



# **The Sales Service API**

**Publish Date: 7 February 2014**

## Table of Contents

1	THE SALES SERVICE API .....	3
1.1	Introduction .....	3
1.2	API Security .....	3
1.2.1	Request headers .....	3
1.2.2	Other remarks .....	7
1.3	Error handling.....	8
1.4	Service Documentation.....	8
1.4.1	InformationElementService .....	8
1.4.2	LocalizedTextService .....	9
1.4.3	MobileProfileService .....	12
1.4.4	PaymentAgreementPaymentService .....	16
1.4.5	PaymentAgreementService .....	18
1.4.6	PaymentMethodService .....	19
1.4.7	ProductClassService .....	20
1.4.8	ProductGroupService .....	22
1.4.9	getProductGroup.....	22
1.4.10	ProductMediumService .....	24
1.4.11	ProductService.....	26
1.4.12	PublicTransportAuthorityService .....	28
1.4.13	RemoteMessageService .....	31
1.4.14	SingleTicketOrderService .....	32
1.4.15	StopService .....	35
1.4.16	SupportedLocalesService .....	35
1.4.17	TermsAndConditionsService .....	37
1.4.18	TimeLinesService .....	38
1.4.19	TravelCardClassicContentService .....	39
1.4.20	TravelCardClassicPurseOrderService .....	51
1.4.21	TravelCardClassicTicketOrderService.....	54
1.4.22	TravelCardClassicTicketPickUpService .....	57
1.4.23	ZoneService .....	65
1.5	Scenario documentation .....	65
1.5.1	Create a mobile profile from a mobile device .....	65
1.5.2	Create a payment agreement.....	66
1.5.3	List products with predefined geography .....	66
1.5.4	List products for a specific geography .....	66
1.5.5	Create a single ticket order from a mobile device.....	66
1.5.6	Payment using a payment agreement .....	67
1.5.7	Payment using CPA (Overcharged SMS) .....	67

## 1 THE SALES SERVICE API

### 1.1 Introduction

The Sales Service API is an API that supports selling travel products for PTAs (Public Transport Authorities). It supports listing available products given the clients capabilities and a given end user's input. A client's capabilities is typically the product medium and payment method, while an end user's input is typically a geography (departure zone and destination zone) or product class (single ticket, period tickets etc).

The API also provides services for retrieving value sets for these capabilities and the geography.

**NOTE:** Clients must always make sure that they can handle that new fields are added to the JSON objects that are returned from the services. If they do not, they will fail if we decide to do so. We will not signal an API change by increasing the API version number in these cases!!!

### 1.2 API Security

Certain parts of the API requires signed requests to use. This is to achieve non-repudiation of service operations that performs writes to the service. These are:

- SingleTicketOrderService
- TravelCardClassicTicketOrderService
- TravelCardClassicPurseOrderService
- TravelCardClassicContentService
- MobileProfileService
- PaymentAgreementService
- PaymentAgreementPaymentService
- All ticket pickup requests

#### 1.2.1 Request headers

There are several security-related header requirements that has to be handled for a client to be able to connect to and use the Fara Sales Service API:

- The client has to provide a Date header with the request.
- The client has to provide a Content-MD5 header with the request.
- The client has to provide the custom header X-Fara-ApiKey with the request to identify himself to the service.
- The client has to provide the custom header X-Fara-Signature with the request to authenticate himself and to secure the content of the request.

##### 1.2.1.1 The Date header

The Date header contains the timestamp for when the request is sent from the client. The date has to be formatted according to RFC 1123. If the given timestamp deviates from the server time by more than five minutes, the request will be rejected.

Example:

Date: Fri, 04 Jan 2013 14:13:00 CET

### 1.2.1.1.1 Related error messages

#### 1.2.1.1.1.1 Missing header

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
...

{
  "code" : "DATE_HEADER_VERIFICATION",
  "message" : "Date header was not present."
}
```

#### 1.2.1.1.1.2 Illegal formatting

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
...

{
  "code" : "DATE_HEADER_VERIFICATION",
  "message" : "Illegal date format in Date header. Please make sure it is
formatted according to RFC-1123 (http://www.ietf.org/rfc/rfc1123.txt)."
}
```

#### 1.2.1.1.1.3 Time deviation

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
...

{
  "code" : "DATE_HEADER_VERIFICATION",
  "message" : "Time skew was 4559379 ms, and the limit was 300000 ms."
}
```

### 1.2.1.2 The Content-MD5 header

The Content-MD5 header must contain a hash of the request body encoded using Base64 ([RFC 1864](#)).

Example:

```
Content-MD5: 1B2M2Y8AsgTpgAmY7PhCfg
```

### 1.2.1.2.1 Related error messages

#### 1.2.1.2.1.1 Missing header

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
...
```

```
{
  "code" : "CONTENT_MD5_HEADER_VERIFICATION",
  "message" : "Content-MD5 header was not present."
}
```

### 1.2.1.2.1.2 Verification failed

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
...
```

```
{
  "code" : "CONTENT_MD5_HEADER_VERIFICATION",
  "message" : "Failed to verify MD5 hash of content."
}
```

### 1.2.1.2.1.3 The X-Fara-ApiKey header

This is a generated api key that functions as the identity of the client. The api key is obtained when registering the client with the service. We use this to locate the correct public key for validating the signature. The api key is actually a UUID (Universally Unique Identifier).

Example:

```
X-Fara-ApiKey: e90604d0-567e-11e2-bcfd-0800200c9a66
```

### 1.2.1.2.2 Related error messages

Error messages related to this header is handled in the next section.

### 1.2.1.3 The X-Fara-Signature header

When the previous headers are added to the request, and the request is complete from the client side, we are ready to generate the last header. The header is based on the three previous headers in addition to some other elements from the request, namely the method, the path and the parameters.

The X-Fara-Signature contains the signature that verifies that the request comes from you as the client.

The signature is generated by using the SHA256WithRSAEncryption algorithm on a strictly defined string. This results in a long string that we later can verify on the server side by using the client's public key and the same strict string that we generate.

The string that is used as the base for the signature generation consists of 6 lines (each ended with the \n newline character, also the last one). Here is an example of how the string may look like:

```
GET
/api/v1.0/ptas
content-md5: 1B2M2Y8AsgTpgAmY7PhCfg
date: Fri, 04 Jan 2013 14:13:00 CET
x-fara-apikey: e90604d0-567e-11e2-bcfd-0800200c9a66
includeImages=true
```

- The first line contains the HTTP method of the request, in uppercase.
- The second line contains the path of the request, in other words the part of the url that identifies the resource you are working on.
- The next three lines is the headers we have mentioned earlier, ordered alphabetically and with the header names in lowercase. The header names are followed by a colon, and yes, the colon must be followed by a space character before the actual header value.
- The last line is the request parameters, url encoded and ordered alphabetically by parameter name. The name/value pairs is separated by & just as in the url.

### 1.2.1.3.1 Pseudocode

```
String stringToSign = uppercase(verb) + "\n" +
    path + "\n" +
    "content-md5: " + md5Header + "\n" +
    "date: " + dateHeader + "\n" +
    "x-fara-apikey: " + apiKey + "\n" +
    sort(lowercase(urlencode(requestParameters))) + "\n";

String signature = base64(sign(stringToSign));
```

#### 1.2.1.3.1.1 Example

```
X-Fara-Signature: hWKK_Rb8qhgzwsaspAI_mScctoR3EqtZmlM5jG9fQ-
bdjeqWHsHNvOk_yPN0AnBjLM0NajSvpCGzQ5lKVxnHu9-
5s5OdOiUVtWdsZSEfm2ny8ATxqEFx9RD7Dj1bZxv0gOQe0ti3fRD0uDmmgppjutQYj1vz_Hv1xPU
zCgFVfMsRfmy3WcqTGCRLgSd-
W7YgOnPG4n3TLDEX3WuY0gjjFRQ2AoABLU1W8IM7jv7G5BrTAiMrWJuqpUDHOIVbGr4zLo6T_U4
wPudZkEAz9gHd2luPN9V7DXVDj_e7cRHlQfiWl3tZ0o051R3xoW78LO-sSMaLYyG6RNrZ-
F6pOj2AgUyB1DyB9FfhLHwiyZH7C4qIvzTQfXypEd0JkrqijvTwdSExPsFtzHZ9dSyM0DB9Gz10
6eSFTKWJfQnaaDbJcnPupqOVIketlSHBib0kOW-
dzHKEvCXeie2eCGhNnGCvZeS0IR2DNtD9Hb5kwMNXh_BnTt37NCmD0znUrS1-res1f
```

### 1.2.1.3.2 Related error messages

#### 1.2.1.3.2.1 Missing X-Fara-ApiKey header

```
HTTP/1.1 403 Forbidden
Content-Type: application/json;charset=UTF-8
...
{
  "code" : "SIGNATURE_HEADER_VALIDATION",
  "message" : "Header 'X-Fara-ApiKey' was not present."
}
```

#### 1.2.1.3.2.2 Missing signature header

```
HTTP/1.1 403 Forbidden
Content-Type: application/json;charset=UTF-8
...
{
```

```

    "code" : "SIGNATURE_HEADER_VALIDATION",
    "message" : "Signature header not present! Please generate a signature
and send the value in the 'X-Fara-Signature' header."
  }

```

#### 1.2.1.3.2.3 Unknown api key

```

HTTP/1.1 403 Forbidden
Content-Type: application/json;charset=UTF-8
...

```

```

{
  "code" : "SIGNATURE_HEADER_VALIDATION",
  "message" : "The value of the 'X-Fara-ApiKey' is not a valid api key.
Make sure you have got it right."
}

```

#### 1.2.1.3.2.4 No public key registered to the api key

```

HTTP/1.1 403 Forbidden
Content-Type: application/json;charset=UTF-8
...

```

```

{
  "code" : "SIGNATURE_HEADER_VALIDATION",
  "message" : "The api key used has no public key registered. Contact the
service provider to resolve this issue."
}

```

#### 1.2.1.4 Signature validation failed

