



We enable docx-as-code

**Advanced Template version: 1.0**

[[doc-title]]

[[doc-subtitle]]

[[doc-type]]

[[doc-company]]

[[doc-company-url]]

Document Number: [[doc-id]]

Version: [[doc-version]]

Release Date: [[doc-release-date]]



We enable docx-as-code

## Document History

Author Name: [[doc-author-name]]  
Author Contact: [[doc-author-email]]

## Document Revision

Version Number	Release Date	Status	Change Description	Author
[[doc-version]]	[[doc-release-date]]	Draft	[[doc-change-description]]	[[doc-author-name]]

## Intellectual Property Rights

This document is exclusive property of [[doc-company]] and all rights are reserved.



## Table of Contents

<b>1 Introduction.....</b>	<b>8</b>
1.1 References.....	8
<b>2 puzzlesCloudMDs.....</b>	<b>9</b>
<b>3 Backticks.....</b>	<b>10</b>
3.1 Backticks MD file.....	10
3.1.1 Backticks in heading.....	10
3.2 End of file.....	10
<b>4 Big.....</b>	<b>11</b>
4.1 Edge Service.....	11
4.1.1 TODO.....	11
4.1.1.1 TODO version 1.0.....	11
4.1.1.2 TODO version 1.0-next.....	11
4.1.2 Test environment and credentials.....	11
4.1.3 Edge Service Registry.....	11
4.1.4 Edge Service Relay.....	12
4.1.5 Edge Service Client.....	12
4.1.6 ReST API.....	12
4.1.6.1 Login.....	12
4.1.6.1.1 Login request.....	12
4.1.6.1.2 Login response.....	13
4.1.6.1.3 Cookies.....	13
4.1.6.2 Logout.....	13
4.1.6.2.1 Logout request.....	13
4.1.6.2.2 Logout response.....	13
4.1.6.3 Voucher list.....	13
4.1.6.3.1 Voucher list request.....	14
4.1.6.3.2 Voucher list response.....	14
4.1.6.4 Voucher detail.....	16
4.1.6.4.1 Voucher detail request.....	17
4.1.6.4.2 Voucher detail response.....	17
4.1.6.5 Voucher order.....	17
4.1.6.5.1 Voucher order request.....	17
4.1.6.5.2 Voucher order response.....	19
4.1.6.6 Voucher order detail.....	19
4.1.6.6.1 Voucher order detail request.....	20
4.1.6.6.2 Voucher order detail response.....	20
4.1.6.7 Provider Text.....	20
4.1.6.7.1 Provider Text request.....	20
4.1.6.7.2 Provider Text response.....	21
4.1.6.8 Content.....	21
4.1.6.8.1 Content request.....	21
4.1.6.8.2 Content response.....	22
4.1.6.9 Verification.....	22
4.1.6.9.1 Verification request by phone number.....	23
4.1.6.9.2 Verification by phone number response.....	23
4.1.6.9.3 Verification request by sms.....	23
4.1.6.9.4 Verification by sms number response.....	23
4.1.6.10 Cookies.....	24



4.1.6.10.1 Cookies request.....	24
4.1.6.10.2 Cookies response.....	24
4.1.6.11 Documentation.....	24
4.1.6.11.1 Documentation request.....	25
4.1.6.11.2 Documentation response.....	25
4.1.6.12 i18 internationalization.....	25
4.1.6.12.1 i18 request.....	25
<b>4.2 Edge Service.....</b>	<b>25</b>
4.2.1 TODO.....	25
4.2.1.1 TODO version 1.0.....	25
4.2.1.2 TODO version 1.0-next.....	26
4.2.2 Test environment and credentials.....	26
4.2.3 Edge Service Registry.....	26
4.2.4 Edge Service Relay.....	26
4.2.5 Edge Service Client.....	26
4.2.6 ReST API.....	27
4.2.6.1 Login.....	27
4.2.6.1.1 Login request.....	27
4.2.6.1.2 Login response.....	27
4.2.6.1.3 Cookies.....	27
4.2.6.2 Logout.....	27
4.2.6.2.1 Logout request.....	27
4.2.6.2.2 Logout response.....	28
4.2.6.3 Voucher list.....	28
4.2.6.3.1 Voucher list request.....	29
4.2.6.3.2 Voucher list response.....	29
4.2.6.4 Voucher detail.....	31
4.2.6.4.1 Voucher detail request.....	32
4.2.6.4.2 Voucher detail response.....	32
4.2.6.5 Voucher order.....	32
4.2.6.5.1 Voucher order request.....	32
4.2.6.5.2 Voucher order response.....	34
4.2.6.6 Voucher order detail.....	34
4.2.6.6.1 Voucher order detail request.....	35
4.2.6.6.2 Voucher order detail response.....	35
4.2.6.7 Provider Text.....	35
4.2.6.7.1 Provider Text request.....	35
4.2.6.7.2 Provider Text response.....	36
4.2.6.8 Content.....	36
4.2.6.8.1 Content request.....	36
4.2.6.8.2 Content response.....	37
4.2.6.9 Verification.....	37
4.2.6.9.1 Verification request by phone number.....	38
4.2.6.9.2 Verification by phone number response.....	38
4.2.6.9.3 Verification request by sms.....	38
4.2.6.9.4 Verification by sms number response.....	38
4.2.6.10 Cookies.....	39
4.2.6.10.1 Cookies request.....	39
4.2.6.10.2 Cookies response.....	39
4.2.6.11 Documentation.....	39
4.2.6.11.1 Documentation request.....	40
4.2.6.11.2 Documentation response.....	40
4.2.6.12 i18 internationalization.....	40
4.2.6.12.1 i18 request.....	40



<b>4.3 Edge Service.....</b>	<b>40</b>
4.3.1 TODO.....	40
4.3.1.1 TODO version 1.0.....	40
4.3.1.2 TODO version 1.0-next.....	41
4.3.2 Test environment and credentials.....	41
4.3.3 Edge Service Registry.....	41
4.3.4 Edge Service Relay.....	41
4.3.5 Edge Service Client.....	41
4.3.6 ReST API.....	42
4.3.6.1 Login.....	42
4.3.6.1.1 Login request.....	42
4.3.6.1.2 Login response.....	42
4.3.6.1.3 Cookies.....	42
4.3.6.2 Logout.....	42
4.3.6.2.1 Logout request.....	42
4.3.6.2.2 Logout response.....	43
4.3.6.3 Voucher list.....	43
4.3.6.3.1 Voucher list request.....	44
4.3.6.3.2 Voucher list response.....	44
4.3.6.4 Voucher detail.....	46
4.3.6.4.1 Voucher detail request.....	47
4.3.6.4.2 Voucher detail response.....	47
4.3.6.5 Voucher order.....	47
4.3.6.5.1 Voucher order request.....	47
4.3.6.5.2 Voucher order response.....	49
4.3.6.6 Voucher order detail.....	49
4.3.6.6.1 Voucher order detail request.....	50
4.3.6.6.2 Voucher order detail response.....	50
4.3.6.7 Provider Text.....	50
4.3.6.7.1 Provider Text request.....	50
4.3.6.7.2 Provider Text response.....	51
4.3.6.8 Content.....	51
4.3.6.8.1 Content request.....	51
4.3.6.8.2 Content response.....	52
4.3.6.9 Verification.....	52
4.3.6.9.1 Verification request by phone number.....	53
4.3.6.9.2 Verification by phone number response.....	53
4.3.6.9.3 Verification request by sms.....	53
4.3.6.9.4 Verification by sms number response.....	53
4.3.6.10 Cookies.....	54
4.3.6.10.1 Cookies request.....	54
4.3.6.10.2 Cookies response.....	54
4.3.6.11 Documentation.....	54
4.3.6.11.1 Documentation request.....	55
4.3.6.11.2 Documentation response.....	55
4.3.6.12 i18n internationalization.....	55
4.3.6.12.1 i18n request.....	55
<b>4.4 Edge Service.....</b>	<b>55</b>
4.4.1 TODO.....	55
4.4.1.1 TODO version 1.0.....	55
4.4.1.2 TODO version 1.0-next.....	56
4.4.2 Test environment and credentials.....	56
4.4.3 Edge Service Registry.....	56
4.4.4 Edge Service Relay.....	56
4.4.5 Edge Service Client.....	56



