



# **CTC Asset – API description for carriers**

## **Version 1.2.1**

This document may not be reproduced or transmitted in any form, in whole or in part, without the express written permission of Trusted Carrier GmbH & Co. KG.

Trusted Carrier GmbH & Co. KG  
Breitenbachstraße 1  
D-60487 Frankfurt  
Telefon: +49 (0) 89 890 569 – 280  
Internet: [www.trusted-carrier.com](http://www.trusted-carrier.com)

# Versions

Version	Date	Type of change
1.2.1	02.07.2024	Added "Asset reverted" notification
1.2	07.12.2023	New CI
1.1	01.08.2023	Addition of Staging-system-exclusive methods
1.0	07.07.2023	First official verison
		-
		-

# Table of contents

1. General information .....	4
1.1 CTC Staging system .....	4
1.2 Structure of requests.....	4
1.3 Structure of answers .....	5
1.4 Vehicle component types & Masterdata fields.....	6
1.5 Technical ownership of a vehicle component.....	6
1.6 “Editing version” & “Published version” of a vehicle component.....	6
1.7 Notifications to external systems & security mode.....	6
2. Methods to edit vehicle component data .....	9
2.1 Create new asset.....	9
2.2 Update asset.....	11
2.3 Upload file.....	13
2.4 Commit Asset .....	15
2.5 Revoke Commit .....	16
2.6 Revert asset.....	17
2.7 Disable asset .....	18
3. Methods to access vehicle component data.....	20
3.1 Get all assets .....	20
3.2 Get editing version.....	21
3.3 Get editing version files.....	22
3.4 Get published version .....	23
3.5 Get published version files .....	25
4. Methods exclusive to Staging system .....	27
4.1 Approve asset.....	27
4.2 Reject asset.....	28
5. Updates to external systems .....	30
5.1 Completed Asset check .....	30
5.2 Reverted Asset check .....	31

# 1. General information

The CTC Asset API is exposing all functions to users that are also available in the CTC web frontend.

## 1.1 CTC Staging system

To test implementing the CTC Asset API, we recommend using the CTC Staging system, which is fully separated from the Production system. Please contact us if you are interested in an account.

The Staging system uses different URIs compared with the Production System.

Example:

Staging: <https://ctcapi-staging.trusted-carrier.com/api/v1/restAPIs/newAsset>

Production: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/newAsset>

From here on, this document only names the Production URIs (except for Staging-only methods mentioned in chapter 4).

## 1.2 Structure of requests

All requests are POST. The header must contain the information about the content type.

If the security mode (see also chapter 1.7) is “Standard”, the authorization happens using the CTC-generated public key in the field “publicKey” within the body.

If the security mode is “JWT”, the token must be part of the “authorization” field within the header and must include your company ID.

Example header (with “JWT”):

```
{  
  "Content-Type": "application/json;charset=UTF-8",  
  "Authorization": "Bearer <JSON Web Token>"  
}
```

Example body (with “Standard”):

```
{  
  "publicKey": "<ctc-generated-publickey>",  
  "data": {  
    <data>  
  }  
}
```

Your keys are available on the page “API settings” in the “Admin” module.

For most methods, the “data” object must contain the unique Asset ID of the vehicle component, which is created when a new vehicle component is created using *newAsset*. The “id” is used to identify the correct vehicle component, which is to be accessed by the method.

In cases where the method can only return information about the current published version, it is also possible to use the licence plate + nationality, or the container number, as identifiers.

When a vehicle component is to be updated, the “data” object must contain the “incrementalVersion” field, which must match the version currently stored in the CTC database. This is to prevent overwriting of newer information. Every time the masterdata of a vehicle component is updated (via API calls *updateAsset* / *uploadFile*, or via Frontend edit), the *incrementalVersion* is increased by 1.

### 1.3 Structure of answers

The answer starts with an error code. If the request was successful, the “code” field is 0. There may be a “warnings” field, if, e.g., some part of the request is no longer supported and only kept for compatibility reasons.

There may be an additional “payload” field, containing details about the vehicle component.

```
{
  "error_code": {
    "code": "0",
    "message": "Success",
    "warnings": "ABC is deprecated"
  },
  "payload": {
    <payload>
  }
}
```

If there are any errors, the “error\_code” field contains a “message” and a “details” field which describes the error.

