

Quick start guide

Version: Octobre 4th 2017

Environment: Sandbox



Welcome

Welcome to the quick start guide for the Secure Container Release application for the port of Antwerp. This quick start guide will show you briefly how you can use the Secure Release API to release, transfer and securely handover full import containers, using our blockchain-based application.

Our product improves the container release process in the Port of Antwerp, to secure the handover of containers on deep sea terminals to the correct receiving party. Currently this release process is based on PIN codes. By transferring the pick-up rights via blockchain, we create a digital twin for that specific container, providing all involved parties with real time data like location, temperature, next transport mode, ... and logging all relevant details (audit trail), throughout its journey to the end customer. By doing so, a unique micro ecosystem is created around a single container where parties involved can exchange information, enabling companies to monetize data they have available. Our solution is safer, simpler, requires less administration and generates valuable data that can be made available to all entitled parties involved in a specific container transport.

Our solution is accessible through apps and portals relying on a distributed blockchain platform with smart contracts. This platform can also be used by 3rd parties. Therefore we are looking for ISV's that offer solutions to the above listed customers and integrators helping to link legacy systems with our platform. We offer these 3rd party developers a blockchain backend to complement their solutions and facilitate transaction sharing with other companies within the ecosystem.

Our solution improves operational efficiency of global transport and make handovers of assets as containers more secure. By transferring pickup rights by blockchain transactions only fully authorised recipients can pick up goods reducing theft, insurance fees and public image damage. Thanks to more and up-to-date information, processes can be optimized.

Using the Secure Container Release backend

The core of the Secure Container Release application are smart contracts, running on a decentralised, permissioned blockchain infrastructure. The smart contracts use a consensus mechanism to make sure security, permissions and data integrity are maintained.

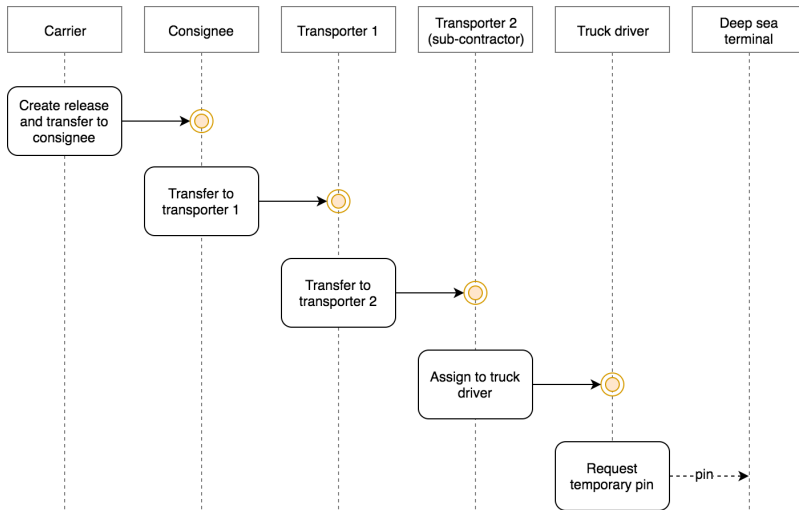
To make things more easy for you, your application(s) can talk to these smart contracts using our API. This API exposes all functions of the smart contracts. In the sections below we explain a typical scenario on how to use the API to securely release full containers.

The basic flow

In the basic flow, the carrier releases the container. To do this, the carrier creates a pickup right and transfers it to the consignee (e.g. a forwarder). The consignee will then transfer the pickup right to a transport company, for instance a transport company. The transport company can transfer the pickup right to another transport company (a sub-contractor) or can use the pickup right to pick up the container on the deep sea terminal. In the latter case, the pickup right is assigned to a truck driver and he can use this to retrieve the container. To do this, the truck driver can generate a temporary pin (or use another kind of identification mechanism, depending on the infrastructure of the terminal).

So, the pickup right is created in the Secure Container Release-application and represent the permission or right to retrieve a container from a deep sea terminal. This pickup right can be transferred to other parties (companies or truck drivers) and then the pickup right will be 'owned' by this party. However, when the pickup right was transferred by you in the past, you will be able to retrieve information about the container, for instance whether it was retrieved. The API to retrieve this kind of information will be added later on.

