

# INSTRUCTION: 'CONNECTING System Interface'

## SERVICE: Hinterland Container Notification - Road

### TARGET GROUP: Road haulier



## Table of contents

|  |          |
|--|----------|
| <b>Introduction .....</b>  | <b>2</b> |
| Why the instruction 'Connecting System Interface'? .....   | 2        |
| Step-by-step plan .....  | 2        |
| <b>APIs for the haulier .....</b>  | <b>3</b> |
| Functionality .....  | 3        |
| TRACKING API .....   | 4        |
| Use cases .....  | 4        |
| 1. Requesting the status of a container from the terminal .....                                    | 5        |
| 2. Pre-notifying a container at the terminal .....   | 5        |
| 3. Notifying the terminal that the container no longer needs to be loaded or unloaded .....        | 7        |
| 4. Receiving information from the Terminal about the handling of the container and the visit ..... | 8        |
| 5. Only performing relevant process steps and inputting relevant data for each terminal .....      | 9        |

# INSTRUCTION: 'CONNECTING System Interface'

## SERVICE: Hinterland Container Notification - Road

### TARGET GROUP: Road haulier



## Introduction

### Why the instruction 'Connecting System Interface'?

This instruction serves to support the customer and the software supplier in quickly and efficiently realising a System Interface connection for the desired service. It contains the step-by-step plan and the information required for a successful realisation.

### Step-by-step plan

| Step  | Explanation  |
|---|--|
| You have requested this connection instruction via the developer portal at <a href="https://developer.portbase.com/services#hinterland">https://developer.portbase.com/services#hinterland</a>  | Any new versions of this document will be published on this portal.  |
| You request the connection application form, fill it out and sign it and then return it to sales@portbase.com (only by the customer, not as a software supplier)  | The application form is available at: <ul style="list-style-type: none"> <li>• <a href="https://www.portbase.com/en/services/">https://www.portbase.com/en/services/</a></li> <li>• Select 'Hinterland Container Notification'</li> <li>• Select 'Subscribe to service'</li> </ul> |
| You will receive your client ID and a client secret for the Hinterland Container Notification environments from Portbase Integration services.  | Your client ID and client secret will have to be included as key data in the API header for authentication and authorisation of the user.  |
| You build and check the APIs based on: <ul style="list-style-type: none"> <li>• the process steps and the coherence between APIs of HCN Road that are described in use cases in these connection instructions</li> <li>• the API documentation describing the data set that you exchange with the Terminal via HCN Road.</li> </ul> | The Hinterland Container Notification APIs have been developed for a JSON payload.<br><br>The API documentation is available here:<br><br><a href="https://hcn.pcs.portbase.com/api-documentation/index.html">https://hcn.pcs.portbase.com/api-documentation/index.html</a>        |
| Once the build has been completed, you test the message in a test environment. The environment that is used to realise the connection will be coordinated with you in collaboration with Portbase Integration services.   |  |
| After successful testing, you will be connected to production by Portbase Integration services  |  |

# INSTRUCTION: 'CONNECTING System Interface'

## SERVICE: Hinterland Container Notification - Road

### TARGET GROUP: Road haulier



## APIs for the road haulier

### Functionality

Regarding hauliers, the user types for the service Hinterland Container Notification (HCN) for the modality 'Road' are:


- Road transport planners - Transport planners at a road transport company who pre-notify the collection or delivery of a container at a terminal
- Drivers - Road hauliers are often self-employed drivers who must be able to enter, view and send information from the truck.

The APIs that can be used by the road haulier's system are listed in the API documentation that is summed up in the following three sections at <https://hcn.pcs.portbase.com/apidocumentation/index.html>. These apply to both the road transport planner and drivers under the heading 'Operator'.

- OPERATOR APIs
- TRACKING APIs
- OTHER APIs

This states:

- which APIs are available to the road transporter for performing the functionality within HCN Road
- which data is exchanged through this with the Terminal via HCN Road

 API requests are not relevant for you as a road haulier if they are not featured in the use cases in the document you are currently reading, even though they are listed in the above-mentioned API documentation. This is because it also features APIs for other modalities. An exception to this are the APIs which are referred to from an API that is relevant, for example Reference data APIs. Furthermore, you do not use APIs from the section DEPRECATED ROAD APIS if a new connection is involved.

