "descriptor": "Venaria"
      }
     ]
    },
    "polyLine": "….",
    "linkName": "A55",
    "distance": 5.1,
    "regularTravellingTime": " 4 ",
    "performanceParameters": {
     "trafficInfoType": "Real-time",
     "delay": " 2 ",
     "speed": " 22 ",
     "performance": "rtm34_4"
    }
   }, {
    "endPoint": {
     "WGS84": {
      "longitude": 45.12495,
      "latitude": 7.63992
     },
     "location_descriptor": {
      "descriptor_type": "loc03_7",
      "descriptor": "Corso Giuseppe Garibaldi"
     }
    },
    "polyLine": "….",
    "distance": 0.7,
    "regularTravellingTime": " 1 "
   }, {
    "endPoint": {
     "WGS84": {
      "longitude": 45.11451,
      "latitude": 7.64410
     },
     "location_descriptor": [
      {
       "descriptor_type": "loc03_7",
       "descriptor": "Corso Giuseppe Garibaldi"
      }, {
       "descriptor_type": "loc03_8",
       "descriptor": "Via Paganelli"
      }
     ]
    },
    "polyLine": "….",
    "linkName": " Corso Giuseppe Garibaldi ",
    "distance": 1.2,
    "regularTravellingTime": " 2 "
   }
  ],
  "trafficEvents": {
   "category": "rtm00_8",
   "link": {
    "rel": "Event",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004"
   }
  },
  "resourceURL": " http://example.com /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01 "
 }
```

```
}
```

D.5 Regular summarized route information request (section 6.5.3.1)

Request:

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01 HTTP/1.1
Accept: application/json
Host: example.com]
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Wed, 19 Oct 2011 16:30:00 GMT

{"route": {
 "travellingTime": " 15 ",
 "distance": " 17.2 ",
 "origin": {
  "WGS84": {
   "longitude": 45.19074,
   "latitude":  7.63441
  },
  "location_descriptor": {
   "descriptor_type": "loc03_7",
   "descriptor": "SP2: Strada Provinciale di Germagnano"
  }
 },
 "segment": [
  {
   "endPoint": {
    "WGS84": {
     "longitude": 45.13028,
     "latitude": 7.69562
    },
    "location_descriptor": {
     "descriptor_type": "loc03_7",
     "descriptor": "A55: Tangenziale di Torino"
    }
   },
   "distance": 10.2,
   "regularTravellingTime": " 8 "
  }, {
   "endPoint": {
    "WGS84": {
     "longitude": 45.12080,
     "latitude": 7.64055
    },
    "location_descriptor": [
     {
      "descriptor_type": "loc03_7",
      "descriptor": "A55: Tangenziale di Torino "
     }, {
      "descriptor_type": "loc03_8",
      "descriptor": "Venaria"
     }
    ]
   },
   "linkName": "A55",
   "distance": 7,
   "regularTravellingTime":  7,
   "performanceParameters": {
    "trafficInfoType": "Real-time",
    "delay": " 4 ",
    "speed": " 22 ",
    "performance": "rtm34_4"
   }
  }}
```

D.6 Create a new area, returning a representation of created resource (section 6.6.5.1)

