

[Documentation] TPP - Nickel APIs access

- I. Prerequisite
 - 1. Api data : customer-authentication-api-v2
 - 2. Api data : customer-banking-api
 - 3. Api data : personal-space-api
- II. Workflows
 - 1. Authentication
 - a. Light authentication
 - b. Full authentication
 - 2. Challenge
 - a. Login challenge
 - b. Password challenge
 - c. SMS challenge
 - 3. Balance
 - 4. Transactions
 - 5. Beneficiary
 - a. Get all beneficiaries
 - b. Add beneficiary
 - c. Delete a beneficiary
 - 6. Outgoing transfer
- III - Appendixes
 - 1. customer-authentication-api-v2 swagger
 - 2. customer-banking-api swagger
 - 3. personal-space-api swagger

I. Prerequisite

1. Api data : customer-authentication-api-v2

URL	https://api.nickel.eu/customer-authentication-api-v2
Endpoints	POST /v1/minimum-viable-authentications POST /v1/sms-code-requests POST /v1/sms-code-verifications
Swagger	Appendix 1

2. Api data : customer-banking-api

URL	https://api.nickel.eu/customer-banking-api
Endpoints	GET /accounts/{number}/operations GET /accounts/{number}/overview
Swagger	Appendix 2

3. Api data : personal-space-api

URL	https://api.nickel.eu/personal-space-api
------------	---

Endpoints	<p>POST /api/v3/authentications</p> <p>POST /api/v3/accounts/{accountNumber}/outgoing-transfers</p> <p>GET /api/v2/accounts/{accountNumber}/beneficiaries</p> <p>POST /api/v2/accounts/{accountNumber}/beneficiaries</p> <p>DELETE /api/v2/accounts/{accountNumber}/beneficiaries/{beneficiaryId}</p> <p>POST /api/v2/challenges/password/validations</p> <p>POST /api/v2/challenges/sms/codes</p> <p>POST /api/v2/challenges/sms/validations</p>
Swagger	Appendix 3

II. Workflows

For **each request**, you must provide these headers :

User-Agent	End customer User-Agent
Session-Id	A session ID representing an identifier of each customer session

On **secured-endpoint**, you may need to sent this :

auth-challenge-token	<p>If a challenge is required, token must be sent inside it on secured-endpoint.</p> <p>If challenges succeed each other, you must provide all the tokens separated by a comma.</p>
----------------------	---

1. Authentication

Authentication is separated into two phases : **light authentication** and **full authentication**. Each authentication is managed via a JWT signed differently.

a. Light authentication

Endpoint

POST /v1/minimum-viable-authentications

Description

It allows you to check the client's identifiers (barcode / access code pair) and to check if the device used is known. If not, a 401 error is returned.

Once the request succeeded (return code 200), the JWT obtained makes it possible to request the customer's balance and transactions. See [Balance](#) and [Transactions](#) sections below.

The light token is valid 15 minutes.

Required header

i These headers **must** be provided while requesting on this endpoint

Header name	Description
tpp-etsi-authorization-number	TPP identification (following ETSI TS 119 49 specification)
tpp-signature-timestamp	Actual timestamp (ISO 8601 format)

digest	POST only. Digest algorithm (at least SHA256)
Signature	Format : keyId=" ", algorithm=" ", headers=" ", signature=" " With : <ul style="list-style-type: none"> • keyId URL of certificate • algorithm algorithm used (at least RSA-SHA256) • headers list of headers signed • signature signature result itself
PSU-IP-Address	Customer (end user) IP Address
PSU-User-Agent	Customer (end user) User-Agent (raw format)

PSU_User_Agent has the same value of User_Agent which is generate by [/minimum-viable-authentications](#) endpoint.

tpp-etsi-authorization-number is an authorization number provided by ETSI organism for all TPP which want to do DSP2.

