ACCEPT Institute

Horizon 2020 ETC 636126

Open Standards & Requirements

Interoperable Traveller Interface

–

Deliverable 9.3

12 April 2018

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 636126.

1 Introduction

1.1 Introduction & Summary

This document describes the Open Standards & Requirements ITI. Deliverable 9.3 and is part of work package 9 '*Interoperable Traveller Interface*'.

Objectives

The objectives of work package 9 are:

- to develop technological standards and connectivity for Travellers Clubs to seamlessly and cost-effectively integrate the online booking, payment, ticketing services and travel alerts of clubs from other regions and countries for their member travellers, which they continue to serve in their own language and according to their own preferences. This deliverable.

- to develop a demonstration back end and reference App that depicts the user experience for a Traveller travelling cross-border and solely abroad. See deliverable 9.2.

Summary

For this deliverable we have developed the relevant open standards and requirements for the interoperable traveller interface (or smartphone app) in order for the app to be able to connect to the central ACCEPT EcoSpace Core. These open standards and requirements are defined as an Application Programming Interfaces (API): the so-called Mobile API.

This API has been implemented by the relevant parties and was demonstrated during the pilots in the ETC project.

2 Content

3 Open Standards & Requirements ITI

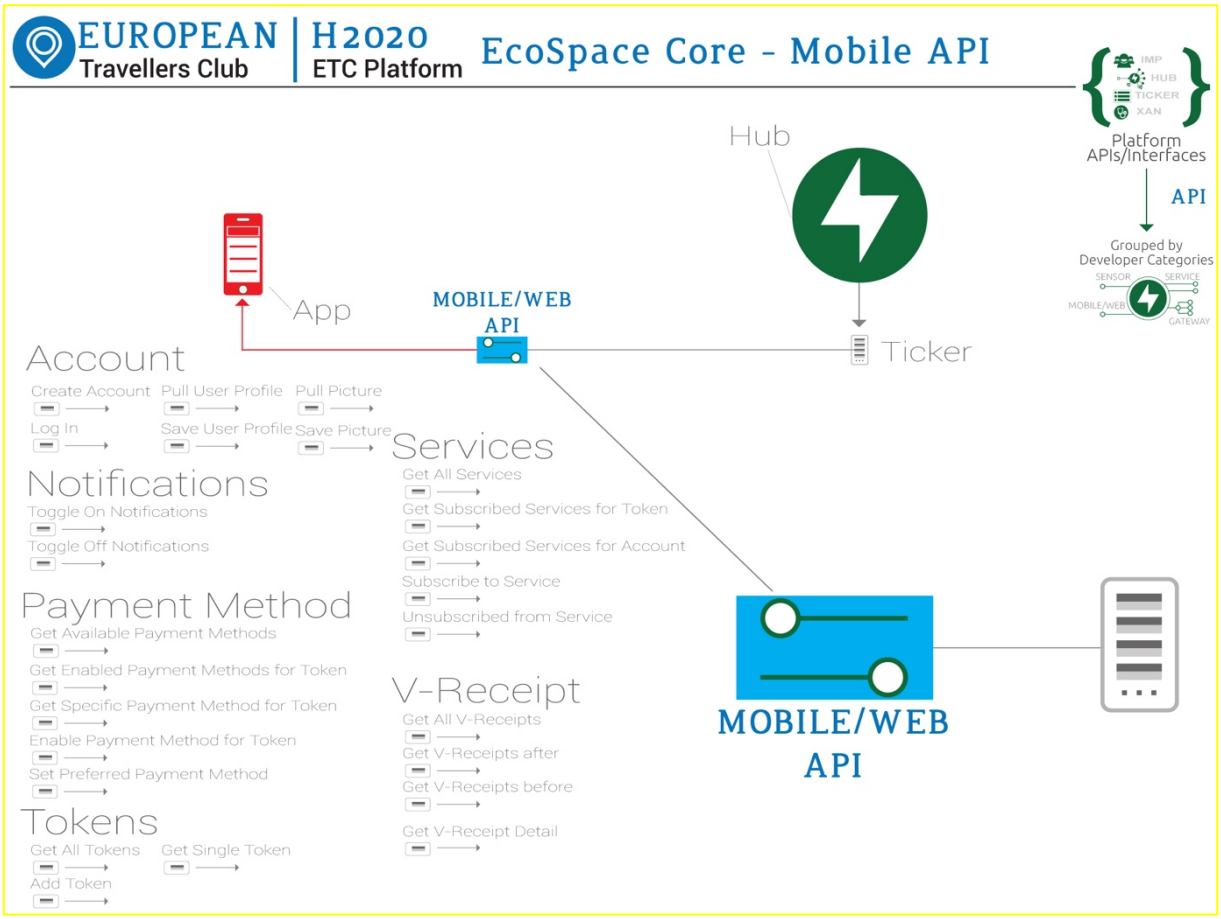
3.1 Mobile API