```

HTTP/1.1 403 Forbidden
Content-Type: application/json;charset=UTF-8
...

```

```

{
  "code" : "SIGNATURE_HEADER_VALIDATION",
  "message" : "Failed to validate signature. You are either using the wrong
private key, or signing the request wrong. On our side, the following was
the canonical request representation used: 'GET\n/api/v1.0/ptas\ncontent-
md5: 1B2M2Y8AsgTpgAmY7PhCfg\ndate: Fri, 04 Jan 2013 16:27:00 GMT\nx-fara-
apikey: RS00001\n\n'"
}

```

### 1.2.2 Other remarks

There are some issues regarding the binary format of the encryption keys on different platforms. The server side used Java.

E.g. for .NET clients, see [here](http://stackoverflow.com/questions/4887286/verify-dsa-signature-generated-by-java-in-c-sharp). (<http://stackoverflow.com/questions/4887286/verify-dsa-signature-generated-by-java-in-c-sharp>)

## 1.3 Error handling

In most cases where the API reports some error, a JSON object is returned in the response body (content type application/json).

The JSON object has the following format:

```
{
  "code" : "CODE",
  "message" : "A description of the error situation"
}
```

## 1.4 Service Documentation

### 1.4.1 InformationElementService

Service providing generic PTA-specific information elements typically used in the mobile application.

#### 1.4.1.1 getInformationElement

Gets a specific information element. The full content is included here.

**Complete name:**

InformationElementService.getInformationElement

**URL:**

```
{apiVersion}/ptas/{ptaShortName}/informationElements/{informationElementId}
```

**HTTP Method:**

GET

**Produces:**

```
application/json
{
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/informationElements/1"
  } ],
  "title" : "Om mobilreise",
  "contentType" : "text/html",
  "elementContent" : "<p>Her er innholdet.</p>",
  "key" : "1"
}
```

#### 1.4.1.1.1 Success:

**HTTP Status:**

200 - OK



### 1.4.1.1.2 Error situations:

**HTTP Status:**

404 - Not Found

If the given information element is not found.

### 1.4.1.2 listInformationElements

Lists all information elements. Note that the actual content is omitted here, and is only returned when requesting a specific information element.

**Complete name:**

InformationElementService.listInformationElements

**URL:**`/apiVersion}/ptas/{ptaShortName}/informationElements`**HTTP Method:**

GET

**Produces:**

application/json

```
[ {
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/informationElements"
  } ],
  "title" : "Om mobilreise",
  "contentType" : "text/html",
  "elementContent" : null,
  "key" : "1"
} ]
```

### 1.4.1.2.1 Success:

**HTTP Status:**

200 - OK

## 1.4.2 LocalizedTextService

Service that provides localized texts by a key. The Accept-Language header (<http://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html#sec5.3>) is used to resolve the locale. The language of the returned content is set in the Content-Language entity header (<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>).

### 1.4.2.1 **getLocalizedText**

Gets a text by it's key localized by the value of the Accept-Language request header. Will return texts localized with the default locale of the PTA if the Accept-Language header is not provided or is not supported by the PTA.

**Complete name:**

LocalizedTextService.getLocalizedText

**URL:**

`/apiVersion}/ptas/{ptaShortName}/localizedTexts/{key}`

**HTTP Method:**

GET

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/localizedTexts/KEY1"
  } ],
  "value" : "This is the text for KEY1.",
  "key" : "KEY1"
}
```

#### 1.4.2.1.1 **Success:**

**HTTP Status:**

200 - OK

#### 1.4.2.1.2 **Error situations:**

**HTTP Status:**

404 - Not Found

Api key not found.

**HTTP Status:**

404 - Not Found

PTA not found.

**HTTP Status:**

404 - Not Found

No text found with the given key.

### 1.4.2.2 listLocalizedTexts

Lists all available texts localized by the value of the Accept-Language request header. Will return texts localized with the default locale of the PTA if the Accept-Language header is not provided or is not supported by the PTA.

**Complete name:**

LocalizedTextService.listLocalizedTexts

**URL:**

`/apiVersion}/ptas/{ptaShortName}/localizedTexts`

**HTTP Method:**

GET

**Produces:**

application/json

```
[ {
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/localizedTexts/KEY1"
  } ],
  "value" : "This is the text for KEY1.",
  "key" : "KEY1"
}, {
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/localizedTexts/KEY2"
  } ],
  "value" : "This is the text for KEY2.",
  "key" : "KEY2"
} ]
```

#### 1.4.2.2.1 Success:

**HTTP Status:**

200 - OK

#### 1.4.2.2.2 Error situations:

**HTTP Status:**

404 - Not Found

Api key not found.

**HTTP Status:**

404 - Not Found

PTA not found.

### 1.4.3 MobileProfileService

Service that provides a profile for mobile application users. A profile is also required for payment agreements and mobile accounts. A mobile profile is created in 3 steps: create (initialize), bind (request activation code) and claim (activate using activation code).

#### 1.4.3.1 claimProfile

Claims a mobile profile validated by an activation code.

**Complete name:**

MobileProfileService.claimProfile

**URL:**

`/apiVersion}/ptas/{ptaShortName}/mobileProfiles`

**HTTP Method:**

PUT

**Consumes:**

`application/vnd.fara.mobile_profile.claim+json;ver=1`

```
{
  "activationCode" : "12345",
  "phoneNumber" : {
    "countryCode" : "47",
    "phoneNumber" : "90000000"
  },
  "password" : "my_Password1"
}
```

**Produces:**

`application/json`

```
{
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/mobileProfiles/1"
  } ],
  "sendEmailReceipt" : true,
  "emailAddress" : null,
  "active" : true,
  "userName" : "ABC-004790000000",
  "mobileProfileId" : "1"
}
```

#### 1.4.3.1.1 Success:

**HTTP Status:**

201 - Created

### 1.4.3.1.2 Error situations:

**HTTP Status:**

400 - Bad Request

There are numerous reasons for this. Details are provided in the response body.

**HTTP Status:**

404 - Not Found

If mobile profile was not found.

### 1.4.3.2 createMobileProfile

Initializes the creation of a new mobile profile.

**Complete name:**

MobileProfileService.createMobileProfile

**URL:**

`/apiVersion/ptas/ptaShortName/mobileProfiles`

**HTTP Method:**

POST

**Consumes:**

application/json

```
{
  "phoneNumber" : {
    "countryCode" : "47",
    "phoneNumber" : "90000000"
  },
  "emailAddress" : "ola.nordmann@norge.no",
  "sendEmailReceipt" : true,
  "password" : "my_Password1"
}
```

#### 1.4.3.2.1 Success:

**HTTP Status:**

204 - No Content

#### 1.4.3.2.2 Error situations:

**HTTP Status:**

400 - Bad Request

The profile already exists, or there are other validation issues with the supplied data. Details are provided in the response body. Code = PROFILE\_ALREADY\_EXIST

**HTTP Status:**

400 - Bad Request

The user connected to the profile is locked. Details are provided in the response body. Code = MOBILE\_PROFILE\_DISABLED\_OR\_LOCKED

**1.4.3.3      getMobileProfile**

Retrieves a mobile profile. If the profile is inactive an error is sent

**Complete name:**

MobileProfileService.getMobileProfile

**URL:**

/{apiVersion}/ptas/{ptaShortName}/mobileProfiles/{mobileProfileId}

**HTTP Method:**

GET

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/mobileProfiles/1"
  } ],
  "sendEmailReceipt" : true,
  "emailAddress" : null,
  "active" : true,
  "userName" : "ABC-004790000000",
  "mobileProfileId" : "1"
}
```

**1.4.3.3.1      Success:****HTTP Status:**

200 - OK

**1.4.3.3.2      Error situations:****HTTP Status:**

404 - Not Found

If the mobile profile was not found.

**1.4.3.4      rebindProfile**

Sends a new activation code to user. Used if you already have a profile, but it's not activated, or the client have forgotten the password.

**Complete name:**

MobileProfileService.rebindProfile

**URL:**

/{apiVersion}/ptas/{ptaShortName}/mobileProfiles

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.mobile\_profile.rebind+json;ver=1

```
{
  "phoneNumber" : {
    "countryCode" : "47",
    "phoneNumber" : "90000000"
  }
}
```

**1.4.3.4.1 Success:****HTTP Status:**

200 - OK

**1.4.3.4.2 Error situations:****HTTP Status:**

400 - Bad Request

There are numerous reasons for this. Details are provided in the response body.

**1.4.3.5 updateMobileProfile**

Updates an existing mobile profile.

**Complete name:**

MobileProfileService.updateMobileProfile

**URL:**

/{apiVersion}/ptas/{ptaShortName}/mobileProfiles/{mobileProfileId}

**HTTP Method:**

PUT

**Consumes:**

application/json

```
{
  "sendEmailReceipt" : true,
  "emailAddress" : null
}
```

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/mobileProfiles/1"
  } ],
  "sendEmailReceipt" : true,
  "emailAddress" : null,
  "active" : true,
  "userName" : "ABC-004790000000",
  "mobileProfileId" : "1"
}
```

**1.4.3.5.1 Success:**

**HTTP Status:**

200 - OK

**1.4.3.5.2 Error situations:**

**HTTP Status:**

404 - Not Found

If the mobile profile was not found.

**1.4.4 PaymentAgreementPaymentService**

Service used to pay using an existing payment agreement. A payment consist of two steps: create and capture.

**1.4.4.1 capturePaymentAgreementPayment**

Effectuates an already existing payment.

**Complete name:**

PaymentAgreementPaymentService.capturePaymentAgreementPayment

**URL:**

`/apiVersion}/ptas/{ptaShortName}/mobileProfiles/{mobileProfileId}/agreements/{agreementName}/payments/{transactionReference}`

**HTTP Method:**

PUT

**Consumes:**

application/json

```
{
  "orderId" : "ABC_UHFPWE"
}
```



**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/mobileProfiles/1/agreements/VI
SA-1234/payments/123456abcdef"
  } ],
  "transactionNo" : 123,
  "transactionReference" : "123456abcdef"
}
```

**1.4.4.1 Success:****HTTP Status:**

200 - OK

**1.4.4.2 createPaymentAgreementPayment**

Creates a new payment. Note that it has to be captured to be effectuated. Captured should be done after the ticket is actually delivered to the buyer.

**Complete name:**

PaymentAgreementPaymentService.createPaymentAgreementPayment

**URL:**

```
/{apiVersion}/ptas/{ptaShortName}/mobileProfiles/{mobileProfileId}/agreements/{agreementName}/
payments
```

**HTTP Method:**

POST

**Consumes:**

application/json

```
{
  "orderId" : "ABC_UHFPWE"
}
```

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/mobileProfiles/1/agreements/VI
SA-1234/payments/123456abcdef"
  } ],
  "transactionNo" : 123,
  "transactionReference" : "123456abcdef"
}
```

#### 1.4.4.2.1 Success:

**HTTP Status:**

201 - Created

### 1.4.5 PaymentAgreementService

Service handling payment agreements. Payment agreements can be used to pay using your credit card in a simple manner, not requiring card details on every payment.