4.4.6 ReST API.....	57
4.4.6.1 Login.....	57
4.4.6.1.1 Login request.....	57
4.4.6.1.2 Login response.....	57
4.4.6.1.3 Cookies.....	57
4.4.6.2 Logout.....	57
4.4.6.2.1 Logout request.....	57
4.4.6.2.2 Logout response.....	58
4.4.6.3 Voucher list.....	58
4.4.6.3.1 Voucher list request.....	59
4.4.6.3.2 Voucher list response.....	59
4.4.6.4 Voucher detail.....	61
4.4.6.4.1 Voucher detail request.....	62
4.4.6.4.2 Voucher detail response.....	62
4.4.6.5 Voucher order.....	62
4.4.6.5.1 Voucher order request.....	62
4.4.6.5.2 Voucher order response.....	64
4.4.6.6 Voucher order detail.....	64
4.4.6.6.1 Voucher order detail request.....	65
4.4.6.6.2 Voucher order detail response.....	65
4.4.6.7 Provider Text.....	65
4.4.6.7.1 Provider Text request.....	65
4.4.6.7.2 Provider Text response.....	66
4.4.6.8 Content.....	66
4.4.6.8.1 Content request.....	66
4.4.6.8.2 Content response.....	67
4.4.6.9 Verification.....	67
4.4.6.9.1 Verification request by phone number.....	68
4.4.6.9.2 Verification by phone number response.....	68
4.4.6.9.3 Verification request by sms.....	68
4.4.6.9.4 Verification by sms number response.....	68
4.4.6.10 Cookies.....	69
4.4.6.10.1 Cookies request.....	69
4.4.6.10.2 Cookies response.....	69
4.4.6.11 Documentation.....	69
4.4.6.11.1 Documentation request.....	70
4.4.6.11.2 Documentation response.....	70
4.4.6.12 i18n internationalization.....	70
4.4.6.12.1 i18n request.....	70
<b>4.5 Edge Service.....</b>	<b>70</b>
4.5.1 TODO.....	70
4.5.1.1 TODO version 1.0.....	70
4.5.1.2 TODO version 1.0-next.....	71
4.5.2 Test environment and credentials.....	71
4.5.3 Edge Service Registry.....	71
4.5.4 Edge Service Relay.....	71
4.5.5 Edge Service Client.....	71
4.5.6 ReST API.....	72
4.5.6.1 Login.....	72
4.5.6.1.1 Login request.....	72
4.5.6.1.2 Login response.....	72
4.5.6.1.3 Cookies.....	72
4.5.6.2 Logout.....	72
4.5.6.2.1 Logout request.....	72
4.5.6.2.2 Logout response.....	73



4.5.6.3 Voucher list.....	73
4.5.6.3.1 Voucher list request.....	74
4.5.6.3.2 Voucher list response.....	74
4.5.6.4 Voucher detail.....	76
4.5.6.4.1 Voucher detail request.....	77
4.5.6.4.2 Voucher detail response.....	77
4.5.6.5 Voucher order.....	77
4.5.6.5.1 Voucher order request.....	77
4.5.6.5.2 Voucher order response.....	79
4.5.6.6 Voucher order detail.....	79
4.5.6.6.1 Voucher order detail request.....	80
4.5.6.6.2 Voucher order detail response.....	80
4.5.6.7 Provider Text.....	80
4.5.6.7.1 Provider Text request.....	80
4.5.6.7.2 Provider Text response.....	81
4.5.6.8 Content.....	81
4.5.6.8.1 Content request.....	81
4.5.6.8.2 Content response.....	82
4.5.6.9 Verification.....	82
4.5.6.9.1 Verification request by phone number.....	83
4.5.6.9.2 Verification by phone number response.....	83
4.5.6.9.3 Verification request by sms.....	83
4.5.6.9.4 Verification by sms number response.....	83
4.5.6.10 Cookies.....	84
4.5.6.10.1 Cookies request.....	84
4.5.6.10.2 Cookies response.....	84
4.5.6.11 Documentation.....	84
4.5.6.11.1 Documentation request.....	85
4.5.6.11.2 Documentation response.....	85
4.5.6.12 i18n internationalization.....	85
4.5.6.12.1 i18n request.....	85

### ***Table of Tables***

Table 1: Document references.....	8
-----------------------------------	---

### ***Table of Figures***

Figure 1: Last Page.....	86
--------------------------	----



## 1 Introduction

### 1.1 References

Referenced documents in creation of this document are listed in the Table 1.

Document Reference	Document Title	Version	Filename
[1]	[[ reference-doc-title ]]		[[ reference-doc-filename ]]
[2]			
[3]			
[4]			
[5]			
[6]			
[7]			
[8]			

Table 1: Document references





We enable docx-as-code

## 2 puzzlesCloudMDs

pc md 1 2





## 4 Big

### 4.1 Edge Service

123456 The Edge Service provides the interface between the public REST API and internal AMQP API.

#### 4.1.1 TODO

##### 4.1.1.1 TODO version 1.0

- ✓ [x] Test environment and credentials
- ✓ [x] specify voucher content fields
- ✓ [x] specify voucher pricing fields
- ✓ [x] specify payment method fields

##### 4.1.1.2 TODO version 1.0-next

- ✓ [ ] refactor entities and apply relationships
- ✓ [ ] provide full resource schemas
- ✓ [ ] enforce validation of resource schemas

#### 4.1.2 Test environment and credentials

TODO

Username	Password	Description
<a href="#">test@test.org</a>	test	just protected
<a href="#">test@gmail.com</a>	test	just testing
testtest	test	qa

#### 4.1.3 Edge Service Registry

The Edge Service Registry is responsible to maintain an index of all currently available Edge Services and configure all its known [Load-Balancers] accordingly.

*As of writing no hosting solution has been chosen yet. Therefor the actual API which is used to configure the load-balancers remains unspecified for now.*

---

All Edge Service Instances **must** register themselves on the registry on startup and un-register on shutdown.



#### 4.1.4 Edge Service Relay

Every Edge Service will be provided with a local relay connecting its instances to the central Message Broker cluster. The message exchange routing of the relay is configured in such a way, that only messages relevant to its connected Edge Services is routed to it from the broker cluster.

[See \[Message Relay\] for more information.](#)

---

#### 4.1.5 Edge Service Client

Each Service that is to be served by the Edge Service is represented by a so-called Edge Service Client. It is responsible to provide the implementation of the public REST API for that given service.

#### 4.1.6 ReST API

The Edge Service API acts as translation layer between the Services and the Frontend. The actual logic is implemented in separate [Edge Service Client]s for each individual Service consumed by the API or API endpoint provided by the API, respectively.

All public-facing endpoints **must** adhere to the [JSONAPI] specification and be subject to strict [Access Control].

##### 4.1.6.1 Login

###### 4.1.6.1.1 Login request

Variables	Description	Example
ICP_SHOP_ID	see [Test environment and credentials]	
ICP_SHOP_PASSPHRASE	see [Test environment and credentials]	
ICP_SHOP_DIGEST	Derived from \${ICP_SHOP_ID} and \${ICP_SHOP_PASSPHRASE}	MzAwMDAwMTY6dG9wLXNIY3JI dC1wYXNzcGhyYXNI



```
POST /api/rest/v1/auth/login HTTP/1.1
Authorization: Basic ${ICP_SHOP_DIGEST}
Accept: application/vnd.api+json
```

#### 4.1.6.1.2 Login response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json
```

#### 4.1.6.1.3 Cookies

Name	Value	Domain	Path	Expires	HttpOnly	Secure
SESSION	MmQ5ZjkxO GltMTM4My0 0M2E0LWI1M DYtYmUwZjY wMzJiZDY5	master.env.a pi.shop1.roc ks	/	Seassion	true	false

#### 4.1.6.2 Logout

##### 4.1.6.2.1 Logout request

Variables	Description	Example
-----------	-------------	---------

```
POST /api/rest/v1/auth/logout HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

##### 4.1.6.2.2 Logout response

```
HTTP/1.1 204 No Content
```

#### 4.1.6.3 Voucher list

See [VoucherWebList] method in the [Mimas integration] documentation for the source of these fields.



#### *4.1.6.3.1 Voucher list request*

```
GET /api/rest/v1/vouchers HTTP/1.1  
Authorization: No Auth  
Accept: application/vnd.api+json
```

#### *4.1.6.3.2 Voucher list response*

```
HTTP/1.1 200 OK  
Content-Type: application/vnd.api+json
```



```
{
  "data": [
    {
      "id": "f634f09f-9a06-4297-a326-0d703938b702",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL10",
        "currency": "EUR",
        "ean-code": "8717472242211",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 10 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 10",
        "value": 10.0,
        "price-total": 11.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": {
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
            "name": "PayPal",
            "provider-name": "Wirecard"
          }
        }
      }
    },
    {
      "id": "a3473e20-19b4-44fe-8389-67a511854759",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL15",
        "currency": "EUR",
        "ean-code": "8717472242129",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 15 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 15",
        "value": 15.0,
        "price-total": 16.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": [
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
```



We enable docx-as-code

```
    "name": "PayPal",  
    "provider-name": "Wirecard"  
  }  
]  
}  
}
```

#### **4.1.6.4 Voucher detail**





#### 4.1.6.4.1 Voucher detail request

```
GET /api/rest/v1/vouchers/:voucher-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.1.6.4.2 Voucher detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "f634f09f-9a06-4297-a326-0d703938b702",
    "type": "voucher",
    "attributes": {
      "code": "LEBALLINNL10",
      "currency": "EUR",
      "ean-code": "8717472242211",
      "encryption-type": 0,
      "kind": 0,
      "name": "Lebara All-in NL 10 EUR",
      "provider-code": "LEBARA",
      "provider-name": "Lebara",
      "short-name": "All-in NL 10",
      "value": 10.0,
      "price-total": 11.99,
      "price-tax": 0.00,
      "price-fee": 1.99,
      "payment-methods": {
        "wirecard-creditcard": {
          "name": "Credit Card",
          "provider-name": "Wirecard",
          "image-url": "https://wirecard.com/logo.png"
        },
        "wirecard-paypal": {
          "name": "PayPal",
          "provider-name": "Wirecard"
        }
      }
    }
  }
}
```

#### 4.1.6.5 Voucher order

See *[VoucherWebReserve]* method in the *[Mimas integration]* documentation for the source of these fields.