So, the basic flow looks like this:



The yellow dot indicates who has the pickup right in every stage of the process. The pickup right is "tokenized" and transferred through the chain to the truck driver who will transport the container. Every party that had the pickup right also retains certain permissions to access data on this container, even when the pickup right was transferred to the next party.

Using Postman to access the API

With [Postman](#), it's very easy to access and test the API:

- Connect to the test-URL: <https://vps10.t-mining.be:4430/tmining/api/v1>
- Add the API-token of the user you want to impersonate in the url with the `api_token`-key (see below)
- Add the fields you want to include in the body as `x-www-form-urlencoded`

Postman provides an easy, graphical interface to test API's:

Key	Value	Description
type	pickup	
isoNumber	TCLU9885571	
blNumber	MSCUZW482401	
location	1742	

In the test-environment (using the url above) you can use a number of tokens to act like different parties in the chain:

Role	User	API token
Carrier	carrier-user-1	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQ6IiwiaWF0IjoxNTA1NDUwNTQzLCJpc3MiOiJULU1pbmluZyJ9.zFyzQoG4RchUZrcVBH9S8rIPgw39ZOx6pXGwnqc4j2Q
Terminal	terminal-user-1	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQ6IiwiaWF0IjoxNTA1NDUwNTQzLCJpc3MiOiJULU1pbmluZyJ9.rwvYt-M7makrJUyE2eIYisbx2NLt8K7uN5WDAXEKp8

Forwarder	forwarder-user-1	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQiOiJ0ZXN0Y2hhaW5fcGFydGljaXBhbnRfMDE3liwiaWF0ljoxtNTA1ND EwNTQ2LjJpc3MiOiJULU1pbmluZyJ9.P8Ct8T3uRq-xEo3s_IrRilfRV0feTSrly6wXJV6LjS0
Trucking company or dispatcher	trucking-user-1	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQiOiJ0ZXN0Y2hhaW5fcGFydGljaXBhbnRfMDE2liwiaWF0ljoxtNTA1ND EwNTUwLjJpc3MiOiJULU1pbmluZyJ9._2dgD5LfiHUZskkDL1opXfGyLA3IKexHQKmcu ya5WNw
Truck driver 1	driver-1	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQiOiJ0ZXN0Y2hhaW5fcGFydGljaXBhbnRfMDE1liwiaWF0ljoxtNTA1ND EwNTU4LjJpc3MiOiJULU1pbmluZyJ9.aQ8H1Dn233OqQVRKCh00hnue-oXFCFJs-rkSVt95QNg
Truck driver 2	driver-2	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQiOiJ0ZXN0Y2hhaW5fcGFydGljaXBhbnRfMDE0liwiaWF0ljoxtNTA1ND ExMDEwLjJpc3MiOiJULU1pbmluZyJ9.x3qlF1SbuikVWgeifx3iNV5oDPB2vmeUTHF1g7aRcLs
Truck driver 3	driver-3	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQiOiJ0ZXN0Y2hhaW5fcGFydGljaXBhbnRfMDEzliwiaWF0ljoxtNTA1NDE xMDEzLjJpc3MiOiJULU1pbmluZyJ9.D-99lj1o5hS8BLn9nwz5JF-HQyHHF9ycLnfeRuNuUUI
Truck driver 4	driver-4	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQiOiJ0ZXN0Y2hhaW5fcGFydGljaXBhbnRfMDEyYliwiaWF0ljoxtNTA1NDE xMDE3LjJpc3MiOiJULU1pbmluZyJ9.pZM-8YIKdxyx41UbnTX8mY3GGR4DrBrplyNgaFI XSYI
Truck driver 5	driver-5	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50SWQiOiJ0ZXN0Y2hhaW5fcGFydGljaXBhbnRfMDE1liwiaWF0ljoxtNTA1NDE xMDI1LjJpc3MiOiJULU1pbmluZyJ9.AYBDpBndKpnD2d0xWGHZiYkJCzADaIL6M5exhd0k-0I