Responses

200 OK	
Response body	<pre>{ "customerId": <CUSTOMER_ID>, "accessToken": "<JWT>", "culture": "<CUSTOMER_CULTURE>", "registrationCountry": "<CUSTOMER_REGISTRATION_COUNTRY>", "customerType": "<CUSTOMER_TYPE>" }</pre>
Description	Customer data is returned with the json web token (JWT)
401 Unauthorized	
Response body	<pre>{ "path": "/v1/minimum-viable-authentications", "message": "error.oldBarcode", "status": 401 }</pre>
Description	The customer provided an old barcode.
401 Unauthorized	

Response body	<pre> { "challengeToken": "<CHALLENGE_TOKEN>", "challengeType": "SMS", "message": "error.authenticationChallenge", "path": "/v1/minimum-viable-authentications", "status": 401 } </pre>
Description	An input from the user is required to confirm the device. A call to the endpoint <code>/v1/sms-code-requests</code> must be done. See Login challenge section below.

401 Unauthorized

Response body	<pre> { "challengeToken": "<CHALLENGE_TOKEN>", "challengeType": "NUDATA_CONSENT", "message": "error.authenticationChallenge", "path": "/v1/minimum-viable-authentications", "status": 401 } </pre>
Description	The customer must accept some consents. To do so, you must inform the customer to connect to the Nickel Application to give consents.

401 Unauthorized

Response body	<pre> { "remainingAttemptCount": <REMAINING_ATTEMPT>, "path": "/v1/minimum-viable-authentications", "message": "error.badCredentials", "status": 401 } </pre>
Description	Barcode and access code do not match. Remaining attempt starts from 3. A 403 is thrown at 0 attempt left, see response below.

403 Forbidden

Response body	<pre> { "path": "/v1/minimum-viable-authentications", "message": "error.tooManyPasswordAttempts", "status": 403 } </pre>
Description	Customer has to wait 24 hours due to many password attempts

403 Forbidden

Response body	<pre> { "path": "/v1/minimum-viable-authentications", "message": "error.customerDisabled", "status": 403 } </pre>
Description	The customer has been disabled. He must contact the customer service

403 Forbidden

Response body	<pre> { "path": "/v1/minimum-viable-authentications", "message": "error.decisionRefused", "status": 403 } </pre>
Description	The customer is not authorized to connect. He must contact the customer service.

b. Full authentication

Endpoint

POST /api/v3/authentications

Description

Take JWT Light to do some conformity check then give full access to the customer personal space through a new JWT (Full).

The full token is valid 15 minutes.

Mandatory headers

The header Personal-Space is mandatory. It's a String used by our backends to modify behaviour and authorizations depending on channel and version of the channel used.

As a TPP, please use the value `web/2.25.TPP` for this header, so we can easily trace your requests.

2. Challenge

a. Login challenge

Endpoints

POST /v1/sms-code-requests

POST /v1/sms-code-verifications

Description

While requesting for light authentication, a 401 Unauthorized error can be obtained if a challenge is required. You must request the challenge by calling the endpoint /v1/sms-code-requests. The customer therefore receives an OTP by SMS, this OTP must be transmitted in the endpoint /v1/sms-code-verifications.