Request:

```
POST /exampleAPI/dynnav/v1/app0001/areas HTTP/1.1
Accept: application/json
Content-Type: application/json
Host: example.com
Date: Wed, 23 Nov 2011 15:00:12 GMT

{"area": {
  "areaDesc": {
   "language": " loc41_30",
   "location_descriptions": {
    "area_reference": {
     "country": "loc40_106",
     "area_tree_version": 1,
     "area_tree_entry": {
      "level": 1,
      "branch": 1,
      "predecessor_branch": 0,
      "area_type": {

       "loc06_8"
      ]
     },
     "area_descriptor": {
      "area_name": "Torino"
     }
    }
   }
  }
 },
 "startValidityTime": " 2011-11-23T16:00:00  ",
 "endValidityTime": " 2011-11-23T20:00:00  ",
 "requestedEventsCategories": [
  "rtm00_8", "rtm00_5"
 ],
 "timeResolution": " 60 "
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/dynnav/v1 v1/app0001/areas/area001
Date: Wed, 23 Nov 2011 15:00:15 GMT

{"area": {
  "areaDesc": {
   "language": " loc41_30",
   "location_descriptions": {
    "area_reference": {
     "country": "loc40_106",
     "area_tree_version": 1,
     "area_tree_entry": {
      "level": 1,
      "branch": 1,
      "predecessor_branch": 0,
      "area_type": {

       "loc06_8"
      ]
     },
     "area_descriptor": {
      "area_name": "Torino"
     }
    }
   }
  }
 },
 "startValidityTime": " 2011-11-23T16:00:00  ",
 "endValidityTime": " 2011-11-23T20:00:00  ",
 "requestedEventsCategories": [
  "rtm00_8", "rtm00_5"
 ],
 "timeResolution": " 60 ",
 "events": {
  "category": "rtm00_8",
  "link": [
   {
    "rel": "Event",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/events/evt002"
   }, {
    "rel": "Event",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004"
   }, {
    "rel": "Event",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/events/evt005"
   }
  ]
 },
 "segmentPerformance": {
  "originPoint": {
   "WGS84": {
    "longitude": 45.14048,
    "latitude":  7.65575
   },
   "location_descriptor": {
    "descriptor_type": "loc03_7",
    "descriptor": "SP2: Strada Provinciale di Germagnano"
   }
  },
  "endPoint": {
   "WGS84": {
    "longitude": 45.13028,
    "latitude": 7.65778
   },
   "location_descriptor": [
    {
     "descriptor_type": "loc03_7",
     "descriptor": "SP2: Strada Provinciale di Germagnano"
    }, {
     "descriptor_type": "loc03_8",
     "descriptor": "A55 Tangenziale di Torino"
    }
   ]
  },
  "linkName": "SP2",
  "distance": 1.1,
  "regularTravellingTime": " 2 ",
  "performanceParameters": [
   {
    "trafficInfoType": "Real-time",
    "time": " 2011-11-23T15:45:00 ",
    "delay": " 1.2 ",
    "speed": " 22 ",
    "performance": "rtm34_4"
   }, {
    "trafficInfoType": "Forecast",
    "time": " 2011-11-23T17:00:00 ",
    "delay": " 0.8 ",
    "speed": " 42 ",
    "performance": "rtm34_2"
   }, {
    "trafficInfoType": "Forecast",
    "time": " 2011-11-23T18:00:00 ",
    "delay": " 0.2 ",
    "speed": " 70 "
   }, {
    "trafficInfoType": "Forecast",
    "time": " 2011-11-23T19:00:00 ",
    "delay": " 0.3 ",
    "speed": " 66 "
   }, {
    "trafficInfoType": "Forecast",
    "time": " 2011-11-23T20:00:00 ",
    "delay": " 0.1 ",
    "speed":  80
   }
  ]
 },
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001 "
}}
```

D.7 Read events and performance parameters related to an area (section 6.7.3.1)

Request:

```
GET /exampleAPI/dynnav/v1/app0001/areas/area001 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Wed, 23 Nov 2011 15:33:00 GMT



{"area": {
  "areaDesc": {
   "language": " loc41_30",
   "location_descriptions": {
    "area_reference": {
     "country": "loc40_106",
     "area_tree_version": 1,
     "area_tree_entry": {
      "level": 1,
      "branch": 1,
      "predecessor_branch": 0,
      "area_type": {

       "loc06_8"
      ]
     },
     "area_descriptor": {
      "area_name": "Torino"
     }
    }
   }
  }
 },
 "startValidityTime": " 2011-11-23T16:00:00  ",
 "endValidityTime": " 2011-11-23T20:00:00  ",
 "requestedEventsCategories": [
  "rtm00_8", "rtm00_5"
 ],
 "timeResolution": " 60 ",
 "events": {
  "category": "rtm00_8",
  "link": [
   {
    "rel": "Event",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/events/evt002"
   }, {
    "rel": "Event",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/events/evt004"
   }, {
    "rel": "Event",
    "href": "http://example.com/exampleAPI/dynnav/v1/app0001/events/evt005"
   }
  ]
 },
 "segmentPerformance": {
  "originPoint": {
   "WGS84": {
    "longitude": 45.14048,
    "latitude":  7.65575
   },
   "location_descriptor": {
    "descriptor_type": "loc03_7",
    "descriptor": "SP2: Strada Provinciale di Germagnano"
   }
  },
  "endPoint": {
   "WGS84": {
    "longitude": 45.13028,
    "latitude": 7.65778
   },
   "location_descriptor": [
    {
     "descriptor_type": "loc03_7",
     "descriptor": "SP2: Strada Provinciale di Germagnano"
    }, {
     "descriptor_type": "loc03_8",
     "descriptor": "A55 Tangenziale di Torino"
    }
   ]
  },
  "linkName": "SP2",
  "distance": 1.1,
  "regularTravellingTime": " 2 ",
  "performanceParameters": [
   {
    "trafficInfoType": "Real-time",
    "time": " 2011-11-23T15:45:00 ",
    "delay": " 1.2 ",
    "speed": " 22 ",
    "performance": "rtm34_4"
   }, {
    "trafficInfoType": "Forecast",
    "time": " 2011-11-23T17:00:00 ",
    "delay": " 0.8 ",
    "speed": " 42 ",
    "performance": "rtm34_2"
   }, {
    "trafficInfoType": "Forecast",
    "time": " 2011-11-23T18:00:00 ",
    "delay": " 0.2 ",
    "speed": " 70 "
   }, {
    "trafficInfoType": "Forecast",
    "time": " 2011-11-23T19:00:00 ",
    "delay": " 0.3 ",
    "speed": " 66 "
   }, {
    "trafficInfoType": "Forecast",
    "time": " 2011-11-23T20:00:00 ",
    "delay": " 0.1 ",
    "speed":  80
   }
  ]
 },
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/areas/area001 "
}}
```

