ICT-2011.8
GET Service Project
2012-318275

PC-Based aggregated transportation planning and control services

12 Dec 2015

The GET Service project (http://www.getservice-project.eu) has received funding from the European Commission under the 7th Framework Programme (FP7) for Research and Technological Development under grant agreement n°2012-318275.

15 December 2015

Project acronym:	GET Service
Project full title:	Service Platform for Green European Transportation
Work package:	3
Document number:	D3.1
Document title:	PC-Based Aggregated Routing Planning and Control Services
Version:	2
Delivery date:	31 July 2014
Actual publication date:	30 Sept 2014, revision 17.12.2015
Dissemination level:	public
Nature:	Prototype
Editor(s) / lead beneficiary:	PTV
Authors(s):	Michael SCHYGULLA, Alexander STUPP
Reviewers:	Wolfgang Burgholzer, Shaya Pourmirza

History

Version	Changes	Authors
0.1	ToC Released	A. Raptopoulos
0.2	Content added	M. Schygulla
0.3	updated	M. Schygulla
0.4	Final draft	M. Schygulla
0.5	Internal Review comments processed	M. Schygulla
0.6	Y2 review comments processed	M. Schygulla
1.0	Final review comments processed	M. Schygulla

Executive Summary

This deliverable presents the PC-based Planning and Control Service Applications and the major principle behind it. As the deliverable's nature is Prototype this document is also combined with the actual application.

The GET Service platform provides transportation planners with the means to plan transportation routes more efficiently and to respond quickly to unexpected events during transportation. To this end, it connects to existing transportation management systems and improves on their performance by enabling the sharing of selected information between transportation partners, logistics service providers and authorities.

The main goal of WP3 is the development of the end-user applications. More specifically this involves the development of transportation planning and control application and the development of the mobile control application.

This deliverable is describing the PC-based transportation planning and control services as they are being developed within the GET Service project. The services and the application are available as prototype components with standardized interfaces to the GET service platform and the different elements and services.

The application for transportation planning is based on the professional tour planning system provided by PTV. Within this system several different use cases can be processed. Tour planning results can be requested and visualized on the map. The system is using various algorithms for routing and planning purposes.

Enhancements of this system are different GET service related interfaces as well as adapted functionality for intermodal routing and planning for the GET service scenarios. The system consists of different components for the calculation and planning. PTV's xServer with its backend-server components has the task to perform routing and planning calculations.

The planning and control application in GET service is being developed within a service oriented architecture approach. Based on the PTV tour planning system Smartour, the intermodal planning system, is being enhanced and further developed.

Through the DMZ, an internet connected subnetwork, at PTV a web-service with a connection to an intermodal router is accessible. This installation allows testing the provided methods without implementation. This web-service is based on a Sunrise implementation.

One of the so-called handler is the *IntermodalHandler* usable after a successful login. As of now the most important methods are *getTestRequest()* and *CalculateRoute()*, both available on the test site.

The method *getTestRequest()* returns an example request (JSON formatted) which can be used as parameter in the *CalculateRoute()* request.

The intermodal planning application with the back-end services for professional transportation routing and planning has been developed further and is available to be connected via a web-service. These interfaces have been set up in frame of the early prototype and can be enhanced for further interaction with other GET Service components and scenarios for the second prototype and integrated architecture.

For testing purposes and the first check of usability and quality of plans/ routes the system is working properly and with sufficient performance.

List of Figures

List of Tables

List of Terms and Abbreviations

Term	Meaning
IM	Intermodal
JSON	Java Script Object Notation
TSC	Transshipment centre

Contents

1 Introduction

This deliverable presents the PC-based Planning and Control Service Applications and the major principle behind it. As the deliverable's nature is Prototype this document is also combined with actual application outcomes. This section provides the background to this deliverable, by presenting the goal of the project as a whole, the goal of the work package of which the deliverable is a part, and the goal of the deliverable itself. Finally, it presents the structure of the remainder of the deliverable.

1.1 Project Goal

The GET Service platform provides transportation planners with the means to plan transportation routes more efficiently and to respond quickly to unexpected events during the transportation. To this end, it connects to existing transportation management systems and improves with their performance by enabling sharing of selected information between transportation partners, logistics service providers and authorities. In particular, the GET Service platform consists of components that: (i) enable the aggregation of information from the raw data that is shared between the partners and the transportation information providers; (ii) facilitate planning and re-planning of transportation based on real-time information; and (iii) facilitate real-time monitoring and control of transportation, as it is being carried out by own resources and partner resources. By providing this functionality, the GET Service platform aims to reduce the number of empty miles that is driven, improve the modal split, and reduce transportation times and slack, as well as response times to unexpected events during the transportation. Thus, it reduces CO2 emissions and improves efficiency.

1.2 Work Package Goal

The main goal of WP3 is the development of the end-user applications. More specifically this involves the development of transportation planning and control application and the development of the mobile control application. The first one concerns the duties and work carried out by the planner and the latter is going to be used by the truck drivers in order to support their work and interact with the GET Service platform.

The WP3 will be responsible for visualizing the information produced by the GET Service core and extensive platform to the corresponding users (planners and truck drivers). It will create the relevant points of interaction between the users and the system. Thus the work that is being carried within this WP is strongly connected with the progress and output of WP2, WP5, WP6 and WP7.

1.3 Deliverable Goal

D3.1 describes the PC-based transportation planning and control services as they are being developed within the GET Service project. The services and the application are available as prototype components with standardized interfaces to the GET service platform and the different elements and services. The main content of this short documentation is an overview about the user interface as well as interface descriptions.

1.4 Deliverable Structure

At first a short usability description of the application is given in Chapter 2. It is some sort of user manual to describe how the intermodal planning application can be connected and used for different scenarios. Secondly the application is being described based on the high level architecture which is related to the early prototype architecture definition (in WP2). The main part of Chapter 3 is the explanation of the implementation and interface via a web-service.

2 Using the PC-Based Application

The application for transportation planning is based upon the professional tour planning system provided by PTV. Within this system several different use cases can be processed. Tour planning results can be requested and visualized on the map. The system is using various algorithms for routing and planning purposes.

Enhancements of this system are different GET service related interfaces as well as adapted functionality for intermodal routing and planning for the GET service scenarios. The system consists of different components for the calculation and planning. PTV's xServer with its backend-server components has the task to perform routing and planning calculations.

Generally the use cases can be divided in two different categories. Firstly there is the routing related functionality which calculates routes from A to B possibly with via-points and considering different infrastructure related information and conditions.

Using *calculateRouteInfo()*, the distance and period of a route will be computed. The internal value "cost" can also be shown but this is only relevant in comparison to alternative routes. This is a simple but fast route calculation.

The second basic use case is planning, which means that a number of orders can be processed when given basic information such as starting/end point, order related information like weight/dimensions and further requirements for loading.

Within the intermodal application the different alternatives for intermodal transport chains between two destinations are being calculated and visualized. Also these alternatives are transferred to the other applications on the GET service platform.

3 Architecture and application design

3.1 High level Architecture – GET service WP3 components

The planning and control application in GET service is being developed within a service oriented architecture approach. Based on the PTV tour planning system Smartour the intermodal planning system is being enhanced and further developed. For the integration in the GET Service project architecture the test system is accessible via a web-service.

