



CTC Wallet – API description for carriers

Version 1.0

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
Telephone: +49 (0) 89 890 569 – 280
Internet: www.trusted-carrier.com

Versions

Version	Date	By	Type of change
1.0	06.03.24	Harald Clemens	First official version

Table of contents

1	Overview	4
1.1	Technical requirements	4
1.2	Configuration in “Admin” module.....	4
2	Methods	7
2.1	Get information about a single driver	7
2.2	Get information about multiple drivers	8
2.3	Create driver	9
2.4	Invite driver	12
2.5	Confirm driver request	14
2.6	Delete request.....	16
2.7	Remove driver.....	16
2.8	Change password	17
2.9	Remove phone number for SMS-TAN process.....	18
2.10	Update driver information.....	19
3	Notifications to external systems.....	21
3.1	Driver accepts association request by the carrier	21
3.2	Driver rejects association request by the carrier	22
3.3	Driver removes an existing association.....	23
3.4	Driver requests an association.....	24
3.5	Driver deletes certificate issued by the carrier	25

1 Overview

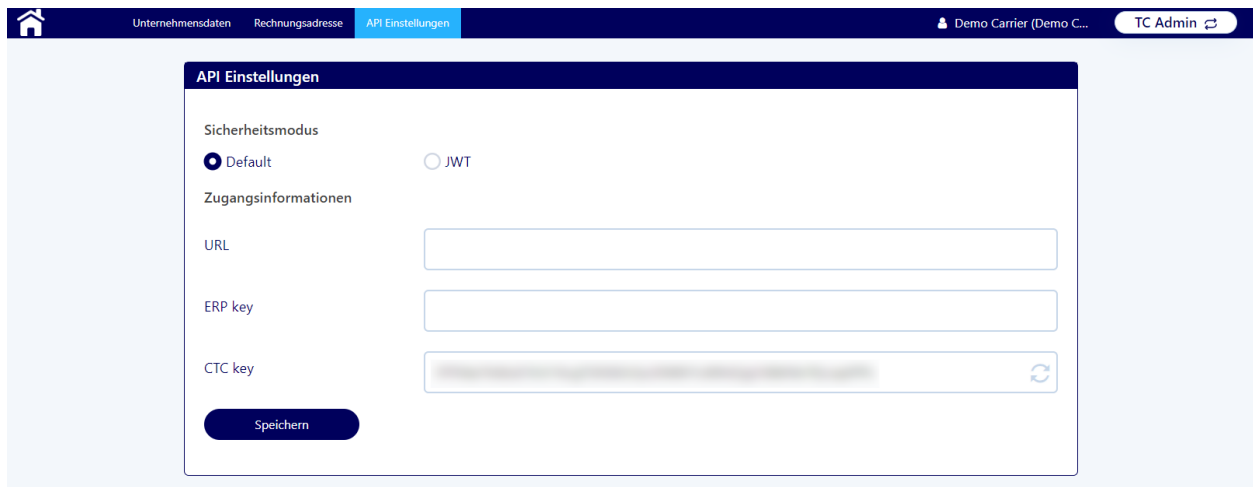
1.1 Technical requirements

The external system requires an online connection and an open port for requests from the CTC backend as well as the ability to send requests to <https://walletapi-staging.trusted-carrier.com/api/restAPIs/v1/carrier> (Staging system) or <https://walletapi.trusted-carrier.com/api/restAPIs/v1/carrier> (Production system).

1.2 Configuration in “Admin” module

1.2.1 API Settings

The communication with an external system can be configured in the “API Settings” page of the “Admin” module.



1.2.2 Security mode

The mode of security can be chosen on the screen “API settings” in the “Admin” module of a CTC account.

1.2.2.1 Default

This is the basic security setting, allowing authorization with pre-shared keys. The pre-shared key must be sent within the body in each request.

Fields:

- 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: The CTC key is automatically generated by CTC. It can only be changed by clicking the “Refresh” icon on the right of the field. The external system needs to send in all requests when communication is initiated by the external system

Example data exchange for a mobile registration:

Header:

```
{  
  "Content-Type": "application/json; charset=UTF-8",  
  "User-Agent": "CTC"  
}
```

Body:

```
{  
  "publicKey": "<ctc-generated public key",  
  "type": "wallet:registration",  
  "payload": {  
    <information of the mobile registration>  
  }  
}
```

1.2.2.2 JWT

This setting uses JWT tokens in the authentication header, which must be sent by both CTC and the external system in each request.