#### 1.4.5.1 completePaymentAgreement

Finalizes the creation of a payment agreement for a mobile profile.

**Complete name:**

PaymentAgreementService.completePaymentAgreement

**URL:**`/apiVersion}/ptas/{ptaShortName}/mobileProfiles/{mobileProfileId}/agreements/pending`**HTTP Method:**

PUT

**Consumes:**

application/json

```
{
  "orderRef" : "1234567890"
}
```

**Produces:**

application/json

```
{
  "agreementName" : "VISA-1234"
}
```

#### 1.4.5.1.1 Success:

**HTTP Status:**

200 - OK

#### 1.4.5.2 createPaymentAgreement

Initializes the creation of a new payment agreement for a mobile profile.

**Complete name:**

PaymentAgreementService.createPaymentAgreement

**URL:**`/apiVersion}/ptas/{ptaShortName}/mobileProfiles/{mobileProfileId}/agreements`

**HTTP Method:**

POST

**Produces:**

application/json

**1.4.5.2.1 Success:****HTTP Status:**

201 - Created

**1.4.5.3 deletePaymentAgreement**

Deletes an existing payment agreement for a mobile profile.

**Complete name:**

PaymentAgreementService.deletePaymentAgreement

**URL:**

/{apiVersion}/ptas/{ptaShortName}/mobileProfiles/{mobileProfileId}/agreements/{agreementName}

**HTTP Method:**

DELETE

**Produces:**

application/json

**1.4.5.3.1 Success:****HTTP Status:**

200 - OK

**1.4.6 PaymentMethodService****1.4.6.1 getPaymentMethod****Complete name:**

PaymentMethodService.getPaymentMethod

**URL:**

/{apiVersion}/ptas/{ptaShortName}/paymentMethods/{paymentMethodId}

**HTTP Method:**

GET

**Produces:**

application/json

```
{
  "links" : [ {
```

```

    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/paymentMethods/6"
  } ],
  "name" : "Mobilkonto",
  "description" : "Mobilkonto",
  "paymentMethodId" : 6
}

```

#### 1.4.6.1.1 Success:

**HTTP Status:**

200 - OK

#### 1.4.6.2 listPaymentMethods

**Complete name:**

PaymentMethodService.listPaymentMethods

**URL:**

`/apiVersion}/ptas/{ptaShortName}/paymentMethods`

**HTTP Method:**

GET

**Produces:**

application/json

```

[ {
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/paymentMethods/6"
  } ],
  "name" : "Mobilkonto",
  "description" : "Mobilkonto",
  "paymentMethodId" : 6
} ]

```

#### 1.4.6.2.1 Success:

**HTTP Status:**

200 - OK

### 1.4.7 ProductClassService

#### 1.4.7.1 getProductClass

**Complete name:**

ProductClassService.getProductClass

**URL:**

`/apiVersion}/ptas/{ptaShortName}/productClasses/{productClassId}`

**HTTP Method:**

GET

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/productClasses/11"
  } ],
  "description" : "STR-Single",
  "productClassId" : 11
}
```

**1.4.7.1.1 Success:****HTTP Status:**

200 - OK

**1.4.7.2 listProductClasses****Complete name:**

ProductClassService.listProductClasses

**URL:**

/{apiVersion}/ptas/{ptaShortName}/productClasses

**HTTP Method:**

GET

**Produces:**

application/json

```
[ {
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/productClasses/11"
  } ],
  "description" : "STR-Single",
  "productClassId" : 11
} ]
```

**1.4.7.2.1 Success:****HTTP Status:**

200 - OK

## 1.4.8 ProductGroupService

### 1.4.9 getProductGroup

Gets a product group that satisfies various parameters.

**Complete name:**

ProductGroupService.getProductGroup

**URL:**

`/apiVersion/ptas/ptaShortName/productGroups/productGroupId`

**HTTP Method:**

GET

**Parameters:**

fromZone

Required:

false

paymentMethod

The payment method the product group has to support. See `PaymentMethodService.listPaymentMethods`.

Required:

true

productClasses

The product classes to get. See `ProductClassService.listProductClasses`.

Required:

false

Default value:

11

productMedium

The product medium the product group has to support. See `ProductMediumService.listProductMedia`.

Required:

true

toZone

Required:

false

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/productGroups/2"
  }, {
    "rel" : "products",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/productGroups/2/products"
  } ],
  "name" : "Enkeltbillett ABC",
  "description" : "Enkeltbillett ABC",
  "quantitySupported" : true,
  "multipleProductsSupported" : true,
  "purseAmountConstraint" : null,
  "productGroupId" : 2
}
```

#### 1.4.9.1.1 Success:

**HTTP Status:**

200 - OK

#### 1.4.9.2 listProductGroups

Lists product groups that satisfies various parameters.

**Complete name:**

ProductGroupService.listProductGroups

**URL:**

/{apiVersion}/ptas/{ptaShortName}/productGroups

**HTTP Method:**

GET

**Parameters:**fromZone

Required:

false

paymentMethod

The payment method the product group has to support. See PaymentMethodService.listPaymentMethods.

Required:

true

productClasses

The product classes to get. See ProductClassService.listProductClasses.

Required:

false

Default value:

11

productMedium

The product medium the product group has to support. See `ProductMediumService.listProductMedia`.

Required:

true

toZone

Required:

false

**Produces:**

application/json

```
[ {
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/productGroups/2"
  }, {
    "rel" : "products",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/productGroups/2/products"
  } ],
  "name" : "Enkeltbillett ABC",
  "description" : "Enkeltbillett ABC",
  "quantitySupported" : true,
  "multipleProductsSupported" : true,
  "purseAmountConstraint" : null,
  "productGroupId" : 2
} ]
```

#### 1.4.9.2.1 Success:

**HTTP Status:**

200 - OK

### 1.4.10 ProductMediumService

#### 1.4.10.1 getProductMedium

**Complete name:**

`ProductMediumService.getProductMedium`

**URL:**

`/apiVersion}/ptas/{ptaShortName}/productMedia/{productMediumId}`



**HTTP Method:**

GET

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/productMedia/4"
  } ],
  "mediumType" : "QR-CODE",
  "cardType" : null,
  "cardApplication" : null,
  "productMediumId" : 4
}
```

**1.4.10.1.1 Success:****HTTP Status:**

200 - OK

**1.4.10.2 listProductMedia****Complete name:**

ProductMediumService.listProductMedia

**URL:**

/{apiVersion}/ptas/{ptaShortName}/productMedia

**HTTP Method:**

GET

**Produces:**

application/json

```
[ {
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/productMedia/4"
  } ],
  "mediumType" : "QR-CODE",
  "cardType" : null,
  "cardApplication" : null,
  "productMediumId" : 4
} ]
```

**1.4.10.2.1 Success:****HTTP Status:**

200 - OK

## 1.4.11 ProductService

### 1.4.11.1 getProduct

**Complete name:**

ProductService.getProduct

**URL:**

`/apiVersion}/ptas/{ptaShortName}/productGroups/{productGroupId}/products/{productId}`

**HTTP Method:**

GET

**Parameters:**

fromZone

Required:

false

paymentMethod

Required:

true

productMedium

Required:

true

toZone

Required:

false

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" :
    "https://mobil.api.fara.no/api/v1.0/ptas/PTA/productGroups/2/products/91%3B
    2%3B94%3D1%2695%3D6%3B95%3D6"
  } ],
  "productClass" : "11",
  "productGroup" : "Enkeltbillett ABC",
  "name" : "Voksen",
  "description" : "Enkeltbillett ABC - Voksen",
  "priceCurrency" : "NOK",
  "priceAmount" : 20.0,
  "priceVatPercentage" : 8.0,
  "priceVatAmount" : 1.6,
  "usageValidityDurationMinutes" : 45,
  "productId" : "91;2;94=1&95=6;95=6"
}
```

**1.4.11.1 Success:****HTTP Status:**

200 - OK

**1.4.11.2 listProducts****Complete name:**

ProductService.listProducts

**URL:**

/{apiVersion}/ptas/{ptaShortName}/productGroups/{productGroupId}/products

**HTTP Method:**

GET

**Parameters:**fromZone

The from zone to use for dynamic geography. Must be used with the 'toZone' parameter to take effect.

Required:

false

paymentMethod

The payment method the product has to support

Required:

true

productMedium

The medium the product has to support

Required:

true

toZone

The to zone to use for dynamic geography. Must be used with the 'fromZone' parameter to take effect.