##### 4.1.6.5.1 Voucher order request

Variables	Description	Example		



```
POST /api/rest/v1/voucher-orders HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "voucher-code": "VODADE15"
      "payment-method": "Wirecard VISA",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "provider-text-url": "https://provider.texts",

      "order-success-redirect-url": "/success",
      "order-failed-redirect-url": "failed",
      "order-canceled-redirect-url": "/cancel"
    }
  }
}
```

field name	type	constrains	description	example
id	UUID	globally unique	globally unique UUID	c91feaf3-5968-4b60-aa5b-62fe7fdd905c
type	string	n/a	type of request, must be "voucher-order", "voucher-order"	
voucher-code	string	n/a	code of the voucher gathered from voucher list	"VODADE15"
payment-method	string	n/a	on of the allowed methods for the voucher, denoted in voucher list response	"Wirecard VISA"
notification-type	string	currently only "email" is supported	type of the notification target	"email"
notification-target	string	n/a	target adress	"[me@my-email.com] (mailto:me@my-email.com)"



field name	type	constraints	description	example
order-success-redirect-url	string	valid URL	Where the wirecard should redirect if payment was successfull	protect
order-failed-redirect-url	string	valid URL	Where the wirecard should redirect if payment fails	protect
order-canceled-redirect-url	string	valid URL	Where the wirecard should redirect if payment was canceled	protect

#### 4.1.6.5.2 Voucher order response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgF26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINNL10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 0.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "0",
      "batch-number": "1234"
    }
  }
}
```

#### 4.1.6.6 Voucher order detail



#### 4.1.6.6.1 Voucher order detail request

```
GET /api/rest/v1/voucher-orders/:order-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.1.6.6.2 Voucher order detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgf26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINN10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 1.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "1611313953175"
    }
  }
}
```

#### 4.1.6.7 Provider Text

See `[GetProviderTexts]` method in the *[Mimas integration]* documentation for the source of these fields.

##### 4.1.6.7.1 Provider Text request

Variables	Description	Example
-----------	-------------	---------



```
GET /api/rest/v1/provider-texts/ HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

GET /api/rest/v1/provider-texts/:provider-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.1.6.7.2 Provider Text response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": [
    {
      "id": "f74bb480-d6df-47ad-8a17-8d7b1b176661",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "title",
        "title": "Produktname",
        "body": "About You-Geschenkkarte"
      }
    },
    {
      "id": "a598bf3b-abe0-4fc4-80fa-135a84510c81",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "subtitle",
        "title": "Slogan",
        "body": "Hol Dir per E-Mail eine About You-Geschenkkarte."
      }
    }
  ]
}
```

#### 4.1.6.8 Content

See [Content Service] for more information on the source of these fields.

##### 4.1.6.8.1 Content request

Variables	Description	Example
-----------	-------------	---------



```
GET /api/rest/v1/content/i18next.json HTTP/1.1
Authorization: No Auth
Accept: application/json
```

#### 4.1.6.8.2 Content response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "en": {
    "world.hello": "Hello, World!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Cancel",
    "radio.options": {
      "option-1": "Selection 1",
      "option-2": "Selection 2",
    }
  },
  "de": {
    "world.hello": "Hallo, Welt!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Abbrechen"
    "radio.options": {
      "option-1": "Auswahl 1",
      "option-2": "Auswahl 2",
    }
  }
}
```

#### 4.1.6.9 Verification

*See for more information on the source of these fields.*



#### 4.1.6.9.1 Verification request by phone number

```
POST /api/rest/v1/auth/phone-number-verification HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "38f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "sms-confirmation-code": "716569"
    }
  }
}
```

#### 4.1.6.9.2 Verification by phone number response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "data": {
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": false,
      "remaining-number-of-attempts": 4,
      "remaining-expiration-time": 0
    }
  }
}
```

#### 4.1.6.9.3 Verification request by sms

```
POST /api/rest/v1/auth/sms-verification-code HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "48f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655"
    }
  }
}
```

#### 4.1.6.9.4 Verification by sms number response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```



```
{
  "data": {
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": true,
      "remaining-number-of-attempts": 2,
      "remaining-delay-time": 20,
      "remaining-expiration-time": 1800
    }
  }
}
```

#### 4.1.6.10 Cookies

##### 4.1.6.10.1 Cookies request

POST /api/rest/v1/cookies HTTP/1.1  
Authorization: No Auth  
Accept: application/vnd.api+json

```
{
  "data": [
    {
      "type": "cookie",
      "id": "shopId",
      "attributes": {
        "name": "shopId",
        "value": "1234"
      }
    },
    {
      "type": "cookie",
      "id": "ns_dev_id",
      "attributes": {
        "name": "ns_dev_id",
        "value": "device_id"
      }
    }
  ]
}
```

##### 4.1.6.10.2 Cookies response

HTTP/1.1 200 OK

#### 4.1.6.11 Documentation

Returns list of Json Schemas for request and response for requested rest endpoint. Use exact endpoint plus schema url parameter. For example, /api/rest/v1/vouchers?schema will return json schemas for /api/rest/v1/vouchers endpoint.





#### 4.1.6.11.1 Documentation request

```
GET /api/rest/v1/vouchers?schema HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.1.6.11.2 Documentation response

Complete Response is omitted because of size , here is just a part :

```
HTTP/1.1 200 OK
```

```
{
  "required": [
    "createdDate",
    "notificationStatus",
    "notificationTarget",
    "notificationType",
    "orderFeeTotal",
    "orderPriceTotal",
    "orderStatus",
    "orderTaxTotal",
    "paymentMethod",
    "paymentProvider",
    "paymentStatus",
    "shortId",
    "voucherCode",
    "voucherCurrency",
    "voucherStatus",
    "voucherValue"
  ]
}
```

#### 4.1.6.12 i18 internationalization

- ✓ Retrieves i18next json based on requested language

##### 4.1.6.12.1 i18 request

```
GET api/rest/v1/content/i18next.json/de HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

## 4.2 Edge Service

The Edge Service provides the interface between the public REST API and internal AMQP API.

### 4.2.1 TODO

#### 4.2.1.1 TODO version 1.0

- ✓ [x] Test environment and credentials



- ✓ [x] specify voucher content fields
- ✓ [x] specify voucher pricing fields
- ✓ [x] specify payment method fields

#### 4.2.1.2 TODO version 1.0-next

- ✓ [ ] refactor entities and apply relationships
- ✓ [ ] provide full resource schemas
- ✓ [ ] enforce validation of resource schemas

#### 4.2.2 Test environment and credentials

TODO

Username	Password	Description
<a href="#">test@test.org</a>	test	just protected
<a href="#">test@gmail.com</a>	test	just testing
testtest	test	qa

#### 4.2.3 Edge Service Registry

The Edge Service Registry is responsible to maintain an index of all currently available Edge Services and configure all its known [Load-Balancers] accordingly.

*As of writing no hosting solution has been chosen yet. Therefor the actual API which is used to configure the load-balancers remains unspecified for now.*

---

All Edge Service Instances **must** register themselves on the registry on startup and un-register on shutdown.

#### 4.2.4 Edge Service Relay

Every Edge Service will be provided with a local relay connecting its instances to the central Message Broker cluster. The message exchange routing of the relay is configured in such a way, that only messages relevant to its connected Edge Services is routed to it from the broker cluster.

*See [Message Relay] for more information.*

---

#### 4.2.5 Edge Service Client

Each Service that is to be served by the Edge Service is represented by a so-called Edge Service Client. It is responsible to provide the implementation of the public REST API for that given service.



#### 4.2.6 ReST API

The Edge Service API acts as translation layer between the Services and the Frontend. The actual logic is implemented in separate [Edge Service Client]s for each individual Service consumed by the API or API endpoint provided by the API, respectively.

All public-facing endpoints **must** adhere to the [JSONAPI] specification and be subject to strict [Access Control].

##### 4.2.6.1 Login

###### 4.2.6.1.1 Login request

Variables	Description	Example
ICP_SHOP_ID	see [Test environment and credentials]	
ICP_SHOP_PASSPHRASE	see [Test environment and credentials]	
ICP_SHOP_DIGEST	Derived from \${ICP_SHOP_ID} and \${ICP_SHOP_PASSPHRASE}	MzAwMDAwMTY6dG9wLXNIY3JI dC1wYXNzcGhyYXNI

```
POST /api/rest/v1/auth/login HTTP/1.1
Authorization: Basic ${ICP_SHOP_DIGEST}
Accept: application/vnd.api+json
```

###### 4.2.6.1.2 Login response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json
```