# INSTRUCTION: 'CONNECTING System Interface'

## SERVICE: Hinterland Container Notification - Road

### TARGET GROUP: Road haulier



## TRACKING API

When you have submitted a container status request or pre-notification to the Terminal, in HCN you next listen to the response from the Terminal via a so-called Tracking API (also called listening API). This listening is a form of so-called long polling. We check whether the action of the terminal has actually resulted in a change in our system. You 'see' the result of this action or what has taken place in HCN in the listening API.

The advantages of this listening API over a deprecated PUSH API (POST API in your system) or a standard short polling (GET API in our system) are:

- You do not realise an endpoint to have data delivered from the terminal. This is safer because there is no entry point on your end for break-ins.
- Let's say that we deliver the data to you via PUSH API, but that this is unsuccessful. For example, because your endpoint is not reachable or it is reachable but the processing fails. In that scenario, HCN does not push the data again and the status update for your system is lost. This drawback is not applicable to the listening API.
- Regarding the processing, you will remain where you left off until you yourself indicate that the processing in your system has been successfully performed. You do this through a so-called Update position. You process the updates at your own pace. This for example allows you to schedule down time and synchronise your system with HCN at a later time.
- With this update position, you can even travel back in time and retroactively listen to updates. This allows you to trace and fix any errors in the processing and process the updates again so that your system displays the correct status.
- HCN sends the complete model. When Portbase adds a functionality, you will see this in the form of an addition to this model. You can ignore new features and have the flexibility to not listen to them until you need to. This means you are not bound by fixed API agreements. The HCN APIs will of course continue to be backwards compatible.

For the detailed operation of the listening API see <https://hcn.pcs.portbase.com/api-documentation/index.html#tag/Tracking-API>

## Use cases

To supplement the functionalities listed in the API documentation, the use cases are featured below, including a description of the logical order in which the APIs must be called in order to realise these use cases. The road haulier can use these use cases to among other things pre-notify containers and hinterland visits at the Terminal via HCN.

NB Unloading and delivering are synonyms for unmounting. Loading and collecting are synonyms for mounting. Both relate to the handling of the container in the hinterland.

# INSTRUCTION: 'CONNECTING System Interface'

## SERVICE: Hinterland Container Notification - Road

### TARGET GROUP: Road haulier



#### 1. Requesting the status of a container from the terminal

**I WANT** to request the status of a container from the terminal (provided the terminal supports this)

**SO THAT** I know what I need to take care of prior to the pre-notification.

NB To determine whether the terminal supports status requests, please see use case 5.

| Process steps   | API section              | API request              |
|---|--------------------------|--------------------------|
| <b>Main scenario:</b><br>1. Submit new status request | Operator API - handlings | Create detached handling |
| <b>Alternative A1:</b> Submit amended status request  | Operator API - handlings | Update detached handling |
| 2. Receive status information (via use case 4)        | Tracking API             | See use case 4           |

#### 2. Pre-notifying a container at the terminal

You can pre-notify a container in the alternative ways listed below. For pre-notification under an existing TAR see option IV. underneath the table below.

NB 1 To determine whether the terminal supports TARs, see use case 5.

NB 2 To determine whether the terminal supports time slots, please consult the functionality matrix document under Port Community service Hinterland Container Notification at <https://www.portbase.com/en/services/>

**I. I WANT** to pre-notify a container:

**SO THAT** the terminal knows that I want to load or unload the container(s) and when and I receive a TAR (if supported by the terminal) and a time slot from the terminal.

I perform this pre-notification:

- i. without first submitting a status request because I already know that I will be loading or unloading the container (Main Scenario or Alternative A1 in the table below) or
- ii. for a container for which I have already requested the status (Alternative A2 or A3)

**II I WANT** to pre-notify the terminal to unload or load the container during a different visit than the one initially specified by me in the pre-notification (Alternative A4)

**SO THAT** I receive a new time slot (if time slots are supported by the terminal) and I know at what time I am expected, thus avoiding unnecessary delay.

**III. I WANT** to inform the terminal of a correction to the pre-notification (if the terminal supports this, Alternative A5)

**SO THAT** I receive the correct status information in response to the pre-notification.

NB To determine whether the terminal supports making updates to a pre-notification, see use case 5.