Required:

false

**Produces:**

application/json

```
[ {  
  "links" : [ {  
    "rel" : "self",
```

```

    "href" :
    "https://mobil.api.fara.no/api/v1.0/ptas/PTA/productGroups/2/products/91%3B
    2%3B94%3D1%2695%3D6%3B95%3D6"
  } ],
  "productClass" : "11",
  "productGroup" : "Enkeltbillett ABC",
  "name" : "Voksen",
  "description" : "Enkeltbillett ABC - Voksen",
  "priceCurrency" : "NOK",
  "priceAmount" : 20.0,
  "priceVatPercentage" : 8.0,
  "priceVatAmount" : 1.6,
  "usageValidityDurationMinutes" : 45,
  "productId" : "91;2;94=1&95=6;95=6"
} ]

```

#### 1.4.11.2.1 Success:

##### HTTP Status:

200 - OK

### 1.4.12 PublicTransportAuthorityService

#### 1.4.12.1 getPublicTransportAuthority

##### Complete name:

PublicTransportAuthorityService.getPublicTransportAuthority

##### URL:

/`{apiVersion}`/ptas/`{ptaShortName}`

##### HTTP Method:

GET

##### Parameters:

includeImages

Required:

false

Default value:

false

##### Produces:

application/json

```

{
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA"
  }, {
    "rel" : "cardGraphics",
    "href" : "https://mobil.api.fara.no/resources/ptaCardGraphics/PTA"
  } ]
}

```

```

}, {
  "rel" : "logo",
  "href" : "https://mobil.api.fara.no/resources/ptaLogos/PTA"
} ],
"name" : "ABC Fylkeskommune",
"shortName" : null,
"companyNo" : 999,
"networkNo" : 99,
"countryCode" : 999,
"countryCallingCode" : "0047",
"currency" : "NOK",
"organisationNumber" : "98765432101",
"financialReportEmail" : null,
"financialReportEmailEnabled" : false,
"geoDescription" : "ABC",
"geoLatitude" : 64.2,
"geoLongitude" : 18.3,
"profileColor" : null,
"information" : "<div>Some information here!</div>",
"logo" : null,
"cardGraphics" : null,
"contactInformationList" : [ ],
"senderAddress" : null,
"ptaId" : "PTA"
}

```

#### 1.4.12.1 Success:

**HTTP Status:**

200 - OK

#### 1.4.12.2 listPublicTransportAuthorities

**Complete name:**

PublicTransportAuthorityService.listPublicTransportAuthorities

**URL:**

`/apiVersion}/ptas`

**HTTP Method:**

GET

**Parameters:**

companyNo

Required:

false

countryCode

Required:

false

includeImages

Required:

false

Default value:

false

networkNo

Required:

false

**Produces:**

application/json

```
[ {
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA"
  }, {
    "rel" : "cardGraphics",
    "href" : "https://mobil.api.fara.no/resources/ptaCardGraphics/PTA"
  }, {
    "rel" : "logo",
    "href" : "https://mobil.api.fara.no/resources/ptaLogos/PTA"
  } ],
  "name" : "ABC Fylkeskommune",
  "shortName" : null,
  "companyNo" : 999,
  "networkNo" : 99,
  "countryCode" : 999,
  "countryCallingCode" : "0047",
  "currency" : "NOK",
  "organisationNumber" : "98765432101",
  "financialReportEmail" : null,
  "financialReportEmailEnabled" : false,
  "geoDescription" : "ABC",
  "geoLatitude" : 64.2,
  "geoLongitude" : 18.3,
  "profileColor" : null,
  "information" : "<div>Some information here!</div>",
  "logo" : null,
  "cardGraphics" : null,
  "contactInformationList" : [ ],
  "senderAddress" : null,
  "ptaId" : "PTA"
} ]
```

#### 1.4.12.2.1 **Success:**

**HTTP Status:**

200 - OK

## 1.4.13 RemoteMessageService

### 1.4.13.1 logRemoteMessage

Creates an entry in a mobile application log on the server.

**Complete name:**

RemoteMessageService.logRemoteMessage

**URL:**

/{{apiVersion}}/clientdevice/errorsituation/

**HTTP Method:**

POST

**Parameters:**

mobileInfo  
Message contents.

Required:

true

mobilePhoneType  
Mobile device model.

Required:

true

mobileVersion  
Mobile application version.

Required:

true

oSVersion  
Mobile OS version.

Required:

true

salesServiceVersion  
Verion of the API used.

Required:

true

typeOfMessage  
Type of message. Currently only 2 are supported: errorMessage and stats. Passing any other value will result in a 'bad request' response.

Required:

true

**Produces:**

application/json

**1.4.13.1.1 Success:**

**HTTP Status:**

200 - OK

**1.4.14 SingleTicketOrderService**

**1.4.14.1 addSingleTicketOrder**

**Complete name:**

SingleTicketOrderService.addSingleTicketOrder

**URL:**

`/apiVersion}/ptas/{ptaId}/singleTicketOrders`

**HTTP Method:**

POST

**Consumes:**

application/json

```
{
  "paymentMethod" : 6,
  "productMedium" : 4,
  "productIds" : [ "91;2;94=1&95=6;95=6" ],
  "receiptEmail" : "ola.nordmann@norge.no",
  "productMediumReference" : null,
  "fromStopIds" : [ 123 ],
  "toStopIds" : [ 456 ]
}
```

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/singleTicketOrders/PTA_UHFPWE"
  } ],
  "paymentMethod" : 6,
  "productMedium" : 4,
  "productIds" : [ "91;2;94=1&95=6;95=6" ],
  "receiptEmail" : null,
  "productMediumReference" : null,
  "fromStopIds" : null,
}
```



```
"toStopIds" : null,
"travelDocumentId" : "PTA_UHFPWE_1367332057509",
"paymentReference" : null,
"pickupUrl" :
"https://mobil.api.fara.no/api/v1.0/qrSingleTickets/PTA_UHFPWE_1367332057509",
"priceAmount" : 20.0,
"currency" : "NOK",
"orderId" : "PTA_UHFPWE"
}
```

#### 1.4.14.1.1 Success:

**HTTP Status:**

201 - Created

#### 1.4.14.1.2 Error situations:

**HTTP Status:**

400 - Bad Request

There are numerous reasons for this type of error: illegal payment method, illegal product medium, illegal product id. The body of the response will contain more information.

#### 1.4.14.2 cancelOrder

Cancels an order. Only available for clients handling the payment themselves.

**Complete name:**

SingleTicketOrderService.cancelOrder

**URL:**

/{apiVersion}/ptas/{ptaId}/singleTicketOrders/{orderId}

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.order.cancel+json;ver=1

```
{
  "reason" : "Payment failed"
}
```

**Produces:**

application/json

#### 1.4.14.2.1 Success:

**HTTP Status:**

200 - OK

### 1.4.14.3 commitOrder

Commits an order. Only available for clients handling the payment themselves.

**Complete name:**

SingleTicketOrderService.commitOrder

**URL:**

`/apiVersion/ptas/{ptaId}/singleTicketOrders/{orderId}`

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.order.commit+json;ver=1

```
{
  "transactionId" : "123456789",
  "terminalId" : "123456789"
}
```

**Produces:**

application/json

#### 1.4.14.3.1 Success:

**HTTP Status:**

200 - OK

### 1.4.14.4 requestEmailReceipt

Request an email receipt for an order.

**Complete name:**

SingleTicketOrderService.requestEmailReceipt

**URL:**

`/apiVersion/ptas/{ptaId}/singleTicketOrders/{orderId}`

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.order.request.email.receipt+json;ver=1

**Produces:**

application/json

#### 1.4.14.4.1 Success:

**HTTP Status:**

200 - OK

### 1.4.15 StopService

#### 1.4.15.1 listStops

**Complete name:**

StopService.listStops

**URL:**`/{apiVersion}/ptas/{ptaShortName}/stops`**HTTP Method:**

GET

**Produces:**

application/json

```
[ {
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/stops/1"
  } ],
  "name" : "City Centre",
  "zoneId" : 1,
  "utmCoordinate" : {
    "zone" : "32V",
    "easting" : 569882,
    "northing" : 7034935
  },
  "wgs84Coordinate" : {
    "longitude" : 10.400648,
    "latitude" : 63.436021
  },
  "stopId" : 1
} ]
```

#### 1.4.15.1.1 Success:

**HTTP Status:**

200 - OK

### 1.4.16 SupportedLocalesService

Service for listing supported locales for a PTA.

#### 1.4.16.1 getSupportedLocale

Gets a supported locale for a PTA.

**Complete name:**

SupportedLocalesService.getSupportedLocale

**URL:**

/{apiVersion}/ptas/{ptaShortName}/supportedLocales/{supportedLocaleId}

**HTTP Method:**

GET

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" :
    "https://mobil.api.fara.no/api/v1.0/ptas/PTA/supportedLocales/nb-no"
  } ],
  "name" : "Norsk Bokmål",
  "default" : true,
  "locale" : "nb-no"
}
```

**1.4.16.1.1 Success:****HTTP Status:**

200 - OK

**1.4.16.1.2 Error situations:****HTTP Status:**

404 - Not Found

Api key not found.

**HTTP Status:**

404 - Not Found

PTA not found.

**1.4.16.2 listSupportedLocales**

Lists all the supported locales for a PTA.

**Complete name:**

SupportedLocalesService.listSupportedLocales

**URL:**

/{apiVersion}/ptas/{ptaShortName}/supportedLocales

**HTTP Method:**

GET

**Produces:**

application/json

```
[ {
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/supportedLocales/en-gb"
  } ],
  "name" : "English (Great Britain)",
  "default" : false,
  "locale" : "en-gb"
}, {
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/supportedLocales/nb-no"
  } ],
  "name" : "Norsk Bokmål",
  "default" : true,
  "locale" : "nb-no"
} ]
```

**1.4.16.2.1 Success:****HTTP Status:**

200 - OK

**1.4.16.2.2 Error situations:****HTTP Status:**

404 - Not Found

Api key not found.

**HTTP Status:**

404 - Not Found

PTA not found.

**1.4.17 TermsAndConditionsService**

Service providing PTA-specific terms and conditions.

**1.4.17.1 getTermsAndConditions**

Gets a specific terms and conditions. The full content is included here.

**Complete name:**

TermsAndConditionsService.getTermsAndConditions

**URL:**

/{apiVersion}/ptas/{ptaShortName}/termsAndConditions

**HTTP Method:**

GET

**Produces:**

application/json

**1.4.17.1.1 Success:**

**HTTP Status:**

200 - OK

**1.4.18 TimeLinesService**

**1.4.18.1 listTimeLines**

Lists the current time line for each domain (BASE, PRODUCT, GEO).