###### 4.2.6.1.3 Cookies

Name	Value	Domain	Path	Expires	HttpOnly	Secure
SESSION	MmQ5ZjkxO GltMTM4My0 0M2E0LWI1M DYtYmUwZjY wMzJiZDY5	master.env.a pi.shop1.roc ks	/	Seassion	true	false

##### 4.2.6.2 Logout

###### 4.2.6.2.1 Logout request

Variables	Description	Example
-----------	-------------	---------



```
POST /api/rest/v1/auth/logout HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### *4.2.6.2.2 Logout response*

```
HTTP/1.1 204 No Content
```

#### **4.2.6.3 Voucher list**

See *[VoucherWebList]* method in the *[Mimas integration]* documentation for the source of these fields.



#### *4.2.6.3.1 Voucher list request*

```
GET /api/rest/v1/vouchers HTTP/1.1  
Authorization: No Auth  
Accept: application/vnd.api+json
```

#### *4.2.6.3.2 Voucher list response*

```
HTTP/1.1 200 OK  
Content-Type: application/vnd.api+json
```



```
{
  "data": [
    {
      "id": "f634f09f-9a06-4297-a326-0d703938b702",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL10",
        "currency": "EUR",
        "ean-code": "8717472242211",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 10 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 10",
        "value": 10.0,
        "price-total": 11.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": {
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
            "name": "PayPal",
            "provider-name": "Wirecard"
          }
        }
      }
    },
    {
      "id": "a3473e20-19b4-44fe-8389-67a511854759",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL15",
        "currency": "EUR",
        "ean-code": "8717472242129",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 15 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 15",
        "value": 15.0,
        "price-total": 16.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": [
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
```



We enable docx-as-code

```
        "name": "PayPal",  
        "provider-name": "Wirecard"  
      }  
    ]  
  }  
}
```

#### **4.2.6.4 Voucher detail**



#### 4.2.6.4.1 Voucher detail request

```
GET /api/rest/v1/vouchers/:voucher-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.2.6.4.2 Voucher detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "f634f09f-9a06-4297-a326-0d703938b702",
    "type": "voucher",
    "attributes": {
      "code": "LEBALLINNL10",
      "currency": "EUR",
      "ean-code": "8717472242211",
      "encryption-type": 0,
      "kind": 0,
      "name": "Lebara All-in NL 10 EUR",
      "provider-code": "LEBARA",
      "provider-name": "Lebara",
      "short-name": "All-in NL 10",
      "value": 10.0,
      "price-total": 11.99,
      "price-tax": 0.00,
      "price-fee": 1.99,
      "payment-methods": {
        "wirecard-creditcard": {
          "name": "Credit Card",
          "provider-name": "Wirecard",
          "image-url": "https://wirecard.com/logo.png"
        },
        "wirecard-paypal": {
          "name": "PayPal",
          "provider-name": "Wirecard"
        }
      }
    }
  }
}
```

#### 4.2.6.5 Voucher order

See *[VoucherWebReserve]* method in the *[Mimas integration]* documentation for the source of these fields.

##### 4.2.6.5.1 Voucher order request

Variables	Description	Example		





```
POST /api/rest/v1/voucher-orders HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "voucher-code": "VODADE15"
      "payment-method": "Wirecard VISA",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "provider-text-url": "https://provider.texts",

      "order-success-redirect-url": "/success",
      "order-failed-redirect-url": "failed",
      "order-canceled-redirect-url": "/cancel"
    }
  }
}
```

field name	type	constrains	description	example
id	UUID	globally unique	globally unique UUID	c91feaf3-5968-4b60-aa5b-62fe7fdd905c
type	string	n/a	type of request, must be "voucher-order", "voucher-order"	
voucher-code	string	n/a	code of the voucher gathered from voucher list	"VODADE15"
payment-method	string	n/a	on of the allowed methods for the voucher, denoted in voucher list response	"Wirecard VISA"
notification-type	string	currently only "email" is supported	type of the notification target	"email"
notification-target	string	n/a	target adress	"[me@my-email.com] (mailto:me@my-email.com)"



field name	type	constraints	description	example
order-success-redirect-url	string	valid URL	Where the wirecard should redirect if payment was successfull	protect
order-failed-redirect-url	string	valid URL	Where the wirecard should redirect if payment fails	protect
order-canceled-redirect-url	string	valid URL	Where the wirecard should redirect if payment was canceled	protect

#### 4.2.6.5.2 Voucher order response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgF26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINNL10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 0.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "0",
      "batch-number": "1234"
    }
  }
}
```

#### 4.2.6.6 Voucher order detail



#### 4.2.6.6.1 Voucher order detail request

```
GET /api/rest/v1/voucher-orders/:order-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.2.6.6.2 Voucher order detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgf26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINN10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 1.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "1611313953175"
    }
  }
}
```

#### 4.2.6.7 Provider Text

See `[GetProviderTexts]` method in the *[Mimas integration]* documentation for the source of these fields.

##### 4.2.6.7.1 Provider Text request

Variables	Description	Example
-----------	-------------	---------



```
GET /api/rest/v1/provider-texts/ HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

GET /api/rest/v1/provider-texts/:provider-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.2.6.7.2 Provider Text response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": [
    {
      "id": "f74bb480-d6df-47ad-8a17-8d7b1b176661",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "title",
        "title": "Produktname",
        "body": "About You-Geschenkkarte"
      }
    },
    {
      "id": "a598bf3b-abe0-4fc4-80fa-135a84510c81",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "subtitle",
        "title": "Slogan",
        "body": "Hol Dir per E-Mail eine About You-Geschenkkarte."
      }
    }
  ]
}
```

#### 4.2.6.8 Content

See [Content Service] for more information on the source of these fields.

##### 4.2.6.8.1 Content request

Variables	Description	Example
-----------	-------------	---------



```
GET /api/rest/v1/content/i18next.json HTTP/1.1
Authorization: No Auth
Accept: application/json
```

#### 4.2.6.8.2 Content response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "en": {
    "world.hello": "Hello, World!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Cancel",
    "radio.options": {
      "option-1": "Selection 1",
      "option-2": "Selection 2",
    }
  },
  "de": {
    "world.hello": "Hallo, Welt!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Abbrechen"
    "radio.options": {
      "option-1": "Auswahl 1",
      "option-2": "Auswahl 2",
    }
  }
}
```

#### 4.2.6.9 Verification

*See for more information on the source of these fields.*



#### 4.2.6.9.1 Verification request by phone number

```
POST /api/rest/v1//auth/phone-number-verification HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "38f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "sms-confirmation-code": "716569"
    }
  }
}
```

#### 4.2.6.9.2 Verification by phone number response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "data": {
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": false,
      "remaining-number-of-attempts": 4,
      "remaining-expiration-time": 0
    }
  }
}
```

#### 4.2.6.9.3 Verification request by sms

```
POST /api/rest/v1/auth/sms-verification-code HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "48f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655"
    }
  }
}
```

#### 4.2.6.9.4 Verification by sms number response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```



```
{
  "data": {
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": true,
      "remaining-number-of-attempts": 2,
      "remaining-delay-time": 20,
      "remaining-expiration-time": 1800
    }
  }
}
```

#### 4.2.6.10 Cookies

##### 4.2.6.10.1 Cookies request

POST /api/rest/v1/cookies HTTP/1.1  
Authorization: No Auth  
Accept: application/vnd.api+json

```
{
  "data": [
    {
      "type": "cookie",
      "id": "shopId",
      "attributes": {
        "name": "shopId",
        "value": "1234"
      }
    },
    {
      "type": "cookie",
      "id": "ns_dev_id",
      "attributes": {
        "name": "ns_dev_id",
        "value": "device_id"
      }
    }
  ]
}
```

##### 4.2.6.10.2 Cookies response

HTTP/1.1 200 OK

#### 4.2.6.11 Documentation

Returns list of Json Schemas for request and response for requested rest endpoint. Use exact endpoint plus schema url parameter. For example, /api/rest/v1/vouchers?schema will return json schemas for /api/rest/v1/vouchers endpoint.



#### 4.2.6.11.1 Documentation request

```
GET /api/rest/v1/vouchers?schema HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.2.6.11.2 Documentation response

Complete Response is omitted because of size , here is just a part :

```
HTTP/1.1 200 OK
```

```
{
  "required": [
    "createdDate",
    "notificationStatus",
    "notificationTarget",
    "notificationType",
    "orderFeeTotal",
    "orderPriceTotal",
    "orderStatus",
    "orderTaxTotal",
    "paymentMethod",
    "paymentProvider",
    "paymentStatus",
    "shortId",
    "voucherCode",
    "voucherCurrency",
    "voucherStatus",
    "voucherValue"
  ]
}
```

#### 4.2.6.12 i18 internationalization

- ✓ Retrieves i18next json based on requested language

##### 4.2.6.12.1 i18 request

```
GET api/rest/v1/content/i18next.json/de HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

### 4.3 Edge Service

The Edge Service provides the interface between the public REST API and internal AMQP API.