Fields:

- 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

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

- Algorithm: 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
 - type: the type of data, e.g., “wallet:registration” for a registration sent using the mobile app
 - jti: unique identifier for the JWT
- 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

Example data exchange for a mobile registration:

Header:

```
{
  "Content-Type": "application/json;charset=UTF-8",
  "User-Agent": "CTC",
  "Authorization": "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJpc3MiOiJodHRwczovL2N0Yy1zdGFnaW5nLnRydXN0ZWQtY2Fycm11c
i5jb20vIiwiaWF0IjoxNzA5NzIzNTIyLjA3OCwiZXhwIjoxNzA5NzIzNTgyLCJ0eXBlIjoid2FsbGV0OnJlZ2lzdHJhdG1
vbiIsImp0aSI6ImMzMWZyYTYkyLWJhNDktNDZkZS1iYjYzLTvkODhkZDVmNjE1MCJ9.CYDdbw4sRGn3CiJk5VsFA-
n_Eezw0qms_Sn0Y-qHa1xQS4sf-
U245pJayHiggrlHbcbRDDRSPwdJa9z_WL3kHSIRUnuextlbX3g5vZ6uN7nqEDafmKyndDfX0pp1CATvqax7VOKdBNNeJzf
7yin_1Nj7xung6wbrGht6rFoshf00ZVld3mRGDnMwkb1mT1_NhA0wQpdt8NF7nF6LzaN_oaLvMdgT99VVKyIv5icvJkR5G
97e_Kp9ei457gN46xUzD_u3mdPZBaNlFR6T2xn3t0V0tauPw9QMQ-
rJ9bXlKRGHi3p1x3qG5EN_BI2MYfA9dw8klpS_4RkYHxABQNQQXg"
}
```

Body:

```
{
  "type": "wallet:registration",
  "payload": {
    <information of the mobile registration>
  }
}
```

2 Methods

2.1 Get information about a single driver

URI: `api/restAPIs/v1/carrier/getDriver`

2.1.1 Functional description

This method returns current available information about a single driver.

Data:

- *userID*: (mandatory)
User ID of the driver account. The ID must belong to either a personal driver account with a completed signup process, or a corporate driver created by the requesting carrier.

2.1.2 Example data exchange

Request sent to CTC Wallet:

```
{
  "publicKey": "ctc-generated-publickey",
  "data": {
    "userID": 1128
  }
}
```

Response from CTC Wallet:

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

```
"payload": {  
  "userID": 1128,  
  "firstName": "Max",  
  "lastName": "Mustermann",  
  "nationality": "DE",  
  "primaryLanguage": "de",  
  "callingCode": "49",  
  "telephoneNumber": "1234567",  
  "dateOfBirth": "19800101",  
  "additionalInformation": "19800101",  
  "language": "en",  
  "brummiCertificate": 1,  
  "status": "accepted",  
  "username": "ErMu"  
  "type": "corporate"  
}
```

2.2 Get information about multiple drivers

URI: `api/restAPIs/v1/carrier/getDrivers`

2.2.1 Functional description

This method returns current available information about multiple drivers associated with the requesting company.

Data:

- *limit*: (mandatory)
Number of drivers that should be in the response. Maximum value: 100
- *offset*: mandatory
Defines the starting position (first position is 0). The ordering is based on the user ID with the lowest ID in position 0

2.2.2 Example data exchange

Request sent to CTC Wallet:

```
{  
  "publicKey": "ctc-generated-publickey",  
  "data": {  
    "limit": 100,  
    "offset": 0  
  }  
}
```



```
}
```

Response from CTC Wallet:

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "drivers": [
      {
        "userID": 208,
        "firstName": "Erika",
        "lastName": "Musterfrau",
        "nationality": "DE",
        "primaryLanguage": null,
        "callingCode": null,
        "telephoneNumber": null,
        "dateOfBirth": "19800429",
        "additionalInformation": null,
        "language": "de",
        "brummiCertificate": 0,
        "status": "accepted",
        "username": "demo-driver@trusted-carrier.com",
        "type": "personal"
      },
      ...
      {
        <last driver>
      }
    ]
  }
}
```

2.3 Create driver

URI: `api/restAPIs/v1/carrier/createDriver`

2.3.1 Functional description

This method creates a new corporate driver account. The answer contains the new user ID of this driver.

For corporate driver accounts, the driver masterdata will be synchronized into the mobile app when the driver logs in into a single-user installation or into a profile of a multi-user installation.

Data:

- *username*: (mandatory)
Must be unique on the CTC platform. Maximum of 30 characters.
- *password*: (mandatory)
The password must have at least: 1 UPPER, 1 lower, 1 special(!@#\$%^&*) and a length of at least 8 characters. The password is stored as a hashed value and is not stored in clear text in any logfiles.
- *newPasswordRequired*: (optional)
Allowed values are "1" if driver must set a new password after the first login, or "0" if that is not required. Default is "0".
- *language*: (optional)
Language that will be set for this account. Value must be in ISO-639-1 format (i.e. 2-letter language code, all lower case, e.g., "de" for German) and be one of the 18 languages supported by CTC. If the field is not sent, the main company language (as defined in "Admin" module) will be used instead.
- *firstName*: (optional)
Given name(s) of the driver. Maximum of 30 characters.
- *lastName*: (optional)
Last name of the driver. Maximum of 30 characters.
- *nationality*: (optional)
Nationality of the driver in ISO 3166-1 alpha-2 format (i.e. 2-letter nationality code, all upper case, e.g., "DE" for Germany)
- *primaryLanguage*: (optional)
Primary language of the driver in ISO-639-1 format (i.e. 2-letter language code, all lower case, e.g., "de" for German)
- *dateOfBirth*: (optional)
Birthdate of the driver (YYYYMMDD). Must be in the past.
- *callingcode*: (optional)
Country calling code without "00" or "+" (e.g., 49 for Germany)
- *telephoneNumber*: (optional)
Telephone number without national trunk prefix (typically a "0"), that is typically used in national calls. Only numbers are valid. Must be between 7 and 15 numbers.
- *additionalInformation*: (optional)
Additional information about the driver. Maximum of 120 characters.

- *brummiCertificate*: (optional)

Allowed values are "1" if you want to send a BGL-Brummi-Card certificate to the driver and "0" otherwise . This is only available for carrier companies with a confirmed membership in a BGL regional organization in the vCard service.

2.3.2 Example data exchange

Request sent to CTC Wallet:

```
{
  "publicKey": "ctc-generated-publickey",
  "data": {
    "username": "demo-driver",
    "password": "Password3#",
    "newPasswordRequired": 1,
    "firstName": "Max",
    "lastName": "Mustermann",
    "nationality": "DE",
    "primaryLanguage": "de",
    "dateOfBirth": "19801101",
    "callingCode": "49",
    "telephoneNumber": "1721234567",
    "additionalInformation": "Additional information about the driver",
    "brummiCertificate": 1
  }
}
```

Response from CTC Wallet:

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "userID": 1467,
    "firstName": "Max",
    "lastName": "Mustermann",
    "nationality": "DE",
    "primaryLanguage": "de",
    "callingCode": "49",
    "telephoneNumber": "1721234567",
    "dateOfBirth": "19801101",
    "additionalInformation": "Additional information",
  }
}
```

```
"language": "de",  
"brummiCertificate": 1,  
"status": "accepted",  
"username": "demo-driver"  
}  
}
```

2.4 Invite driver

URI: `api/restAPIs/v1/carrier/inviteDriver`

2.4.1 Functional description

This method invites a driver with a personal driver account to be associated with the company.

The workflow depends on the entered email, the role of an existing account and the current association status.

- Email is unknown: an invitation email is sent to the email address to download the mobile app CTC Wallet and create a new account. The driver can confirm the request after logging in into the mobile app. The status of the association is then "pending".
- If the email is known on the CTC platform:
 - If the email belongs a user with a non-driver role, the API responds with an error
 - If the email belongs to a user with a personal driver account:
 - If this driver already has a confirmed association, the API responds with an a error
 - If this driver requested an association, inviteDriver will confirm the request (i.e. it will do the same as confirmDriverRequest). The status of the association is then "accepted".
 - If this driver has no association yet, a push notification is sent to this driver. The status of the association is then "pending".

The association request is valid for 7 days and will be automatically deleted afterwards if the driver does not react in that time period.

For personal driver accounts, the driver masterdata will not be synchronized into the mobile app.

