

# INFORMATION TECHNOLOGIES FOR SHIFT TO RAIL

## D4.4 – Trip Tracker Core Integration Report

Due date of deliverable: 31/08/2016

Actual submission date: 26/09/2016

Leader of this Deliverable: Petr Buchniček (OLTIS Group)

Reviewed: Y

<b>Document status</b>		
Revision	Date	Description
0	19/05/2016	Template creation
1	31/08/2016	Test campaign execution report
2	12/09/2016	Adopted general deliverable template
3	21/09/2016	Final version after TMC approval

<b>Project funded from the European Union's Horizon 2020 research and innovation programme</b>		
<b>Dissemination Level</b>		
<b>PU</b>	Public	X
<b>CO</b>	Confidential, restricted under conditions set out in Model Grant Agreement	
<b>CI</b>	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/05/2015

Duration: 30 months

## **EXECUTIVE SUMMARY**

---

This document presents an integration report on the core release of the Trip Tracker module of the IT2Rail project. After a brief description of tested system configuration, settings and requirements, the document is focused on the validation of the Trip Tracker's functionality implemented so far. First, all the intended tests are described in detail. Reports on their execution are completed in the subsequent chapter.

The terminology used in this document is based on the IT2Rail ontology. The following applicable documents are referenced in this deliverable:

- D4.1 – Trip Tracker Ontology document
- D4.2 – Trip Tracker Specifications document

## TABLE OF CONTENTS

---

Executive Summary .....	2
List of Figures .....	4
List of Tables .....	4
1. Introduction .....	5
2. Test materials description .....	5
2.1 Basic configuration .....	5
2.1.1 Infrastructure and hardware .....	5
2.1.2 Setup & Configuration .....	5
2.1.3 Tested system .....	6
2.1.4 System data parameters .....	6
2.1.5 Simulators .....	6
2.1.6 Personnel .....	6
3. Test descriptions .....	7
3.1 Tracking activation .....	7
3.1.1 Request tracking (simulated calls) .....	7
3.1.2 Request tracking (demonstration) .....	8
4. Test execution .....	9
4.1 Tracking activation .....	9
4.1.1 Request tracking (simulated calls) .....	10
4.1.2 Request tracking (demonstration) .....	10

---

## LIST OF FIGURES

No figures are introduced in this document.

---

## LIST OF TABLES

Table 1: Request tracking (simulated calls) – description.....	7
Table 2: Request tracking (simulated calls) – expected results .....	8
Table 3: Request tracking (demonstration) – description.....	8
Table 4: Request tracking (demonstration) – expected results.....	9
Table 5: Request tracking (simulated calls) – summary .....	10
Table 6: Request tracking (simulated calls) – observed results .....	10
Table 7: Request tracking (demonstration) – summary .....	10
Table 8: Request tracking (demonstration) – observed results.....	11

## 1. INTRODUCTION

The objective of the C-REL test strategy is to validate the functionality developed for this release. In particular, the test concerns the functionality of the journey tracking request. The test can be executed using relevant web service, exposed by the TrackingActivation component of the Trip Tracker. The goals of the test are:

- To verify the readiness of RequestJourneyTracking method, i.e. to test the availability of relevant web service for external clients.
- To check the interface and to demonstrate communication between Travel Companion and Trip Tracker, focusing on triggering of journey tracking.

Our test campaign will also investigate error conditions for the web service, e.g. the calls of tested method with missing or empty parameters. However, no changes of a basic configuration of the TrackingActivation component are expected and no stress tests are scheduled in this campaign.

## 2. TEST MATERIALS DESCRIPTION

This chapter lists all the assets required to perform the test campaign. The test is associated with a basic configuration of the tested system. The configuration itself, including its parameters and resources needed to conduct the tests, are described in the following subchapters.

### 2.1 BASIC CONFIGURATION

In order to process the journey tracking requests successfully, the TrackingActivation component requires to be configured properly. The configuration consists of infrastructure and hardware settings (minimum system requirements) and also the provided service settings (application setup needed to ensure its correct operation). The description does not include specification of any other components on which the TrackingActivation component may rely during its future use.