The Mobile API is used to connect a consumer facing Mobile App or Web Site to the EcoSpace Core. The mobile API supports multiple user centric calls. These calls allow a User to create an account, add tokens to their account, select services, select available payment methods, and see the resulting transactions created by consumption thereof.

The Mobile API supports message exchange via a REST interface currently supporting XML & JSON message structures. Due to the message size overhead of XML, JSON format is recommended.

3.2 Message Overview

There are a number of messages that a Mobile App or Website can GET or POST to the EcoSpace Core via the Mobile API.



3.2.1 Account Messages

For User Accounts

- Create Account
- Oauth Login
- User Profile
- Save Picture
- Pull Picture

3.2.2 Notifications

- Toggle On Push Notifications
- Toggle Off Push Notifications

3.2.3 Payment Methods Messages

- Get Payment Methods
- Enable Payment Method for Token
- Get Enabled Payment Methods For Token
- Get Specific Payment Method for Token
- Set Preferred Payment Method for Token

3.2.4 Token Messages

- Get All Tokens
- Get Single Token Details
- Add Token

3.2.5 ID Based Services Messages

- Get All Services
- Get All Service Subscriptions for Token
- Get All Service Subscriptions for Account
- Subscribe to Service
- UnSubscribe from Service

3.2.6 Ticker Data Messages

- Get All V-Receipts
- Get V-Receipt Details
- Get V-Receipts Created After
- Get V-Receipts Created Before

3.2.7 ETC Mobile API Web Page

The Mobile API for the ETC Test Environment also has a public API Web Page, to help developers with reference code examples and to allow test driving of their own code samples. While the page is public there are certain certificates and permissions required before developers can set up and test their own calls. ETC may be contacted to arrange this.