D.8 Create a new subscription, returning a representation of created resource (section 6.8.5.1)

Request:

```
POST /exampleAPI/dynnav/v1/app0001/subscriptions HTTP/1.1
Accept: application/json
Content-Type: application/json
Host: example.com
Date: Wed, 23 Oct 2011 17:45:00 GMT

{"subscription": {
  "callbackReference": {
   "notifyURL": "http://application.example.com/notifications/DynNavNotification"
  },
  "link": {
   "rel": "Trip",
   "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"
  },
  "trackingProc": " true ",
  "deviceLocationURI": " http://locationserver.example.com/hu4u43b780c "}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
```

```
Location: http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001
Date: Wed, 23 Oct 2011 17:45:05 GMT

{"subscription": {
 "callbackReference": {
  "notifyURL": "http://application.example.com/notifications/DynNavNotification"
 },
 "link": {
  "rel": "Trip",
  "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"
 },
 "trackingProc": " true ",
 "deviceLocationURI": " http://locationserver.example.com/hu4u43b780c "
}}
```

D.9 Modify subscription settings (section 6.9.4.1)

Request:

```
PUT /exampleAPI/dynnav/v1/app0001/subscriptions/sub001 HTTP/1.1
Accept: application/json
Host: example.com

{"subscription": {
 "callbackReference": {
  "notifyURL": "http://application.example.com/notifications/DynNavNotification"
 },
 "link": [
  {
   "rel": "Trip",
   "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"
  }, {
   "rel": "Area",
   "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/area001"
  }
 ],
 "trackingProc": " true ",
 "deviceLocationURI": " http://locationserver.example.com/hu4u43b780c ",
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001 "
}}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Wed, 23 Nov 2011 16:13:00 GMT

{"subscription": {
 "callbackReference": {
  "notifyURL": "http://application.example.com/notifications/DynNavNotification"
 },
 "link": [
  {
   "rel": "Trip",
   "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001"
  }, {
   "rel": "Area",
   "href": "http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/area001"
  }
 ],
 "trackingProc": " true ",
 "deviceLocationURI": " http://locationserver.example.com/hu4u43b780c ",
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001 "}}
```

D.10 Notification of available updates (section 6.10.5.1)

Request:

```
POST /notifications/DynNavNotification HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: application.example.com

{"notification":
  "callbackReference": {
   "notifyURL": "http://application.example.com/notifications/DynNavNotification"
  },
  "link": [
   {
    "rel": "Trip",
    "href": "http://example.com/exampleAPI/dynnav/v1app0001/trips/trip001"
   }, {
    "rel": "Area",
    "href": "http://example.com/exampleAPI/dynnav/v1app0001/trips/trip001/area001"
   }
  ],
  "trackingProc": " true ",
  "deviceLocationURI": " http://locationserver.example.com/hu4u43b780c ",
  "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/subscriptions/sub001 "
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.11 Retrieve all events (section 6.11.3.1)

Request:

```
GET /exampleAPI/dynnav/v1/app0001/events HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Wed, 16 Nov 2011 16:13:00 GMT