Releasing the container (for carriers)

The carrier can release a container by creating a container-contract and transferring it to the consignee. To do this the carrier executes a POST-call to the releases-webservice:

POST
https://vps10.t-mining.be:4430/tmining/api/v1/releases?api_token=eyJhbGciOiJIUzI1NiIsInR5...
Params
Send
Save

Authorization
Headers (2)
Body
Pre-request Script
Tests
Cookies
Code

form-data
x-www-form-urlencoded
raw
binary

Key-Value Edit

```

originalId:2
isoNumber:QSTU1234578
blNumber:QS5634
containerType:20GP
loadStatus:full
grossWeight:3000
seals:none
portOfLoading:CNSGP
portOfDestination:BEANR
placeOfDischarge:1742 MPET DEURGANCKTERMINAL WEST
containerOwner:Carrier
consignee:Forwarder
validFrom:2017-09-12T07:48:00Z
validUntil:2017-12-07T00:00:00Z
messageNumber:none

```

Body
Cookies
Headers (4)
Tests
Status: 200 OK
Time: 1901 ms
Size: 122 B

So, in the body we include this data:

```

originalId:2
isoNumber:QSTU1234578
blNumber:QS5634
containerType:20GP
loadStatus:full
grossWeight:3000
seals:none
portOfLoading:CNSGP
portOfDestination:BEANR
placeOfDischarge:1742 MPET DEURGANCKTERMINAL WEST
containerOwner:Carrier
consignee:Forwarder
validFrom:2017-09-12T07:48:00Z
validUntil:2017-12-07T00:00:00Z
messageNumber:none

```

This will do three things:

- Create a container object, which can later be retrieved by calling the containers-service
- Creates a pickup right for this container object, which can be retrieved by calling the pickup_rights-service
- Transfer this pickup right to the consignee indicated in the body

Please note:

- You have to use the carrier-token to do this
- You have to use "x-www-form-urlencoded"-parameters in the body (you can use "Bulk edit" to input these more easily)
- The service will return a status-code 200 of this succeeded
- The service will return status-code 500 when the container is already known in the system for this bill of lading
- If you want to use other ISO-codes for the container type, you can retrieve the known types using the container_types-webservice
- The codes of the terminal and companies (Forwarder, Carrier, ..) are case-sensitive.
- The originalID and messageNumber were specifically added for the proof-of-concept and will be depreciated in the next version. You can leave these empty.

Afterwards, you can retrieve the data of the pickup right using the pickup_rights-service. If you call a GET-operation on /api/v1/pickup_rights/, you get the current start of the pickup right in the body. This can be used to retrieve the state of the pickup right, after other parties updated the data. The terminal can for instance update the availability date, the carrier the block, etc.

We created a pickup right above, so let's check whether it is returned by the pickup_rights-service:

GET

https://vps10.t-mining.be:4430/tmining/api/v1/pickup_rights?api_token=eyJhbGciOiJIUzI1Ni...

Params

Authorization

Headers (2)

Body

Pre-request Script

Tests

Type

No Auth

Body

Cookies

Headers (6)

Tests

Status: 200 OK

Pretty

Raw

Preview

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

[

{

"id": 1,

"containerId": 2,

"transferrerId": 0,

"pickupLocationId": 38,

"createdAt": "1970-01-18T10:10:06.817Z",

"validFrom": "2017-09-12T07:48:00.000Z",

"validUntil": "2017-12-07T00:00:00.000Z",

"blocked": false,

"available": false,

"transfereeId": 2,

"assigneeId": 0,

"pickedUp": false,

"delivered": false

},

{

"id": 2,

"containerId": 3,

"transferrerId": 0,

As you can see the pickup rights are returned, but some data is only indicated by id's. This is the case for:

Field	Refers to	Service
containerId	a container object	GET /api/v1/containers
transferrerId	an organization (company)	GET /api/v1/organizations
transfereeId	an organization (company)	GET /api/v1/organizations
assigneeId	a user (a truck driver)	GET /api/v1/users
pickupLocationId	a location (a quay)	GET /api/v1/locations

So, you can retrieve these data by calling the corresponding services, e.g. for the containerId:

GET

https://vps10.t-mining.be:4430/tmining/api/v1/containers?api_token=eyJhbGciOiJIUzI1NiIsIn...