With reference to the GET Service overall architecture and the GET subsystem components as described in D2.2.2 the WP3 components can be allocated to the client platform side and the client devices, a planner is using its planning system to elaborate valid planning options for the transportation chain which can be chosen in frame of the given scenarios. For the transportation execution phase the mobile device provides different functionalities like truck navigation and track and trace.

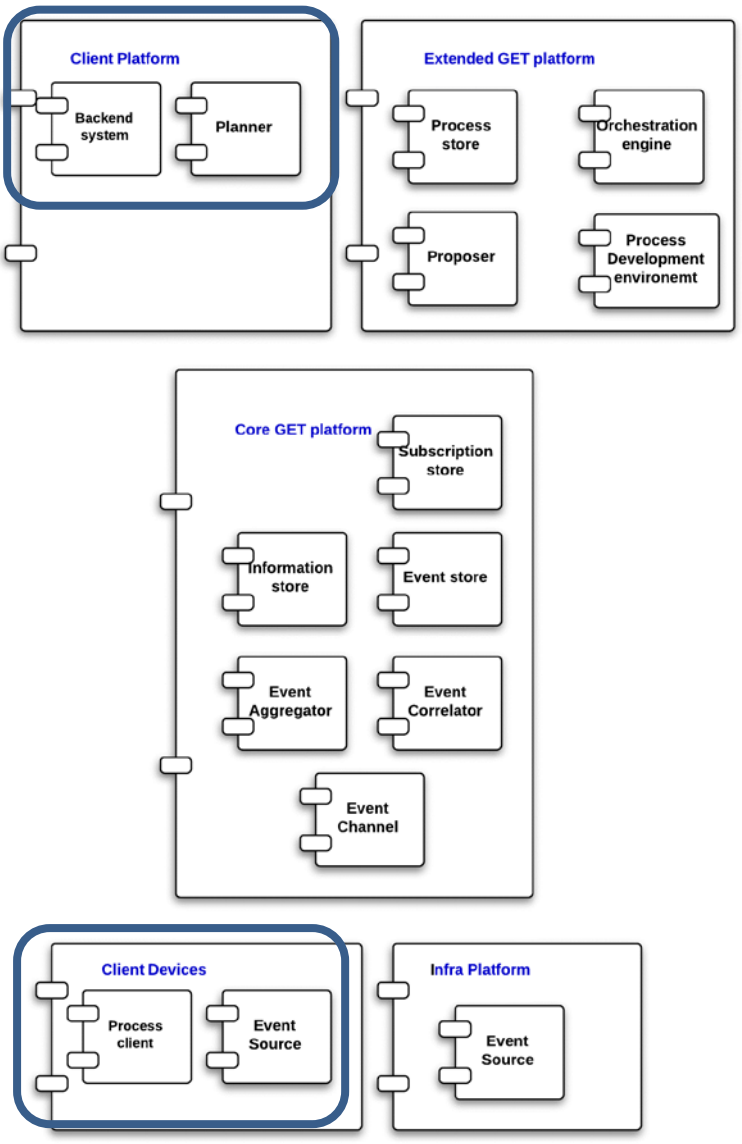


Figure 1: GET Service subsystem components – perspective WP3

Zooming into the client platform the planning architecture for the WP3 developments can be shown in the following picture. Basic component is the PTV xServer which allows to request and retrieve different routings and transport planning alternatives for different transport modes. In frame of the GET Service project a dedicated xServer System environment has been set up, further developed in terms of functionalities and data handling and provided via web-interfaces to the GET Service platform. The front end for the planner, based on PTV Smartour has been enhanced and enabled to plan and visualize different intermodal transportation alternatives and plans regarding the different scenarios. The interaction between the planning/ routing application and the GET Service environment respectively the different use cases and scenarios can be depicted in the following figure:

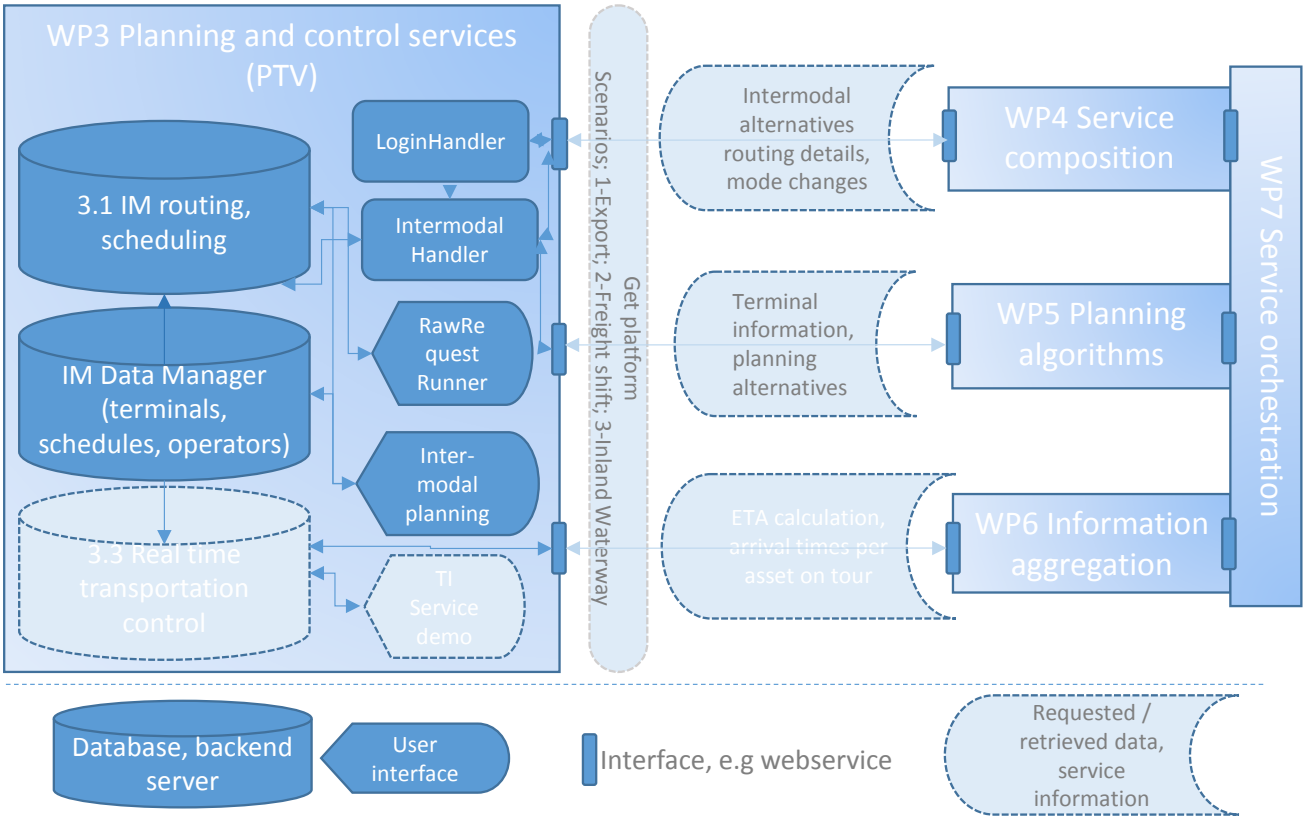


Figure 2: WP3 overall architecture with components, interfaces and interaction with GET scenarios/ WPs

WP3 planning and control services consist of three main backend components, the routing and scheduling server, the data manager and the real time application for ETA calculation. This backend service is part of the tasks in 3.3, therefore the related services are marked in light grey. In 3.1 main purpose was to develop the intermodal functionalities, the interfaces for the interaction with the other WPs and the user interfaces for testing. ETA related developments are being described in D3.3. The use cases and Get scenarios supported by WP3 are being described later on in the following chapters.