{"eventList": {
 "event": [
  {
   "rtMessage": {
    "message_id": 124,
    "version_number": 1,
    "message_generation_time": "2002-04-03T13:40:00Z",
    "severity_factor": "rtm31_2",
    "Comment": " Location is A811 at Drymen ",
    "location_container": {
     "language": "loc41_30",
     "location_coordinates": {
      "location_type": "loc01_5",
      "location_point": {
       "WGS84": {
        "longitude": -4.45451,
        "latitude": 56.05573
       },
       "location_descriptor": [
        {
         "descriptor_type": "loc03_7",
         "descriptor": "A811"
        }, {
         "descriptor_type": "loc03_8",
         "descriptor": "A809"
        }, {
         "descriptor_type": "loc03_24",
         "descriptor": "Dumbarton"
        }, {
         "descriptor_type": "loc03_24",
         "descriptor": "Stirling"
        }
       ]
      }
     }
    },
    "Comment": " Temporary traffic lights ",
    "facilities_performance": {
     "traffic_control": {
      "traffic_control_type": "rtm42_11",
      "traffic_control_status": "rtm43_12",
      "position": {

       "rtm10_37"
      ]
     }
    }
   }
  },
  "resourceURL": " http://example.com /exampleAPI/dynnav/v1/app0001/events/evt001 "
 }, {
```

```
 "rtMessage": {
  "message_id": 123,
  "version_number": 1,
  "message_generation_time": "2002-04-03T13:03:00Z",
  "severity_factor": "rtm31_4",
  "Comment": " Location is on A12 in Brentford, Essex ",
  "location_container": {
   "language": "loc41_30",
   "location_coordinates": {
    "location_type": "loc01_5",
    "location_point": {
     "WGS84": {
      "longitude": -0.1337,
      "latitude": 51.52641
     },
     "location_descriptor": [
      {
       "descriptor_type": "loc03_7",
       "descriptor": "A12"
      }, {
       "descriptor_type": "loc03_8",
       "descriptor": "A128"
      }, {
       "descriptor_type": "loc03_24",
       "descriptor": "Brentwood"
      }, {
       "descriptor_type": "loc03_25",
       "descriptor": "Essex"
      }
     ]
    },
    "direction": {
     "direction_type": "loc02_2"
    }
   }
  },
  "Comment": " Accident in thick fog involving 50 vehicles ",
  "accidents": {
   "number_of": 1,
   "position": {

    "rtm10_37"
   ]
  },
  "vehicles": {
   "number_of": 50,
   "vehicle_problem": {

    "rtm03_22"
   ]
  }
 }
},
"visibility": {
 "obscurity": {
  "obscurity_problem": "rtm17_2",
    "visibility_distance": 20
   }
  },
  "network_conditions": {
   "position": {

    "rtm10_37"
   ]
  },
  "restriction": {

   "rtm49_1"
  ]
 }
}
}
}
],
"resourceURL": " http://example.com /exampleAPI/dynnav/v1/app0001/events "
}}
```

D.12 Retrieve a traffic event (section 6.12.3.1)

Request:

```
GET /exampleAPI/dynnav/v1/app0001/events/evt002 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Wed, 16 Nov 2011 16:11:00 GMT

{"event": {
  "rtMessage": {
   "message_id": 123,
   "version_number": 1,
   "message_generation_time": "2002-04-03T13:03:00Z",
   "severity_factor": "rtm31_4",
   "Comment": " Location is on A12 in Brentford, Essex ",
   "location_container": {
    "language": "loc41_30",
    "location_coordinates": {
     "location_type": "loc01_5",
     "location_point": {
      "WGS84": {
       "longitude": -0.1337,
       "latitude": 51.52641
      },
      "location_descriptor": [
       {
        "descriptor_type": "loc03_7",
        "descriptor": "A12"
       }, {
        "descriptor_type": "loc03_8",
        "descriptor": "A128"
       }, {
        "descriptor_type": "loc03_24",
        "descriptor": "Brentwood"
       }, {
        "descriptor_type": "loc03_25",
        "descriptor": "Essex"
       }
      ]
     },
     "direction": {
      "direction_type": "loc02_2"
     }
    }
   },
   "Comment": " Accident in thick fog involving 50 vehicles ",
   "accidents": {
    "number_of": 1,
    "position": {

     "rtm10_37"
    ]
   },
   "vehicles": {
    "number_of": 50,
    "vehicle_problem": {

     "rtm03_22"
    ]
   }
  }
 },
 "visibility": {
  "obscurity": {
   "obscurity_problem": "rtm17_2",
   "visibility_distance": 20
  }
 },
 "network_conditions": {
  "position": {

   "rtm10_37"
  ]
 },
 "restriction": {

  "rtm49_1"
 ]
}
}
},
"resourceURL": " http://example.com /exampleAPI/dynnav/v1/app0001/events/evt002 "
```

```
}}
```

D.13 Create a new list of POIs related to a route, returning a representation of created resource (section 6.13.5.1)

Request:

```
POST /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/ HTTP/1.1
Accept: application/json
Content-Type: application/json
Host: example.com
Date: Wed, 23 Nov 2011 18:00:12 GMT

{"poiList": {
    "category": " urn:service:fuel ",
    "routeRange": " 0 "
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01
Date: Wed, 23 Nov 2011 18:00:12 GMT

{"poiList": {
  "category": " urn:service:fuel",
  "routeRange": " 0 ",
  "pois": {
   "poi:poi": {
    "id": "http://www.dynnavExamples.org/pois/45343489",
    "poi:label": {
     "term": "XX Petrol Sattion"
    },
    "poi:description": {
     "poi:value": "XX Petrol Station, Food and Drinks"
    },
    "poi:category": {
     "term": "service:fuel",
     "scheme": "http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03",
     "poi:value": "seat of a first-order administrative division"
    },
    "poi:link": {
     "term": "related",
     "href": "http://www.geonames.org/maps/google_7.678992_45.062735.html",
     "type": "text/html",
     "scheme": "http://www.iana.org/assignments/link-relations/link-relations.xml"
    },
    "poi:location": {
     "poi:point": {
      "term": "centroid",
      "poi:Point": {
       "srsName": "http://www.opengis.net/def/crs/EPSG/0/4326",
       "poi:posList": "7.678992 45.062735"
     }
    }
   }
  }
 },
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01"}}
```

D.14 Read the POIs of a list related to a route (section 6.14.3.1)

Request:

```
GET /exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Wed, 30 Nov 2011 17:00:10 GMT

{"poiList": {
 "category": " urn:service:fuel",
 "routeRange": " 0 ",
 "pois": {
  "poi:poi": {
   "id": "http://www.dynnavExamples.org/pois/45343489",
   "poi:label": {
    "term": "XX Petrol Sattion"
   },
   "poi:description": {
    "poi:value": "XX Petrol Station, Food and Drinks"
   },
   "poi:category": {
    "term": "service:fuel",
    "scheme": "http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03",
    "poi:value": "seat of a first-order administrative division"
   },
   "poi:link": {
    "term": "related",
    "href": "http://www.geonames.org/maps/google_7.678992_45.062735.html",
    "type": "text/html",
    "scheme": "http://www.iana.org/assignments/link-relations/link-relations.xml"
   },
   "poi:location": {
    "poi:point": {
     "term": "centroid",
     "poi:Point": {
      "srsName": "http://www.opengis.net/def/crs/EPSG/0/4326",
      "poi:posList": "7.678992 45.062735"
     }
    }
   }
  }
 },
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01"    }},
   "resurceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01",
   "routeRange": " 0 "
}}
```

D.15 Create a new list of POIs related to an area (section 6.15.5.1)

Request:

```
POST /exampleAPI/dynnav/v1/app0001/poiAreas HTTP/1.1
Accept: application/json
Content-Type: application/json
Host: example.com
Date: Wed, 23 Nov 2011 18:00:12 GMT

{"poiList": {
  "category": " urn:service:fuel",
  "area": {
   "language": "loc41_30",
   "location_coordinates": {
    "location_type": "loc01_1",
    "location_point": {
     "WGS84": {
      "longitude": 7.688992,
      "latitude": 45.062935,
      "expansion": {
       "radius_of_circle": 1000
      }
     }
    }
   }
  }
 }}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/dynnav/v1/app0001/poiAreas/poiL01
Date: Wed, 23 Nov 2011 18:00:12 GMT

{"poiList": {
  "category": " urn:service:fuel",
  "area": {
   "language": "loc41_30",
   "location_coordinates": {
    "location_type": "loc01_1",
    "location_point": {
     "WGS84": {
      "longitude": 7.688992,
      "latitude": 45.062935,
      "expansion": {
       "radius_of_circle": 1000
      }
     }
    }
  }
 },
 "pois": {
  "poi:poi": {
   "id": "http://www.dynnavExamples.org/pois/45343489",
   "poi:label": {
    "term": "XX Petrol Sattion"
   },
   "poi:description": {
    "poi:value": "XX Petrol Station, Food and Drinks"
   },
   "poi:category": {
    "term": "service:fuel",
    "scheme": "http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03",
    "Text": "\r\t\t\r\t"
   },
   "poi:link": {
    "term": "related",
    "href": "http://www.geonames.org/maps/google_7.678992_45.062735.html",
    "type": "text/html",
    "scheme": "http://www.iana.org/assignments/link-relations/link-relations.xml"
   },
   "poi:location": {
    "poi:point": {
     "term": "centroid",
     "poi:Point": {
      "srsName": "http://www.opengis.net/def/crs/EPSG/0/4326",
      "poi:posList": "7.678992 45.062735"
     }
    }
   }
  }
 },
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01"}}
```

D.16 Read the POIs of a list related to an area (section 6.16.3.1)

Request:

```
GET /exampleAPI/dynnav/v1/app0001/poiAreas/poiL01 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Wed, 30 Nov 2011 17:56:10 GMT

{"poiList": {
 "category": " urn:service:fuel",
 "area": {
  "language": "loc41_30",
  "location_coordinates": {
   "location_type": "loc01_1",
   "location_point": {
    "WGS84": {
     "longitude": 7.688992,
     "latitude": 45.062935,
     "expansion": {
      "radius_of_circle": 1000
     }
    }
   }
  }
 },
 "pois": {
  "poi:poi": {
   "id": "http://www.dynnavExamples.org/pois/45343489",
   "poi:label": {
    "term": "XX Petrol Sattion"
   },
   "poi:description": {
    "poi:value": "XX Petrol Station, Food and Drinks"
   },
   "poi:category": {
    "term": "service:fuel",
    "scheme": "http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03",
    "Text": "\r\t\t\r\t"
   },
   "poi:link": {
    "term": "related",
    "href": "http://www.geonames.org/maps/google_7.678992_45.062735.html",
    "type": "text/html",
    "scheme": "http://www.iana.org/assignments/link-relations/link-relations.xml"
   },
   "poi:location": {
    "poi:point": {
     "term": "centroid",
     "poi:Point": {
      "srsName": "http://www.opengis.net/def/crs/EPSG/0/4326",
      "poi:posList": "7.678992 45.062735"
     }
    }
   }
  }
 },
 "resourceURL": " http://example.com/exampleAPI/dynnav/v1/app0001/trips/trip001/routes/rt01/pois/poiL01"}}
```

Appendix E. Operations mapping (Informative)

As this specification does not have a baseline specification, this appendix is empty.

Appendix F. Light-weight resources (Informative)

As this version of the specification does not define any light-weight resources, this Appendix is empty.

Appendix G. Authorization aspects (Normative)

None specified in this version of the specification.

Appendix H. Partial Route Encoding Schema

The section provides an overview of the partial route encoding schema. Partial route encoding schema is used to access in an efficient way to traffic information related to a set of route that share common segments. Note that the use of partial route information is limited to Smart ND scenario where the application uploads on the server estimated routes, see section 5.3.2.

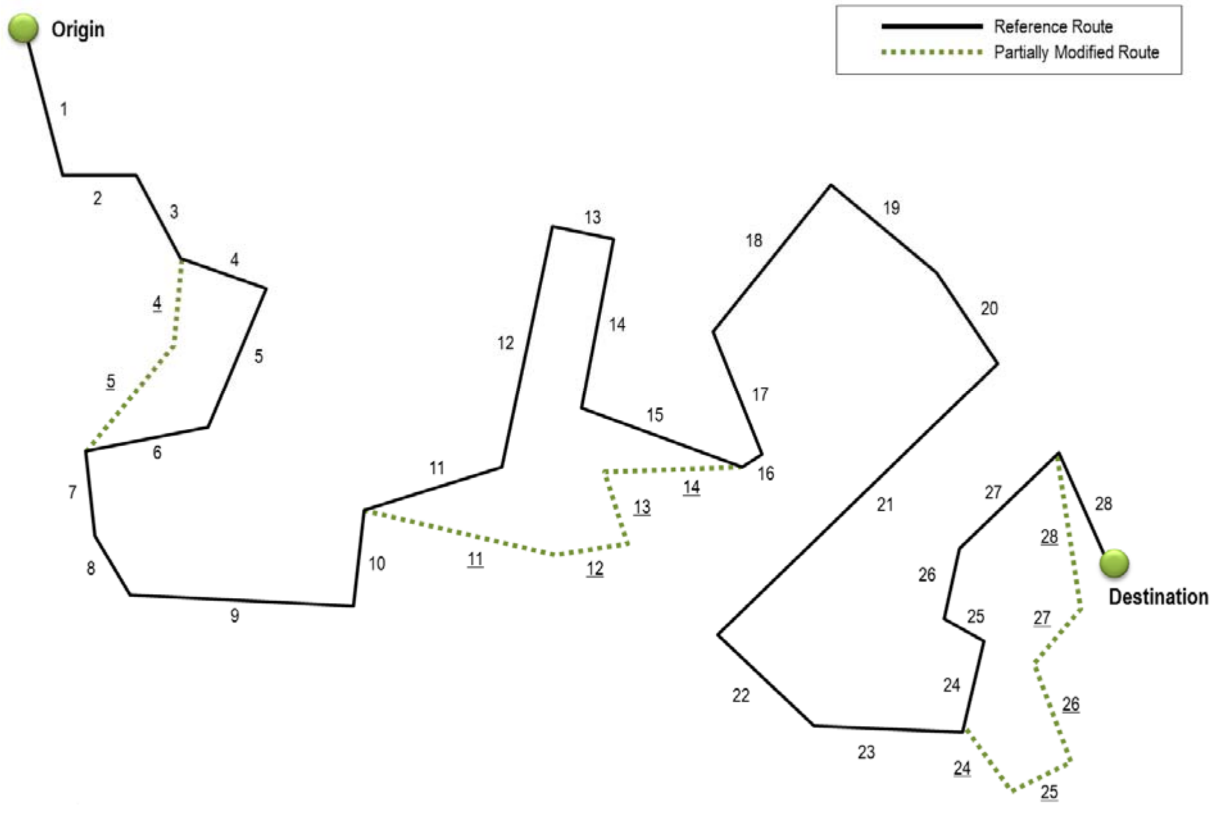

Figure 5 Example of Partial Route Description

As showed in Figure 5, the application can choose to provide as route information only the sequence of segments that is changed compared to a reference route, previously uploaded on the server. In the example of Figure 5, the sequence of segments between 4-5, 11-13 and 24-28 is modified compared to the reference route. Multiple deviations (in the figure, there are 3 deviations) may be included in partial route description. Information to merge the partial route with the reference route is provided in the partial route resource: in details the following parameters defined in the *route* structure are used:

- the *firstSegment* carries the information of the index of the first changed segment in the reference route information for each single deviation of the partial route (The 4th, 11st and 24th segments in the reference route in the Figure 5);