3.2.8 ETC Mobile API Web Page by Test Environment

VDV TravelScheme DEHUB

https://oti-test-dehub.cloudapp.net/Mobile

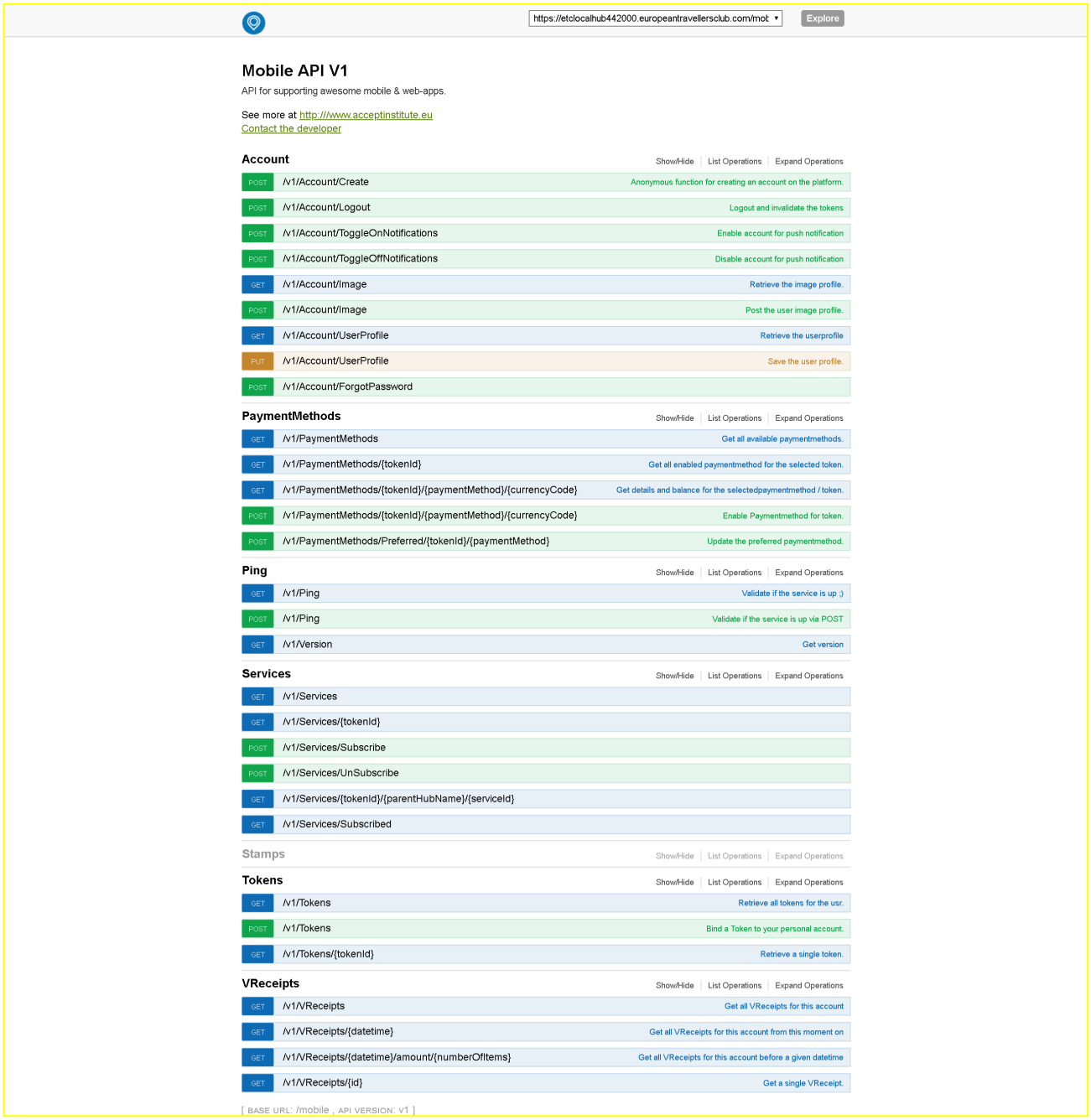
VDV TravelScheme LUXHUB

https://oti-test-luxhub.cloudapp.net/Mobile

OV-chipkaart TravelScheme NLHUB

https://oti-test-nlhub.cloudapp.net/Mobile

Example of Mobile API Web Page



3.3 Type Glossary

The following data types are used in this specification.

Field	Description / Value
GUID	Expressed as a String as in the following example: “00000000-0000-0000-0000-000000000000”

Array	Refers to an array that is specified in a separate table in this document.
Object	Refers to a data object that is specified in a separate table in this document.
Number	A non decimal value as in the following example 42
String	Represents character strings, sequence of zero or more characters as in the following example “Technology”
Dictionary	Collection of keys and values, as used in property bags.
Bool	Espressed as: True or False

3.4 Message Construction Conventions

The following data types are used in this specification.

Field	Description / Value
Control Characters	May not be used in any field elements.
UTF-8	Encoding is required for all field elements.

3.5 String Conversion Guidelines for Complex Types

Field	Description / Value
DateTime	Expressed as a string using the following convention: “20151210191159000+0000” Local timestamp format: yyyyMMddHHmmssfff concatenated with the timezone. Format [+/-]HHmm
Longitude	Expressed as a string using the following convention: “4.887156” The field value must be a valid WGS 84 longitude value from -180 to 180.
Latitude	Expressed as a string using the following convention: “52.390878” The field value must be a valid WGS 84 latitude.

Byte []	Expressed as hexadecimal strings

3.6 Message Formats

XML or JSON

3.7 Message Transport

HTTPS REST

3.8 Account Messages

User Account related calls.

3.8.1 Create Account

Use API Post: /V1/Account/Create

With Message Body containing the following fields:

Message Body for API Post

Field	Type	Description
Email	String	Valid Email Address Max length: 254
Password	string	Max length: 255
ConfirmPassword	string	Max length: 255

Expected response

Field	Type	Description
Success	Boolean	True =Success, False = Error
Response	String	Extra information if user account creation failed.