#### 4.3.1 TODO

##### 4.3.1.1 TODO version 1.0

- ✓ [x] Test environment and credentials





- ✓ [x] specify voucher content fields
- ✓ [x] specify voucher pricing fields
- ✓ [x] specify payment method fields

#### 4.3.1.2 TODO version 1.0-next

- ✓ [ ] refactor entities and apply relationships
- ✓ [ ] provide full resource schemas
- ✓ [ ] enforce validation of resource schemas

#### 4.3.2 Test environment and credentials

TODO

Username	Password	Description
<a href="#">test@test.org</a>	test	just protected
<a href="#">test@gmail.com</a>	test	just testing
testtest	test	qa

#### 4.3.3 Edge Service Registry

The Edge Service Registry is responsible to maintain an index of all currently available Edge Services and configure all its known [Load-Balancers] accordingly.

*As of writing no hosting solution has been chosen yet. Therefor the actual API which is used to configure the load-balancers remains unspecified for now.*

---

All Edge Service Instances **must** register themselves on the registry on startup and un-register on shutdown.

#### 4.3.4 Edge Service Relay

Every Edge Service will be provided with a local relay connecting its instances to the central Message Broker cluster. The message exchange routing of the relay is configured in such a way, that only messages relevant to its connected Edge Services is routed to it from the broker cluster.

*See [Message Relay] for more information.*

---

#### 4.3.5 Edge Service Client

Each Service that is to be served by the Edge Service is represented by a so-called Edge Service Client. It is responsible to provide the implementation of the public REST API for that given service.



#### 4.3.6 ReST API

The Edge Service API acts as translation layer between the Services and the Frontend. The actual logic is implemented in separate [Edge Service Client]s for each individual Service consumed by the API or API endpoint provided by the API, respectively.

All public-facing endpoints **must** adhere to the [JSONAPI] specification and be subject to strict [Access Control].

##### 4.3.6.1 Login

###### 4.3.6.1.1 Login request

Variables	Description	Example
ICP_SHOP_ID	see [Test environment and credentials]	
ICP_SHOP_PASSPHRASE	see [Test environment and credentials]	
ICP_SHOP_DIGEST	Derived from \${ICP_SHOP_ID} and \${ICP_SHOP_PASSPHRASE}	MzAwMDAwMTY6dG9wLXNIY3JI dC1wYXNzcGhyYXNI

```
POST /api/rest/v1/auth/login HTTP/1.1
Authorization: Basic ${ICP_SHOP_DIGEST}
Accept: application/vnd.api+json
```

###### 4.3.6.1.2 Login response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json
```

###### 4.3.6.1.3 Cookies

Name	Value	Domain	Path	Expires	HttpOnly	Secure
SESSION	MmQ5ZjkxO GltMTM4My0 0M2E0LWI1M DYtYmUwZjY wMzJiZDY5	master.env.a pi.shop1.roc ks	/	Seassion	true	false

##### 4.3.6.2 Logout

###### 4.3.6.2.1 Logout request

Variables	Description	Example
-----------	-------------	---------



```
POST /api/rest/v1/auth/logout HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.3.6.2.2 Logout response

```
HTTP/1.1 204 No Content
```

#### 4.3.6.3 Voucher list

See *[VoucherWebList]* method in the *[Mimas integration]* documentation for the source of these fields.



#### 4.3.6.3.1 *Voucher list request*

```
GET /api/rest/v1/vouchers HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.3.6.3.2 *Voucher list response*

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```



```
{
  "data": [
    {
      "id": "f634f09f-9a06-4297-a326-0d703938b702",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL10",
        "currency": "EUR",
        "ean-code": "8717472242211",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 10 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 10",
        "value": 10.0,
        "price-total": 11.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": {
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
            "name": "PayPal",
            "provider-name": "Wirecard"
          }
        }
      }
    },
    {
      "id": "a3473e20-19b4-44fe-8389-67a511854759",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL15",
        "currency": "EUR",
        "ean-code": "8717472242129",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 15 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 15",
        "value": 15.0,
        "price-total": 16.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": [
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
```



We enable docx-as-code

```
        "name": "PayPal",  
        "provider-name": "Wirecard"  
      }  
    ]  
  }  
]  
}
```

#### **4.3.6.4 Voucher detail**



#### 4.3.6.4.1 Voucher detail request

```
GET /api/rest/v1/vouchers/:voucher-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.3.6.4.2 Voucher detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "f634f09f-9a06-4297-a326-0d703938b702",
    "type": "voucher",
    "attributes": {
      "code": "LEBALLINNL10",
      "currency": "EUR",
      "ean-code": "8717472242211",
      "encryption-type": 0,
      "kind": 0,
      "name": "Lebara All-in NL 10 EUR",
      "provider-code": "LEBARA",
      "provider-name": "Lebara",
      "short-name": "All-in NL 10",
      "value": 10.0,
      "price-total": 11.99,
      "price-tax": 0.00,
      "price-fee": 1.99,
      "payment-methods": {
        "wirecard-creditcard": {
          "name": "Credit Card",
          "provider-name": "Wirecard",
          "image-url": "https://wirecard.com/logo.png"
        },
        "wirecard-paypal": {
          "name": "PayPal",
          "provider-name": "Wirecard"
        }
      }
    }
  }
}
```

#### 4.3.6.5 Voucher order

See [VoucherWebReserve] method in the [Mimas integration] documentation for the source of these fields.

##### 4.3.6.5.1 Voucher order request

Variables	Description	Example		



```
POST /api/rest/v1/voucher-orders HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "voucher-code": "VODADE15",
      "payment-method": "Wirecard VISA",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "provider-text-url": "https://provider.texts",

      "order-success-redirect-url": "/success",
      "order-failed-redirect-url": "failed",
      "order-canceled-redirect-url": "/cancel"
    }
  }
}
```

field name	type	constrains	description	example
id	UUID	globally unique	globally unique UUID	c91feaf3-5968-4b60-aa5b-62fe7fdd905c
type	string	n/a	type of request, must be "voucher-order", "voucher-order"	
voucher-code	string	n/a	code of the voucher gathered from voucher list	"VODADE15"
payment-method	string	n/a	on of the allowed methods for the voucher, denoted in voucher list response	"Wirecard VISA"
notification-type	string	currently only "email" is supported	type of the notification target	"email"
notification-target	string	n/a	target adress	"[me@my-email.com] (mailto:me@my-email.com)"





field name	type	constraints	description	example
order-success-redirect-url	string	valid URL	Where the wirecard should redirect if payment was successfull	protect
order-failed-redirect-url	string	valid URL	Where the wirecard should redirect if payment fails	protect
order-canceled-redirect-url	string	valid URL	Where the wirecard should redirect if payment was canceled	protect

#### 4.3.6.5.2 Voucher order response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgF26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINNL10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 0.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "0",
      "batch-number": "1234"
    }
  }
}
```

#### 4.3.6.6 Voucher order detail



#### 4.3.6.6.1 Voucher order detail request

```
GET /api/rest/v1/voucher-orders/:order-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.3.6.6.2 Voucher order detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgf26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINN10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 1.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "1611313953175"
    }
  }
}
```

#### 4.3.6.7 Provider Text

See `[GetProviderTexts]` method in the *[Mimas integration]* documentation for the source of these fields.

##### 4.3.6.7.1 Provider Text request

Variables	Description	Example
-----------	-------------	---------



```
GET /api/rest/v1/provider-texts/ HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

GET /api/rest/v1/provider-texts/:provider-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.3.6.7.2 Provider Text response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": [
    {
      "id": "f74bb480-d6df-47ad-8a17-8d7b1b176661",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "title",
        "title": "Produktname",
        "body": "About You-Geschenkkarte"
      }
    },
    {
      "id": "a598bf3b-abe0-4fc4-80fa-135a84510c81",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "subtitle",
        "title": "Slogan",
        "body": "Hol Dir per E-Mail eine About You-Geschenkkarte."
      }
    }
  ]
}
```

#### 4.3.6.8 Content

See [Content Service] for more information on the source of these fields.