**Complete name:**

TimeLinesService.listTimeLines

**URL:**

`/apiVersion}/ptas/{ptaShortName}/timeLines`

**HTTP Method:**

GET

**Produces:**

application/json

```
[ {
  "domain" : "BASE",
  "timelineVersion" : 1,
  "periods" : [ {
    "startDate" : "2014-01-01",
    "endDate" : "2099-12-31",
    "dataSetVersion" : 1
  } ]
}, {
  "domain" : "GEO",
  "timelineVersion" : 1,
  "periods" : [ {
    "startDate" : "2014-01-01",
    "endDate" : "2099-12-31",
    "dataSetVersion" : 1
  } ]
}, {
  "domain" : "PRODUCT",
  "timelineVersion" : 1,
  "periods" : [ {
    "startDate" : "2014-01-01",
    "endDate" : "2099-12-31",
    "dataSetVersion" : 1
  } ]
} ]
```

```
} ]
```

#### **1.4.18.1.1 Success:**

##### **HTTP Status:**

200 - OK

#### **1.4.18.1.2 Error situations:**

##### **HTTP Status:**

404 - Not Found

Either the PTA does not exist, or the API key is unknown. Check the error code and message provided in the response body.

### **1.4.19 TravelCardClassicContentService**

Service for extracting travel card content based on binary data input.

#### **1.4.19.1 extractTravelCardClassicContent**

Extract content of a MIFARE Classic travel card.

Request:

The travel card data is mandatory input.

In addition there are some optional settings for adjusting behaviour.

Time zone setting regulates date-time values, and default value is 'Europe/Stockholm'.

Language setting regulates any textual response, and default value is 'sv' (Swedish).

Include products not status OK setting (default false) regulates whether to include products with invalid status (e.g. 'BLOCKED')

The travel card will often contain old products that are not relevant any more, and will normally not be shown to the customer, but there is a setting for including products that is not valid now or in the future (default false).

There are also settings for including compact or verbose string representations of the products.

These string representations are intended for test/example purposes (not production), and may be removed in future versions of the API.

Response:

The response contains card ID, card status (OK,PENDING,DISABLED,BLOCKED) and an array of varying products.

The different classes of products has varying content (i.e. polymorphic behaviour.)

The product classes are: 'storedValueRight', 'storedTravelRight', 'customerEntitlement' and 'chargeToAccount'.

The product has a type (STR\_SINGLE, STR\_PUNCH, STR\_PERIOD, STR\_SCHOOL, STR\_SINGLE\_AUTO\_SVR, STR\_SINGLE\_AUTO\_CTA, STR\_GROUP, SVR\_STANDARD, CTA\_STANDARD, CUE\_BEST\_PRICE) that also affects how the product is to be interpreted.

Stored travel right (STR) products has a geography restricting the geographic area of validity within zones.

A 'fromZoneToZone' restricts validity to be between a start and end zone.

A 'zoneIntevalList' restricts validity to a series of start and end zones.

**Complete name:**

TravelCardClassicContentService.extractTravelCardClassicContent

**URL:**

`/apiVersion/classicTravelCardContent/`

**HTTP Method:**

POST

**Consumes:**

application/json

```
{
  "travelCardData" : {
    "sectors" : [ {
      "sectorNo" : 0,
      "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "NWk7FnGIBABDJUf/AAcDBQ=="
      }, {
        "blockNo" : 1,
        "bytes" : "AAADLRSYAVInAQACAABs1w=="
      }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAA=="
      }, {
        "blockNo" : 3,
        "bytes" : "qqqqqqqqqHeIQbu7u7u7uw=="
      } ]
    }, {
      "sectorNo" : 1,
      "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "BgAABQAACgAACHAAAAApg=="
      }, {
        "blockNo" : 1,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAcW=="
      }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAcW=="
      }, {
        "blockNo" : 3,
```



```

    "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 2,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "oEIEAABAAAAAAAAAAAAQAq1A=="
  }, {
    "blockNo" : 1,
    "bytes" : "oEIEAABAAAAAAAAAAAAQAq1A=="
  }, {
    "blockNo" : 2,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 3,
    "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 3,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "hAAAAAAAAAAAAAAAAAAAAAKw=="
  }, {
    "blockNo" : 1,
    "bytes" : "hAAAAAAAAAAAAAAAAAAAAAKw=="
  }, {
    "blockNo" : 2,
    "bytes" : "hAAAAAAAAAAAAAAAAAAAAAKw=="
  }, {
    "blockNo" : 3,
    "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 4,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "hAAAAAAAAAAAAAAAAAAAAAKw=="
  }, {
    "blockNo" : 1,
    "bytes" : "hAAAAAAAAAAAAAAAAAAAAAKw=="
  }, {
    "blockNo" : 2,
    "bytes" : "hAAAAAAAAAAAAAAAAAAAAAKw=="
  }, {
    "blockNo" : 3,
    "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 5,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 1,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 2,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAA=="
  }, {

```

```

        "blockNo" : 3,
        "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
}, {
    "sectorNo" : 6,
    "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 1,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 3,
        "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
}, {
    "sectorNo" : 7,
    "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 1,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 3,
        "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
}, {
    "sectorNo" : 8,
    "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 1,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 3,
        "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
}, {
    "sectorNo" : 9,
    "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 1,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    } ]
} ]

```

```

    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 10,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 2,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 11,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 2,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 12,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 2,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 13,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 2,

```

```

    "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 3,
    "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 14,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 1,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 2,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 3,
    "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 15,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 1,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 2,
    "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 3,
    "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
  } ]
} ]
},
"timeZone" : "Europe/Stockholm",
"language" : "sv",
"includeProductsNotStatusOk" : null,
"includeProductsNotValidForUseNowOrFuture" : null,
"includeCompactProductStrings" : null,
"includeVerboseProductStrings" : null
}

```

**Produces:**

application/json

```

{
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/classicTravelCardContent"
  } ],
  "travelCardId" : "372992309",
  "travelCardStatus" : "OK",
  "productList" : [ {
    "storedValueRight" : {
      "productType" : "SVR_STANDARD",

```

```

"displayNameExtraShort" : "Reskassa",
"displayNameShort" : "Reskassa",
"displayNameLong" : "Reskassa",
"exampleDisplayTextCompact" : "Reskassa 123.00",
"exampleDisplayTextVerbose" : "Reskassa: Reskassa balans 123.00",
"companyNo" : 180,
"productStatus" : "OK",
"notValidBeforeTime" : null,
"notValidAfterTime" : 1698620398000,
"purseBalance" : 123.00,
"validForUseNowOrFuture" : true
}
}, {
  "storedTravelRight" : {
    "productType" : "STR_SINGLE",
    "displayNameExtraShort" : "EnkB",
    "displayNameShort" : "Enk.bilj",
    "displayNameLong" : "Enkelbiljett Kalmar",
    "exampleDisplayTextCompact" : "3 Skolung och 2 Vuxen, Enkelbiljett
Kalmar, börja användas innan 2014-02-07 ut dagen",
    "exampleDisplayTextVerbose" : "Biljett: 3 Skolung och 2 Vuxen,
Enkelbiljett Kalmar, zonintervall: Tyllinge/Kasinge-Överum/Björnsholm,
börja användas innan 2014-02-07 ut dagen",
    "companyNo" : 180,
    "productStatus" : "OK",
    "notValidBeforeTime" : null,
    "notValidAfterTime" : null,
    "startUseBeforeTime" : 1391813998000,
    "passengersList" : [ {
      "categoryId" : 1,
      "categoryName" : "Vuxen",
      "count" : 2
    }, {
      "categoryId" : 156,
      "categoryName" : "Skolung",
      "count" : 3
    } ],
    "duration" : null,
    "durationUnitId" : null,
    "durationUnitText" : "",
    "travelsAvailableTotal" : 1,
    "travelsDoneToday" : 0,
    "travelsDoneTotal" : 0,
    "geography" : {
      "zoneIntervalList" : {
        "zoneIntervalList" : [ {
          "startZone" : {
            "zoneId" : 1,
            "zoneName" : "Tyllinge/Kasinge"
          },
          "endZone" : {
            "zoneId" : 99,
            "zoneName" : "Överum/Björnsholm"
          }
        } ]
      }
    }
  },
  "validForUseNowOrFuture" : true
}

```

```

    }, {
      "storedTravelRight" : {
        "productType" : "STR_SINGLE",
        "displayNameExtraShort" : "EnkB",
        "displayNameShort" : "Enk.bilj",
        "displayNameLong" : "Enkelbiljett",
        "exampleDisplayTextCompact" : "Vuxen, Enkelbiljett, Kastlösa-
Mörbylånga, börja användas innan 2014-02-07 ut dagen",
        "exampleDisplayTextVerbose" : "Biljett: 1 Vuxen, Enkelbiljett, från
zone Kastlösa till zone Mörbylånga, börja användas innan 2014-02-07 ut
dagen",
        "companyNo" : 180,
        "productStatus" : "OK",
        "notValidBeforeTime" : null,
        "notValidAfterTime" : null,
        "startUseBeforeTime" : 1391813998000,
        "passengersList" : [ {
          "categoryId" : 1,
          "categoryName" : "Vuxen",
          "count" : 1
        } ],
        "duration" : null,
        "durationUnitId" : null,
        "durationUnitText" : "",
        "travelsAvailableTotal" : 1,
        "travelsDoneToday" : 0,
        "travelsDoneTotal" : 0,
        "geography" : {
          "fromZoneToZone" : {
            "startZone" : {
              "zoneId" : 6,
              "zoneName" : "Kastlösa"
            },
            "endZone" : {
              "zoneId" : 7,
              "zoneName" : "Mörbylånga"
            }
          }
        }
      },
      "validForUseNowOrFuture" : true
    }
  ], {
    "storedTravelRight" : {
      "productType" : "STR_PERIOD",
      "displayNameExtraShort" : "30dlän",
      "displayNameShort" : "30dgrlän",
      "displayNameLong" : "30 dagars resperiod län",
      "exampleDisplayTextCompact" : "Studerande, 30 dagars resperiod län,
flexibel start, börja användas innan 2014-02-07 ut dagen, giltig 30 dagar",
      "exampleDisplayTextVerbose" : "Period: 1 Studerande, 30 dagars
resperiod län, zonintervall: Tyllinge/Kasinge-Överum/Björnsholm, Kalmar
tätort, Oskarshamn tätort, Västervik tätort, flexibel start, börja användas
innan 2014-02-07 ut dagen, giltig 30 dagar",
      "companyNo" : 180,
      "productStatus" : "OK",
      "notValidBeforeTime" : null,
      "notValidAfterTime" : null,
      "startUseBeforeTime" : 1391813998000,
      "passengersList" : [ {