# INSTRUCTION: 'CONNECTING System Interface'

## SERVICE: Hinterland Container Notification - Road

### TARGET GROUP: Road haulier



| Process steps   | API section   | API request   |
|---|---|---|
| I.i. Directly pre-notifying a container without first making a status request   |   |   |
| <b>Main scenario:</b><br>1. Submitting new pre-notification for a <i>new</i> visit  | Operator API - road-only<br>or<br>Operator API – voyages<br>Operator API – visits<br>Operator API - handlings | 1a. Pre-notify new handling<br>or<br>1a. Create voyage (modality Road)<br>1b. Create visit (modality Road)<br>1c. Create handling (preNotification=true and fill in existing visitId)<br>see [1] below table                    |
| <b>Alternative A1:</b> Submitting new pre-notification for an <i>existing</i> visit   | Operator API - handlings  | Create handling (preNotification=true and fill in existing visit ID)  |
| I.ii. Pre-notifying a container for which I have already requested the status   |   |   |
| <b>Alternative A2:</b> Attaching an individual status request to a <i>new</i> visit and converting it to a pre-notification       | Operator API - road-only<br>or<br>Operator API – voyages<br>Operator API – visits<br>Operator API - handlings | A2a. Pre-notify detached handling<br>or<br>A2a. Create voyage (modality Road)<br>A2b. Create visit (modality Road)<br>A2c. Move any handling (preNotification=true and fill in nextVoyageId/nextVisitId)<br>see [1] below table |
| <b>Alternative A3:</b> Attaching an individual status request to an <i>existing</i> visit and converting it to a pre-notification | Operator API - handlings  | Move any handling (preNotification=true and fill in nextVoyageId/nextVisitId)   |
| II. Loading or unloading a container during a different visit than the one initially pre-notified by me                           |   |   |
| <b>Alternative A4:</b> Submitting an existing pre-notification to an <i>existing</i> visit  | Operator API - handlings  | Move any handling (preNotification=true and fill in nextVoyageId/nextVisitId)   |
| III. Correcting pre-notification  |   |   |
| <b>Alternative A5:</b> Changing pre-notification  | A5a. Operator API – handlings and/or<br>A5b. Operator API - visits  | A5a. Update handling and/or<br>A5b. Update visit (if you want to adjust the ETA. Note: this applies to all handlings attached to the visit)   |
| Continuation <b>Main scenario</b>   |   |   |
| 2. Receiving status information, extra container handling data, TAR and time slot (via use case 4)                                | Tracking API  | See use case 4  |

[1] If you also use or are going to use other modes of transport than Road in the hinterland, you can also perform the process steps with alternative APIs. In this way, you keep the implementation of APIs consistent for all modes of transport.

# INSTRUCTION: 'CONNECTING System Interface'

## SERVICE: Hinterland Container Notification - Road

### TARGET GROUP: Road haulier



**IV. I WANT** to pre-notify the terminal that I want to load or unload an extra container during that visit using a Truck Appointment Reference (TAR) that I have already received for a truck visit **SO THAT** I receive a new time slot for that one single visit for all pre-notified containers (if the terminal supports time slots) and I know at what time I am expected, thus avoiding unnecessary delay.

1. First, pre-notify one container for a new visit (Main Scenario or Alternative A2 from the table above)
2. Next, specify other containers that are to be unloaded and loaded during this existing visit by means of the visit ID (Alternative A1, A3 or A4).  
You can do this **without** needing to wait for the TAR from the terminal (via use case 4). This is permitted, but not necessary. HCN unburdens you and ensures that the containers are pre-notified at the terminal under the same TAR.
3. In conclusion, you will receive status information, extra data related to the *container handling*, a TAR and time slot via use case 4.

### 3. Notifying the terminal that the container no longer needs to be loaded or unloaded

**I WANT** to notify the terminal that I will no longer be loading or unloading the container (if the terminal supports this)

**SO THAT** my overview with pre-notifications is up-to-date and the terminal is not confronted with a *no-show*.

NB To determine whether the terminal supports the cancellation of pre-notifications, please see use case 5.

| Process steps  | API section   | API request   |
|--|---|---|
| <b>Main scenario:</b><br>1. Cancellation pre-notification  | Operator API – handlings<br>Or<br>Operator API - visits | 1. Cancel any handling<br>or<br>1. Cancel visit; all pre-notifications of the visit will be cancelled. You do not need to perform step 1. |
| <b>Alternative 1:</b> Execute A4 in use case 2 ( <i>loading/unloading container during a different visit</i> ): Submit existing pre-notification on an <i>existing</i> visit | Operator API - handlings                                | Move any handling (preNotification=true and fill in nextVoyageId/nextVisitId)   |
| <b>Alternative 2:</b> Detach pre-notification from an existing visit   | Operator API - handlings                                | Detach handling   |