Responses

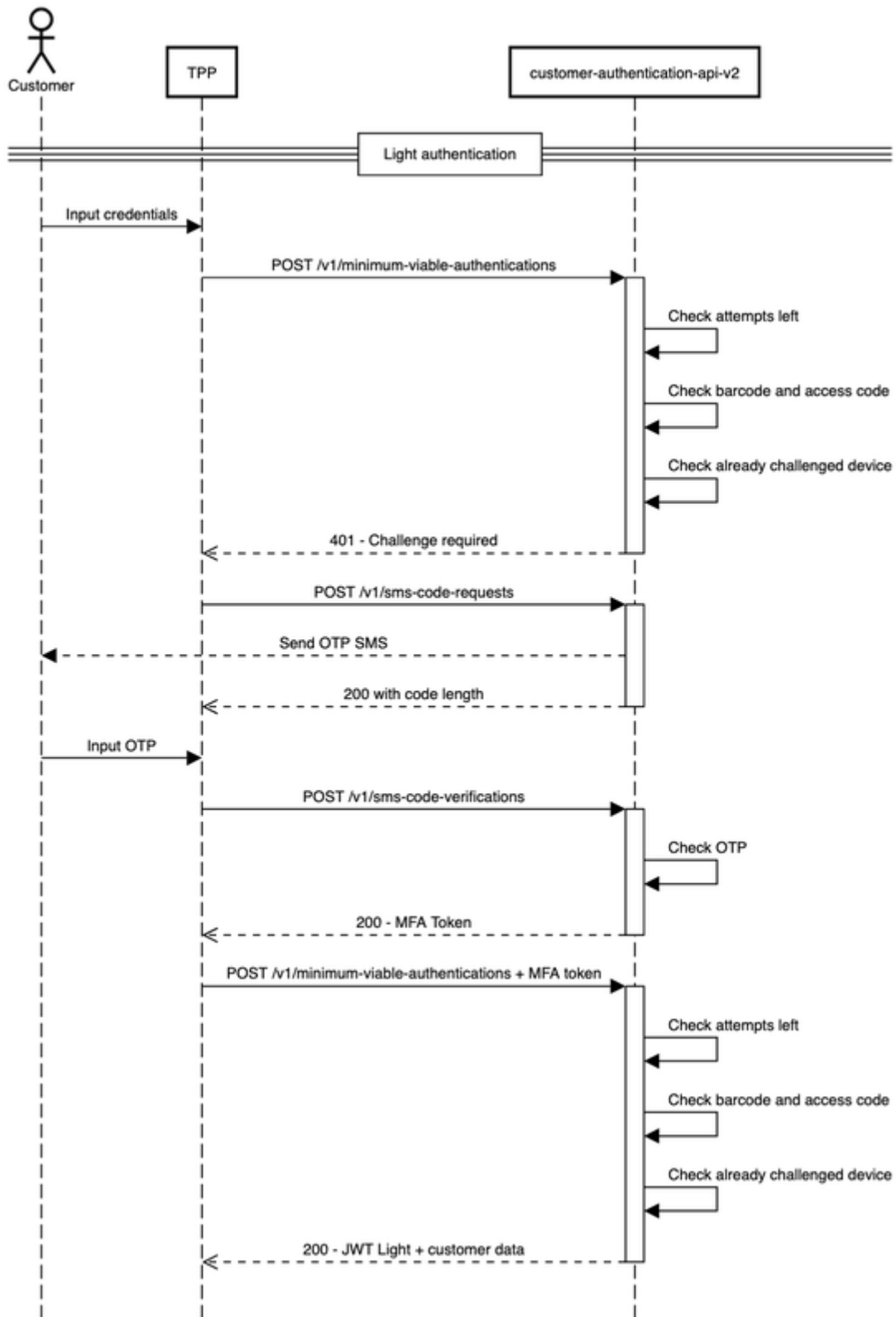
200 OK	
Response body	<pre>{ "mfaToken": "<MFA_TOKEN>" }</pre>
Description	Return MFA token to provide on the next light authentication attempt
400 Bad request	
Response body	<pre>{ "path": "/v1/sms-code-verifications", "message": "error.expiredChallenge", "status": 400 }</pre>
Description	Challenge OTP expired
401 Unauthorized	
Response body	<pre>{ "path": "/v1/sms-code-verifications", "message": "error.validation", "remainingAttempts": <REMAINING_ATTEMPTS>, "status": 401 }</pre>

Description	OTP failed, wrong code provided. Remaining attempts starts from 3. A 400 is thrown at 0 attempt left, see response below.
-------------	---