3.8.2 Oauth Login

Creates login session with short term and long term tokens. Login is based on OAuth2 specs. Following parameters are required:

Use API Post: /Auth/Token

With Message Body containing the following fields:

Message Body for API Post

Field	Type	M	Description
Username	String	Y	Email address submitted
Password	String	Y	Pwd
Grant_type	String	Y	Fixed value = 'password'
Client_id	String	Y	Application Identifier provided by EcoSpace
Client_Secret	String	N	Application Password provided by EcoSpace
Device_Id	String	N	Optional, identify device when logged in on multiple devices

Expected response

Short term and Long term token – see http://OAuth.net/documentation

3.8.3 Pull User Profile

Pull a User Profile.

API Get

Use API Get: /V1/Account/UserProfile

Expected response

Field	Type	Description
FirstName	String	Max Length = 50
MiddelName	String	Max Length = 25
LastName	String	Max Length = 60
DateOfBirth	DateTime	Date
MobilePhoneNumber	String	Max Length = 20
Address	String	Max Length = 250
ZipCode	String	Max Length = 10
BankAccount	String	Max Length = 27
SWIFTCode	String	Max Length = 8, see ISO 9362
City	String	Max Length = 163
Country	String	Max Length = 36

3.8.4 Save User Profile

Save a User Profile.

Use API Post: /V1/Account/UserProfile

With Message Body containing the following fields:

Message Body for API Post

Field	Type	Description
FirstName	String	Max Length = 50
MiddelName	String	Max Length = 25
LastName	String	Max Length = 60
DateOfBirth	DateTime	Date
MobilePhoneNumber	String	Max Length = 20
Address	String	Max Length = 250
ZipCode	String	Max Length = 10
BankAccount	String	Max Length = 27
SWITFCode	String	Max Length = 8, see ISO 9362
City	String	Max Length = 163
Country	String	Max Length = 100

For HTTP response codes see 0

3.8.5 Pull User Picture

Pull a User Picture.

API Get

Use API Get: /V1/Account/Image

Expected response

Returns a byte array, with a HTTP Header of Image/png

3.8.6 Save User Picture

Save a User Picture.

API Post

Use API Post: /V1/Account/Image

With Multipart content type and binary presentation of the file included in the body.

Expected response

Response: HttpCode: 202, 409

For HTTP response codes see 0

3.8.7 Toggle On Notifications

Register device for push notifications. Push notifications are sent to the phone when a new V-Receipt/Ticker Item is available. This service can be toggled on or off by the user.

Use API Post: /V1/Account/ToggleOnNotifications

With Message Body containing the following fields:

Message Body for API Post

Field	Type	M	Description
Platform	String	Y	E.g. Android, Ios, WindowsPhone, used to determine notification route
RegistrationID	String	Y	Identifier for the Notification Platform to broadcast to.

Expected response

Response: HttpCode: 200, 400 or 500

For HTTP response codes see 0

3.8.8 Toggle Off Notifications

UnRegister device for push notifications. Push notifications are sent to the phone when a new V-Receipt/Ticker Item is available. This service can be toggled on or off by the user.

Use API Post: /V1/Account/ ToggleOffNotifications

With Message Body containing the following fields:

Message Body for API Post

Field	Type	M	Description
RegistrationID	String	Y	Identifier for the Notification Platform to broadcast to.

Expected response

Response: HttpCode: 200, 400 or 500

For HTTP response codes see 0

3.9 Payment Method Messages

3.9.1 Get Available Payment Methods

Pulls a list of available Payment methods for user to select.

API Post

Use API Get: /V1/PaymentMethods

Expected response

Returns collection of available payment methods, See Data Object Payment Method 0 for description of field elements.