3.2 Basic Use Cases and GET specific scenarios

The application has three main use cases. Basic use case is the simple routing request for a transportation route from A to B. Results can contain intermodal routing options as well as single mode routing results depending on the relation, distance and time parameters. When it comes to order management, transport alternatives per order can be requested from the system. Multiple order assignments to different alternatives are based on tariffs, schedules and other order related parameters. In the next section the implementation, respectively interface connection between GET Service platform and the webservices at PTV are being described. In addition, a user interface for routing and planning visualisation is being presented.

Within GET Service three concrete scenarios have been set up and technically demonstrated. The developments within WP3 support the scenarios in different ways. For the partners in GET service several interfaces have been developed and provided to give the developers the possibility to easily test system procedures and to enable efficient interface programming.

Scenario	WP4	WP5	WP6	WP7	WP3	Test application
1 – Export	**Request routing options for inter-modal transport**	Request for routes	Deadline alert prediction	GET platform integration and vizuali-sation	Provision of routing data and intermodal routes	Intermodal handler, RawRequest runner
2 – freight shift	Provision of process model	Request routing options, provide transport plan, use	**Request ETA values for assets in tour, event pro-cessing**	GET platform front end vizuali-sation	ETA calculation, provision of ETA values	TI service demo, ETA service PTV drive-andarrive
3 – Inland Waterway		**Request routing options. Provide transp. plan**	Event pro-cessing and provision of lock status	GET platform front end vizuali-sation	Provision of alternatives, terminal infos	Intermodal handler, RawRequest runner

Table 1: GET scenarios, role of the different WPs

For the scenario demonstration the developments in the different work packages realised the technical interaction between the components. In Scenario 1 routing alternatives for intermodal transportation between terminals for road, rail and barge transport were requested from the WP3 routing and planning service. The routing results could be vizualised on the GET platform front end. For the freight shift scenario (road, air freight) the PTV ETA service were called to retrieve the estimated time of arrival of specific trucks on tour. The ETA service has been implemented as business solution suitable for different user cases of road transport of larger fleets. The Inland

Waterway scenario required the provision of different transportation planning alternatives especially for the transport modes barge and road between origin and destination locations in different European Countries. Before implementation of the interfaces partners could test the application as mentioned in the table.

4 Accessibility Information

4.1 General Information

Through the central server infrastructure at PTV a web-service with a connection to an intermodal router is accessible. This installation allows testing the provided methods without implementation. This web-service is based on a Sunrise implementation.

From here, as can be seen in Figure 3, one can get an overview of the implemented parts and their latest update.

The link “html” opens up the documentation and help area, which will be explained in section 4.5.2. To be able to test the procedure the handlers *LoginHandler* and *IntermodalHandler* are needed. *IntermodalHandler* can only be used after the successful login through *LoginHandler*.

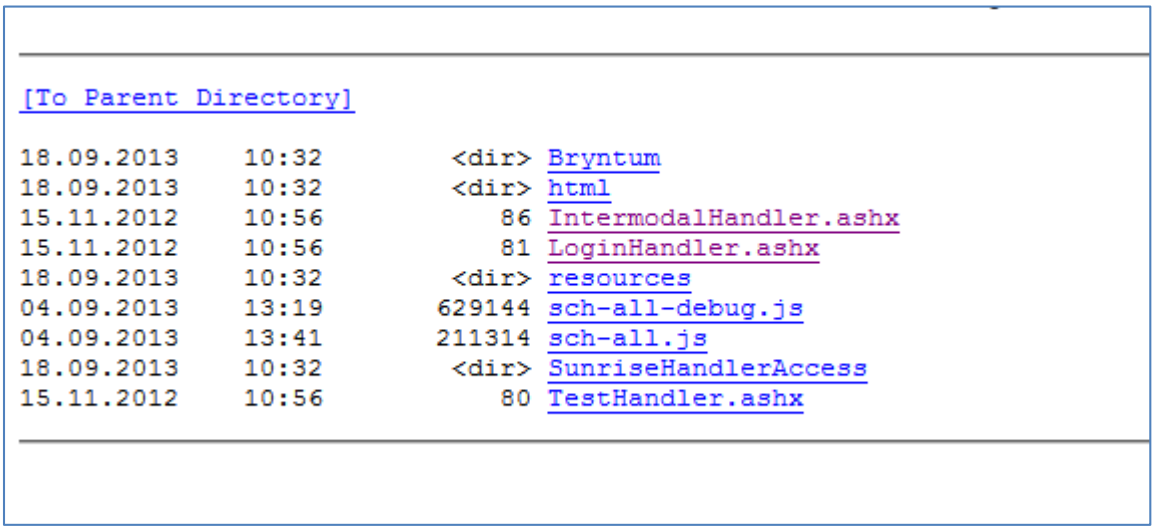


Figure 3: The WebService: all available Handler

4.2 LoginHandler

The *LoginHandler* is used to demonstrate how the login works. For this reason the following figures show the existing options. The options to change the user password, receive the admin contact information and to log in are the options presented in the screen shot, Figure 4. In order to check the default settings, a test page is provided (Figure 5). Here one can select the method to be tested and fill in the “request parameters”. The test starts with the test button. If for example the login should be tested one only needs to enter the default password, user and client information to receive a positive answer for a correct login. Other options for a test are *LoginIsValid*, Logout and others.

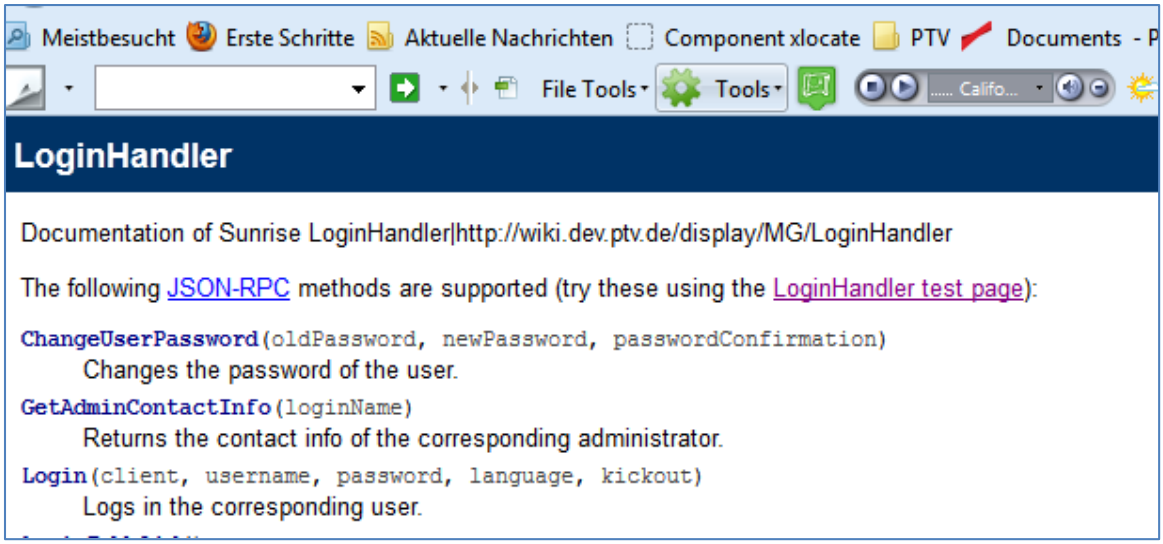


Figure 4: The LoginHandler

Each handler provides a test page (Figure 5).

