

# D 4.2.3. – INTERFACE DEVELOPMENT KIT

## WP4- 4.2. – SOFTWARE DEVELOPMENT

10/01/2022

Partner: PP9 – RATHMANN d.o.o.  
Authors: Robert Borošak, Damir Jurić, Alessandro Caponi  
Email: [robert.borosak@rathmann.hr](mailto:robert.borosak@rathmann.hr), [damir.juric@rathmann.hr](mailto:damir.juric@rathmann.hr), [alessandro.caponi@poistory.it](mailto:alessandro.caponi@poistory.it)

<b>Project Acronym</b>	E-CHAIN
<b>Project ID Number</b>	10048282
<b>Project Title</b>	Enhanced Connectivity and Harmonization of data for the Adriatic Intermodal Network
<b>Priority Axis</b>	4 - Maritime Transport
<b>Specific objective</b>	4.1 - Improve the quality, safety and environmental sustainability of marine and coastal transport services and nodes by promoting multimodality in the programme area
<b>Work Package Number</b>	N
<b>Work Package Title</b>	WP name
<b>Activity Number</b>	N.N
<b>Activity Title</b>	Act name
<b>Partner in Charge</b>	PPn – Name
<b>Partners involved</b>	LP - Municipality of Ancona PP1 - Amatori Interestate SRL PP2 - Brusutti S.R.L. PP3 - G.M.T. S.P.A. PP4 - University of Trieste PP5 - University Of Rijeka, Faculty Of Maritime Studies Rijeka PP6 – Prosoft d.o.o. PP7 - Jadrolinija PP8 - City of Split PP9 - Rathmann d.o.o.
<b>Status</b>	Final
<b>Distribution</b>	Public

## VERSION CONTROL

Date	Version	Prepared by	Responsible	Approved by	Revision	Comment
08/11/2021	Draft 1	R. Borošak	PP9	D. Jurić	D. Jurić	/
16/12/2021	Draft 2	A. Caponi	PP9	R. Borošak	R. Borošak	/
10/07/2022	Final	R. Borošak/A. Caponi	PP9	D. Jurić	D. Jurić	/

## TABLE OF CONTENTS

Version control.....	2
Table of Contents .....	3
Acronyms / Abbreviations .....	4
<b>TECHNICAL DOCUMENTATION .....</b>	<b>5</b>
1.    DEFINITIONS.....	5
2.    AUTHORIZATION.....	6
3.    E-CHAIN TIMETABLE SERVICE ENDPOINTS.....	7
3.1.    AGENT ENDPOINT .....	7
3.1.1.    REGISTER AGENT .....	7
3.1.2.    REGISTERED AGENT CONNECTIONS.....	8
3.2.    STATIONS ENDPOINT .....	9
3.2.1.    REGISTER STATIONS .....	9
3.2.2.    REGISTERED STATIONS .....	10
3.3.    CONNECTIONS ENDPOINT.....	11
3.3.1.    REGISTER CONNECTIONS.....	11
3.3.2.    SEARCH .....	13
3.3.3.    SEARCHNEAREST .....	14
3.4.    VEHICLE TYPES .....	16
4.    E-CHAIN PLAN WEB SERVICE .....	17
4.1.    MESSAGE PLAN.....	17
4.2.    START MESSAGE PLAN .....	17

## ACRONYMS / ABBREVIATIONS

ACRONYM	DEFINITION
PP	Project partners
PT	Project Team
TC	Technical task coordinator
WP	Work package
IT	Information Technologies
API	Application Programming Interface
JSON	JavaScript Object Notation

## TECHNICAL DOCUMENTATION

This is the technical documentation of E-CHAIN web services. There're two clusters of services:

- 1) **E-CHAIN timetable service** aims to provide a way for agents to register and add their stations and timetables. The web service also provides a way to search for all available (unimodal or intermodal) routes between registered stations. The web service is JSON-based.
- 2) **E-CHAIN plan web service** aims to provide a way for travellers to register to a message plan to get information about a trip.

Swagger documentation of E-chain timetable service is available on:

<https://f4bi.rathmann.hr/echain/swagger/index.html>

### 1. DEFINITIONS

**Hub** – In order for the service to detect that there is a possibility of transfer between voyages at a station, it is necessary to introduce the term „Hub“. Since agents are not aware of other agents' stations and their timetables, each station should be assigned a hub before registering stations on the service. A hub is identified by its name. By default, every station is its own hub with the same name. For stations with the possibility of transfer between voyages, the same hub name should be assigned to multiple stations (of multiple agents).

Take the port of Ancona for example. An agent offering ship transport may have a station named Port Ancona. An agent offering a bus transportation may have a station named Central Station Ancona. For a service to detect that a passenger can transfer from bus to ship in Ancona, both stations (Port

Ancona and Central Station Ancona) should be registered with the same Hub. Currently, this needs to be done manually during the phase of data pre-processing (before registering stations on the service).