#### 2.1.1 Infrastructure and hardware

The minimum system requirements (runtime environment) are Windows 7 operating system (and above) with .NET Framework 4.6.1 (and above) installed. No further installation is needed. The distributable archive containing all necessary application files could be simply extracted to any chosen folder.

The TrackingActivation component is represented by a standalone console application implementing HTTP/JSON web service with RequestJourneyTracking remote method.

#### 2.1.2 Setup & Configuration

The related configuration file is based on XML syntax and editable by any common text editor.

There are two custom settings:

1. Value at *AppLogDirectory* setting – absolute or relative path to folder where application log will be written.
2. Value at *Address* setting – IP address (or host name) and TCP port at local machine where the web service has to be hosted (listening).

The application must be run by the user with appropriate user rights for:

- launching executable files,
- writing to application log file,
- addressing registers.

Please mind also personal or local net firewall restrictions.

If the console application is set to listening at localhost machine at port 8080, remote method is available at URI: *http://localhost:8080/api/RequestJourneyTracking*

### 2.1.3 Tested system

This test campaign aims at testing of the journey tracking request functionality, provided as a web service, exposed by the TrackingActivation component of the Trip Tracker module.

### 2.1.4 System data parameters

The RequestJourneyTracking remote method accepts JSON object passed by HTTP POST request with three string values as input parameters called:

- *userID*,
- *trip2track*,
- *activationType*.

HTTP request header value Content-Type has to be "application/json".

The method returns JSON object with one string value called *activationResponseMessage*. HTTP response header value Content-Type is "application/json; charset=utf-8".

String "OK" is returned when the method call is successful, otherwise an error message is returned.

### 2.1.5 Simulators

At the first stage of testing, incoming journey tracking requests will be simulated by the REST/JSON plug-in in the internet browser (e.g. Postman plug-in in Google Chrome).

At the second stage, the real communication between Travel Companion and Trip Tracker will be tested, thus the interface operation will be demonstrated by the test.

### 2.1.6 Personnel

The personel required to run this test campaign are qualified people with high professional knowledge and background on Information Technology and Computer Science.

### 3. TEST DESCRIPTIONS

This chapter describes the test case that will be executed before the Core Release of the Trip Tracker.

#### 3.1 TRACKING ACTIVATION

The aim of TrackingActivation component is to reflect Traveller's decision to track the journey. When such a decision is made, the Travel Companion sends a request to the Trip Tracker in order to trigger tracking of given journey. TrackingActivation implements an interface to Travel Companion, which is designed to retrieve all necessary information required to enable tracking activation. The main objective of the test case is to test the correct communication between Travel Companion and Trip Tracker (in terms of triggering of journey tracking), including the provision of required input parameters. Otherwise the journey tracking cannot be activated, so the request is rejected. That's why both possibilities (provision of correct and/or incorrect parameters) have to be tested.

##### 3.1.1 Request tracking (simulated calls)

This test will verify the readiness of RequestJourneyTracking method through simulated calls. Using the suitable plug-in in the internet browser, the tester repeatedly calls the RequestJourneyTracking remote method and sends the input parameters to the application, observing responses it returns.

4.1.1a Request tracking	
Method Of Test	Demonstration
Type of test	Manual
Objective	To verify the readiness of RequestJourneyTracking method
Description	The method in question will be called repeatedly with correct/wrong parameters from WP4 testing environment (simulated calls)
Status	
% passed	
Regression	
Test Case Tester	

Table 1: Request tracking (simulated calls) – description

4.1.1a Request tracking				
ID of step	Step description	Expected result	Observed result	State
1	Correct parameters	OK		
2	Wrong parameters	Bad request		
3	Correct parameters	OK		
4	Correct parameters	OK		

**Table 2: Request tracking (simulated calls) – expected results**

Input JSON object example:

```
{
  "trip2track": "journey",
  "userID": "123456",
  "activationType": "activate"
}
```

Returned JSON object example:

```
{"activationResponseMessage": "OK"}
{"activationResponseMessage": "input parameter trip2track is missing or empty"}
```

### 3.1.2 Request tracking (demonstration)

This test aims at verification of the interface and communication between Travel Companion and Trip Tracker. The only additional precondition of this test is the ability of Travel Companion to call the remote method exposed by the Trip Tracker. Other preconditions are same as in the previous test (simulated calls).