# INSTRUCTION: 'CONNECTING System Interface'

## SERVICE: Hinterland Container Notification - Road

### TARGET GROUP: Road haulier



#### 4. Receiving information from the Terminal about the handling of the container and the visit

**I WANT** to receive status information on the status request (if the terminal supports this) **SO THAT** I know what I need to take care of prior to the pre-notification.

and

**I WANT** to receive status information in response to the pre-notification **SO THAT** I know whether I am allowed to load or unload the container prior to arrival at the terminal, thus preventing unnecessary delay.

and

**I WANT** to receive extra data from the container handling

**SO THAT** among other things, I know when I may unload the container at the terminal without additional costs (after Cargo opening) and when I have to unload the container before Cargo closing or, for a container that needs to be loaded, the period within which I am permitted to load the container at the terminal (within the Commercial window).

and

**I WANT** to among other things receive the container number, time of handling and licence plate of the truck used for the loading or unloading of the container

**SO THAT**

at the moment of gate-in/gate-out, this so-called loading/unloading confirmation provides me with proof of which container I have discharged or loaded, with which mode of transport this occurred and at which terminal.

NB. 1. To determine whether the terminal supports a gate-in/gate-out, please consult the functionality matrix document under Port Community service Hinterland Container Notification at <https://www.portbase.com/en/services/>

| Process steps   | API section  | API request   |
|---|--------------|---|
| 1. Receiving:<br>a. updated status information on the status requests<br>b. updated status information on the pre-notifications<br>c. extra data <i>container handling</i><br>d. TAR and time slot for the visits<br>e. unloading/loading confirmations | Tracking API | 1. Listen to updates:<br>a. from the handling Declaration in the detached Handlings<br>b. from the handling Declaration of the handlings of the visits in the road Voyages<br>c1. from the handling Status in the detached Handlings<br>c2. from the handling Status of the handlings of the visits in the road Voyages<br>d. from the planned Visit Data of the visits in the road Voyages<br>e. of the handling Result of the handlings of the visits in the road Voyages<br>2. Update position<br><i>see [2] below table</i> |

NB.2. For API requests Reset position and Disconnects client see [API documentation](#).

[2] Updates can be offered consolidated in batches via one single Response to the List to update API Request. This means that you only perform an Update position after you have successfully processed all updates related to all use cases within the batch of all visits and handlings, otherwise you will miss updates in your processing. An Update position can be compared to a bookmark for remembering where you left off. This way, you can 'read' at your own pace.



# INSTRUCTION: 'CONNECTING System Interface'

## SERVICE: Hinterland Container Notification - Road

### TARGET GROUP: Road haulier



#### 5. Only performing relevant process steps and inputting relevant data for each terminal

**I WANT** to know whether the terminal supports correcting a pre-notification or cancelling a pre-notification before I submit a status request.

**SO THAT** I do not perform any unnecessary work and know which terminal I need to inform outside of the system when the pre-notification needs to be corrected or cancelled.

NB To determine *how* you can best inform the terminal outside of the system in these cases, please consult the functionality matrix document under Port Community service Hinterland Container Notification at <https://www.portbase.com/en/services/>

Before I submit a status request or pre-notification I WANT to know which data I can or need to enter per type of operation for the container for each terminal

SO THAT I do not perform unnecessary work and do not receive unnecessary error messages after I have submitted the pre-notification or status request

NB The types of handlings per mode of transport are: full/unload, full/load, empty/unload and empty/load

Example: There are terminals that support the so-called 'loading on size type'. In that case, an empty container that is to be loaded may be pre-notified under omission of the container number, stating the container size/type and, of course, the booking number. In that case, the container number is reported back by the Terminal via use case 4 (step 1e. received loading/unloading confirmations).

| Process steps   | API section        | API request   |
|---|--------------------|---|
| 1. Request which process steps are supported per terminal | Reference Data API | Get terminal  |
| 2. Request which data is relevant for each terminal       | Reference Data API | Get terminal (data from terminal Settings per handling type (loading/full Booleans) per mode of transport |