**Connection** – a term that represents an item in an agent’s timetable. A connection is defined on departure and arrival stations with its departure and arrival time, trip id, vehicle type, CO2 emission and length (distance between departure and arrival stations).

**TripData** – in order to send the right plan the service need some information from the user, such as user contact (email or phone number), trip departure and arrival city and time, selected plan.

## 2. AUTHORIZATION

Endpoints that are used for registering data (Agents, Stations, Connections) are protected using basic authorization. In order to be authorized, the client needs to send an authorization header:

*Authorization: Basic RUNoYWluOmVDaDQxbI9pbnRlcnlZw==*

## 3. E-CHAIN TIMETABLE SERVICE ENDPOINTS

### 3.1. AGENT ENDPOINT

#### 3.1.1. REGISTER AGENT

The first step is to register an agent using POST method on an Agent endpoint.

POST <https://f4bi.rathmann.hr/echain/api/agent>

Authorization: Basic RUNoYWluOmVDaDQxbI9pbnRlcnlzZw==

Content-Type: application/json

```
{  
  
  "name": "Jadrolinija",  
  
  "urlTemplate" :  
  
  "https://www2.jadrolinija.hr/Voyager2Web/#/?voy={tripId}&from={departureHub}&to={arrivalHub}&departure={departureTime}&lang=hr",  
  
  "timeFormat" : "mmdyyyyhhMM",  
  
  "stationIdentifier" : "id"  
  
}
```

- Name: agent name.
- UrlTemplate: used when creating a URL for ticket purchase. URL parameter should have (but not required) a placeholder for tripld, departure hub, arrival hub and departure time in square



brackets. If any of those parameters are not contained in the URL template, they are omitted when creating a ticket URL.

- Time format: how to format the departure time
- StationIdentifier: value used for the station when creating a URL. Supported values are Id and Name.

To update agent parameters, send another request on the same endpoint with updated parameter values.

### 3.1.2. REGISTERED AGENT CONNECTIONS

To get the registered connections for a specific agent use GET method

GET <https://f4bi.rathmann.hr/echain/api/agent/{agentName}/connections>

Authorization: Basic RUNoYWluOmVDaDQxbI9pbnRlcnIzZw==

Content-Type: application/json

- AgentName: the name of the agent

The method returns all connections of an agent where departure time is greater than current time.

## 3.2. STATIONS ENDPOINT

### 3.2.1. REGISTER STATIONS

After the agent has been registered it can register stations on the Stations endpoint. The endpoint accepts an agent name and a list of stations.

POST <https://f4bi.rathmann.hr/echain/api/stations>

Authorization: Basic RUNoYWluOmVDaDQxbI9pbnRlcnlZzw==

Content-Type: application/json

```
{  
  "agent": "Jadrolinija",  
  "stations": [  
    {  
      "latitude": "43.5044659",  
      "longitude": "16.4379773",  
      "name": "Split",  
      "hubName": "Split",  
      "id": "SPL",  
      "countryCode": "HR"  
    },  
    {  
      "latitude": "43.6205968",  
      "longitude": "13.5108701",
```

```
"name": "Port Ancona",  
"hubName": "Ancona",  
"id": "ANC",  
"countryCode" : "IT"  
}  
]  
}
```

- Agent: the name of the agent who is registering the stations
- Latitude: station latitude
- Longitude: station longitude
- Name: agent's station name
- Hub: hub name (if left empty, the station's name is used)
- Id: agent's station id
- CountryCode: (two-letter) country in which the station is located

### 3.2.2. REGISTERED STATIONS

To get the list of all registered stations on the service use:

GET <https://f4bi.rathmann.hr/echain/api/stations/registered>

### 3.3. CONNECTIONS ENDPOINT

#### 3.3.1. REGISTER CONNECTIONS

After the station registration, an agent can register its connections.

POST <https://f4bi.rathmann.hr/echain/api/connections>

Authorization: Basic RUNoYWluOmVDaDQxbI9pbNRIcnIzZw==

Content-Type: application/json

```
{  
  "agent": "Jadrolinija",  
  "connections": [  
    {  
      "departureStationId": "SPL",  
      "departureTime": "2021-11-05T11:15:00",  
      "arrivalStationId": "ANC",  
      "arrivalTime": "2021-11-05T23:25:00",  
      "vehicleType": "Ship",  
      "tripId": "12975",  
      "co2Emission": "123",  
      "lengthKm": "418"  
    },  
    {  
      "departureStationId": "ANC",
```

```
"departureTime": "2021-11-06T09:30:00",  
"arrivalStationId": "SPL",  
"arrivalTime": "2021-11-06T19:45:00",  
"vehicleType": "Ship",  
"tripId": "12977",  
"co2Emission": "125",  
"lengthKm" : "432"  
}  
]  
}
```