4.1.1b Request tracking	
<b>Method Of Test</b>	Demonstration
<b>Type of test</b>	Automated
<b>Objective</b>	To demonstrate the processing of journey tracking request
<b>Description</b>	The interface and communication between TC and TT will be tested, focusing on triggering of journey tracking
<b>Status</b>	
<b>% passed</b>	
<b>Regression</b>	
<b>Test Case Tester</b>	

**Table 3: Request tracking (demonstration) – description**



4.1.1b Request tracking				
ID of step	Step description	Expected result	Observed result	State
1	Correct parameters	OK		
2	Correct parameters	OK		
3	Wrong parameters	Bad request		
4	Wrong parameters	Bad request		
5	Correct parameters	OK		
6	Correct parameters	OK		
7	Correct parameters	OK		
8	Correct parameters	OK		
9	Correct parameters	OK		
10	Wrong parameters	Bad request		
11	Correct parameters	OK		
12	Correct parameters	OK		
13	Wrong parameters	Bad request		
14	Correct parameters	OK		
15	Wrong parameters	Bad request		
16	Wrong parameters	Bad request		
17	Correct parameters	OK		
18	Correct parameters	OK		
19	Correct parameters	OK		
20	Correct parameters	OK		

**Table 4: Request tracking (demonstration) – expected results**

## 4. TEST EXECUTION

The results of both tests execution are summarized in this chapter.

### 4.1 TRACKING ACTIVATION

The main objective of the test case, as well as the detailed description of executed tests, are specified in previous chapters. In this chapter, the achieved results of executed tests are described. The first test aimed at verification of the readiness of tested RequestJourneyTracking method (through simulated calls), while the second test verified the real communication between Travel Companion and Trip Tracker.

#### 4.1.1 Request tracking (simulated calls)

4.1.1a Request tracking	
Method Of Test	Demonstration
Type of test	Manual
Objective	To verify the readiness of RequestJourneyTracking method
Description	The method in question will be called repeatedly with correct/wrong parameters from WP4 testing environment (simulated calls)
Status	OK
% passed	100%
Regression	NA
Test Case Tester	OLTIS Group

Table 5: Request tracking (simulated calls) – summary

4.1.1a Request tracking				
ID of step	Step description	Expected result	Observed result	State
1	Correct parameters	OK	OK	Passed
2	Wrong parameters	Bad request	Bad request	Passed
3	Correct parameters	OK	OK	Passed
4	Correct parameters	OK	OK	Passed

Table 6: Request tracking (simulated calls) – observed results

#### 4.1.2 Request tracking (demonstration)

4.1.1b Request tracking	
Method Of Test	Demonstration
Type of test	Automated
Objective	To demonstrate the processing of journey tracking request
Description	The interface and communication between TC and TT will be tested, focusing on triggering of journey tracking
Status	OK
% passed	100%
Regression	NA
Test Case Tester	SNCF + OLTIS Group

Table 7: Request tracking (demonstration) – summary

4.1.1b Request tracking				
ID of step	Step description	Expected result	Observed result	State
1	Correct parameters	OK	OK	Passed
2	Correct parameters	OK	OK	Passed
3	Wrong parameters	Bad request	Bad request	Passed
4	Wrong parameters	Bad request	Bad request	Passed
5	Correct parameters	OK	OK	Passed
6	Correct parameters	OK	OK	Passed
7	Correct parameters	OK	OK	Passed
8	Correct parameters	OK	OK	Passed
9	Correct parameters	OK	OK	Passed
10	Wrong parameters	Bad request	Bad request	Passed
11	Correct parameters	OK	OK	Passed
12	Correct parameters	OK	OK	Passed
13	Wrong parameters	Bad request	Bad request	Passed
14	Correct parameters	OK	OK	Passed
15	Wrong parameters	Bad request	Bad request	Passed
16	Wrong parameters	Bad request	Bad request	Passed
17	Correct parameters	OK	OK	Passed
18	Correct parameters	OK	OK	Passed
19	Correct parameters	OK	OK	Passed
20	Correct parameters	OK	OK	Passed

**Table 8: Request tracking (demonstration) – observed results**