Example:

```
{
  "error_code": {
    "code": "4001",
    "message": "Invalid parameters",
    "details": "Asset not found"
  }
}
```

If the request is not successful, no changes to a vehicle component are done.

## 1.4 Vehicle component types & Masterdata fields

You can find more detailed information about the vehicle component types existing on the CTC platform here: <https://www.trusted-carrier.com/documentation/asset-fields.pdf>

The file contains:

- Vehicle component types
- TypeID of each vehicle component type
- Available masterdata fields for each vehicle component type
- Name (slugs) of each masterdata field, including valid inputs for each field

## 1.5 Technical ownership of a vehicle component

When a vehicle component is created, the company of the user becomes the “technical” owner on the CTC platform. While the “real” owner of a vehicle component (i.e. the information as written in the vehicle registration) may be different, the technical ownership is always given to the vehicle component creator.

API access to editing vehicle components is restricted to the technical owner.

The technical ownership can not be transferred to another company.

## 1.6 “Editing version” & “Published version” of a vehicle component

The “published version” is the most-recent approved version a vehicle component. When a location queries for vehicle information, only the information in the “published” version is provided.

When there are updates to a vehicle component, an “editing version” is automatically created, if it does not exist already.

While a vehicle component is in status “waiting for review”, no changes can be done until the Support approves/rejects the vehicle component or the request for approval is revoked.

## 1.7 Notifications to external systems & security mode

When a support process for a vehicle component is finished, the corresponding information is sent to your system, if an endpoint has been defined on the screen “API settings” in the “Admin module” of your CTC account (see chapter 5).

You can choose the mode of security on the “API settings” screen:

- “Standard”: This is the basic security setting, allowing authorization with a pre-shared key sent in the body of the request.
  - URL: In this field, the endpoint of the external system can be specified. If this field has a value, the ERP key field must also be filled
  - ERP key: CTC will send the value entered here towards the external system, when communication is initiated by CTC
  - CTC key: CTC creates a key, that the external system needs to send in all requests when communication is initiated by the external system
- “JWT”: This extends the security with JSON Web Tokens (JWT) in the authentication header, which must be sent by both CTC and the external system in each request.
  - URL: In this field, the endpoint of the external system can be specified. If this field has a value, the ERP JWT key field must also be filled
  - CTC JWT key: CTC will create a private/public keypair in RSA2048 format
  - ERP JWT key: The value entered here must be in RSA2048 & ASN.1 format  
Each location will be have security key generated by CTC

If JWT is enabled:

- Both carrier and CTC APIs will be required to send an authorization header with the value: Bearer <JWT token>
- JWT settings
  - RS512
  - Contents of the payload
    - iss: sender URL
      - CTC will use one of the following: <https://ctc.trusted-carrier.com/> (production system), <https://ctc-staging.trusted-carrier.com/> (staging system)
      - While “iss” is not a mandatory value, the external system is advised to provide it and it will be stored in the CTC logs
    - cid: your company ID
    - iat: unixtime seconds
      - Timestamp when the token was issued
      - Must be before current time
    - exp: unixtime seconds
      - Timestamp when the token expires
      - Maximum of 60 seconds into the future
- Example token sent by an external system:
  - `eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MjAwMDc5NjYsImklhdCI6MTcyMDAwNzkwNiwiiaXNzIjoieW91cmNvbXBhbnkuY29tliwiY2lkIjoxMjN9.HZSfVBOIvDSS3vRnTF-4CaV6OF4GEZF5ETIRwhc9qihFrHfV7nnv1V25n3laiTXtsdvqJP5GKuWkflummqOqq7WxU7Du-9WUOdyexQAt-gllS6nl7ojKGde7TqZ6JvIC-OE-f6guVkw2mWJkycSFe_g7QBEeQ9bXa3SlzW9dydQEQx_fo8W3DLJau5bn-vANhVLP5HEo-6YEd0aomph7Oxvtrt9h5BiOONtUThy-`

jpnBk\_kRO9vyblOwcOQYk0\_pIHHGRTNEhGGOxIjhokU5p9qlsc-  
P1TiPQIC8FgleW3ydwOxJt5\_-g\_YrqzrhcOEdlss\_DzDkYP7\_PQWXNZTidg