400 Bad request

Response body	<pre>{ "path": "/v1/sms-code-verifications", "message": "error.tooManyChallenges", "status": 400 }</pre>
Description	Customer has to wait 24 hours due to many OTP attempts

Diagram



b. Password challenge

Endpoint

POST /api/v2/challenges/password/validations

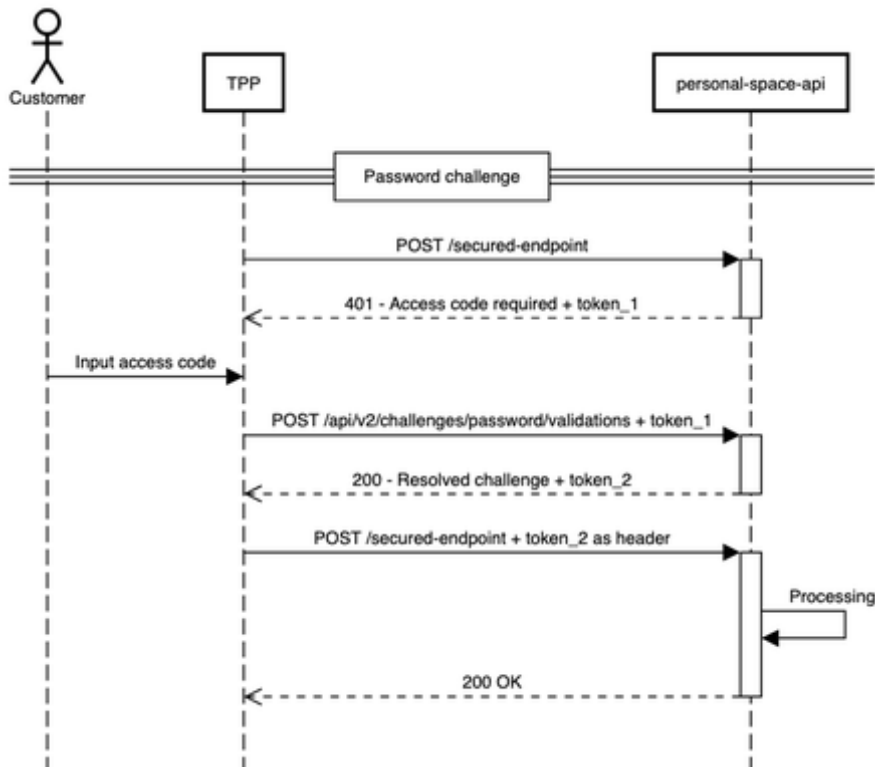
Description

When a challenge type PASSWORD is required, customer is prompt to enter his access code to proceed. Coupled with the challenge token sent by the endpoint that returned 401, you will gain a new token to unlock this one. This last token must be sent in the header named *auth-challenge-token*.

Responses

200 OK	
Response body	<pre>{ "token": "<CHALLENGE_RESOLVED_TOKEN>" }</pre>
Description	Return a token representing a challenge resolved. Use this token on the secured endpoint that gave you a 401 error.
400 Bad request	
Response body	<pre>{ "status": 400, "path": "/api/v2/challenges/password/validations", "message": "error.validation", "token": "<TOKEN>", "remainingAttempts": <REMAINING_ATTEMPT> }</pre>
Description	Wrong access code provided. Remaining attempts starts from 3. At 0 attempt left, you have to wait 24 hours to try again.

Diagram



c. SMS challenge

Endpoints

POST /api/v2/challenges/sms/codes

POST /api/v2/challenges/sms/validations

Description

While requesting on some secured endpoint, a 401 Unauthorized error can be obtained if a challenge SMS is required. You must request the challenge by calling the endpoint /api/v2/challenges/sms/codes. The customer therefore receives an OTP by SMS, this OTP must be transmitted in the endpoint /api/v2/challenges/sms/validations. A new token is then returned, this token must be sent as header *auth-challenge-token* when requesting on the first secured endpoint.