- the *lastSegment* carries the information of the index of the last changed segment in the reference route information for each single deviation of the partial route (The 6th, 15th and 27th segments in the reference route of Figure 5);
- the *numSegment* carries the information about the number of segments that constitutes each single deviation (2,4 and 5 in the example of Figure 5, respectively);

If a reference route is removed from the server using the DELETE method, a partial route resource that refers to the reference route should be encoded with complete sequence of segments. In order to keep the consistency, this procedure is automatically operated by the server.

In case the performance parameters provided by the server for uploaded partial route are not better than those of a route already submitted, the application may choose to remove the last uploaded partial route information using the DELETE operation: the application will iteratively re-estimate and upload alternative routes to find the best one with respect to reported traffic conditions. However, in order to minimize the throughput over the wireless connection and avoid inefficient loops, the application can request traffic information in the area where the re-estimation is occurred, as described in step 9 of chap. 5.3.2.

Appendix I. Updated Route Information to 3rd Party

This section describes the procedure for providing updated trip and/or route information in case the destination is defined using the 3rd party ID in the lightweight ND, following the movement of the 3rd party, whose position is tracked by the DynNav server using an external location application. Whenever the DynNav application uploads its position, the DynNav server acquires the 3rd party's position and checks whether or not the 3rd party's position is changed. If the 3rd party's position is changed, the DynNav server will creates the updated information based on the uploaded DynNav application's position and/or the changed 3rd party's position. The route to the 3rd party scenario is described in the sequence diagram of the section 5.3.1.

According to the changed 3rd party's position, the updated information can be of type: (1) updated destination, (2) additional segment, (3) alternative route. The detailed descriptions are provided in sections below.

(1) Updated destination

In case the changed 3rd party's position is still inside the last segment of the ongoing route, the application is notified of updated destination information. In this case, the route information is unchanged.

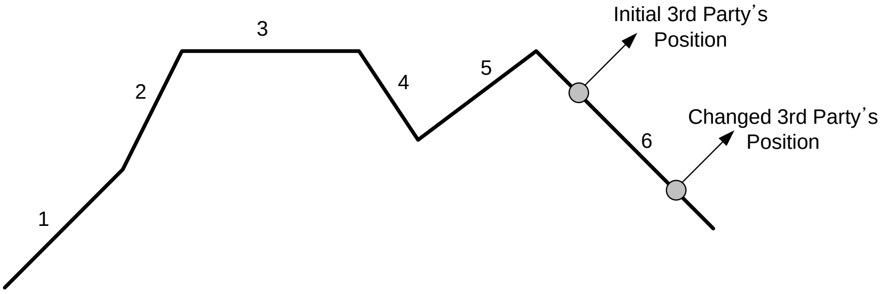

Figure 6 Example of Updated Destination

In the Figure 6, the changed 3rd party's position is inside the last segment of the ongoing route (Segment 6). In this case, the DynNav server notifies the application of the updated destination, and the application accesses to the updated destination (the changed 3rd party's position) in the Trip resource.