```

```

    "categoryId" : 3,
    "categoryName" : "Studerande",
    "count" : 1
  } ],
  "duration" : 30,
  "durationUnitId" : 2,
  "durationUnitText" : "dagar",
  "travelsAvailableTotal" : null,
  "travelsDoneToday" : 0,
  "travelsDoneTotal" : 0,
  "geography" : {
    "zoneIntervalList" : {
      "zoneIntervalList" : [ {
        "startZone" : {
          "zoneId" : 1,
          "zoneName" : "Tyllinge/Kasinge"
        },
        "endZone" : {
          "zoneId" : 99,
          "zoneName" : "Överum/Björnsholm"
        }
      }, {
        "startZone" : {
          "zoneId" : 340,
          "zoneName" : "Kalmar tätort"
        },
        "endZone" : {
          "zoneId" : 340,
          "zoneName" : "Kalmar tätort"
        }
      }, {
        "startZone" : {
          "zoneId" : 610,
          "zoneName" : "Oskarshamn tätort"
        },
        "endZone" : {
          "zoneId" : 610,
          "zoneName" : "Oskarshamn tätort"
        }
      }, {
        "startZone" : {
          "zoneId" : 950,
          "zoneName" : "Västervik tätort"
        },
        "endZone" : {
          "zoneId" : 950,
          "zoneName" : "Västervik tätort"
        }
      }
    ]
  }
},
"validForUseNowOrFuture" : true
}
}, {
  "storedTravelRight" : {
    "productType" : "STR_PUNCH",
    "displayNameExtraShort" : "10 l",
    "displayNameShort" : "10reslän",
    "displayNameLong" : "10 resor län",

```

```

"exampleDisplayTextCompact" : "Vuxen, 10 resor län, giltig till 2014-08-06 ut dagen, 7 punch återstående",
"exampleDisplayTextVerbose" : "Punch: 1 Vuxen, 10 resor län, zonintervall: Tyllinge/Kasinge-Överum/Björnsholm, Kalmar tätort, Oskarshamn tätort, Västervik tätort, giltig till 2014-08-06 ut dagen, 7 punch återstående (från total 10)",
"companyNo" : 180,
"productStatus" : "OK",
"notValidBeforeTime" : null,
"notValidAfterTime" : 1407362398000,
"startUseBeforeTime" : null,
"passengersList" : [ {
  "categoryId" : 1,
  "categoryName" : "Vuxen",
  "count" : 1
} ],
"duration" : null,
"durationUnitId" : null,
"durationUnitText" : "",
"travelsAvailableTotal" : 10,
"travelsDoneToday" : 1,
"travelsDoneTotal" : 3,
"geography" : {
  "zoneIntervalList" : {
    "zoneIntervalList" : [ {
      "startZone" : {
        "zoneId" : 1,
        "zoneName" : "Tyllinge/Kasinge"
      },
      "endZone" : {
        "zoneId" : 99,
        "zoneName" : "Överum/Björnsholm"
      }
    }, {
      "startZone" : {
        "zoneId" : 340,
        "zoneName" : "Kalmar tätort"
      },
      "endZone" : {
        "zoneId" : 340,
        "zoneName" : "Kalmar tätort"
      }
    }, {
      "startZone" : {
        "zoneId" : 610,
        "zoneName" : "Oskarshamn tätort"
      },
      "endZone" : {
        "zoneId" : 610,
        "zoneName" : "Oskarshamn tätort"
      }
    }, {
      "startZone" : {
        "zoneId" : 950,
        "zoneName" : "Västervik tätort"
      },
      "endZone" : {
        "zoneId" : 950,
        "zoneName" : "Västervik tätort"
      }
    }
  ]
}

```



```

    }
  } ]
}
},
  "validForUseNowOrFuture" : true
}
}, {
  "storedTravelRight" : {
    "productType" : "STR_SCHOOL",
    "displayNameExtraShort" : "Skol",
    "displayNameShort" : "Skol",
    "displayNameLong" : "Västervik Kommunskolekort en resa",
    "exampleDisplayTextCompact" : "Ungdom, Västervik Kommunskolekort en
resa, börja användas innan 2014-02-07 ut dagen",
    "exampleDisplayTextVerbose" : "Skola: 1 Ungdom, Västervik
Kommunskolekort en resa, zonintervall: Tyllinge/Kasinge-Loftahammar,
Blankaholm-Överum/Björnsholm, Västervik tätort, börja användas innan 2014-
02-07 ut dagen",
    "companyNo" : 180,
    "productStatus" : "OK",
    "notValidBeforeTime" : null,
    "notValidAfterTime" : null,
    "startUseBeforeTime" : 1391813998000,
    "passengersList" : [ {
      "categoryId" : 64,
      "categoryName" : "Ungdom",
      "count" : 1
    } ],
    "duration" : null,
    "durationUnitId" : null,
    "durationUnitText" : "",
    "travelsAvailableTotal" : null,
    "travelsDoneToday" : 0,
    "travelsDoneTotal" : 0,
    "geography" : {
      "zoneIntervalList" : {
        "zoneIntervalList" : [ {
          "startZone" : {
            "zoneId" : 1,
            "zoneName" : "Tyllinge/Kasinge"
          },
          "endZone" : {
            "zoneId" : 3,
            "zoneName" : "Loftahammar"
          }
        } ],
        "startZone" : {
          "zoneId" : 90,
          "zoneName" : "Blankaholm"
        },
        "endZone" : {
          "zoneId" : 99,
          "zoneName" : "Överum/Björnsholm"
        }
      },
      "startZone" : {
        "zoneId" : 950,
        "zoneName" : "Västervik tätort"
      }
    }
  },

```

```

    "endZone" : {
      "zoneId" : 950,
      "zoneName" : "Västervik tätort"
    }
  } ]
}
},
"validForUseNowOrFuture" : true
}
}, {
"storedTravelRight" : {
  "productType" : "STR_SINGLE_AUTO_SVR",
  "displayNameExtraShort" : "Best-PriceK",
  "displayNameShort" : "Best-PriceK",
  "displayNameLong" : "Bestprice Kalmar",
  "exampleDisplayTextCompact" : "Vuxen, Bestprice Kalmar, Västervik
tätort, giltig till 2023-11-13 ut dagen",
  "exampleDisplayTextVerbose" : "Auto resor: 1 Vuxen, Bestprice Kalmar,
inom zon Västervik tätort, giltig till 2023-11-13 ut dagen",
  "companyNo" : 180,
  "productStatus" : "OK",
  "notValidBeforeTime" : null,
  "notValidAfterTime" : 1699916398000,
  "startUseBeforeTime" : null,
  "passengersList" : [ {
    "categoryId" : 1,
    "categoryName" : "Vuxen",
    "count" : 1
  } ],
  "duration" : null,
  "durationUnitId" : null,
  "durationUnitText" : "",
  "travelsAvailableTotal" : 0,
  "travelsDoneToday" : 0,
  "travelsDoneTotal" : 0,
  "geography" : {
    "fromZoneToZone" : {
      "startZone" : {
        "zoneId" : 950,
        "zoneName" : "Västervik tätort"
      },
      "endZone" : {
        "zoneId" : 950,
        "zoneName" : "Västervik tätort"
      }
    }
  },
  "validForUseNowOrFuture" : true
}
}, {
"customerEntitlement" : {
  "productType" : "CUE_BEST_PRICE",
  "displayNameExtraShort" : "Rabattavtal",
  "displayNameShort" : "Rabattavtal",
  "displayNameLong" : "Rabattavtal",
  "exampleDisplayTextCompact" : "Rabattavtal nivå 42",
  "exampleDisplayTextVerbose" : "Bästa pris: Rabattavtal
(Rabattavtalsprodukt) genomsnittliga nivån 42",
  "companyNo" : 180,

```

```

    "productStatus" : "OK",
    "notValidBeforeTime" : null,
    "notValidAfterTime" : 1698620398000,
    "bestPriceAverageLevel" : 42,
    "validForUseNowOrFuture" : true
  }
}, {
  "chargeToAccount" : {
    "productType" : "CTA_STANDARD",
    "displayNameExtraShort" : "Före",
    "displayNameShort" : "Företag",
    "displayNameLong" : "Företagskort",
    "exampleDisplayTextCompact" : "Belasta konto, Företag: 000000001005",
    "exampleDisplayTextVerbose" : "Belasta konto: Företagskort:
000000001005",
    "companyNo" : 180,
    "productStatus" : "OK",
    "notValidBeforeTime" : null,
    "notValidAfterTime" : 1698620398000,
    "accountNumber" : "000000001005",
    "validForUseNowOrFuture" : true
  }
} ]
}

```

#### 1.4.19.1.1 Success:

**HTTP Status:**

200 - OK

#### 1.4.19.1.2 Error situations:

**HTTP Status:**

400 - Bad Request

Invalid input. Response body may contain details (in English) about the cause.

### 1.4.20 TravelCardClassicPurseOrderService

Service for handling travel card purse orders (adding money to the purse).

#### 1.4.20.1 addClassicTravelCardPurseOrder

Place a MIFARE Classic travel card purse sales order. Product medium must be 102 or omitted. Product medium reference is the card number.