For this particular API call the following fields are not relevant for this call, and may be ignored:

- Enabled
- TokenValue
- TokenTypeID
- OrderID
- Suspended

Data Object: Payment Method

Field	Type	Description
Id	long	Payment Method ID assigned by EcoSpace
Name	String	Payment Method Name Max Length = 20
Displayname	String	Payment Method Display Name Max Length = 20
IconUrl	String	URL to icon
RequiredUserData	Collection of strings Collection of profile categories	Optional, contains the required userprofile information that is shared with the payment method when the paymentmethod is enabled for the user.
Enabled	Boolean	True
Suspended	Boolean	True = Suspended by Payment Method, False = Not
TokenValue	String	Optional
TokenTypeId	Int	Optional – default = 0
OrderId	Int	Optional – default = 0

3.9.2 Get Enabled Payment Methods for Token

Pulls a list of enabled Payment Methods for a specific Token ID.

Use API Get: /V1/PaymentMethods/{TokenValue}

With the following parameters:

Parameters for API Post

Parameter	Type	M	Description
TokenValue	String	Y	Token ID Value

Expected Response

See Payment Method Data Object 0

3.9.3 Get Specific Payment Method for Token

Pulls specific enabled Payment methods for a specific Token ID.

Use API Get: /V1/PaymentMethods/{TokenValue}/{PaymentMethod}/{CurrencyCode}

With the following parameters:

Parameters for API Get

Parameter	Type	M	Description
TokenValue	String	Y	Token ID Value
PaymentMethod	String	Y	Payment Method Name Max Length = 20
CurrencyCode	String	Y	ISO 4217 Currently only 'EUR' supported CHAR 3

Expected Response

Field	Type	Description
Id	long	Payment Method ID assigned by EcoSpace
Name	String	Payment Method Name Max Length = 20
Displayname	String	Payment Method Display Name Max Length = 20
IconUrl	String	URL to icon
RequiredUserData	Collection of strings Collection of profile categories	Optional, contains the required userprofile information that is shared with the payment method when the paymentmethod is enabled for the user.
Enabled	Boolean	True
TokenValue	String	Optional
TokenTypeId	Int	Optional – default = 0
OrderId	Int	Optional – default = 0
CurrencyCode	CHAR	ISO 4217 Currently only 'EUR' supported CHAR 3
Balance	Nullable Int	Available balance of Payment Method In cents

3.9.4 Enable Payment Method for Token

Enables a Payment method for a specific Token ID.

Use API Post: /V1/PaymentMethods/{TokenValue}/{PaymentMethod}/{CurrencyCode}

With the following parameters:

Parameters for API Get

Parameter	Type	M	Description
TokenValue	String	Y	Token ID Value
PaymentMethod	String	Y	Payment Method Name Max Length = 20
CurrencyCode	String	Y	ISO 4217 Currently only 'EUR' supported CHAR 3

Expected Response

Expected Response HTTP Status Codes: 202, 404, 409

For HTTP response codes see 0

3.9.5 Set Preferred Payment Method

Sets a user's preferred payment method for a specific Token ID.

Use API Post: /V1/PaymentMethods/Preferred/{TokenValue}/{PaymentMethod}

With the following parameters:

Parameters for API Post

Parameter	Type	M	Description
TokenValue	String	Y	Token ID Value
PaymentMethod	String	Y	Payment Method Name Max Length = 20

Expected Response

Expected Response HTTP Status Codes: 201, 409, 404

For HTTP response codes see 0

3.10 Token Messages

For managing a User's Token IDs and Token bearing form factors.

3.10.1 Get All tokens

Returns collection of 'owned' a.k.a. user bound tokens.

API Post

Use API Get: /V1/Tokens

Expected Response

Returns collection of owned tokens, See Data Object Token 0 for description of field elements.

Data Object: Token

Field	Type	Description
TokenId	long	Number of the tokenset returned
Tokens	List<TokenIdentifier>	See object 0
Description	String	Optional user description of the token

Data Object: Tokenidentifier

Field	Type	Description
TokenType	string	MFA, GST, EPAN, UID, Barcode, Biometric
TokenValue	string	Unique to abovementioned Token Type e.g. serial number, Token ID, Max Length 255

3.10.2 Get single token

Pulls details of a particular Token.

Use API Get: /V1/Tokens/{TokenValue}

With the following parameters:

Parameters for API Get

Parameter	Type	M	Description
TokenId	long	Y	Token ID Value

Expected Response

See Data Object Token 0

3.10.3 Add Token to Token Collection

Adds a new Token to Token Collection.