Responses

200 OK

Response body

```

{
  "token": "<CHALLENGE_RESOLVED_TOKEN>"
}
  
```

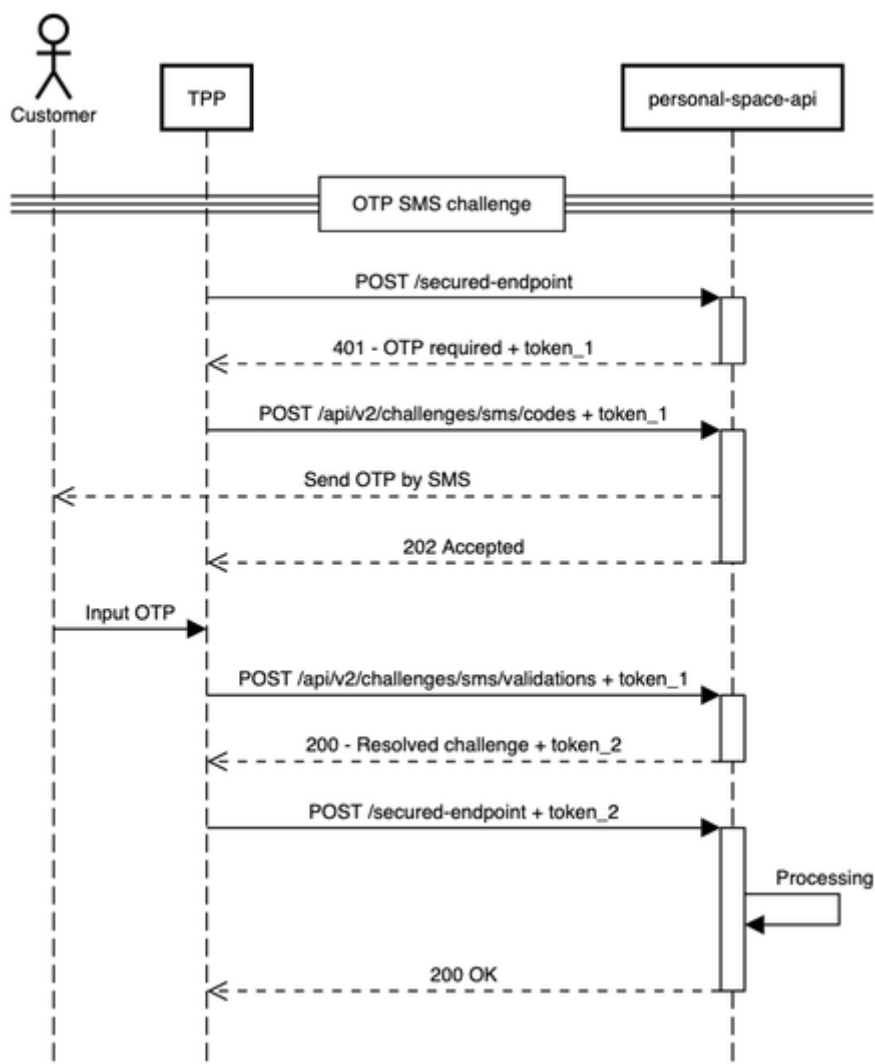
Description

Return a token representing a challenge resolved. Use this token on the secured endpoint that gave you a 401 error.

400 Unauthorized

Response body	<pre> { "path": "/api/v2/challenges/sms/validations", "message": "error.validation", "remainingAttempts": <REMAINING_ATTEMPTS>, "token": "<TOKEN>", "status": 400 } </pre>
Description	OTP failed, wrong code provided. Remaining attempts starts from 3. At 0, you will not be able to resolve the OTP.

Diagram



3. Balance

i This endpoint is secured by **LIGHT** JWT


Endpoint

GET /accounts/{number}/overview

Description

Endpoint to retrieve the balance of the customer. The balance amount returned must be divided by 100.