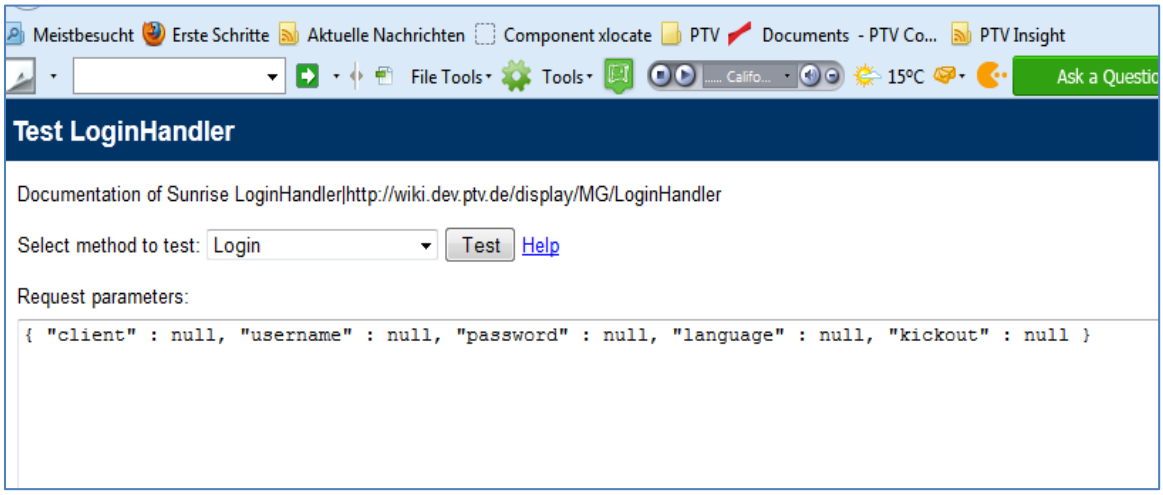


Figure 5: LoginHandler Testpage

.

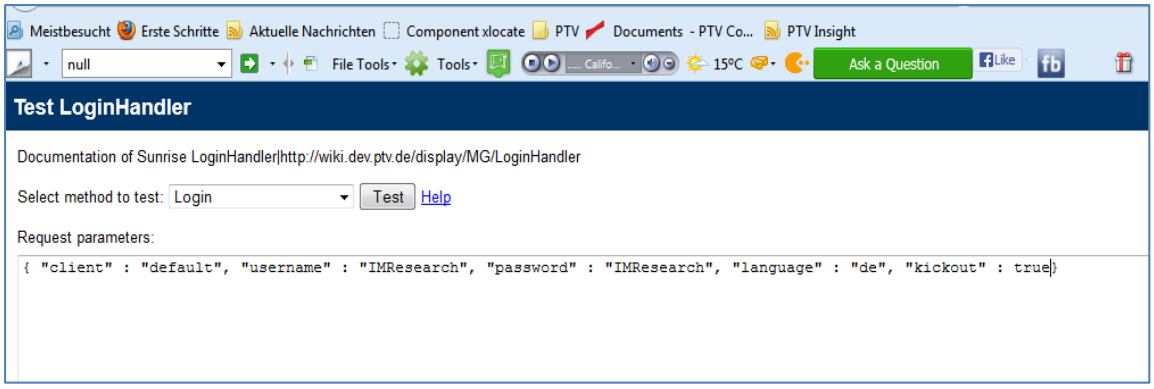


Figure 6: Request with data

The test result of a successful completion can be seen in Figure 7.

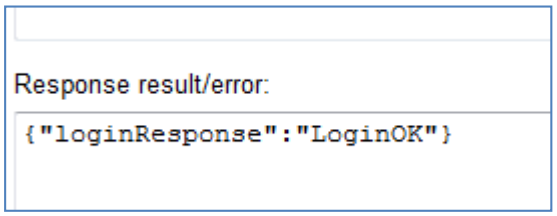


Figure 7: Successful login result in JSON

4.3 IntermodalHandler

Another handler is the *IntermodalHandler* usable after a successful login. Different methods for retrieving intermodal transport information are available. As of now the most important methods are *getTestRequest()* and *CalculateRoute()*, both available on the test site(See yellow markup in Figure 8).

Here the different methods for requesting and retrieving test results of different route options as well as transportation modes and terminals are being listed.

The method *getTestRequest()* returns an example request (JSON formatted) which can be used as parameter in the *CalculateRoute()* request. Other methods can also be seen in the screenshot.

The functionality of the simple routing request is to search for intermodal transport chains between locations for a certain start or end time, respecting given chain composition restrictions (e.g. mode, operator). Results are based on IM data.

The simple routing request returns a set of composed transport chains between these two locations (addresses or geo-locations), which match the applied routing restrictions for a certain starting time.

A transport chain can consist of a single or multiple transport leg elements. The transport chain can include only a single or multiple transport modes.

Transport modes supported are:

- Road
- Rail
- Inland Water Ways (Barge)

- Air
- Deep Sea & Short Sea

Routing restrictions supported are:

- Transport mode
- Transport Service Operator
- Accompanied Transport
- Unaccompanied Transport
- Maximum Costs allowed

A simple routing result can either be based on a road routing or a timetable based transport service or a combination of both. Depending on the transport service availability announced in the IM data, the xIntermodal result set consists of a single or multiple transport chain results.

Transport chain results of a simple routing request consider the given start time of the request and deliver different chain alternatives accordingly. Each alternative presented in the results consists of a unique service composition (with respect to chain composition and operator).

Each transport chain proposal results in itinerary and KPIs (km, travel-time and costs - € and CO_2; both based on given data sets or data models.

For valid routing options the following use case and data requirements are given:

- Valid IM Data (terminals, operators, regular services incl. costs, terminal transfer modalities)
- Digital Road Map
- Digital networks per mode of transport (optional)

As input data the system needs at least:

- Origin address (with coordinates)
- Destination address (with coordinates)
- Starting time
- Restrictions (optional)

The output information contains the

- List of routes (= transport alternative), each of them consisting of a sequence of leg descriptions containing
  - Station (Origin address, Terminal or destination address)
  - Results concerning the inbound leg to the actual station such as
    - Mode
    - Regular service
    - Times (Arrival time, handling time, departure time, …)
    - costs
    - KPIs (distance, duration, emission, …)

Figure 8: The IntermodalHandler methods

For enhanced testing scenarios and usability for the different partners a specific user interface called “Raw request runner” has been set up. The user interface is being described in the following section.