Data:

- *email*: (mandatory)
Email of the driver account

- *language*: (optional)
Language that will be used in the email to invite drivers who do not have an account yet. Value must be in ISO-639-1 format (i.e. 2-letter language code, all lower case, e.g., "de" for German) and be one of the 18 languages supported by CTC. If the field is not sent, the main company language (as defined in "Admin" module) will be used instead.
- *firstName*: (optional)
Given name(s) of the driver. Maximum of 30 characters.
- *lastName*: (optional)
Last name of the driver. Maximum of 30 characters.
- *nationality*: (optional)
Nationality of the driver in ISO 3166-1 alpha-2 format (i.e. 2-letter nationality code, all upper case, e.g., "DE" for Germany)
- *primaryLanguage*: (optional)
Primary language of the driver in ISO-639-1 format (i.e. 2-letter language code, all lower case, e.g., "de" for German)
- *dateOfBirth*: (optional)
Birthdate of the driver (YYYYMMDD). Must be in the past.
- *callingcode*: (optional)
Country calling code without "00" or "+" (e.g., 49 for Germany)
- *telephoneNumber*: (optional)
Telephone number without national trunk prefix (typically a "0"), that is typically used in national calls. Only numbers are valid. Must be between 7 and 15 numbers.
- *additionalInformation*: (optional)
Additional information about the driver. Maximum of 120 characters.
- *brummiCertificate*: (optional)
Allowed values are "1" if you want to send a BGL-Brummi-Card certificate to the driver and "0" otherwise . This is only available for carrier companies with a confirmed membership in a BGL regional organization in the vCard service.

2.4.2 Example data exchange

Request sent to CTC Wallet:

```
{  
  "publicKey": "ctc-generated-publickey",  
  "data": {  
    "email": "demo-driver@trusted-carrier.com",  
    "language": "de",  
  }  
}
```

```
"firstName": "Max",
"lastName": "Mustermann",
"nationality": "DE",
"primaryLanguage": "de",
"dateOfBirth": "19801101",
"callingCode": "49",
"telephoneNumber": "1721234567",
"additionalInformation": "Additional information about the driver",
"brummiCertificate": 1
}
}
```

Response from CTC Wallet:

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "userID": null,
    "firstName": "Max",
    "lastName": "Mustermann",
    "nationality": "DE",
    "primaryLanguage": "de",
    "callingCode": "49",
    "telephoneNumber": "1721234567",
    "dateOfBirth": "19801101",
    "additionalInformation": "Additional information about the driver",
    "language": "de",
    "brummiCertificate": 1,
    "status": "invited",
    "requestExpiresAt": 1702034110.216,
    "username": "demo-driver@trusted-carrier.com ",
    "type": "personal"
  }
}
```

2.5 Confirm driver request

URI: `api/restAPIs/v1/carrier/confirmDriverRequest`

2.5.1 Functional description

This method confirms a pending association request from users with a personal driver account. This option is available for users, if the carrier has activated it on the Wallet "Settings" page.

The status of the association changes to "accepted".

Data:

- *userID*: (mandatory)
User ID of the driver account. If there is no pending request from this account, the API will return an error.

2.5.2 Example data exchange

Request sent to CTC Wallet:

```
{
  "publicKey": "ctc-generated-publickey",
  "data": {
    "userID": 1128
  }
}
```

Response from CTC Wallet:

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "userID": 1128,
    "firstName": "Max",
    "lastName": "Mustermann",
    "nationality": "DE",
    "primaryLanguage": "de",
    "callingCode": "49",
    "telephoneNumber": "1234567",
    "dateOfBirth": "19800101",
    "additionalInformation": "19800101",
    "language": "en",
    "brummiCertificate": 1,
    "status": "accepted",
    "username": "demo-driver@trusted-carrier.com",
    "type": "personal"
  }
}
```

```
}
```

2.6 Delete request

URI: `api/restAPIs/v1/carrier/deleteRequest`

2.6.1 Functional description

This method cancels pending association requests.

This applies to both:

- Carrier has sent an association request via email (using the frontend or using *inviteDriver*)
- Driver has sent an association request using the mobile app

Data:

- *email*: (mandatory)
Email of the driver account. If there is no pending request, the API will return an error.

2.6.2 Example data exchange

Request sent to CTC Wallet:

```
{
  "publicKey": "ctc-generated-publickey",
  "data": {
    "email": "demo-driver@trusted-carrier.com"
  }
}
```

Response from CTC Wallet:

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

2.7 Remove driver

URI: `api/restAPIs/v1/carrier/removeDriver`

2.7.1 Functional description

This method deletes confirmed associations.