- Agent: the name of the agent who is registering the connections
- DepartureStationId: agent's departure station id
- DepartureTime: time of departure
- ArrivalStationId: agent's arrival station id
- ArrivalTime: time of arrival
- VehicleType: type of vehicle used
- TripId: agent's trip id
- CO2Emission: CO2 emission of the vehicle on the connection
- LengthKm: Distance between the departure and the arrival station

### 3.3.2. SEARCH

To search for available routes between stations use:

POST <https://f4bi.rathmann.hr/echain/api/connections/search>

Content-Type: application/json

```
{  
  
  "departureHubName": "Split",  
  
  "arrivalHubName" : "Ancona",  
  
  "departureTime" : "2021-11-05T08:30:00",  
  
  "maxTransferWaitTimeMinutes" : 60,  
  
  "maxRouteDurationDays" : 2,  
  
  "vehicleTypes" : ["Ship", "Bus"]  
  
}
```

- DepartureHubName: name of the departure hub
- ArrivalHubName: name of the arrival hub
- DepartureTime: time of departure
- MaxTransferWaitTimeMinutes: maximum time (in minutes) that is allowed to wait between voyage transfers
- MaxRouteDurationDays: maximum time (in days) that is allowed for the duration of an entire trip

- VehicleTypes: types of vehicles that are allowed on the trip

The result of the search is a list of available routes.

Each route object defines total length, total duration, total CO2 emission and a list of voyages on the route.

Each voyage contains a ticket URL (where the user can purchase a ticket for the selected voyage), length, duration, CO2 emission, agent name, vehicle type used on the voyage and a trip id.

Each voyage contains one or more connections.

### 3.3.3. SEARCHNEAREST

To search for available routes between stations without knowing the stations in advance you can use this API endpoint. The API will find the nearest departure and arrival station based on the input latitude and longitude and then perform a search for available routes between stations. The result of the search is the same as in the Search endpoint.

POST <https://f4bi.rathmann.hr/echain/api/connections/searchnearest>

Content-Type: application/json

```
{  
  
  "departureLatitude": "43.456252",  
  
  "departureLongitude": "13.415919",  
  
  "departureTime" : "2021-11-05T08:30:00",
```

```
"arrivalLatitude": "44.110614",  
  
"arrivalLongitude": "15.308840",  
  
"maxTransferWaitTimeMinutes": 60,  
  
"maxRouteDurationDays": 2,  
  
"vehicleTypes": ["Ship", "Bus"]  
  
}
```

- DepartureLatitude: latitude of the departure point
- DepartureLongitude: longitude of the departure point
- ArrivalLatitude: latitude of the arrival point
- ArrivalLongitude: longitude of the arrival point
- DepartureTime: time of departure
- MaxTransferWaitTimeMinutes: maximum time (in minutes) that is allowed to wait between voyage transfers
- MaxRouteDurationDays: maximum time (in days) that is allowed for the duration of an entire trip
- VehicleTypes: types of vehicles that are allowed on the trip



### 3.4. VEHICLE TYPES

To get all available vehicle types, use:

GET <https://f4bi.rathmann.hr/echain/api/vehicleTypes/registered>

## 4. E-CHAIN PLAN WEB SERVICE

### 4.1. MESSAGE PLAN

In order to start message plans a plan id is needed. This endpoint returns all published message plans in e-chain, with name and id.

GET <https://server1.entourance.com/api/public/plan>

### 4.2. START MESSAGE PLAN

Two endpoints are available to start a plan, based on recipient selected from the user (email or phone number). SMS sender must be enabled, to send messages via phone.

POST [https://server1.entourance.com/api/public/plan/phone/{phone\\_number}](https://server1.entourance.com/api/public/plan/phone/{phone_number})

POST <https://server1.entourance.com/api/public/plan/email/{email}>

Content-Type: application/json

```
{  
  
  "from_date": "2022-04-10",  
  
  "from_location": "ancona",  
  
  "language": "it_IT",  
  
  "plan_id": "23",  
  
  "privacy_disclosure": true,
```

```
"to_date": "2022-04-12",  
  
"to_location": "split"  
  
}
```

- From\_date: date of departure
- From\_location: city of departure. Must be a city of departure mapped in the selected plan
- Language: language of the user. English will be used as fallback If selected language is not available
- Plan\_id: id of the plan to be started
- Privacy\_disclosure: privacy acceptance
- To\_date: date of arrival
- To\_location: city of arrival. Must be a city of departure mapped in the selected plan