Use API Post: /V1/Tokens

With Message Body containing the following fields:

Message Body for API Post

Field	Type	Description
TokenType	string	MFA, GST, EPAN, UID, Barcode, Biometric
TokenValue	string	Unique to abovementioned Token Type e.g. serial number, Token ID, Max Length 255

Description	String	Optional user description of the token
ActivationCode	String	Token Activation Code

Expected Response

Expected Response HTTP Status Codes: 201, 409, 400

For HTTP response codes see 0

3.11 Services Messages

For managing a User's Service Subscriptions

3.11.1 Get All Services

Returns collection of available Services

API Get

Use API Get: /V1/Services

Expected Response

Returns collection of services, See Data Object Services 0 for description of field elements.

For this particular API call the following fields are not relevant for this call, and may be ignored:

- TokenValue
- TokenType

Data Object: Service

Field	Type	Description
ServiceId	long	Service ID provided by Local EcoSpace
ServiceName	string	Service Name assigned by Service Provider
ParentHubName	String	Service can be related with a foreignhub, if so the foreignhub is mentioned. E.g. LUXHUB, DEHUB, NLHUB
TokenValue	String	Optional: Unique to abovementioned Token Type e.g. serial number, Token ID, Max Length 255
TokenType	String	Optional: MFA, GST, EPAN, UID, Barcode, Biometric
OptInUserData	String []	Please see OptInUserData 0

OptInUserData Categories & Fields

Field	Category	Type	Description
FirstName	Name	String	Max Length = 50

MiddelName	Name	String	Max Length = 25
LastName	Name	String	Max Length = 60
DateOfBirth	Birthdate	DateTime	Date
MobilePhoneNumber	Contact	String	Max Length = 20
Address	Address	String	Max Length = 250
ZipCode	Address	String	Max Length = 10
BankAccount	Financial	String	Max Length = 27
SWITFCode	Financial	String	Max Length = 8, see ISO 9362
Name on Bank Account	Financial	String	Max Length = 140
City	Address	String	Max Length = 163
Region	Address	String	Max Length = 50
Country	Address	String	Max Length = 100
EMail	Contact	String	Max Length = 254
Picture	Picture	PictureFile	Data Object

3.11.2 Get All Subscribed Services for Token

Pulls list of subscribed services for a given token. Including Foreign Services at Interoperable Services from other Travel Schemes.

Use API Get: /V1/Services/{TokenValue}

With the following parameters:

Parameters for API Get

Parameter	Type	M	Description
TokenValue	String	Y	Token ID Value

Expected Response

Data Object: SubscribedService

Field	Type	Description
ServiceId	Long	Service ID provided by Local EcoSpace
ServiceName	String	Service Name assigned by Service Provider
ParentHubName	String	Service can be related with a foreignhub, if so the foreignhub is mentioned. E.g. LUXHUB, DEHUB, NLHUB
TokenValue	String	Optional: Unique to abovementioned Token Type e.g. serial number, Token ID, Max Length 255
TokenType	String	Optional: MFA, GST, EPAN, UID, Barcode, Biometric
Subscribed	Boolean	True = Subscribed, False = Not Subscribed

Suspended	Boolean	True = Suspended by Service, False = Not

3.11.3 Get All Subscribed Services for Account

Pulls list of subscribed services for all tokens a user account has bound to it.
Including Foreign Services a.k.a. Interoperable Services from other Travel Schemes.

API Get
Use API Get: /V1/Services/Subscribed

Expected Response
Collection of Subscribed Services
See Data Object: SubscribedServices 0

3.11.4 Subscribe to Service

To subscribe to a Service for a specific Token ID.

Use API Post: /V1/Services/Subscribe
With following message body

Message Body for API Post
See data object service 0

Expected Response
Expected Response HTTP Status Codes: 201, 400, 500
For HTTP response codes see 0

3.11.5 UnSubscribe from Service

To unsubscribe from a Service for a specific Token ID.

Use API Post: /V1/Services/UnSubscribe
With following message body

Message Body for API Post
See data object service 0

Expected Response
Expected Response HTTP Status Codes: 201, 400, 500
For HTTP response codes see 0

3.12 V-Receipt Messages

Methods for pulling V-Receipts from the Ticker.