- Decoded red part (header):

```
{  
  "alg": "RS512",  
  "typ": "JWT"  
}
```
- Decoded green part (payload):

```
{  
  "exp": 1720007966,  
  "iat": 1720007906,  
  "iss": "yourcompany.com",  
  "cid": 123  
}
```
- Blue part is to verify the signature
- The request will be rejected with a 401 response if:
  - Authorization header is missing or malformed, and the account has JWT enabled
  - JWT isn't verifiable using the public key provided by location
  - JWT is expired



## 2. Methods to edit vehicle component data

### 2.1 Create new asset

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/newAsset>

#### Functional description

This method creates a new vehicle component in status “Unapproved changes”. The value of “incrementalVersion” is set to “1”.

If a vehicle component with the same identifiers already exists in CTC Asset, a new vehicle component will still be created. The existing vehicle component will remain unchanged.

Data:

- *type*: mandatory  
Numeric value of the vehicle component type (see <https://www.trusted-carrier.com/documentation/asset-fields.pdf> for the numeric values)
- *licence\_number*: mandatory  
Vehicle registration plate or container number (available characters: A-Z, 0-9, -, Ä, Ö, Ü; capital letters only; no spaces)
- *nationality*: mandatory for non-container types, non-applicable for container types  
Nationality of the vehicle component in ISO 3166-1 alpha-2 format
- *owner*: optional.  
If “owner” field does not exist, the 5 sub-fields “name”, “street”, “zipCode”, “city”, “country” will be automatically filled with the information provided in the vCard service.  
If “owner” field exists, all 5 sub-fields must exist too.

If a new vehicle component is successfully created, the answer contains the unique Asset ID of the vehicle component, which is then used as the identifier in further methods to get/update data.

While there is no strict technical enforcement of certain licence plate / container number patterns, please adhere to the following rules:

- Use the “-“ for German licence plates to distinguish between the city code and the following group of letters (e.g., use “M-AB1234” for a German vehicle registration plate)
- Do not use “-“ for any other nationalities (e.g., use “R945DX” for a Dutch vehicle registration plate)
- Do not use “-“ in container numbers to separate the control number (e.g, use “CTUU1234567”)

#### Example data exchange

User story: Create a new semi-truck (type 100) with licence plate "C-TC9208" (DE) and set a custom vehicle owner

Request sent to CTC Asset:

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "type": 100,
    "licence_number": "C-TC9208",
    "nationality": "DE",
    "owner": {
      "name": "Custom owner",
      "street": "Musterstraße 12",
      "zipCode": "12345",
      "city": "City",
      "country": "DE"
    }
  }
}
```

Response from CTC Asset:

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "vehicle": {
      "id": 2224,
      "type": 100,
      "incrementalVersion": 1,
      "createdAt": 1687939396.318,
      "updatedAt": 1687939396.408,
      "status": "pending",
      "owner": {
        "id": 2220,
        "name": "Custom owner",
        "street": "Musterstraße 12",
        "zipCode": "12345",
        "city": "City",
        "country": "DE"
      }
    },
    "currentVersionID": 2437,
    "currentVersion": {
      "id": 2437,
      "createdAt": 1687939396.339,
      "updatedAt": 1687939396.339,
    }
  }
}
```

```
"status": "pending",
"reviewDetails": null,
"fields": {
  "licence_number": "C-TC9208",
  "nationality": "DE",
  "vehicle_type": 100,
  "vehicle_owner": "{\"name\": \"Custom owner\", \"street\": \"Musterstraße 12\", \"zipCode\": \"12345\", \"city\": \"City\", \"country\": \"DE\"}"
}
}
}
```

## 2.2 Update asset

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/updateAsset>

### Functional description

This method updates masterdata of an existing vehicle component. It is not possible to update vehicle components that are technically owned by other platform participants.

*updateAsset* increases the value of “incrementalVersion” by 1.

Data:

- *id*: mandatory  
Unique ID of the vehicle component
- *incrementalVersion*: mandatory  
Current version of this vehicle component
- *fields*: mandatory  
All fields that should be updated. If a field is not mentioned in “fields”, its data will remain unchanged. To delete field information, send an empty string "".