##### 4.3.6.8.1 Content request

Variables	Description	Example
-----------	-------------	---------



```
GET /api/rest/v1/content/i18next.json HTTP/1.1
Authorization: No Auth
Accept: application/json
```

#### 4.3.6.8.2 Content response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "en": {
    "world.hello": "Hello, World!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Cancel",
    "radio.options": {
      "option-1": "Selection 1",
      "option-2": "Selection 2",
    }
  },
  "de": {
    "world.hello": "Hallo, Welt!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Abbrechen"
    "radio.options": {
      "option-1": "Auswahl 1",
      "option-2": "Auswahl 2",
    }
  }
}
```

#### 4.3.6.9 Verification

*See for more information on the source of these fields.*



#### 4.3.6.9.1 Verification request by phone number

```
POST /api/rest/v1//auth/phone-number-verification HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "38f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "sms-confirmation-code": "716569"
    }
  }
}
```

#### 4.3.6.9.2 Verification by phone number response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "data": {
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": false,
      "remaining-number-of-attempts": 4,
      "remaining-expiration-time": 0
    }
  }
}
```

#### 4.3.6.9.3 Verification request by sms

```
POST /api/rest/v1/auth/sms-verification-code HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "48f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655"
    }
  }
}
```

#### 4.3.6.9.4 Verification by sms number response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```



```
{
  "data": {
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": true,
      "remaining-number-of-attempts": 2,
      "remaining-delay-time": 20,
      "remaining-expiration-time": 1800
    }
  }
}
```

#### 4.3.6.10 Cookies

##### 4.3.6.10.1 Cookies request

POST /api/rest/v1/cookies HTTP/1.1  
Authorization: No Auth  
Accept: application/vnd.api+json

```
{
  "data": [
    {
      "type": "cookie",
      "id": "shopId",
      "attributes": {
        "name": "shopId",
        "value": "1234"
      }
    },
    {
      "type": "cookie",
      "id": "ns_dev_id",
      "attributes": {
        "name": "ns_dev_id",
        "value": "device_id"
      }
    }
  ]
}
```

##### 4.3.6.10.2 Cookies response

HTTP/1.1 200 OK

#### 4.3.6.11 Documentation

Returns list of Json Schemas for request and response for requested rest endpoint. Use exact endpoint plus schema url parameter. For example, /api/rest/v1/vouchers?schema will return json schemas for /api/rest/v1/vouchers endpoint.



#### 4.3.6.11.1 Documentation request

```
GET /api/rest/v1/vouchers?schema HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.3.6.11.2 Documentation response

Complete Response is omitted because of size , here is just a part :

```
HTTP/1.1 200 OK
```

```
{
  "required": [
    "createdDate",
    "notificationStatus",
    "notificationTarget",
    "notificationType",
    "orderFeeTotal",
    "orderPriceTotal",
    "orderStatus",
    "orderTaxTotal",
    "paymentMethod",
    "paymentProvider",
    "paymentStatus",
    "shortId",
    "voucherCode",
    "voucherCurrency",
    "voucherStatus",
    "voucherValue"
  ]
}
```

#### 4.3.6.12 i18 internationalization

- ✓ Retrieves i18next json based on requested language

##### 4.3.6.12.1 i18 request

```
GET api/rest/v1/content/i18next.json/de HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

## 4.4 Edge Service

The Edge Service provides the interface between the public REST API and internal AMQP API.

### 4.4.1 TODO

#### 4.4.1.1 TODO version 1.0

- ✓ [x] Test environment and credentials



- ✓ [x] specify voucher content fields
- ✓ [x] specify voucher pricing fields
- ✓ [x] specify payment method fields

#### 4.4.1.2 TODO version 1.0-next

- ✓ [ ] refactor entities and apply relationships
- ✓ [ ] provide full resource schemas
- ✓ [ ] enforce validation of resource schemas

#### 4.4.2 Test environment and credentials

TODO

Username	Password	Description
<a href="#">test@test.org</a>	test	just protected
<a href="#">test@gmail.com</a>	test	just testing
testtest	test	qa

#### 4.4.3 Edge Service Registry

The Edge Service Registry is responsible to maintain an index of all currently available Edge Services and configure all its known [Load-Balancers] accordingly.

*As of writing no hosting solution has been chosen yet. Therefor the actual API which is used to configure the load-balancers remains unspecified for now.*

---

All Edge Service Instances **must** register themselves on the registry on startup and un-register on shutdown.

#### 4.4.4 Edge Service Relay

Every Edge Service will be provided with a local relay connecting its instances to the central Message Broker cluster. The message exchange routing of the relay is configured in such a way, that only messages relevant to its connected Edge Services is routed to it from the broker cluster.

*See [Message Relay] for more information.*

---

#### 4.4.5 Edge Service Client

Each Service that is to be served by the Edge Service is represented by a so-called Edge Service Client. It is responsible to provide the implementation of the public REST API for that given service.





#### 4.4.6 ReST API

The Edge Service API acts as translation layer between the Services and the Frontend. The actual logic is implemented in separate [Edge Service Client]s for each individual Service consumed by the API or API endpoint provided by the API, respectively.

All public-facing endpoints **must** adhere to the [JSONAPI] specification and be subject to strict [Access Control].

##### 4.4.6.1 Login

###### 4.4.6.1.1 Login request

Variables	Description	Example
ICP_SHOP_ID	see [Test environment and credentials]	
ICP_SHOP_PASSPHRASE	see [Test environment and credentials]	
ICP_SHOP_DIGEST	Derived from \${ICP_SHOP_ID} and \${ICP_SHOP_PASSPHRASE}	MzAwMDAwMTY6dG9wLXNIY3JI dC1wYXNzcGhyYXNI

```
POST /api/rest/v1/auth/login HTTP/1.1
Authorization: Basic ${ICP_SHOP_DIGEST}
Accept: application/vnd.api+json
```

###### 4.4.6.1.2 Login response

<pre>HTTP/1.1 201 Created Content-Type: application/vnd.api+json</pre>
--

###### 4.4.6.1.3 Cookies

Name	Value	Domain	Path	Expires	HttpOnly	Secure
SESSION	MmQ5ZjkxO GltMTM4My0 0M2E0LWI1M DYtYmUwZjY wMzJiZDY5	master.env.a pi.shop1.roc ks	/	Seassion	true	false

##### 4.4.6.2 Logout

###### 4.4.6.2.1 Logout request

Variables	Description	Example
-----------	-------------	---------



```
POST /api/rest/v1/auth/logout HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### *4.4.6.2.2 Logout response*

```
HTTP/1.1 204 No Content
```

#### **4.4.6.3 Voucher list**

See *[VoucherWebList]* method in the *[Mimas integration]* documentation for the source of these fields.



#### *4.4.6.3.1 Voucher list request*

```
GET /api/rest/v1/vouchers HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### *4.4.6.3.2 Voucher list response*

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```



```
{
  "data": [
    {
      "id": "f634f09f-9a06-4297-a326-0d703938b702",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL10",
        "currency": "EUR",
        "ean-code": "8717472242211",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 10 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 10",
        "value": 10.0,
        "price-total": 11.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": {
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
            "name": "PayPal",
            "provider-name": "Wirecard"
          }
        }
      }
    },
    {
      "id": "a3473e20-19b4-44fe-8389-67a511854759",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL15",
        "currency": "EUR",
        "ean-code": "8717472242129",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 15 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 15",
        "value": 15.0,
        "price-total": 16.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": [
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
```



We enable docx-as-code

```
        "name": "PayPal",  
        "provider-name": "Wirecard"  
      }  
    ]  
  }  
}
```

#### **4.4.6.4 Voucher detail**



#### 4.4.6.4.1 Voucher detail request

```
GET /api/rest/v1/vouchers/:voucher-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.4.6.4.2 Voucher detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "f634f09f-9a06-4297-a326-0d703938b702",
    "type": "voucher",
    "attributes": {
      "code": "LEBALLINNL10",
      "currency": "EUR",
      "ean-code": "8717472242211",
      "encryption-type": 0,
      "kind": 0,
      "name": "Lebara All-in NL 10 EUR",
      "provider-code": "LEBARA",
      "provider-name": "Lebara",
      "short-name": "All-in NL 10",
      "value": 10.0,
      "price-total": 11.99,
      "price-tax": 0.00,
      "price-fee": 1.99,
      "payment-methods": {
        "wirecard-creditcard": {
          "name": "Credit Card",
          "provider-name": "Wirecard",
          "image-url": "https://wirecard.com/logo.png"
        },
        "wirecard-paypal": {
          "name": "PayPal",
          "provider-name": "Wirecard"
        }
      }
    }
  }
}
```

#### 4.4.6.5 Voucher order

See *[VoucherWebReserve]* method in the *[Mimas integration]* documentation for the source of these fields.