**Complete name:**

TravelCardClassicPurseOrderService.addClassicTravelCardPurseOrder

**URL:**

`/v{apiVersion}/ptas/{ptaId}/classicTravelCardPurseOrders`

**HTTP Method:**

POST

**Consumes:**

application/json

```
{
  "paymentMethod" : 9,
  "productMedium" : 102,
  "receiptEmail" : null,
  "productMediumReference" : "372992309",
  "amount" : 500
}
```

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/classicTravelCardTickets/PTA_UHFPWE"
  } ],
  "paymentMethod" : 9,
  "productMedium" : 102,
  "productIds" : [ "91;70;95=9&96=500;95=9" ],
  "receiptEmail" : null,
  "productMediumReference" : "372992309",
  "fromStopIds" : null,
  "toStopIds" : null,
  "travelDocumentId" : "PTA_UHFPWE_1367332057509",
  "paymentReference" : null,
  "pickupUrl" :
"https://mobil.api.fara.no/api/v1.0/classicTravelCardTickets/PTA_UHFPWE_1367332057509",
  "priceAmount" : 500.0,
  "currency" : "SEK",
  "orderId" : "PTA_UHFPWE"
}
```

**1.4.20.1.1 Success:**

**HTTP Status:**

201 - Created

**1.4.20.1.2 Error situations:**

**HTTP Status:**

400 - Bad Request

There are numerous reasons for this type of error: illegal payment method, illegal product medium, illegal product id. The body of the response will contain more information.

### 1.4.20.2 cancelOrder

Cancels an order. Only available for clients handling the payment themselves.

**Complete name:**

TravelCardClassicPurseOrderService.cancelOrder

**URL:**

`/v1/ptas/{ptaId}/classicTravelCardPurseOrders/{orderId}`

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.order.cancel+json;ver=1

```
{
  "reason" : "Payment failed"
}
```

**Produces:**

application/json

#### 1.4.20.2.1 Success:

**HTTP Status:**

200 - OK

### 1.4.20.3 commitOrder

Commits an order. Only available for clients handling the payment themselves.

**Complete name:**

TravelCardClassicPurseOrderService.commitOrder

**URL:**

`/v1/ptas/{ptaId}/classicTravelCardPurseOrders/{orderId}`

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.order.commit+json;ver=1

```
{
  "transactionId" : "123456789",
  "terminalId" : "123456789"
}
```

**Produces:**

application/json

**1.4.20.3.1 Success:****HTTP Status:**

200 - OK

**1.4.20.4 requestEmailReceipt**

Request an email receipt for an order.

**Complete name:**

TravelCardClassicPurseOrderService.requestEmailReceipt

**URL:**

/{apiVersion}/ptas/{ptaId}/classicTravelCardPurseOrders/{orderId}

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.order.request.email.receipt+json;ver=1

**Produces:**

application/json

**1.4.20.4.1 Success:****HTTP Status:**

200 - OK

**1.4.21 TravelCardClassicTicketOrderService**

Service for handling MIFARE Classic travel card orders.

**1.4.21.1 addTravelCardClassicOrder**

Place a MIFARE Classic travel card ticket sales order. Product medium must be 102 or omitted. Product medium reference is the card number.

**Complete name:**

TravelCardClassicTicketOrderService.addTravelCardClassicOrder

**URL:**

/{apiVersion}/ptas/{ptaId}/classicTravelCardOrders

**HTTP Method:**

POST

**Consumes:**

application/json

```
{
  "paymentMethod" : 9,
  "productMedium" : 102,
  "productIds" : [ "91;64;79=30&94=1&95=9;95=9" ],
  "receiptEmail" : null,
  "productMediumReference" : "372992309",
  "fromStopIds" : null,
  "toStopIds" : null
}
```

**Produces:**

application/json

```
{
  "links" : [ {
    "rel" : "self",
    "href" :
"https://mobil.api.fara.no/api/v1.0/ptas/PTA/classicTravelCardOrders/PTA_UH
FPWE"
  } ],
  "paymentMethod" : 9,
  "productMedium" : 102,
  "productIds" : [ "91;64;79=30&94=1&95=9;95=9" ],
  "receiptEmail" : null,
  "productMediumReference" : "372992309",
  "fromStopIds" : null,
  "toStopIds" : null,
  "travelDocumentId" : "PTA_UHFPWE_1367332057509",
  "paymentReference" : "830840578197969209@12345",
  "pickupUrl" :
"https://mobil.api.fara.no/api/v1.0/classicTravelCardTickets/PTA_UHFPWE_136
7332057509",
  "priceAmount" : 1120.0,
  "currency" : "SEK",
  "orderId" : "PTA_UHFPWE"
}
```

**1.4.21.1.1 Success:**
**HTTP Status:**

201 - Created

**1.4.21.1.2 Error situations:**
**HTTP Status:**

400 - Bad Request

There are numerous reasons for this type of error: illegal payment method, illegal product medium, illegal product id. The body of the response will contain more information.

### 1.4.21.2 cancelOrder

Cancels an order. Only available for clients handling the payment themselves.

**Complete name:**

TravelCardClassicTicketOrderService.cancelOrder

**URL:**

`/apiVersion/ptas/{ptaId}/classicTravelCardOrders/{orderId}`

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.order.cancel+json;ver=1

```
{
  "reason" : "Payment failed"
}
```

**Produces:**

application/json

#### 1.4.21.2.1 Success:

**HTTP Status:**

200 - OK

### 1.4.21.3 commitOrder

Commits an order. Only available for clients handling the payment themselves.

**Complete name:**

TravelCardClassicTicketOrderService.commitOrder

**URL:**

`/apiVersion/ptas/{ptaId}/classicTravelCardOrders/{orderId}`

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.order.commit+json;ver=1

```
{
  "transactionId" : "123456789",
  "terminalId" : "123456789"
}
```

**Produces:**

application/json



**1.4.21.3.1 Success:****HTTP Status:**

200 - OK

**1.4.21.4 requestEmailReceipt**

Request an email receipt for an order.

**Complete name:**

TravelCardClassicTicketOrderService.requestEmailReceipt

**URL:**

/{apiVersion}/ptas/{ptaId}/classicTravelCardOrders/{orderId}

**HTTP Method:**

PUT

**Consumes:**

application/vnd.fara.order.request.email.receipt+json;ver=1

**Produces:**

application/json

**1.4.21.4.1 Success:****HTTP Status:**

200 - OK

**1.4.22 TravelCardClassicTicketPickUpService**

Service for handling pickup (card write commands) of travel card orders.

**1.4.22.1 pickUpTravelCardClassicTicket**

Pick up of a MIFARE Classic travel card ticket.

**Complete name:**

TravelCardClassicTicketPickUpService.pickUpTravelCardClassicTicket

**URL:**

/{apiVersion}/classicTravelCardTickets/{travelDocumentId}

**HTTP Method:**

POST

**Consumes:**

application/json

{

```

"cardData" : {
  "sectors" : [ {
    "sectorNo" : 0,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "NWk7FnGIBABDJUf/AAcDBQ=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAADLRSYAVInAQACAABs1w=="
    }, {
      "blockNo" : 2,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIQbu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 1,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "BgAABQAACgAACHAAAAApg=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAcW=="
    }, {
      "blockNo" : 2,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAcW=="
    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 2,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "oEIEAABAAAAAAAAAAAAQAq1A=="
    }, {
      "blockNo" : 1,
      "bytes" : "oEIEAABAAAAAAAAAAAAQAq1A=="
    }, {
      "blockNo" : 2,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 3,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "hAAAAAAAAAAAAAAAAAAAAAKw=="
    }, {
      "blockNo" : 1,
      "bytes" : "hAAAAAAAAAAAAAAAAAAAAAKw=="
    }, {
      "blockNo" : 2,
      "bytes" : "hAAAAAAAAAAAAAAAAAAAAAKw=="
    }, {
      "blockNo" : 3,

```

```

    "bytes" : "qqqqqqqqqHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 4,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "hAAAAAAAAAAAAAAAAAAAAKw=="
  }, {
    "blockNo" : 1,
    "bytes" : "hAAAAAAAAAAAAAAAAAAAAKw=="
  }, {
    "blockNo" : 2,
    "bytes" : "hAAAAAAAAAAAAAAAAAAAAKw=="
  }, {
    "blockNo" : 3,
    "bytes" : "qqqqqqqqqHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 5,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "AAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 1,
    "bytes" : "AAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 2,
    "bytes" : "AAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 3,
    "bytes" : "qqqqqqqqqHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 6,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "AAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 1,
    "bytes" : "AAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 2,
    "bytes" : "AAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 3,
    "bytes" : "qqqqqqqqqHeIALu7u7u7uw=="
  } ]
}, {
  "sectorNo" : 7,
  "blocks" : [ {
    "blockNo" : 0,
    "bytes" : "AAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 1,
    "bytes" : "AAAAAAAAAAAAAAAAAAAA=="
  }, {
    "blockNo" : 2,
    "bytes" : "AAAAAAAAAAAAAAAAAAAA=="
  }, {

```

```

        "blockNo" : 3,
        "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
}, {
    "sectorNo" : 8,
    "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 1,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 3,
        "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
}, {
    "sectorNo" : 9,
    "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 1,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 3,
        "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
}, {
    "sectorNo" : 10,
    "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 1,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 3,
        "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
}, {
    "sectorNo" : 11,
    "blocks" : [ {
        "blockNo" : 0,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 1,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
        "blockNo" : 2,
        "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    } ]
} ]

```

```

    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 12,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 2,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 13,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 2,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 14,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 2,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 3,
      "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
    } ]
  }, {
    "sectorNo" : 15,
    "blocks" : [ {
      "blockNo" : 0,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 1,
      "bytes" : "AAAAAAAAAAAAAAAAAAAAAAAAA=="
    }, {
      "blockNo" : 2,

```

```

        "bytes" : "AAAAAAAAAAAAAAAAAAAAAA=="
      }, {
        "blockNo" : 3,
        "bytes" : "qqqqqqqqqeHeIALu7u7u7uw=="
      } ]
    } ]
  },
  "cardId" : "372992309"
}

```

**Produces:**

application/json

```

{
  "links" : [ {
    "rel" : "self",
    "href" :
    "https://mobil.api.fara.no/api/v1.0/classicTravelCardTickets/PTA_UHFPWE_1367332057509"
  } ],
  "salesOrderId" : "PTA_UHFPWE",
  "currency" : "SEK",
  "priceAmount" : 1120.0,
  "vatPercentage" : 6.0,
  "vatAmount" : 67.2,
  "paymentReference" : "830840578197969209@12345",
  "serverTimestamp" : 1391758649824,
  "usageValidFromTime" : null,
  "usageValidUntilTime" : null,
  "products" : [ {
    "productClass" : "14",
    "productGroup" : "30 dagars resperiod län",
    "name" : "Vuxen",
    "description" : "30 dagars resperiod län - Vuxen",
    "priceCurrency" : "SEK",
    "priceAmount" : 1120.0,
    "priceVatPercentage" : 6.0,
    "priceVatAmount" : 67.2,
    "usageValidityDurationMinutes" : null,
    "productId" : "91;64;79=30&94=1&95=9;95=9"
  } ],
  "cardId" : "372992309",
  "cardCommands" : {
    "id" : "372992309",
    "commandList" : [ {
      "sequence" : 1,
      "cmd" : "write_key",
      "addrHex" : "1F",
      "authKey" : "A",
      "aidForAuthKey" : 0,
      "aidForKeysToWrite" : 180,
      "dataHex" : "AAAAAAAAAAAA78778800BBBBBBBBBBBB"
    }, {
      "sequence" : 2,
      "cmd" : "write_key",
      "addrHex" : "23",
      "authKey" : "A",
      "aidForAuthKey" : 0,
    }
  ]
}

```