The values of each field to be updated have to conform with the requirements as defined in “Annex 1: Driver + Vehicle Master data v2.2.xlsx”, tab “Vehicle master data, column “Validation Details”. There may also be cross-dependencies between fields. E.g., setting the number of chambers of a tank requires also setting the volume and the number of surge plates per chamber.

If a field was updated, an existing PDF assignment for the field is removed. A new PDF must be uploaded to add an assignment before a commit is possible.

### Example data exchange

User story: Set/update expiration date for field "General inspection"

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "id": 2224,
    "incrementalVersion": 1,
    "fields": {
      "general_inspection": "20251231"
    }
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "vehicle": {
      "id": 2224,
      "type": 100,
      "incrementalVersion": 2,
      "createdAt": 1687939396.318,
      "updatedAt": 1687939465.185,
      "status": "pending",
      "owner": {
        "id": 2220,
        "name": "Custom owner",
        "street": "Musterstraße 12",
        "zipCode": "12345",
        "city": "City",
        "country": "DE"
      }
    },
    "currentVersionID": 2437,
    "currentVersion": {
      "id": 2437,
      "createdAt": 1687939396.339,
      "updatedAt": 1687939465.023,
      "status": "pending",
      "reviewDetails": null,
      "fields": {
        "licence_number": "C-TC9208",
        "nationality": "DE",

```

```
        "vehicle_type": 100,  
        "vehicle_owner": "{ \"name\": \"Custom owner\", \"street\": \"Musterstraße 12\",  
        \"zipCode\": \"12345\", \"city\": \"City\", \"country\": \"DE\" }",  
        "general_inspection": "20251231"  
    }  
},  
"files": []  
}  
}
```

## 2.3 Upload file

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/uploadFile>

### Functional description

This method uploads a document, which can be assigned to validate a selection of fields. Every document must validate at least 1 field at the time of the upload. If a document has no more assignments, it is automatically removed.

Allowed document types are PDF, JPG and PNG. The maximum size of each document is 5 megabytes.

*uploadFile* increases the value of “incrementalVersion” by 1.

Data:

- *id*: mandatory  
Unique ID of the vehicle component
- *incrementalVersion*: mandatory  
Current version of this vehicle component
- *fields*: mandatory  
Array of 1 or more fields, that the document validates
- *name*: mandatory  
File name
- *content*: mandatory  
Document in base64 encoding

If a field was updated, an existing PDF assignment for the field is removed. A new PDF must be uploaded to add an assignment before a commit is possible.

### Example data exchange

User story: Upload a document that validates the fields "Vehicle type", "Licence number", "Nationality", "General inspection" and the 5 vehicle owner fields

Request sent to CTC Asset

```
{
  "publicKey" : "<ctc-generated-publickey>",
  "data" :
  {
    "id" : 2224,
    "incrementalVersion" : 3,
    "fields" : [ "vehicle_type",
"licence_number", "nationality", "general_inspection", "vehicle_owner"],
    "name": "Test.pdf",
    "content": "<base-64 encoded document>"
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "vehicle": {
      "id": 2224,
      "type": 100,
      "incrementalVersion": 4,
      "createdAt": 1687939396.318,
      "updatedAt": 1687941314.256,
      "status": "pending",
      "owner": {
        "id": 2220,
        "name": "Custom owner",
        "street": "Musterstraße 12",
        "zipCode": "12345",
        "city": "City",
        "country": "DE"
      }
    },
    "currentVersionID": 2437,
    "currentVersion": {
      "id": 2437,
      "createdAt": 1687939396.339,
      "updatedAt": 1687940357.922,
      "status": "pending",
      "reviewDetails": null,
      "fields": {
```