4.4 Raw request runner for interface testing

In the following screen shots a short documentation about the interface with PTV xServer for the GET Service project is being described:

- On the PTV Server environment the GET specific xintermodal server is being provided.
- The data base behind includes as schedules complete available data set from PTV
- The connection for external access can be done via http://80.146.239.130:50160/

For testing purposes the xServer Raw Request Runner can be used. The user interface and the usability is being described in the following screenshots:

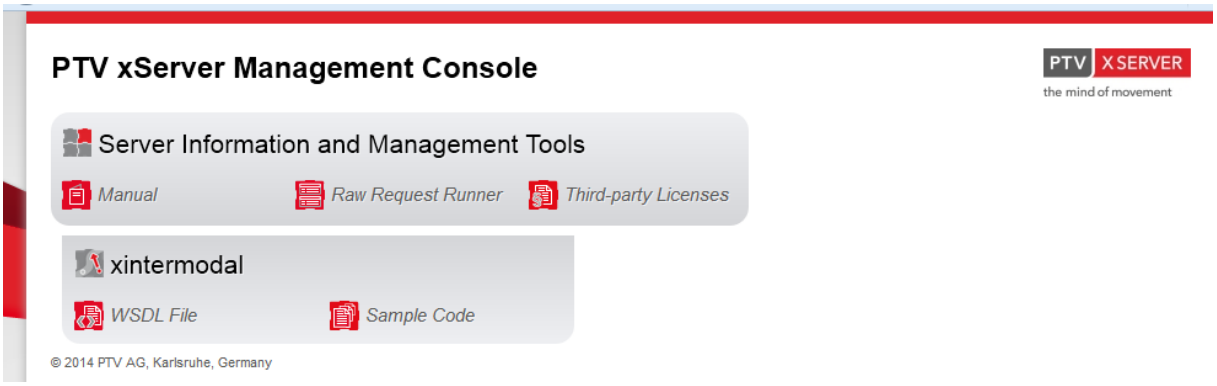


Figure 9: xServer Management Console

Click on „RawRequestRunner“ :

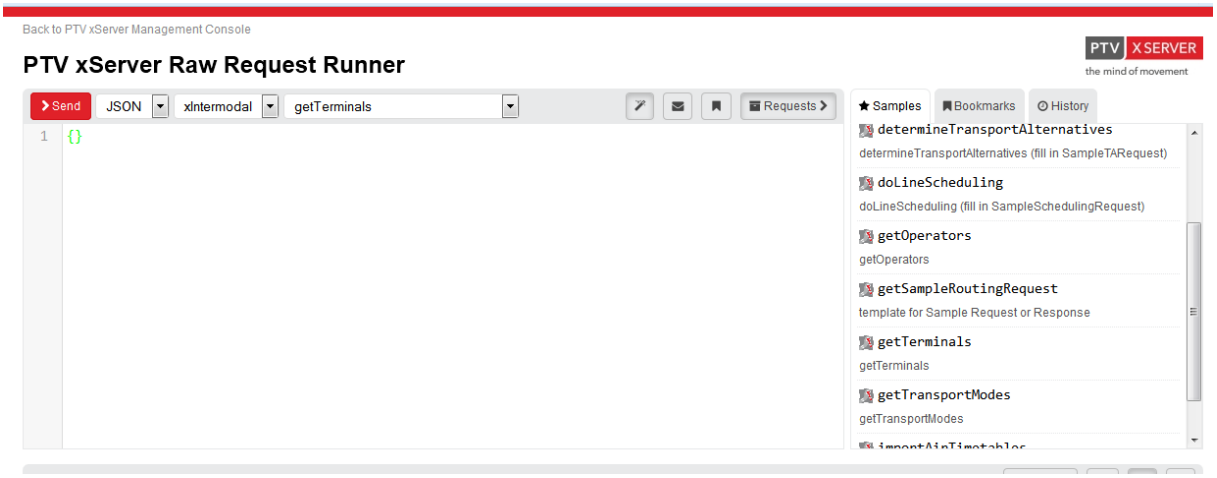


Figure 10: Start screen of Raw Request Runner

- „*Samples“ are various examples for Requests which can be started,
- e.g. click on „getTerminals“ then on „>Send“.
- Results:

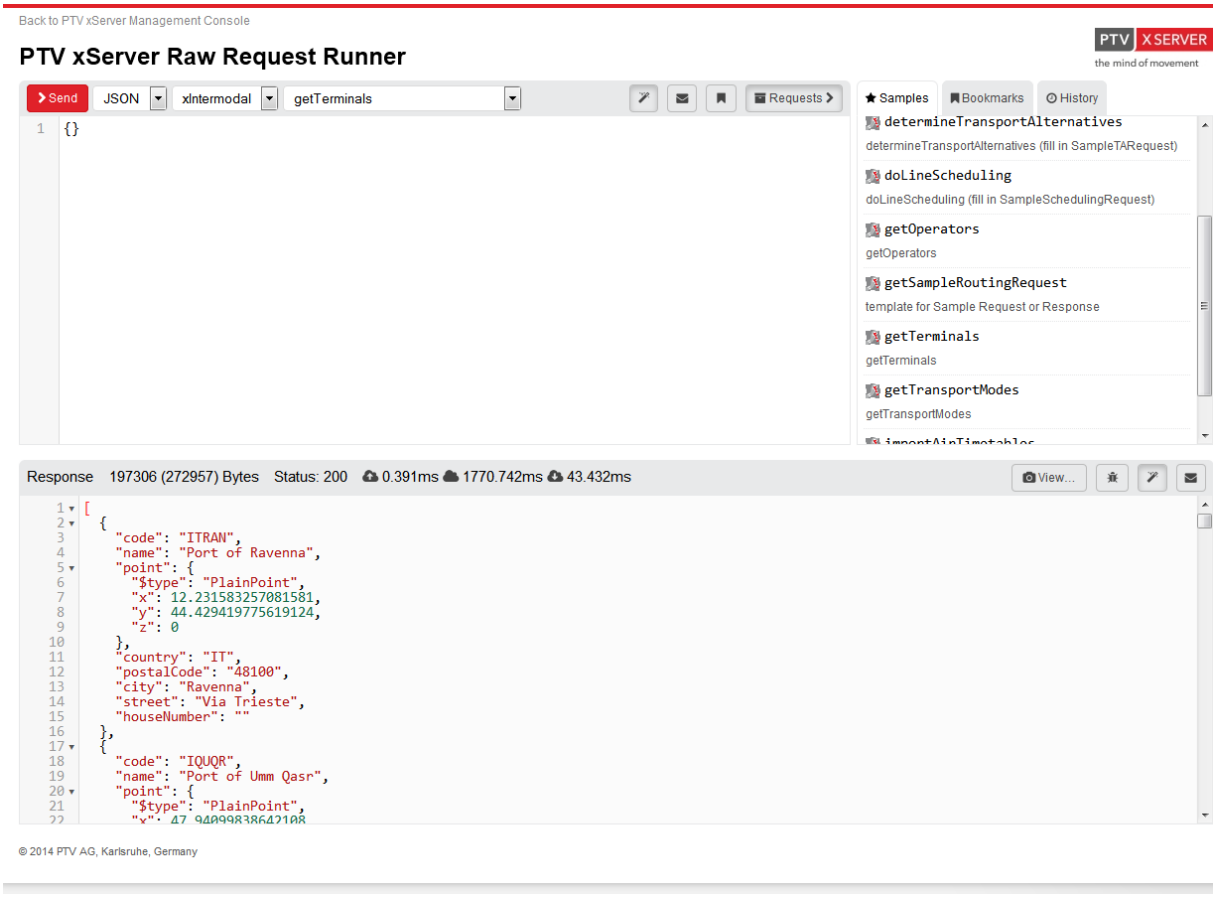


Figure 11: Raw request runner; response example

To start a routing request pls follow the process:

Click in „*Samples“ on „getSampleRoutingRequest“ and then on „>Send“

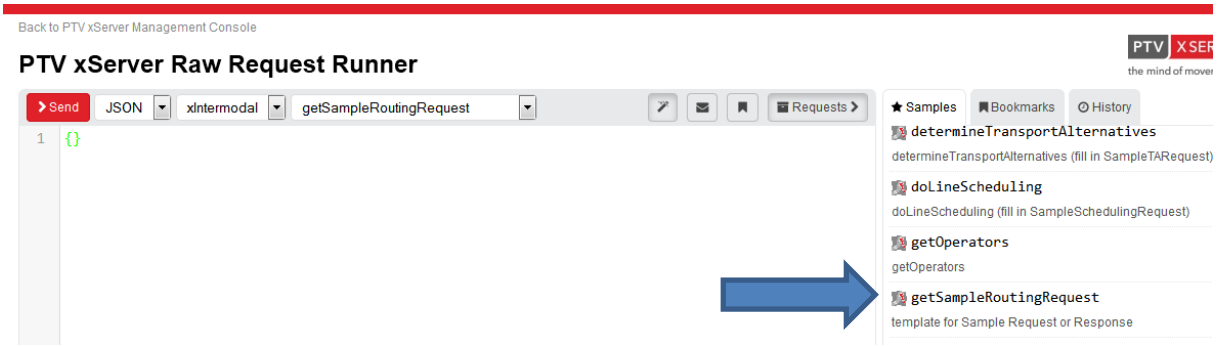


Figure 12: User guidance

- The answer is a JSON-example, how the routing request should look like:

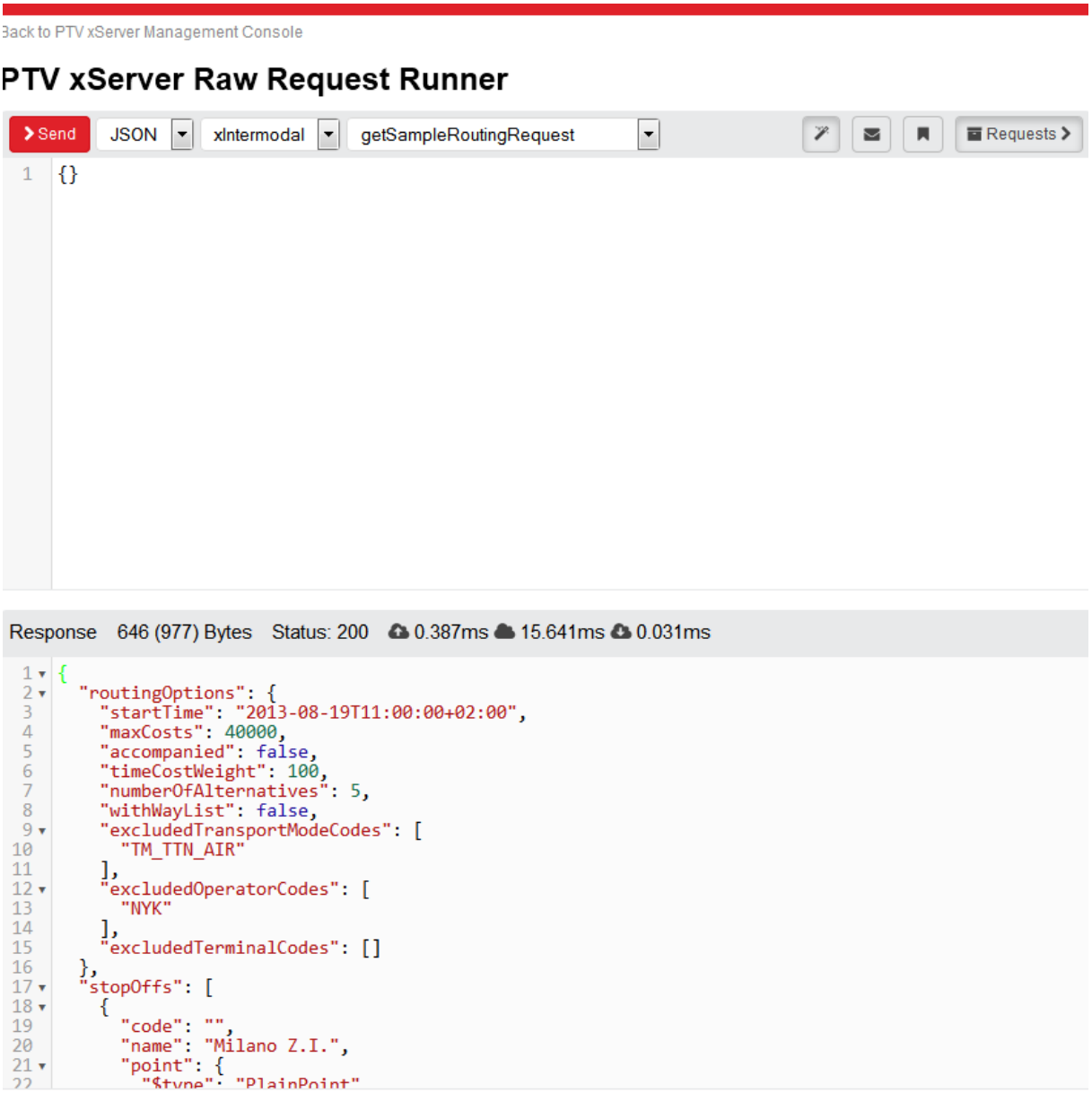


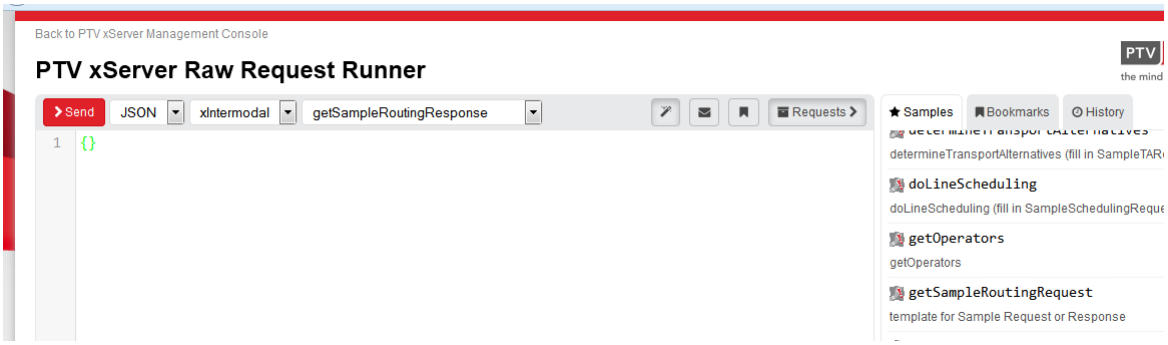
Figure 13: JSON example

This example can be copied and pasted, changed and then inserted in the routing request.