(2) Additional segment

In case the additional segment is needed to reach the changed 3rd party's position after the user of application reaches the destination of the ongoing route, the application is notified of the updated destination information and the related route information.

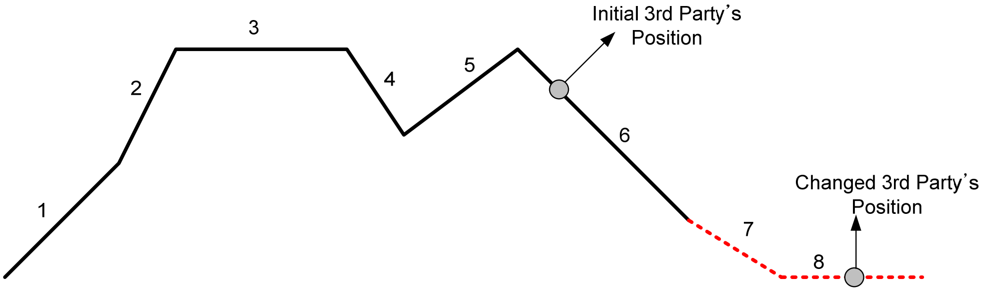

Figure 7 Example of Additional Segment

In the Figure 7, the changed 3rd party's position is outside the ongoing route (Segment 1~6), and additional segments are needed to reach the 3rd party (Segment 7 and 8). In this case, the DynNav server notifies the application of the additional segments, and the application accesses to the updated destination (the changed 3rd party's position) in the Trip resource and the additional segments in the route resource. The additional segments are added to the ongoing route and the updated route and trip resources are notified to the application.

(3) Alternative route

In case the DynNav server calculates an alternative route based on the ongoing position of the application user and the changed 3rd party position, the application is notified of updated destination information and related route information.

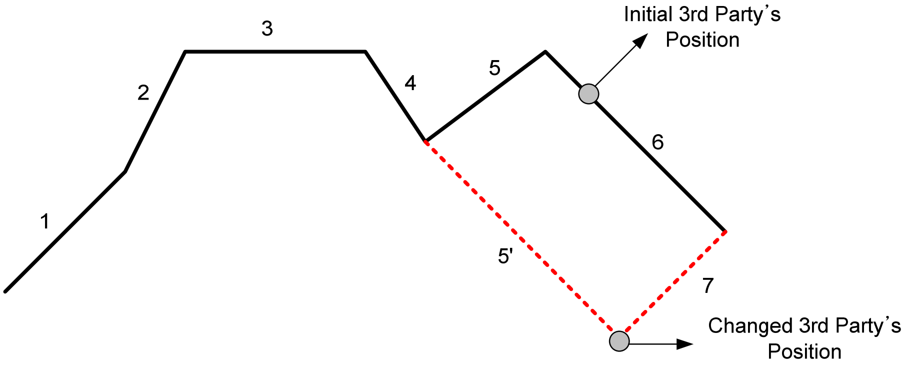

Figure 8 Example of Alternative Route

In the Figure 8, (in this example, it is assumed that the user of application does not enter the segment 5 yet.) the changed 3rd party's position is outside the ongoing route (Segment 1~6), and additional segment is needed to reach the 3rd party (Segment 7). However the DynNav server also calculates the alternative route (segment 5) to the changed 3rd party's position. In this case, the DynNav server notifies the application of the alternative route: the application accesses to the updated destination (the changed 3rd party's position) in the Trip resource and read the new alternative route.

Different notification policies can be implemented for the each updated information type described above: in case of alternative route (scenario 3), the DynNav server notifies the application of the updated information as soon as it is available. In case of updated destination (scenario 1) and additional segment (scenario 2), in order to decrease the number of interaction between the DynNav server and the application, the DynNav server notifies the application of the updated information when the application is in the vicinity of the destination. To detect when the user of application arrives in the vicinity of the destination, the DynNav server set the positionUpdate element in the Segment structure. (For example, to detect the user of application arrives in the vicinity of the destination; the DynNav server set the positionUpdate element in the segment 5 or segment 6 of the figure 6.) Where to request a position update procedure using positionUpdate element is implementation dependent. The user position information uploaded by the DynNav application is used by the DynNav server to check whether the user is in the vicinity of the destination.