```
    "licence_number": "C-TC9208",
    "nationality": "DE",
    "vehicle_type": 100,
    "vehicle_owner": "{\"name\": \"Custom owner\", \"street\": \"Musterstraße 12\\
\", \"zipCode\": \"12345\", \"city\": \"City\", \"country\": \"DE\"}",
    "general_inspection": "20251231"
  }
},
"files": [
  {
    "id": 912,
    "name": "Test.pdf",
    "fields": [
      "vehicle_type",
      "licence_number",
      "nationality",
      "vehicle_owner",
      "general_inspection"
    ],
    "url": "<URL of the uploaded document>"
  }
]
}
}
```

## 2.4 Commit Asset

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/commitAsset>

### Functional description

This method starts the approval process for a vehicle component. If the conditions for an approval process are met, the vehicle component status changes to “Waiting for approval”.

All fields with expiration values must be in the future. If a (manufacturer) location is defined, all fields according to the location profile must be filled. All fields with information must be validated with a document.

If a vehicle component is in status “Waiting for approval”, it can not be edited until the commit is approved by the CTC support, or if the commit is revoked.

Committing an asset does not increase the value of *incrementalVersion*.

Data:

- *id*: mandatory  
Unique ID of the vehicle component
- *incrementalVersion*: mandatory  
Current version of this vehicle component
- *locationList*: mandatory  
Array of 0 or more location IDs. If the array is not empty, all required fields as defined by the locations requirement profiles have to be filled. Locations, which have defined a requirement profile for a vehicle component type, and their respective location IDs are displayed in the area "Select location to display masterdata requirements" on the editing page of a vehicle component.

## Example data exchange

User story: Start the approval process for a vehicle component, which must meet the requirements as defined by the location with ID 54

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "id": 2224,
    "incrementalVersion": 4,
    "locationList": [54]
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  }
}
```

## 2.5 Revoke Commit

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/revokeCommit>

### Functional description

This method is for revoking the commit of a vehicle component. This is only possible for vehicle components in status "waiting for approval", i.e. if the CTC support has not done the approval process yet.

Revoking a commit does not increase the value of *incrementalVersion*.



Data:

- *id*: mandatory  
Unique ID of the vehicle component
- *incrementalVersion*: mandatory  
Current version of this vehicle component

## Example data exchange

User story: Revoke the commit of a vehicle component

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "id": 2224,
    "incrementalVersion": 4
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  }
}
```

## 2.6 Revert asset

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/revertAsset>

### Functional description

This method is to revert an asset to its last published (approved) version. The editing version is removed, therefore all edits since the vehicle component was approved are deleted.

This is only possible for vehicle components in status “Unapproved changes”, which have been approved before.

The value of *incrementalVersion* is not reduced.

Data:

- *id*: mandatory  
Unique ID of the vehicle component

- *incrementalVersion*: mandatory  
Current version of this vehicle component

## Example data exchange

User story: Revoke the commit of a vehicle component

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "id": 2224,
    "incrementalVersion": 4
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  }
}
```

## 2.7 Disable asset

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/disableAsset>

### Functional description

This method is to set a vehicle component status to inactive. A vehicle component can not be re-activated after inactivation.

Note: A vehicle component is automatically set to “inactive”, if a different vehicle component with the same identifiers (licence number and nationality) is approved by the CTC support.

Due to regulatory obligations, vehicle components are only fully deleted after such obligations to retain data have expired. For details, please consult our data privacy policy.

Data:

- *id*: mandatory  
Unique ID of the vehicle component
- *incrementalVersion*: mandatory  
Current version of this vehicle component

## Example data exchange

User story: Disable a vehicle component

Request sent to CTC Asset

```
{  
  "publicKey": "<ctc-generated-publickey>",  
  "data": {  
    "id": 2224,  
    "incrementalVersion": 4  
  }  
}
```

Response from CTC Asset

```
{  
  "error_code": {  
    "code": "0",  
    "message": "Success"  
  }  
}
```

## 3. Methods to access vehicle component data

### 3.1 Get all assets

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/getMyAssets>

#### Functional description

This method returns all vehicle components that are technically owned.

Data:

- *limit*: mandatory  
Number of vehicle components that should be in the response
- *offset*: mandatory  
Defines the starting position (first position is 0). The ordering is based on the vehicle component ID with the oldest vehicle component in position 0

#### Example data exchange

User story: Get 10 vehicle components of my company, ignoring the first 20 vehicle components

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "limit": 10,
    "offset": 20
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "vehicles": [
      {
        <data of first result>
      },
      ...
      ,{
        <data of last result>
      }
    ]
  }
}
```

```
    }  
  ]  
}  
}
```

## 3.2 Get editing version

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/getEditingVersion>

### Functional description

This method accesses the current (editing) version a vehicle component. If a vehicle component is in status “approved”, there is no editing version.

Data:

- *id*: mandatory  
Unique ID of the vehicle component

### Example data exchange

User story: Get the current editing version of a vehicle component

Request sent to CTC Asset