For corporate driver accounts, this deletes the association as well as the user account and all associated personal data. The user will not be able to login into the mobile app anymore.

For personal driver accounts, only the association is deleted.

Data:

- *userID*: (mandatory)
User ID of the driver account. If there is no confirmed association, the API will return an error.

2.7.2 Example data exchange

Request sent to CTC Wallet:

```
{
  "publicKey": "ctc-generated-publickey",
  "data": {
    "userID": 1128
  }
}
```

Response from CTC Wallet:

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

2.8 Change password

URI: `api/restAPIs/v1/carrier/changePassword`

2.8.1 Functional description

This method changes the password of a corporate driver account.

For personal driver accounts, this method is not available.

Data:

- *userID*: (mandatory)
User ID of the account
- *password*: (mandatory)
The password must have at least: 1 UPPER, 1 lower, 1 special(!@#\$\$%^&*) and a length of at least 8 characters. The password is stored as a hashed value and is not stored in clear text in any logfiles.
- *newPasswordRequired*: (optional)
Allowed values are "1" if driver must set a new password after the next login, or "0" if that is not required. Default is "0".

2.8.2 Example data exchange

Request sent to CTC Wallet:

```
{
  "publicKey": "ctc-generated-publickey",
  "data": {
    "userID": 1128,
    "password": "Password3#",
    "newPasswordRequired": 1,
  }
}
```

Response from CTC Wallet:

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

2.9 Remove phone number for SMS-TAN process

URI: `api/restAPIs/v1/carrier/removePhoneNumber`

2.9.1 Functional description

This method is to remove the phone number stored on account level of corporate driver accounts. The phone number is used for receiving SMS-TANs which are needed to login into a profile of a Multi-user installation of the mobile app.

Users with personal driver accounts can reset the phone number using their email address.

Data:

- *userID*: (mandatory)
User ID of the account

2.9.2 Example data exchange

Request sent to CTC Wallet:

```
{
  "publicKey": "ctc-generated-publickey",
  "data": {
    "userID": 1128
  }
}
```

Response from CTC Wallet:

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

2.10 Update driver information

URI: `api/restAPIs/v1/carrier/updateDriver`

2.10.1 Functional description

This method is used to update information about a driver with a pending or confirmed association.

If a driver requested an association, calling this method confirms the driver's request (i.e. it does the same as *confirmDriverRequest*)

Data:

- *userID*: (mandatory)
User ID of the account
- *username*: (optional)
User name can be changed (only for corporate driver accounts). Maximum of 30 characters.
- All other fields: see *createDriver* / *inviteDriver* methods.

If a field is not sent, the existing information is preserved.

To delete the existing information of a field, an empty string ("") must be sent. Fields that cannot be deleted: *userID*, *username*, *language*

2.10.2 Example data exchange

Request sent to CTC Wallet:

```
{
  "publicKey": "ctc-generated-publickey",
  "data": {
    "userID": 1128,
    "username": "demo-driver2",
    "additionalInformation": "",
    "brummiCertificate": 1
  }
}
```

Response from CTC Wallet:

```
{
  "error_code": {
    "code": "0",
    "message": "Success"
  },
  "payload": {
    "userID": 1128,
    "language": "de",
    "firstName": "Max",
    "lastName": "Mustermann",
    "nationality": "DE",
    "primaryLanguage": "de",
    "callingCode": "49",
    "telephoneNumber": "1721234567",
    "dateOfBirth": "19801101",
  }
}
```

```
"additionalInformation": null,  
"brummiCertificate": 1,  
"status": "accepted",  
"username": "demo-driver2"  
}  
}
```

3 Notifications to external systems

Notifications are only sent if an endpoint and a key are entered in the “Admin” module.

Changes of the status of an association are only possible for associations with drivers with a “Personal” account.

3.1 Driver accepts association request by the carrier

3.1.1 Functional description

When the carrier sends an association request to the driver (via frontend or API), the driver receives an email (if they do not have an account yet) or a push notification into their mobile app (if they already have an account).

Here, the driver **accepts** the association request, thus setting the association status to “accepted”.

Note: The same notification is sent if a driver logs in into a Multi-user installation without having a prior association with the carrier.