After „*Samples“ , „calculateRoute“ and then click on „>Send“ you can see the following:

Figure 14: Calculate Route function for samples

Please insert the copied routing request instead of „{}“ and then click on „>Send“ . You should get then the results. While inserting the request the address does not need to be correct, relevant are only the coordinates. Similar to that it is possible to retrieve a sample routing response via „getSampleRoutingResponse“. Therefore choose again in „*Samples“ the “getSampleRoutingRequest” and choose additionally in the third combobox above the „getSampleRoutingResponse”.



The response is then:

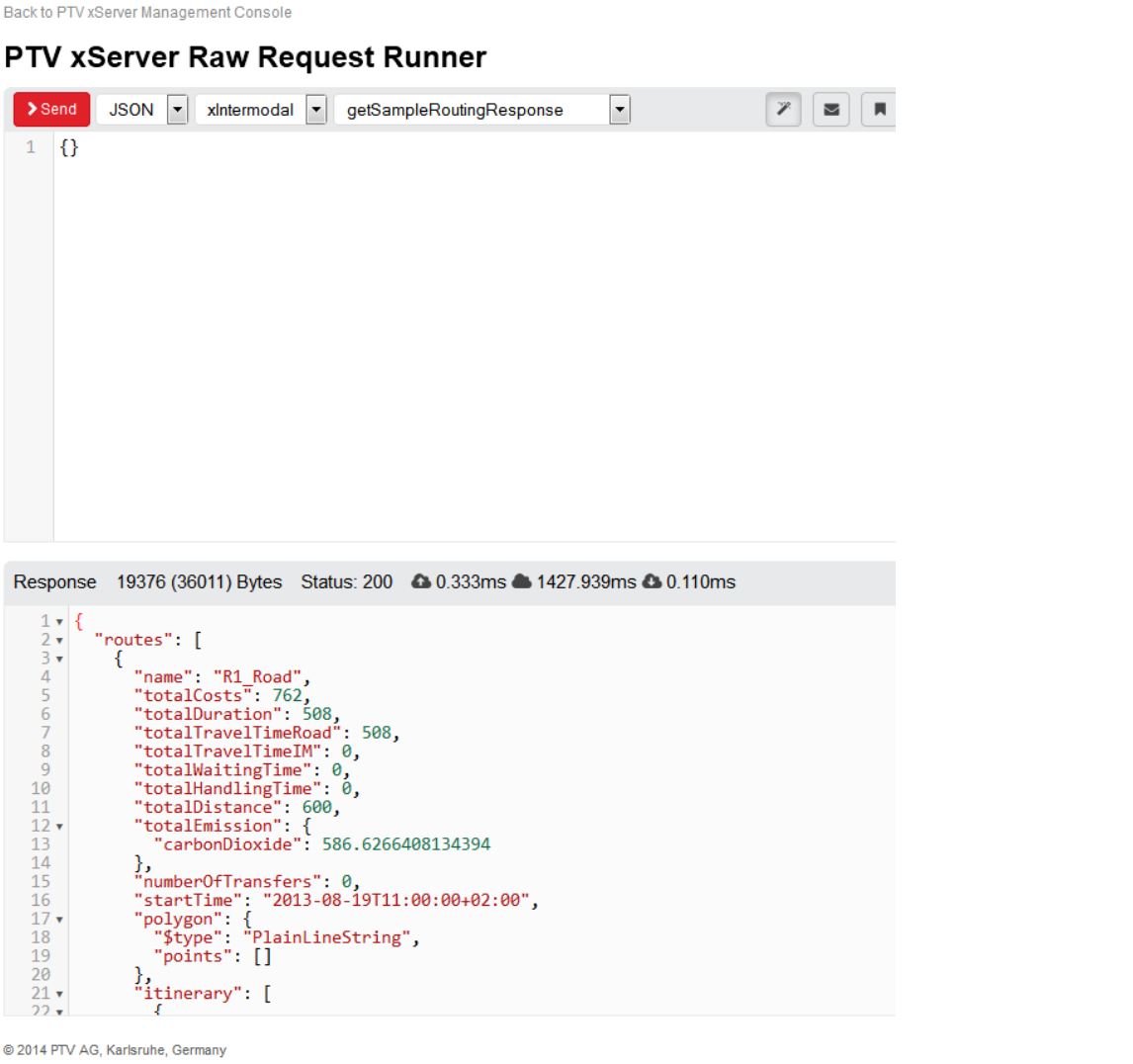


Figure 15: Result for “getSampleRoutingResponse”

4.5 Documentation, screenshots of PC-based intermodal routing

4.5.1 Screenshot examples of intermodal route planning

The intermodal route planner can be used with a user interface for entering addresses or terminals, routing results are being listed with the different transport legs and transshipment locations. Additionally the route is being visualized on a map.

After entering addresses like for example Karlsruhe in Germany and Rotterdam in Netherlands (see yellow markup in Figure 16**)** and the call *CalculateRoute()* we receive 4 routes as a result (the blue circle). The screenshot shows the routes with some information like the distance in km.

