# CTC Asset – API description for locations

**Version 1.1.1**



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

# 1. Versions

| Version | Date | Type of change |
|---|---|---|
| 1.1.1 | 02.02.2024 | Correction in responses for getVehicleData / getMultipleVehiclesData |
| 1.1 | 07.12.2023 | New CI |
| 1.0 | 28.09.2022 | First official verison |
| | | - |
| | | - |
| | | |

# Table of contents



# 2. General information

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

## 2.1 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.
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.

## 2.2 Structure of requests

All requests are POST.

The header must contain the information about the content type and the JWT token, if JWT is activated (see chapter 2.5 for choosing the security mode).

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

The content of the body consists of the CTC-generated public key, followed by a “data” object.

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

Your public key is available on the page “API settings” in the “Admin” module.

## 2.3 Structure of answers

The answer starts with an error code. If the request was successful, the “code” field is 0. There may be an additional “payload” field, containing details about the vehicle component.

```
{
    "error_code": {
        "code": "0",
        "message": "Success"
    },
    "payload": {
        <payload>
    }
}
```

If there are any errors, the “error_code” field contains a “message” and a “details” field which describe 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.

## 2.4 Vehicle component types & Masterdata fields

You can find more detailed information about the vehicle component types existing on the CTC platform and their fields in the file “Vehicle Master Data vX.X.xlsx”

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

## 2.5 External system & 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 as well:

- “Default”: This is the basic security setting, allowing authorization with a key.
    - 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 JWT tokens in the authentication header, which must be sent by both CTC and the external system in each request.
    - 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 location and CTC APIs will be required to send an authorization header with the value: Bearer <JWT token>
- JWT settings
    - RS512
    - payload
        - iss: sender URL
            - CTC will use one of the following: https://ctc.trusted-carrier.com/ (production system), https://ctc-staging.trusted-carrier.com/ (test system)

        - While “payload” is not a mandatory value, the external system is advised to provide it and it will be stored in the CTC logs
    - 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
- 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

# 3. Methods to access vehicle component data

## 3.1 Get data for a single vehicle component

URI: https://ctcapi.trusted-carrier.com/api/v1/restAPIs/getVehicleData

### Functional description

*getVehicleData* returns information about the latest published version of a vehicle component, if such a component exists in the CTC Asset database. CTC Asset will also return data if the vehicle status is “expired”.

The request must contain either the CTC Asset ID, the VIN number (for road-capable vehicles), the combination of licence plate and nationality (for road-capable vehicles) or the container number (for containers).

The request can be customized by adjusting the “fields” parameter so that CTC returns only specific values.

Data:

- Mandatory identificator: 1 one of:
  - *id*: CTC Asset ID
  - *vin*: Vehicle identification number
  - *licence_number*: Licence plate or container number
  - *nationality*: 2-character ISO code
- *fields*: (mandatory)
  - Value = “all” returns all available fields (including empty fields)
  - Value = “profile” returns all fields as per manufacturer configuration. If profile does not exist or is not active, an error message is returned.
  - Value = “custom” requires sending a list of fields that should be returned.
- “*list_of_fields*” (mandatory if fields value is “custom”):
  Array of strings of field name slugs. See the document for the masterdata field names

### Example data exchange

Request from external system based on licence plate & nationality:

```
{
    "publicKey": "<ctc-generated-publickey>",
    "data": {
        "licence_number": "M-XX123", // licence plate of the vehicle
        "nationality": "DE", // nationality as a 2-character ISO code
        "fields": "profile" // request according to manufacturer-specific component profile
    }
```

```
}
```

Request from external system based on Asset ID:

```
{
    "publicKey": "<ctc-generated-publickey>",
    "data": {
        "id": "1234", // component ID of the vehicle
        "fields": "all" // request all available fields
    }
}
```

Request from external system based on VIN:

```
{
    "publicKey": "<ctc-generated-publickey>",
    "data": {
        "vin": "W0L000051T2123456", // component ID of the vehicle
        "fields": "custom", // request a custom list of fields
        "list_of_fields": [
            "general_inspection",
            "safety_inspection",
            "empty_weight"
        ]
    }
}
```

Response from CTC Asset (if vehicle is found)

```
{
    "error_code": {
        "code": "0",
        "message": "Success"
    },
    "payload": {
        "id": 2029,
        "status": "active",
        "approvedVersionID": 2435,
        "type": 100,
        "fields": {
            "licence_number": "M-XX123",
            "nationality": "DE",
            "general_inspection": "20251231",
            "safety_inspection": "20251130",
            "vehicle_type": 100,
            "max_weight": 40000,
            "vehicle_owner": "{\"name\":\"TC
Demo\",\"createdAt\":1604573202,\"updatedAt\":1676455796,\"street\":\"Demostreet
35\",\"zipCode\":\"12345\",\"city\":\"Demo City\",\"country\":\"DE\",\"id\":2159}"
```

```
        },
        "owner": {
            "id": 2218,
            "name": "TC Demo",
            "street": "Demostreet 35",
            "zipCode": "12345",
            "city": "Demo City",
            "country": "DE"
        }
    }
}
```

## 3.2 Get data for a set of vehicle components

URI: https://ctcapi.trusted-carrier.com/api/v1/restAPIs/getMultipleVehiclesData

### Functional description

*getMultipleVehiclesData* allows querying for multiple vehicle components (up to 10) at once.

The request structure is similar to getVehicleData. In the response, the order of the response array is the same as the order of the request array. If a vehicle component cannot be found, *null* is returned for this vehicle component..

### Example data exchange

Request from external system based on licence plate & nationality:

```
{
    "publicKey": "<ctc-generated-publickey>",
    "data": [
        {
            "licence_number": "M-XX123", // licence plate of the vehicle (e.g., semi-truck)
            "nationality": "DE",
            "fields": "profile"
        },
        {
            "licence_number": "M-XX124", // licence plate of the vehicle (e.g., semi-trailer
(curtain))
            "nationality": "DE",
            "fields": "profile"
        }
    ]
}
```

Response from CTC Asset (vehicles 1 was found, vehicle 2 was not found)

```
{
    "error_code": {
        "code": "0",
        "message": "Success"
    },
    "payload": [{
        "id": 2029,
        "status": "active",
        "approvedVersionID": 2435,
        "type": 100,
        "fields": {
            "licence_number": "M-XX123",
            "nationality": "DE",
            "general_inspection": "20251231",
            "safety_inspection": "20251130",
            "vehicle_type": 100,
            "max_weight": 40000,
            "vehicle_owner": "{\"name\":\"TC
Demo\",\"createdAt\":1604573202,\"updatedAt\":1676455796,\"street\":\"Demostreet
35\",\"zipCode\":\"12345\",\"city\":\"Demo City\",\"country\":\"DE\",\"id\":2159}"
        },
        "owner": {
            "id": 2218,
            "name": "TC Demo",
            "street": "Demostreet 35",
            "zipCode": "12345",
            "city": "Demo City",
            "country": "DE"
        }
    ,null
    ]
}
```