```

    "aidForKeysToWrite" : 180,
    "dataHex" : "AAAAAAAAAAAA78778800BBBBBBBBBBBB"
  }, {
    "sequence" : 3,
    "cmd" : "write",
    "addrHex" : "1C",
    "authKey" : "B",
    "aidForAuthKey" : 180,
    "aidForKeysToWrite" : null,
    "dataHex" : "878348222D30020080C0F00550002560"
  }, {
    "sequence" : 4,
    "cmd" : "write",
    "addrHex" : "1D",
    "authKey" : "B",
    "aidForAuthKey" : 180,
    "aidForKeysToWrite" : null,
    "dataHex" : "6D701210000000000000000000000000"
  }, {
    "sequence" : 5,
    "cmd" : "write",
    "addrHex" : "1E",
    "authKey" : "B",
    "aidForAuthKey" : 180,
    "aidForKeysToWrite" : null,
    "dataHex" : "8801600900000000000000001EE1170000"
  }, {
    "sequence" : 6,
    "cmd" : "write",
    "addrHex" : "20",
    "authKey" : "B",
    "aidForAuthKey" : 180,
    "aidForKeysToWrite" : null,
    "dataHex" : "8026000000000030090000000000E074"
  }, {
    "sequence" : 7,
    "cmd" : "write",
    "addrHex" : "0D",
    "authKey" : "B",
    "aidForAuthKey" : 1,
    "aidForKeysToWrite" : null,
    "dataHex" : "84C3D715222D00000000001007581B8D"
  }, {
    "sequence" : 8,
    "cmd" : "write",
    "addrHex" : "04",
    "authKey" : "B",
    "aidForAuthKey" : 1,
    "aidForKeysToWrite" : null,
    "dataHex" : "0600000500000A00000A1000B4C00883"
  }, {
    "sequence" : 9,
    "cmd" : "write",
    "addrHex" : "05",
    "authKey" : "B",
    "aidForAuthKey" : 1,
    "aidForKeysToWrite" : null,
    "dataHex" : "B41000B4C008B4100000000000000065"
  }, {

```

```

    "sequence" : 10,
    "cmd" : "write",
    "addrHex" : "06",
    "authKey" : "B",
    "aidForAuthKey" : 1,
    "aidForKeysToWrite" : null,
    "dataHex" : "00000000000000000000000000000000B"
  }, {
    "sequence" : 11,
    "cmd" : "write",
    "addrHex" : "08",
    "authKey" : "B",
    "aidForAuthKey" : 1,
    "aidForKeysToWrite" : null,
    "dataHex" : "A042041100C08000000000000000001001875"
  }, {
    "sequence" : 12,
    "cmd" : "write",
    "addrHex" : "09",
    "authKey" : "B",
    "aidForAuthKey" : 1,
    "aidForKeysToWrite" : null,
    "dataHex" : "A042041100C08000000000000000001001875"
  } ]
},
"travelDocumentId" : "PTA_UHFPWE_1367332057509"
}

```

**1.4.22.1.1 Success:**

**HTTP Status:**  
200 - OK

**1.4.22.1.2 Error situations:**

**HTTP Status:**  
400 - Bad Request

Invalid input.

**HTTP Status:**  
410 - Gone

No pending pickup.

**HTTP Status:**  
419 - Insufficient Space On Resource

Not enough free space to put product(s) on travel card.



## 1.4.23 ZoneService

### 1.4.23.1 listZones

**Complete name:**

ZoneService.listZones

**URL:**

`/apiVersion}/ptas/{ptaShortName}/zones`

**HTTP Method:**

GET

**Produces:**

application/json

```
[ {
  "links" : [ {
    "rel" : "self",
    "href" : "https://mobil.api.fara.no/api/v1.0/ptas/PTA/zones/1"
  } ],
  "name" : "City Centre",
  "zoneId" : 1
} ]
```

#### 1.4.23.1.1 Success:

**HTTP Status:**

200 - OK

## 1.5 Scenario documentation

### 1.5.1 Create a mobile profile from a mobile device

This scenario describes how to create a mobile profile. A mobile profile in this case is a protected server side storage that are connected to a given mobile phone number. A mobile profile is required to be able to register a credit card agreement to pay for tickets. The creation of a mobile profile is a multi step process:

- 1 Use the [MobileProfileService.createMobileProfile](#) service operation to initialize the creation of a mobile profile. If a 400 (Bad request) is returned, you need to check the error code provided. If the error code is PROFILE\_ALREADY\_EXIST, a mobile profile for the given mobile phone number already exists, and you can access the mobile profile directly if the username and password is known, or you can go to the next step to reset the password for the mobile profile. If the error code is MOBILE\_PROFILE\_DISABLED\_OR\_LOCKED, you will have to abort the process and tell the end user to contact support. For other return codes or error codes, abort the operation. On success (204 No Content), go to the next step.
- 2 Use the [MobileProfileService.rebindProfile](#) operation to initialize a rebind of the mobile profile. This will generate an SMS to the given mobile phone number with an activation code. When the mobile

application user has received this activation code, the activation code can be used in the next step.

- 3 Use the [MobileProfileService.claimProfile](#) operation to actually claim the mobile profile using the activation code and the password that is to be used for it. The username for the mobile profile is returned in the response.

When the mobile profile has been created using the above operations, the client will be able to access the mobile profile and its sub-resources, e.g. paymentAgreements. The [MobileProfileService.getMobileProfile](#) operation is protected, and you need to provide the username and password for the mobile profile as request headers, namely x-fara-username and x-fara-pwd respectively.

### 1.5.2 Create a payment agreement

To create a payment agreement you actually need to perform an actual purchase for security reasons. This purchase will be credited as soon as the agreement has been authorized. The creation of a payment agreement is thus a multi-step process. And as the [PaymentAgreementService](#) is a sub-resource of the [MobileProfileService](#), you have to provide the username and password of the mobile profile in the x-fara-username and x-fara-pwd request headers respectively.

- 1 Use the [PaymentAgreementService.createPaymentAgreement](#) operation to initialize the creation of a payment agreement. The response includes a URL to a web page that the user should be directed to provide his credit card details. The API integrator needs to capture the orderRef query parameter from the href attribute of the submit button in the payment form, and use it in the next step.
- 2 Use the [PaymentAgreementService.completePaymentAgreement](#) operation to complete the payment agreement creation using the orderRef from the last step. You should store the agreementName, because it must be used when using the payment agreement to pay for tickets later on.

### 1.5.3 List products with predefined geography

Listing only products with predefined geography is a matter of using the [ProductGroupService.listProductGroups/ProductService.listProducts](#) operation without the toZone and fromZone query parameters specified.

### 1.5.4 List products for a specific geography

Listing products for a specific geography is a matter of using the [ProductGroupService.listProductGroups/ProductService.listProducts](#) operation with the toZone and fromZone query parameters specified. Values for fromZone and toZone is obtained by using [ZoneService.listZones/StopService.listStops](#). It is up to the API integrator if zones or stops are used to specify the geography. The relation between them is that every stop belongs to one zone, and one zone contains many stops.

### 1.5.5 Create a single ticket order from a mobile device

An order is created in the context of a payment method and a product medium. As the payment method can affect the price of products, and products may have restrictions on what product medium they are available on, you will need to list the available products using the same values for these

parameters and you use when you create the order. The typical payment method for a mobile device is either a "Credit card mobile" (a payment agreement) or "SMS" (CPA, overcharged SMS). The product medium is typically a QR-code. An order for a single ticket order is created using the [SingleTicketOrderService.addSingleTicketOrder](#). On success, the operation will return a pickup url that can be used to download the actual ticket. However, this is not possible before the order is payed. So, depending on the payment method used, follow that scenario descriptions below.

After successfully paying the order, you should be able to pickup the actual ticket representation using the provided URL. Pickup for a QR based ticket may be customized by providing some optional parameters with the pickup request. These are:

- **fileFormat** - (png, jpg / jpeg), default is png
- **errorCorrectionLevel** - (l(ow), m(edium), q(uartile), h(igh)), default is l(ow)
- **widthAsPixelCount** - positive integer, default is 116
- **heightAsPixelCount** - positive integer, default is 116
- **doReturnVisualInspectionAnimation** - boolean (true|false), default is true

### 1.5.6 Payment using a payment agreement

Payment using an existing payment agreement is a two-step process. The payment is initialized using [PaymentAgreementPaymentService.createPaymentAgreementPayment](#). The credit card will be checked and the money will be reserved. The client is now allowed to download the ticket (pickup), and after successful pickup, the payment has to be finalized using [PaymentAgreementPaymentService.capturePaymentAgreementPayment](#)

### 1.5.7 Payment using CPA (Overcharged SMS)

A payment using CPA is initialized by the mobile device by sending a SMS to a specific number (check with your service provider) with a code word and the order id. After having sent the SMS, the device must poll the pickup URL regularly to check if the payment has been confirmed. The pickup service uses the following protocol to signal the current state of the payment:

- 1 402 (Payment Required) with error code PAYMENT\_PENDING: Payment is still in progress. Try again in a little while.
- 2 402 (Payment Required) with error code PAYMENT\_NOT\_INITIALIZED: Payment is not in progress yet. Make sure the SMS was sent correctly. Then try again in a little while.
- 3 410 (Gone): The ticket has already been picked up. Is this a fraud attempt?
- 4 403 (Forbidden): The payment was rejected. No need to try again!