Params

Authorization

Headers (2)

Body

Pre-request Script

Tests

Type

No Auth

Body

Cookies

Headers (6)

Tests

Status: 200 OK

Pretty

Raw

Preview

JSON

```

17 {
18   "id": 2,
19   "isoNumber": "QSTU1234578",
20   "blNumber": "QS5634",
21   "containerType": "20GP",
22   "loadStatus": "full",
23   "grossWeight": 3000,
24   "seals": "none",
25   "portOfLoading": "CNSGP",
26   "portOfDestination": "BEANR",
27   "placeOfDischargeId": 38,
28   "containerOwnerId": 4,
29   "creatorOrganizationId": 4,
30   "blocked": false
31 },

```

Please note that we did not change the token until now. This was not necessary since the carrier can also retrieve the information of the containers and pickup rights that were created by him/her. If you change the token to the token of the forwarder, you will get this list too since all containers are transferred to the (one and only) forwarder in this environment.

Transferring the pickup right (for forwarders and transport companies)

The consignee (most of the time the freight forwarder) can transfer the pickup right to the transport company that will execute the transport. But the transport company can also transfer the pickup right to another transport company. This is for instance useful when the transport order will be executed by a subcontractor. For the consignee, this process is transparent in the sense that all information will flow back to the forwarder, irrespective of who did the transport. And in the other way: every update of the forwarders or carrier, will be sent to the correct transport company.

As indicated above, the list of pickup rights that are currently assigned to you (or your customer) can be retrieved using the pickup_rights-API.

Please make sure you change the token to the token of the forwarder, at this stage of the process!

So, we execute a GET-operation on the pickup_rights-service:

GET ▼ https://vps10.t-mining.be:4430/tmining/api/v1/pickup_rights?api_token=eyJhbGciOiJIUzI1NiI... Params

Authorization Headers (2) Body Pre-request Script Tests

Type No Auth ▼

Body Cookies Headers (6) Tests Status: 200 OK

Pretty Raw Preview JSON ▼

```

1 [
2   {
3     "id": 1,
4     "containerId": 2,
5     "transferrerId": 0,
6     "pickupLocationId": 38,
7     "createdAt": "1970-01-18T10:10:06.817Z",
8     "validFrom": "2017-09-12T07:48:00.000Z",
9     "validUntil": "2017-12-07T00:00:00.000Z",
10    "blocked": false,
11    "available": false,
12    "transfereeId": 2,
13    "assigneeId": 0,
14    "pickedUp": false,
15    "delivered": false
16  },
17  {
18    "id": 2,
19    "containerId": 3,
20    "transferrerId": 0
  }
]

```

This API will return all pickup rights assigned to you, so you can use this list to loop over the containers and transfer them one by one to the correct transport company. The details of the container, organisations and locations can be retrieved using the services listed in the former section.

In this case we will transfer the pickup right to the trucking company, who has id 5. This can be done by calling the transfer-operation on a pickup right:

PUT ▼ https://vps10.t-mining.be:4430/tmining/api/v1/pickup_right/1/transfer?api_token=eyJhbGciOiJIUzI1NiI... Params Send ▼

Authorization Headers (2) Body ● Pre-request Script Tests

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

	Key	Value	Description	..
<input checked="" type="checkbox"/>	transfereeld	5		
	New key	Value	Description	

Body Cookies Headers (4) Tests Status: 200 OK Time: 997 ms

So, with this call, we transferred the pickup right with id 1 (as indicated in the url) to the organisation with organisation id 5 (as indicated in the body). Please note that you have to use a PUT-operation here.

If we now retrieve the pickup rights, we will see that the transfereeld has indeed changed to 5:

GET

https://vps10.t-mining.be:4430/tmining/api/v1/pickup_rights?api_token=eyJhbGciOiJIUzI1NiI...