##### 4.4.6.5.1 Voucher order request

Variables	Description	Example		



```
POST /api/rest/v1/voucher-orders HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "voucher-code": "VODADE15"
      "payment-method": "Wirecard VISA",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "provider-text-url": "https://provider.texts",

      "order-success-redirect-url": "/success",
      "order-failed-redirect-url": "failed",
      "order-canceled-redirect-url": "/cancel"
    }
  }
}
```

field name	type	constrains	description	example
id	UUID	globally unique	globally unique UUID	c91feaf3-5968-4b60-aa5b-62fe7fdd905c
type	string	n/a	type of request, must be "voucher-order", "voucher-order"	
voucher-code	string	n/a	code of the voucher gathered from voucher list	"VODADE15"
payment-method	string	n/a	on of the allowed methods for the voucher, denoted in voucher list response	"Wirecard VISA"
notification-type	string	currently only "email" is supported	type of the notification target	"email"
notification-target	string	n/a	target adress	"[me@my-email.com] (mailto:me@my-email.com)"



field name	type	constraints	description	example
order-success-redirect-url	string	valid URL	Where the wirecard should redirect if payment was successfull	protect
order-failed-redirect-url	string	valid URL	Where the wirecard should redirect if payment fails	protect
order-canceled-redirect-url	string	valid URL	Where the wirecard should redirect if payment was canceled	protect

#### 4.4.6.5.2 Voucher order response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgF26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINNL10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 0.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "0",
      "batch-number": "1234"
    }
  }
}
```

#### 4.4.6.6 Voucher order detail





#### 4.4.6.6.1 Voucher order detail request

```
GET /api/rest/v1/voucher-orders/:order-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.4.6.6.2 Voucher order detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgf26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINN10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 1.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "1611313953175"
    }
  }
}
```

#### 4.4.6.7 Provider Text

See `[GetProviderTexts]` method in the *[Mimas integration]* documentation for the source of these fields.

##### 4.4.6.7.1 Provider Text request

Variables	Description	Example
-----------	-------------	---------



```
GET /api/rest/v1/provider-texts/ HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

GET /api/rest/v1/provider-texts/:provider-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.4.6.7.2 Provider Text response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": [
    {
      "id": "f74bb480-d6df-47ad-8a17-8d7b1b176661",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "title",
        "title": "Produktname",
        "body": "About You-Geschenkkarte"
      }
    },
    {
      "id": "a598bf3b-abe0-4fc4-80fa-135a84510c81",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "subtitle",
        "title": "Slogan",
        "body": "Hol Dir per E-Mail eine About You-Geschenkkarte."
      }
    }
  ]
}
```

#### 4.4.6.8 Content

See [Content Service] for more information on the source of these fields.

##### 4.4.6.8.1 Content request

Variables	Description	Example
-----------	-------------	---------



```
GET /api/rest/v1/content/i18next.json HTTP/1.1
Authorization: No Auth
Accept: application/json
```

#### 4.4.6.8.2 Content response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "en": {
    "world.hello": "Hello, World!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Cancel",
    "radio.options": {
      "option-1": "Selection 1",
      "option-2": "Selection 2",
    }
  },
  "de": {
    "world.hello": "Hallo, Welt!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Abbrechen"
    "radio.options": {
      "option-1": "Auswahl 1",
      "option-2": "Auswahl 2",
    }
  }
}
```

#### 4.4.6.9 Verification

*See for more information on the source of these fields.*



#### 4.4.6.9.1 Verification request by phone number

```
POST /api/rest/v1//auth/phone-number-verification HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "38f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "sms-confirmation-code": "716569"
    }
  }
}
```

#### 4.4.6.9.2 Verification by phone number response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "data": {
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": false,
      "remaining-number-of-attempts": 4,
      "remaining-expiration-time": 0
    }
  }
}
```

#### 4.4.6.9.3 Verification request by sms

```
POST /api/rest/v1/auth/sms-verification-code HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "48f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655"
    }
  }
}
```

#### 4.4.6.9.4 Verification by sms number response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```



```
{
  "data": {
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": true,
      "remaining-number-of-attempts": 2,
      "remaining-delay-time": 20,
      "remaining-expiration-time": 1800
    }
  }
}
```

#### 4.4.6.10 Cookies

##### 4.4.6.10.1 Cookies request

POST /api/rest/v1/cookies HTTP/1.1  
Authorization: No Auth  
Accept: application/vnd.api+json

```
{
  "data": [
    {
      "type": "cookie",
      "id": "shopId",
      "attributes": {
        "name": "shopId",
        "value": "1234"
      }
    },
    {
      "type": "cookie",
      "id": "ns_dev_id",
      "attributes": {
        "name": "ns_dev_id",
        "value": "device_id"
      }
    }
  ]
}
```

##### 4.4.6.10.2 Cookies response

HTTP/1.1 200 OK

#### 4.4.6.11 Documentation

Returns list of Json Schemas for request and response for requested rest endpoint. Use exact endpoint plus schema url parameter. For example, /api/rest/v1/vouchers?schema will return json schemas for /api/rest/v1/vouchers endpoint.



#### 4.4.6.11.1 Documentation request

```
GET /api/rest/v1/vouchers?schema HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.4.6.11.2 Documentation response

Complete Response is omitted because of size , here is just a part :

```
HTTP/1.1 200 OK
```

```
{
  "required": [
    "createdDate",
    "notificationStatus",
    "notificationTarget",
    "notificationType",
    "orderFeeTotal",
    "orderPriceTotal",
    "orderStatus",
    "orderTaxTotal",
    "paymentMethod",
    "paymentProvider",
    "paymentStatus",
    "shortId",
    "voucherCode",
    "voucherCurrency",
    "voucherStatus",
    "voucherValue"
  ]
}
```

#### 4.4.6.12 i18 internationalization

- ✓ Retrieves i18next json based on requested language

##### 4.4.6.12.1 i18 request

```
GET api/rest/v1/content/i18next.json/de HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

## 4.5 Edge Service

The Edge Service provides the interface between the public REST API and internal AMQP API.

### 4.5.1 TODO

#### 4.5.1.1 TODO version 1.0

- ✓ [x] Test environment and credentials



- ✓ [x] specify voucher content fields
- ✓ [x] specify voucher pricing fields
- ✓ [x] specify payment method fields

#### 4.5.1.2 TODO version 1.0-next

- ✓ [ ] refactor entities and apply relationships
- ✓ [ ] provide full resource schemas
- ✓ [ ] enforce validation of resource schemas

#### 4.5.2 Test environment and credentials

TODO

Username	Password	Description
<a href="#">test@test.org</a>	test	just protected
<a href="#">test@gmail.com</a>	test	just testing
testtest	test	qa

#### 4.5.3 Edge Service Registry

The Edge Service Registry is responsible to maintain an index of all currently available Edge Services and configure all its known [Load-Balancers] accordingly.

*As of writing no hosting solution has been chosen yet. Therefor the actual API which is used to configure the load-balancers remains unspecified for now.*

---

All Edge Service Instances **must** register themselves on the registry on startup and un-register on shutdown.

#### 4.5.4 Edge Service Relay

Every Edge Service will be provided with a local relay connecting its instances to the central Message Broker cluster. The message exchange routing of the relay is configured in such a way, that only messages relevant to its connected Edge Services is routed to it from the broker cluster.

*See [Message Relay] for more information.*

---

#### 4.5.5 Edge Service Client

Each Service that is to be served by the Edge Service is represented by a so-called Edge Service Client. It is responsible to provide the implementation of the public REST API for that given service.



#### 4.5.6 ReST API

The Edge Service API acts as translation layer between the Services and the Frontend. The actual logic is implemented in separate [Edge Service Client]s for each individual Service consumed by the API or API endpoint provided by the API, respectively.

All public-facing endpoints **must** adhere to the [JSONAPI] specification and be subject to strict [Access Control].

##### 4.5.6.1 Login

###### 4.5.6.1.1 Login request

Variables	Description	Example
ICP_SHOP_ID	see [Test environment and credentials]	
ICP_SHOP_PASSPHRASE	see [Test environment and credentials]	
ICP_SHOP_DIGEST	Derived from \${ICP_SHOP_ID} and \${ICP_SHOP_PASSPHRASE}	MzAwMDAwMTY6dG9wLXNIY3JI dC1wYXNzcGhyYXNI

```
POST /api/rest/v1/auth/login HTTP/1.1
Authorization: Basic ${ICP_SHOP_DIGEST}
Accept: application/vnd.api+json
```

###### 4.5.6.1.2 Login response

<pre>HTTP/1.1 201 Created Content-Type: application/vnd.api+json</pre>
--

###### 4.5.6.1.3 Cookies

Name	Value	Domain	Path	Expires	HttpOnly	Secure
SESSION	MmQ5ZjkxO GltMTM4My0 0M2E0LWI1M DYtYmUwZjY wMzJiZDY5	master.env.a pi.shop1.roc ks	/	Seassion	true	false

##### 4.5.6.2 Logout

###### 4.5.6.2.1 Logout request

Variables	Description	Example
-----------	-------------	---------





```
POST /api/rest/v1/auth/logout HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### *4.5.6.2.2 Logout response*

```
HTTP/1.1 204 No Content
```

#### **4.5.6.3 Voucher list**

See *[VoucherWebList]* method in the *[Mimas integration]* documentation for the source of these fields.