```
{  
  "publicKey": "<ctc-generated-publickey>",  
  "data": {  
    "id": 2224  
  }  
}
```

Response from CTC Asset

```
{  
  "error_code": {  
    "code": "0",  
    "message": "Success"  
  },  
  "payload": {  
    "vehicle": {  
      "id": 2224,  
      "type": 100,  
      "incrementalVersion": 6,  
      "createdAt": 1687939396.318,  
      "updatedAt": 1687941402.522,  
      "status": "committed",  
      "owner": {
```

```

    "id": 2220,
    "name": "Custom owner",
    "street": "Musterstraße 12",
    "zipCode": "12345",
    "city": "City",
    "country": "DE"
  },
  "currentVersionID": 2437,
  "currentVersion": {
    "id": 2437,
    "createdAt": 1687939396.339,
    "updatedAt": 1687940357.922,
    "status": "pending",
    "reviewDetails": null,
    "fields": {
      "licence_number": "C-TC9208",
      "nationality": "DE",
      "vehicle_type": 100,
      "vehicle_owner": "{\"name\":\"Custom owner\", \"street\":\"Musterstraße 12\", \"zipCode\":\"12345\", \"city\":\"City\", \"country\":\"DE\"}",
      "general_inspection": "20251231"
    }
  }
}

```

### 3.3 Get editing version files

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/getEditingVersionFiles>

#### Functional description

This method accesses the files of the current (editing) version a vehicle component. If a vehicle component is in status “approved”, there is no editing version.

Data:

- *id*: mandatory  
Unique ID of the vehicle component
- *content*: mandatory  
If value is 0, then the response contains a temporary link to the document. The link is valid for 30 minutes.  
If value is 1, then the response also contains the document in base64 encoding

The response contains an array of files, including the information which fields are validated by each document.

## Example data exchange

User story: Get the files of the current editing version of a vehicle component

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "id": 2224,
    "content": 1
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "files": [
      {
        "id": 912,
        "name": "Test.pdf",
        "fields": [
          "vehicle_type",
          "licence_number",
          "nationality",
          "vehicle_owner",
          "general_inspection"
        ],
        "url": "<URL of the document>",
        "content": "<base-64 encoded document>"
      }
    ]
  }
}
```

## 3.4 Get published version

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/getPublishedVersion>

### Functional description

This method accesses the most recent published version a vehicle component. If a vehicle component has never been approved by the support, there is no published version.

If there have been any updates to the published version, they are not part of the response. For this information, `getEditingVersion` has to be used.

Data:

- `id`: mandatory  
Unique ID of the vehicle component

## Example data exchange

User story: Get the most recent published version of a vehicle component

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "id": 2224
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "vehicle": {
      "id": 2224,
      "type": 100,
      "incrementalVersion": 6,
      "createdAt": 1663156238.764,
      "updatedAt": 1687878525.56,
      "status": "pending",
      "owner": {
        "name": "Custom owner",
        "street": "Musterstraße 12",
        "zipCode": "12345",
        "city": "City",
        "country": "DE"
      },
    },
    "approvedVersionID": 2403,
    "approvedVersion": {
      "id": 2403,
    }
  }
}
```



```

    "createdAt": 1681986540.892,
    "updatedAt": 1681986556.282,
    "status": "approved",
    "reviewDetails": "",
    "fields": {
      "licence_number": " C-TC9208",
      "nationality": "DE",
      "vehicle_type": 100,
      "max_weight": 40000,
      "general_inspection": "20231231",
      "safety_inspection": "20231231",
      "vehicle_owner": "{\"name\":\"Custom owner\",\"street\":\"Musterstraße 12\",
    },
    "lastRequestedPublishedVersionID": null
  }
}

```

### 3.5 Get published version files

URI: <https://ctcapi.trusted-carrier.com/api/v1/restAPIs/getPublishedVersionFiles>

#### Functional description

This method accesses the files of the most recent published version a vehicle component. If a vehicle component has never been approved by the support, there is no published version.

Data:

- *id*: mandatory  
Unique ID of the vehicle component
- *content*: mandatory  
If value is 0, then the response contains a temporary link to the document. The link is valid for 30 minutes.  
If value is 1, then the response also contains the document in base64 encoding

The response contains an array of files, including the information which fields are validated by each document.

#### Example data exchange

User story: Get the files of the most recent published version of a vehicle component

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "id": 2029,
    "content": 0
  }
}
```

### Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "files": [
      {
        "id": 885,
        "name": "Test.pdf",
        "fields": [
          "vehicle_type",
          "licence_number",
          "nationality",
          "max_weight",
          "general_inspection",
          "safety_inspection",
          "vehicle_owner"
        ],
        "url": "<URL of the document>"
      }
    ]
  }
}
```

## 4. Methods exclusive to Staging system

The methods listed in this chapter are made available on the Staging system to allow testing the complete process. Thus, it is not necessary to invoke the manual approval/rejection process for vehicle components by the CTC support staff like on the production system. Therefore you are able to approve or reject a vehicle component by yourself.

**These methods are not available on the production system!**

The methods will trigger sending out the same emails / API notifications, that are sent during a manual review.

### 4.1 Approve asset

URI: <https://ctcapi-staging.trusted-carrier.com/api/v1/restAPIs/approveAsset>

#### Functional description

This method approves a vehicle component, that is currently in status "waiting for review". A new published version is created. The status changes to "approved". The editing version is deleted.

If another vehicle component with the same unique identifiers (licence plate + nationality or container number) was published, this other vehicle component is automatically disabled.

*approveAsset* increases the value of "incrementalVersion" by 1.

Data:

- *id*: mandatory  
Unique ID of the vehicle component

#### Example data exchange

User story: Approve a vehicle component to create a new published using the Staging-only method *approveAsset*.

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "id": 2029
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  }
}
```

## 4.2 Reject asset

URI: <https://ctcapi-staging.trusted-carrier.com/api/v1/restAPIs/rejectAsset>

### Functional description

This method rejects a vehicle component, that is currently in status “waiting for review”. The status changes to “rejected”. No changes to the editing version are made.

*rejectAsset* does not increase the value of *incrementalVersion*.

Data:

- *id*: mandatory  
Unique ID of the vehicle component
- *reviewDetails*: mandatory  
Reason for denial (max. 350 characters)

### Example data exchange

User story: Reject the approval of a vehicle component by using the Staging-only method *rejectAsset*.

Request sent to CTC Asset

```
{
  "publicKey": "<ctc-generated-publickey>",
  "data": {
    "id": 2029,
    "reviewDetails": "<reason for the rejection>"
  }
}
```

Response from CTC Asset

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  }
}
```

}

## 5. Updates to external systems

When an endpoint is defined in the “Admin” module, notifications are sent as the result of changes to an asset.

### Header sent by CTC (with activated JWT):

```
{  
  "Content-Type": "application/json;charset=UTF-8",  
  "User-agent": "CTC",  
  "Authorization": "Bearer <JWT-Token>"  
}
```

### 5.1 Completed Asset check

When a support process is finished (approval or rejection), the result of the support process is sent towards that endpoint. In case of a rejection, the reason is stated.

#### Body sent by CTC (approval):

```
{  
  "publicKey": "<your entered key>",  
  "type": "asset:reviewCompleted",  
  "payload": {  
    "id": 2224,  
    "incrementalVersion": 14,  
    "status": "approved",  
    "licence_number": "C-TC9208",  
    "nationality": "DE"  
  }  
}
```

#### Body sent by CTC (rejection):

```
{  
  "publicKey": "<your entered key>",  
  "type": "asset:reviewCompleted",  
  "payload": {  
    "id": 2224,  
    "incrementalVersion": 13,  
    "status": "rejected",  
    "licence_number": "C-TC9208",  
    "nationality": "DE",  
    "reviewDetails": "<reason for the rejection>"  
  }  
}
```

## 5.2 Reverted Asset check

In the case of an erroneous review result, the result of a support process needs to be reverted. The status of the asset returns to “committed” (“waiting for approval” in the frontend). If the original review result was “accepted”, the latest published version is unpublished.

### Body sent by CTC (rejection):

```
{
  "publicKey": "<your entered key>",
  "type": "asset:reviewReverted",
  "payload": {
    "id": 2224,
    "incrementalVersion": 13,
    "status": "committed",
    "licence_number": "C-TC9208",
    "nationality": "DE"
  }
}
```