Params

Send

AuthorizationHeaders (2)BodyPre-request ScriptTests

TypeNo Auth

BodyCookiesHeaders (6)TestsStatus: 200 OKTime: 2037 ms

PrettyRawPreviewJSON

```
1  [
2  {
3    "id": 1,
4    "containerId": 2,
5    "transferrerId": 4,
6    "pickupLocationId": 38,
7    "createdAt": "1970-01-18T10:10:06.817Z",
8    "validFrom": "2017-09-12T07:48:00.000Z",
9    "validUntil": "2017-12-07T00:00:00.000Z",
10   "blocked": false,
11   "available": false,
12   "transfereeId": 5,
13   "assigneeId": 0,
14   "pickedUp": false,
15   "delivered": false
16 },
17 {
18   "id": 2,
19   "containerId": 3.
```

Assigning the pickup right to a driver (for trucking companies)

To assign the pickup right to a truck driver, the assign-webservice is used. This will assign the pickup right to a truck driver, who can only use it to retrieve the container from the terminal. All other operations (revoking, assigning to another driver, etc) are reserved for the truck company.

Please make sure you change the token to the token of the transport company, at this stage of the process!

To get the truck driver, we retrieve the users from the platform and select the users with type "driver":

GET https://vps10.t-mining.be:4430/tmining/api/v1/users?api_token=eyJhbGciOiJIUzI1NiIsInR5cCI6I...

Authorization Headers (2) Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (6) Tests Status: 200 OK Time: 1267 ms

Pretty Raw Preview JSON

```

42 {
43   "id": 5,
44   "userName": "driver-1",
45   "firstName": "x5",
46   "surName": "y5",
47   "email": "z5",
48   "organizationId": 5,
49   "type": "Driver",
50   "owner": "F5477298AC6A3DC2AC412417449A09FDB89578CE"
51 },
52 {
53   "id": 6,
54   "userName": "driver-2",
55   "firstName": "x6"

```

And then we can assign the pickup right (with id 1) to the driver (with id 5):

PUT https://vps10.t-mining.be:4430/tmining/api/v1/pickup_right/1/assign?api_token=eyJhbGciOiJIUzI1NiIsInR5cCI6I...

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> assigneeld	5	
New key	Value	Description

Body Cookies Headers (4) Tests Status: 200 OK Time: 1256 ms

For barge and rail, this step is not necessary. The pickup right is linked to the company, and this is sufficient.

Retrieving the container on the terminal

When the truck driver arrives on the terminal, he has to retrieve a temporary pin. This can be done with the request-service. To get the list of containers assigned to that specific truck driver, again, the pickup rights-service can be used, with the token of that particular driver. Then, the pickup right id can be used to request a temporary pin:

GET https://vps10.t-mining.be:4430/tmining/api/v1/pickup_right/1/request?api_token=eyJhbGciOi... Params Send

Authorization Headers (2) Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (6) Tests Status: 200 OK Time: 1868 ms

Pretty Raw Preview JSON

```
1 {
2   "temporaryPincode": 66153,
3   "temporaryPincodeValidUntil": "2017-09-21T09:09:43.388Z"
4 }
```

As you can see, the temporary pin is generated and it has a very limited validity (half an hour at the moment). The format of the temporary pin can change so we suggest to make sure 10 digits are available on the screen.

The temporary pin will be checked by the terminal, after the driver has typed in the pin in a terminal on-site. When the pins match, the container is delivered to the truck driver.

At the moment, you can request a new temporary pin at any time.

In the future, the use of temporary pins will probably be discouraged in favour of other means of identification (biometric devices, etc). However, at the moment, the temporary pin provides an easy way to ensure compatibility with the existing systems of the terminals.

Other common scenario's

Revoking a pickup right

When a pickup right is transferred to another party and that party has not assigned the pickup right to a driver yet, the pickup right can be revoked and transferred to another party, using the `revoke_transfer-webservice`:

PUT /api/v1/pickup_right/{id}/revoke_transfer

Reassigning a pickup right to another driver

The re-assign a pickup right to another truck driver, the truck company has to revoke the assignment and assign it again to the new driver:

PUT /api/v1/pickup_right/{id}/revoke_assignment