#### *4.5.6.3.1 Voucher list request*

```
GET /api/rest/v1/vouchers HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### *4.5.6.3.2 Voucher list response*

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```



```
{
  "data": [
    {
      "id": "f634f09f-9a06-4297-a326-0d703938b702",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL10",
        "currency": "EUR",
        "ean-code": "8717472242211",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 10 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 10",
        "value": 10.0,
        "price-total": 11.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": {
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
            "name": "PayPal",
            "provider-name": "Wirecard"
          }
        }
      }
    },
    {
      "id": "a3473e20-19b4-44fe-8389-67a511854759",
      "type": "voucher",
      "attributes": {
        "code": "LEBALLINNL15",
        "currency": "EUR",
        "ean-code": "8717472242129",
        "encryption-type": 0,
        "kind": 0,
        "name": "Lebara All-in NL 15 EUR",
        "provider-code": "LEBARA",
        "provider-name": "Lebara",
        "short-name": "All-in NL 15",
        "value": 15.0,
        "price-total": 16.99,
        "price-tax": 0.00,
        "price-fee": 1.99,
        "payment-methods": [
          "wirecard-creditcard": {
            "name": "Credit Card",
            "provider-name": "Wirecard",
            "image-url": "https://wirecard.com/logo.png"
          }
          "wirecard-paypal": {
```



We enable docx-as-code

```
        "name": "PayPal",  
        "provider-name": "Wirecard"  
      }  
    ]  
  }  
]  
}
```

#### **4.5.6.4 Voucher detail**



#### 4.5.6.4.1 Voucher detail request

```
GET /api/rest/v1/vouchers/:voucher-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.5.6.4.2 Voucher detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "f634f09f-9a06-4297-a326-0d703938b702",
    "type": "voucher",
    "attributes": {
      "code": "LEBALLINNL10",
      "currency": "EUR",
      "ean-code": "8717472242211",
      "encryption-type": 0,
      "kind": 0,
      "name": "Lebara All-in NL 10 EUR",
      "provider-code": "LEBARA",
      "provider-name": "Lebara",
      "short-name": "All-in NL 10",
      "value": 10.0,
      "price-total": 11.99,
      "price-tax": 0.00,
      "price-fee": 1.99,
      "payment-methods": {
        "wirecard-creditcard": {
          "name": "Credit Card",
          "provider-name": "Wirecard",
          "image-url": "https://wirecard.com/logo.png"
        },
        "wirecard-paypal": {
          "name": "PayPal",
          "provider-name": "Wirecard"
        }
      }
    }
  }
}
```

#### 4.5.6.5 Voucher order

See *[VoucherWebReserve]* method in the *[Mimas integration]* documentation for the source of these fields.

##### 4.5.6.5.1 Voucher order request

Variables	Description	Example		



```
POST /api/rest/v1/voucher-orders HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "voucher-code": "VODADE15"
      "payment-method": "Wirecard VISA",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "provider-text-url": "https://provider.texts",

      "order-success-redirect-url": "/success",
      "order-failed-redirect-url": "failed",
      "order-canceled-redirect-url": "/cancel"
    }
  }
}
```

field name	type	constrains	description	example
id	UUID	globally unique	globally unique UUID	c91feaf3-5968-4b60-aa5b-62fe7fdd905c
type	string	n/a	type of request, must be "voucher-order", "voucher-order"	
voucher-code	string	n/a	code of the voucher gathered from voucher list	"VODADE15"
payment-method	string	n/a	on of the allowed methods for the voucher, denoted in voucher list response	"Wirecard VISA"
notification-type	string	currently only "email" is supported	type of the notification target	"email"
notification-target	string	n/a	target adress	"[me@my-email.com] (mailto:me@my-email.com)"



field name	type	constraints	description	example
order-success-redirect-url	string	valid URL	Where the wirecard should redirect if payment was successfull	protect
order-failed-redirect-url	string	valid URL	Where the wirecard should redirect if payment fails	protect
order-canceled-redirect-url	string	valid URL	Where the wirecard should redirect if payment was canceled	protect

#### 4.5.6.5.2 Voucher order response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgF26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINNL10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 0.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "0",
      "batch-number": "1234"
    }
  }
}
```

#### 4.5.6.6 Voucher order detail



#### 4.5.6.6.1 Voucher order detail request

```
GET /api/rest/v1/voucher-orders/:order-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.5.6.6.2 Voucher order detail response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "data": {
    "id": "c91feaf3-5968-4b60-aa5b-62fe7fdd905c",
    "type": "voucher-order",
    "attributes": {
      "notification-status": "pending",
      "confirmation-status": "pending",
      "notification-type": "email",
      "notification-target": "me@my-email.com",
      "payment-provider": "wirecard",
      "payment-redirect-url": "https://wpp-test.wirecard.com/?wPaymentToken=y93bki675x2CdKP8-h0mLD2zDgf26ptjxSNRiKFiLM",
      "payment-session-id": "4b7c0aec-3db3-4cd2-9b22-4d41fb964fdc",
      "payment-status": "pending",
      "payment-method": "creditcard",
      "voucher-code": "LEBALLINN10",
      "voucher-currency": "EUR",
      "voucher-value": 10.0
      "voucher-status": "pending",
      "order-price-total": 11.99,
      "order-tax-total": 0.00,
      "order-fee-total": 1.99,
      "order-status": "pending",
      "serial-number": "1234",
      "confirmation-date": "1611313953175"
    }
  }
}
```

#### 4.5.6.7 Provider Text

See `[GetProviderTexts]` method in the *[Mimas integration]* documentation for the source of these fields.

##### 4.5.6.7.1 Provider Text request

Variables	Description	Example
-----------	-------------	---------





```
GET /api/rest/v1/provider-texts/ HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json

GET /api/rest/v1/provider-texts/:provider-id HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.5.6.7.2 Provider Text response

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "data": [
    {
      "id": "f74bb480-d6df-47ad-8a17-8d7b1b176661",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "title",
        "title": "Produktname",
        "body": "About You-Geschenkkarte"
      }
    },
    {
      "id": "a598bf3b-abe0-4fc4-80fa-135a84510c81",
      "type": "voucher-text",
      "attributes": {
        "provider-code": "ABOUTYOU",
        "kind": "subtitle",
        "title": "Slogan",
        "body": "Hol Dir per E-Mail eine About You-Geschenkkarte."
      }
    }
  ]
}
```

#### 4.5.6.8 Content

See [Content Service] for more information on the source of these fields.

##### 4.5.6.8.1 Content request

Variables	Description	Example
-----------	-------------	---------



```
GET /api/rest/v1/content/i18next.json HTTP/1.1
Authorization: No Auth
Accept: application/json
```

#### 4.5.6.8.2 Content response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "en": {
    "world.hello": "Hello, World!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Cancel",
    "radio.options": {
      "option-1": "Selection 1",
      "option-2": "Selection 2",
    }
  },
  "de": {
    "world.hello": "Hallo, Welt!",
    "example.text": "Lorem ipsum dolor sit amet...",
    "button.cancel": "Abbrechen"
    "radio.options": {
      "option-1": "Auswahl 1",
      "option-2": "Auswahl 2",
    }
  }
}
```

#### 4.5.6.9 Verification

*See for more information on the source of these fields.*



#### 4.5.6.9.1 Verification request by phone number

```
POST /api/rest/v1/auth/phone-number-verification HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "38f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "sms-confirmation-code": "716569"
    }
  }
}
```

#### 4.5.6.9.2 Verification by phone number response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "data": {
    "type": "phone-number-verification",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": false,
      "remaining-number-of-attempts": 4,
      "remaining-expiration-time": 0
    }
  }
}
```

#### 4.5.6.9.3 Verification request by sms

```
POST /api/rest/v1/auth/sms-verification-code HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

```
{
  "data": {
    "id": "48f4dfef-9baa-4302-831f-e41df4ce4084",
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655"
    }
  }
}
```

#### 4.5.6.9.4 Verification by sms number response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```



```
{
  "data": {
    "type": "sms-verification-code",
    "attributes": {
      "phone-number": "+381691994655",
      "ok": true,
      "remaining-number-of-attempts": 2,
      "remaining-delay-time": 20,
      "remaining-expiration-time": 1800
    }
  }
}
```

#### 4.5.6.10 Cookies

##### 4.5.6.10.1 Cookies request

POST /api/rest/v1/cookies HTTP/1.1  
Authorization: No Auth  
Accept: application/vnd.api+json

```
{
  "data": [
    {
      "type": "cookie",
      "id": "shopId",
      "attributes": {
        "name": "shopId",
        "value": "1234"
      }
    },
    {
      "type": "cookie",
      "id": "ns_dev_id",
      "attributes": {
        "name": "ns_dev_id",
        "value": "device_id"
      }
    }
  ]
}
```

##### 4.5.6.10.2 Cookies response

HTTP/1.1 200 OK

#### 4.5.6.11 Documentation

Returns list of Json Schemas for request and response for requested rest endpoint. Use exact endpoint plus schema url parameter. For example, /api/rest/v1/vouchers?schema will return json schemas for /api/rest/v1/vouchers endpoint.



#### 4.5.6.11.1 Documentation request

```
GET /api/rest/v1/vouchers?schema HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```

#### 4.5.6.11.2 Documentation response

Complete Response is omitted because of size , here is just a part :

```
HTTP/1.1 200 OK
```

```
{
  "required": [
    "createdDate",
    "notificationStatus",
    "notificationTarget",
    "notificationType",
    "orderFeeTotal",
    "orderPriceTotal",
    "orderStatus",
    "orderTaxTotal",
    "paymentMethod",
    "paymentProvider",
    "paymentStatus",
    "shortId",
    "voucherCode",
    "voucherCurrency",
    "voucherStatus",
    "voucherValue"
  ]
}
```

#### 4.5.6.12 i18 internationalization

✓ Retrieves i18next json based on requested language

##### 4.5.6.12.1 i18 request

```
GET api/rest/v1/content/i18next.json/de HTTP/1.1
Authorization: No Auth
Accept: application/vnd.api+json
```



We enable docx-as-code



Figure 1: Last Page