3.1.2 Example data exchange

Notification sent by CTC Wallet:

```
{  
  "publicKey": "ERP-publickey",  
  "type": "wallet:driver",  
  "payload": {  
    "userID": 1128,  
    "firstName": "Max",  
    "lastName": "Mustermann",  
    "nationality": "DE",  
    "primaryLanguage": "de",  
    "callingCode": "49",  
    "telephoneNumber": "1234567",  
    "dateOfBirth": "19800101",  
  }  
}
```

```
    "additionalInformation": "19800101",
    "language": "en",
    "brummiCertificate": 1,
    "status": "accepted",
    "username": "demo-driver@trusted-carrier.com"
    "type": "personal"
  }
}
```

Expected response:

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

3.2 Driver rejects association request by the carrier

3.2.1 Functional description

When the carrier sends an association request to the driver (via frontend or API), the driver receives an email (if they do not have an account yet) or a push notification into their mobile app (if they already have an account).

Here, the driver **rejects** the association request, thus setting the association status to “removed” and deleting the association with all its associated data. The driver will not appear in future calls of *getDriver* / *getDrivers*.

3.2.2 Example data exchange

Notification sent by CTC Wallet:

```
{
  "publicKey": "ERP-publickey",
  "type": "wallet:driver",
  "payload": {
    "userID": 1128,
    "firstName": null,
    "lastName": null,
    "nationality": null,
    "primaryLanguage": null,
    "callingCode": null,
  }
}
```

```
    "telephoneNumber": null,
    "dateOfBirth": null,
    "additionalInformation": null,
    "language": null,
    "brummiCertificate": 0,
    "status": "removed",
    "username": "demo-driver@trusted-carrier.com"
    "type": "personal"
  }
}
```

Expected response:

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

3.3 Driver removes an existing association

3.3.1 Functional description

When there is a confirmed association between a carrier and a driver, both have the option to delete the association.

When the driver deletes the association, the association status is set to “removed” and all its associated data is deleted. The driver will not appear in future calls of *getDriver* / *getDrivers*.

Note: The same notification is sent if a driver account is deleted, which then also deletes the association.

3.3.2 Example data exchange

Notification sent by CTC Wallet:

```
{
  "publicKey": "ERP-publickey",
  "type": "wallet:driver",
  "payload": {
    "userID": 1128,
    "firstName": null,
    "lastName": null,
    "nationality": null,
    "primaryLanguage": null,
  }
}
```

```
"callingCode": null,
"telephoneNumber": null,
"dateOfBirth": null,
"additionalInformation": null,
"language": null,
"brummiCertificate": 0,
"status": "removed",
"username": "demo-driver@trusted-carrier.com"
"type": "personal"
}
}
```

Expected response:

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

3.4 Driver requests an association

3.4.1 Functional description

Drivers can request an association with a carrier, if the carrier has chosen to allow this on the “Settings” page in the “Wallet” module.

If there already is a pending association by the carrier for the same driver, the association request is automatically confirmed. I.e. the same notification as in chapter 5.1. is sent.

If there is no pending association, an association request with status “driverRequested” is created and a notification is sent. The association request is valid for 7 days and will be automatically deleted afterwards if the carrier does not react in that time period.

3.4.2 Example data exchange

Notification sent by CTC Wallet:

```
{
  "publicKey": "ERP-publickey",
  "type": "wallet:driver",
  "payload": {
    "userID": 1128,
    "primaryLanguage": null,
    "callingCode": null,

```



```
"telephoneNumber": null,
"dateOfBirth": null,
"status": "driverRequested",
"language": "en",
"brummiCertificate": 0,
"expiresAt": 1700054744.949,
"username": "demo-driver@trusted-carrier.com",
"type": "personal"
}
}
```

Expected response:

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

3.5 Driver deletes certificate issued by the carrier

3.5.1 Functional description

When a driver deletes a certificate issued by the carrier (e.g., “BGL-Brummi-Card” certificate that is available for BGL-certified carriers), a notification is sent with the updated information.

3.5.2 Example data exchange

Notification sent by CTC Wallet:

```
{
  "publicKey": "ERP-publickey",
  "type": "wallet:driver",
  "payload": {
    "userID": 1128,
    "firstName": "Max",
    "lastName": "Mustermann",
    "nationality": "DE",
    "primaryLanguage": "de",
    "callingCode": "49",
    "telephoneNumber": "1234567",
    "dateOfBirth": "19800101",

```

```
"additionalInformation": "19800101",  
"language": "en",  
"brummiCertificate": 0,  
"status": "accepted",  
"username": "demo-driver@trusted-carrier.com"  
"type": "personal"  
}  
}
```

Expected response:

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