3.12.1 Get all Vreceipts for account

API Get

Use API Get: /V1/VReceipts/

Expected Response

VReceiptItemContainer (see 0)

VReceiptItemContainer

Field	Type	Description
PaymentMethods	Dictionary	Key = Id, value = paymentmethodname
CurrencyCodes	Dictionary	Key = Id, value = Currencycode (3 chars)
ServiceTokens	Dictionary	Key = Id, value = servicetokenValue
Items	Collection of VirtualReceiptOverviewItem	Data Object 0

VirtualReceiptOverviewItem

Field	Type	Description
Id	Guid	VReceiptReferenceId
IconURL	String	Link to Icon for ticker
ItemTile	String	Title of tickeritem
LineItemDescription1	String	
LineItemDescription2	String	
ReceiptDate	DateTime	Date of the VReceipt(creation)
CurrencyCode	Integer	Refers to currencycode in dictionary in VReceiptItemContainer
Amount	Integer	Amount of the VReceipt
CostDescription	String	
StampsAvailable	Boolean	
ServiceToken	Integer	Refers to servicetoken in dictionary in VReceiptItemContainer
PaymentMethod	Integer	Refers to paymentmethod in dictionary in VReceiptItemContainer

3.12.2 V-Receipts with ´after´ specification

Methods for pulling V-Receipts from the Ticker after a given datetime.

Use API Get: /V1/VReceipts/{datetime}

With the following parameters:

Parameters for API Get

Parameter	Type	M	Description
DateTime	Long	Y	Format 'yyyyMMddHHmmss'

Expected Response

VReceiptItemContainer (See 0)

3.12.3 V-Receipts with ´before´ specification

Methods for pulling V-Receipts from the Ticker before a given datetime.

Use API Get: /V1/VReceipts/{datetime}/amount/{numberOfItems}

With the following parameters:

Parameters for API Get

Parameter	Type	M	Description
DateTime	Long	Y	Format 'yyyyMMddHHmmss'
NumberOfItems	Integer	Y	Defines maximum of items to be retrieved, should have value between 1 and 200.

Expected Response

VReceiptItemContainer (See 0)

3.12.4 V-ReceiptDetails

Methods for pulling V-Receiptdetails from the Ticker for a specific VReceipt

Use API Get: /V1/VReceipts/{id}

With the following parameters:

Parameters for API Get

Parameter	Type	M	Description
Id	Guid	Y	Id of the VReceipt

Expected Response

VirtualReceiptItem

DataObject VirtualReceiptItem

Field	Type	Description
Id	Guid	VReceiptReferenceId

Template	String	Name of the template used to visually format the Vreceiptdetails
ServicetokenValue	Guid	
ServiceTokenDescription	String	Refers to the alias of the token registered.
ReceiptDate	DateTime	Date of the Vreceipt(creation)
CostDescription	String	
Merchant	Complex type of type Merchant	
Location	Location	
VoucherIds	Collection of Guid	Links to possible available vouchers
PaymentTransaction	PaymentTransaction	0
ServiceTransaction	ServiceTransaction	See 0
LineItems	A collection of VirtualReceiptLineItems	See 0

Data Object: Merchant

Field	Type	Description
Name	String	Name of the Merchant
Slogan	String	Slogan of the Merchant
VisualCodes	MerchantCode	See 0
WebsiteURL	String	Link to the website of the Merchant
BannerURL	String	Link to the MerchantBannerImage

Data Object: MerchantCode

Field	Type	Description
Value	String	Value of the Code
Type	String	Type of the code (QR/ WiFi / …)

Data Object: Location

Field	Type	Description
Longitude	Decimal	DD.dddddd°
Latitude	Decimal	DD.dddddd°
Address	String	Textual representation of the location (i.e. street)

Data Object: PaymentTransaction

Field	Type	Description
PaymentMethodUsed	PaymentMethod	See **Fout! Verwijzingsbron niet gevonden.**
TransactionNumber	String	PaymentMethodTransactionNumber
HelpedByPictureURL	String	URL to picture of the person behind the counter
HelpedByName	String	Name of the employee
CurrencyCode	String	