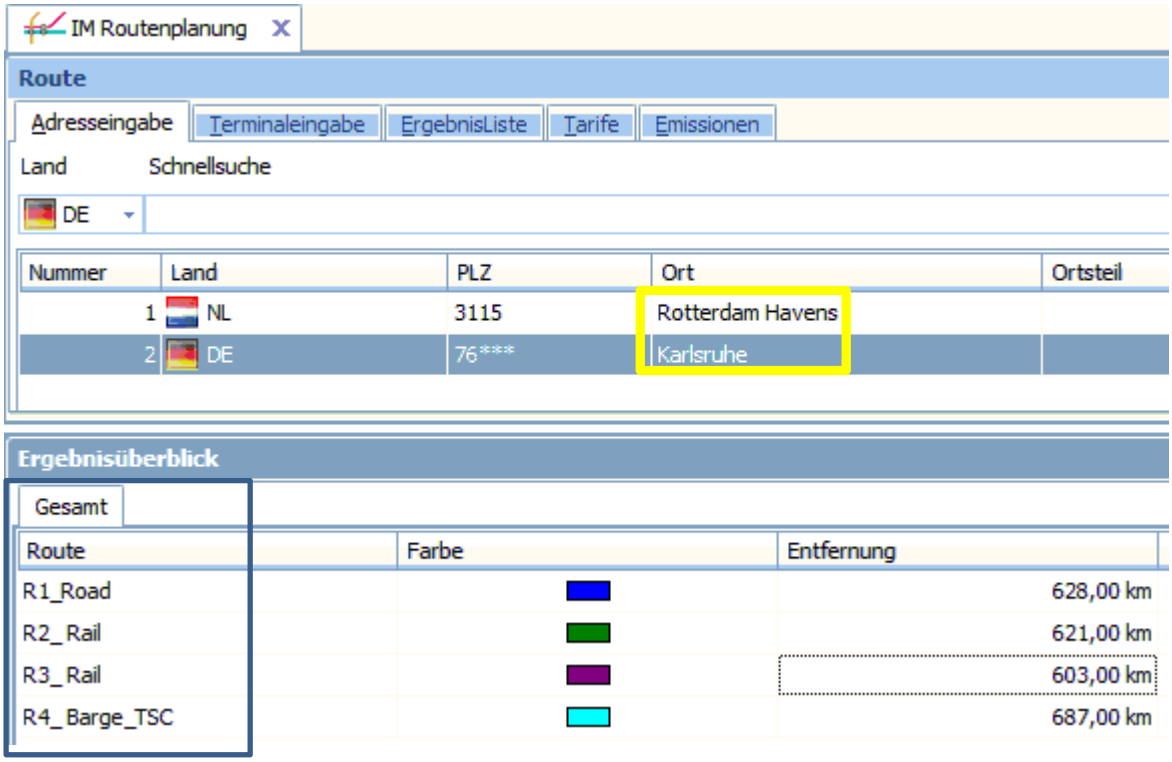


Figure 16: Addresses and route overview

The screenshot below indicates another route for a rail connection between Karlsruhe and Oslo in detail. In the left column the first icon is a truck, the second is a crane for up- or unloading, followed by the symbol for rail and arrows, indicating a transfer in a Transshipment Center (e.g. Port of Rotterdam).

Figure 17: A route in detail

In detail this route starts with a truck delivering the goods to Karlsruhe Hgbf (main cargo station). There the cargo is loaded onto a train, which terminates in Hamburg. After arriving the goods are transferred in a Transshipment Center until finally a truck brings the goods to their final destination in Oslo.

The next screenshot shows the map with the different route options of the example.

Figure 18: Map with the routes

4.5.2 The Online Help

An online documentation is accessible at http://80.146.239.188/IMResearch/Utility/html/. It is still under construction, but will include both documentation and help functions, after completion.

As an example for the help options Figure 19 shows a screenshot with the menu options for further information.

Future tasks are to reduce the online help to the needed topics and to describe all methods and parameters depending on the *IntermodalHandler*.

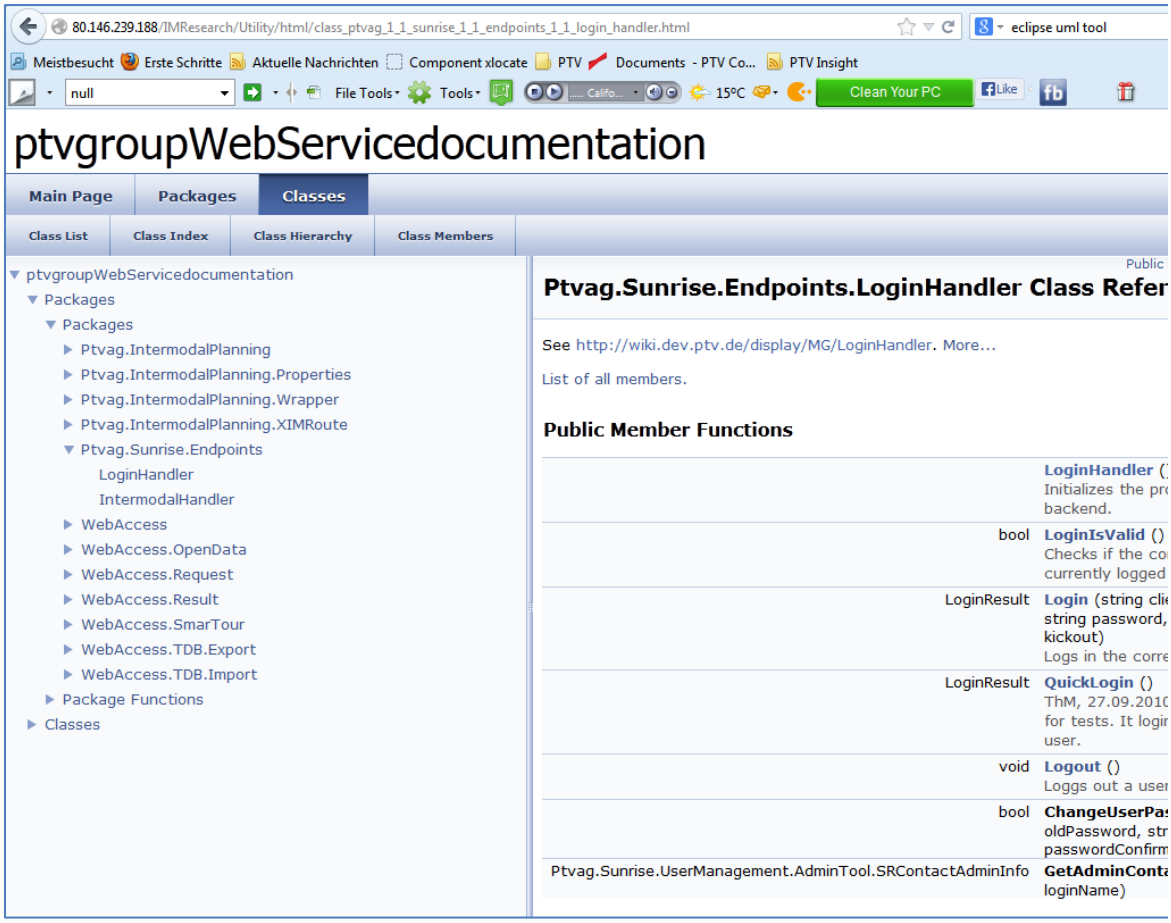


Figure 19: The main online help entry

5 Conclusions and Next Steps

The intermodal planning application with the back end services for professional transportation routing and planning has been developed further and is available to be connected via a web-service. These interfaces have been set up in frame of the early prototype and can be enhanced for further interaction with other GET Service components and scenarios for the second prototype and the integrated architecture.

For testing purposes and the first check of usability and quality of plans/routes the system is working properly and with sufficient performance.

For the second prototype the test system have been enhanced and further developed in order to be connected to the scenarios/components. Interfaces have been updated, the services and scenario related data for different transportation alternatives have been integrated. The reference to the GET Service platform and the related common interface have been considered.

Furthermore the data provision and information flow between the different components have to be analysed and checked regarding compatibility and comparability of planning results.

In the meantime the scenario related data and use cases are being implemented and tested.

Another issue is the interaction with and implementation of the mobile application. Within WP3 PTV has provided a professional truck navigator with standardized interface for the implementation of the mobile application, further information on this can be found in D3.2: This can be used for the final project stage with the real-time approach for re-planning and event handling during transport execution.