Data Object: ServiceTransaction

Field	Type	Description
EcoHubName	String	Ecohubname of the ecohub hosting the service
BannerURL	String	URL to servicebanner
ServiceURL	String	URL to webpage of service
Name	String	Name of the Service
TransactionNumber	String	ServiceTransactionNumber
ServiceId	Long	Ecohub-service registrationnumber
IconURL	String	URL to the serviceicon
SupportPhoneNumber	String	Phonenumber of the support-department of the service

Data Object: VirtualReceiptLineItem

Field	Type	Description
OrderId	Int	
VATPercentage	Decimal	Optional
VAT	Int	Optional
Price	Int	Includes VAT
NumberOfItems	Int	Optional
ExtraInformation	String	
Description	String	
DateTime	Long	Time represented in UnixTime

3.13 Response Codes

Response Code	Description / Example

200	OK
201	Created
202	Accepted
400	Bad request – Invalid model (data send to the API does not meet minimal requirements)
400	Bad request - serviceId needs to be a positive number.
400	Bad request – invalid sensor (sensorid is unknown)
400	Bad request – no certificate available (signature is set, but certificate is not known for this sensorid)
400	Bad request – no signature available (no signature is set, but is required by sensorconfiguration)
400	Bad request – invalid signature (signature does not match the data provided)
500	Internal server error – general error.
409	Conflict – Object we try to create already exists. (example: en account with known emailaddress)

3.14 Message Signing

To mitigate MITM attacks, all communication to and from the Hub should be signed.

All outbound messages from the HUB are signed by default. It is up to the receiving party to verify these signatures.

While all inbound messages should be signed, the Hub can be configured to allow unsigned messages to be accepted.

This allows exceptions to be made for external services and devices that are unable to sign messages on a case by case basis.

The Signature employs the following fields in a message:

3.15 Data object: Signature

Field	Type	Description / Value
Signature	String	Base64 encoded ECDSA signature over the concatenated message data values
SignatureThumbprint	String	Base64 encoded fingerprint of the receiving party's certificate. The fingerprint is a SHA1 hash of the public key.

3.15.1 Signature Creation & Verification

Signature Creation

Messages are signed using the following procedure.

Payload Creation

1. Per included field: concatenate the field length (ranges from 0 through 99999) + the fieldvalue.

1.1 If Non Mandatory Object like 'PropertyBag' is empty or null: object is skipped.

1.2 If Non Mandatory Object is filled: include children (items)

1.3 If Non Mandatory Type is empty like string: include length of zero and value empty.

1.4 If Non Mandatory Type is null like string or nullable integer: skipped.

1.5 For each key value combination like dictionaries and/or propertybags, first the key and then the value is included in the signature input.

1.6 For other collection types like arrays and lists all the values (in order) are included in the signature input.

2. Concatenate all fields based on the order mentioned in the models above.

Payload Hashing

3. A Signature is created by running the the binary representation of the field concatenation mentioned above through a SHA256 hash using the private key of the Hub-Side, Sensor- or Service-Side certificate.

Payload Signing

The signature itself comprises the ASN.1 DER structured r and s fields of the signature:

```
ECDSASignature ::= SEQUENCE {
   r  INTEGER,
   s  INTEGER
}
```

The ECDSA signature SHALL be verified according to [X9.62, §7.4], using SHA256 as the hash function and elliptic curve domain parameters as specified by [FIPS 186-4, § D.1.2.3] with fixed Curve-ID secp256r1. The public key to be used is the Hub public key for Hub outbound messages.

If the signature consists of zero bytes (0x00), the receiving party shall decline the transaction.

Signature Verification

Step 1:

The SignatureThumbprint shall be used by the counterparty to select the appropriate stored key.

Step 2:

Recreate Payload from message.

See Payload Creation 0

Step 3

Hash Payload

See Payload Hashing 0

Step 4:

Verify Signature

The signature itself comprises the ASN.1 DER structured r and s fields of the signature:

ECDSASignature ::= SEQUENCE {

r INTEGER,

s INTEGER

}

The ECDSA signature SHALL be verified according to [X9.62, §7.4], using SHA256 as the hash function and elliptic curve domain parameters as specified by [FIPS 186-4, § D.1.2.3] with fixed Curve-ID secp256r1. The public key to be used is the Hub public key.

If the signature consists of zero bytes (0x00), the receiving party shall decline the transaction unless otherwise configured.