4. Transactions

 This endpoint is secured by **LIGHT** JWT

Endpoint


GET /accounts/{number}/operations

Description

Search method on customer's transactions.

5. Beneficiary

a. Get all beneficiaries

 This endpoint is secured by **FULL** JWT


Endpoint

GET /api/v2/accounts/{accountNumber}/beneficiaries

Description

Retrieve all beneficiaries. A beneficiary identifier is mandatory to make an outgoing transfer.

b. Add beneficiary

 This endpoint is secured by **FULL** JWT

Endpoint

POST /api/v2/accounts/{accountNumber}/beneficiaries

Description

Add a new beneficiary

Responses


401 Unauthorized

Response body	<pre> { "status":401, "path": "/api/v2/accounts/<ACCOUNT_NUMBER> /beneficiaries", "message": "error. additionalAuthenticationRequired", "challenge": { "type": "PASSWORD", "reason": "ADD_BENEFICIARY", "token": "<PASSWORD_CHALLENGE_TOKEN>" } } </pre>
Description	A password challenge is required to perform this action. See Password challenge section above.

401 Unauthorized

Response body	<pre> { "status": 401, "path": "/api/v2/accounts/<ACCOUNT_NUMBER> /beneficiaries", "message": "error. additionalAuthenticationRequired", "challenge": { "type": "SMS", "reason": "ADD_BENEFICIARY", "token": "<SMS_CHALLENGE_TOKEN>" } } </pre>
Description	A SMS challenge is required to perform this action. See SMS challenge section above.

c. Delete a beneficiary

 This endpoint is secured by **FULL JWT**

Endpoint

DELETE /api/v2/accounts/{accountNumber}/beneficiaries/{beneficiaryId}

Description

Beneficiary is necessary to make an outgoing transfer. You can retrieve the list of beneficiaries, add a beneficiary or delete one.

6. Outgoing transfer

i This endpoint is secured by **FULL JWT**

Endpoint

POST /api/v3/accounts/{accountNumber}/outgoing-transfers

Description

A beneficiary must be provided while doing an outgoing transfer (see [Beneficiary](#) above).

III - Appendixes

1. customer-authentication-api-v2 swagger

```
{
  "swagger": "2.0",
  "info": {
    "description": "Api documentation about the customer's authentication",
    "version": "1.0",
    "title": "customer-authentication-api",
    "termsOfService": "urn:tos",
    "contact": {},
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0"
    }
  },
  "host": "api.nickel.eu",
  "basePath": "/customer-authentication-api-v2",
  "tags": [
    {
      "name": "customer-authentication-resource",
      "description": "Customer Authentication Resource"
    }
  ],
  "paths": {
    "/v1/minimum-viable-authentications": {
      "post": {
        "tags": [
          "customer-authentication-resource"
        ],
        "summary": "getMinimumViableToken",
        "operationId": "getMinimumViableTokenUsingPOST",
        "consumes": [
          "application/json"
        ],
        "produces": [
```

```

    "*/*"
  ],
  "parameters": [
    {
      "name": "nudetect-payload",
      "in": "header",
      "description": "nudetect-payload",
      "required": false,
      "type": "string"
    },
    {
      "in": "body",
      "name": "payload",
      "description": "payload",
      "required": true,
      "schema": {
        "$ref": "#/definitions/AuthenticationPayload"
      }
    }
  ],
  "responses": {
    "200": {
      "description": "OK",
      "schema": {
        "$ref": "#/definitions/AuthenticationResponseView"
      }
    }
  }
},
"/v1/sms-code-requests": {
  "post": {
    "tags": [
      "customer-authentication-resource"
    ],
    "summary": "sendSmsCode",
    "operationId": "sendSmsCodeUsingPOST",
    "consumes": [
      "application/json"
    ],
    "produces": [
      "*/*"
    ],
    "parameters": [
      {
        "name": "challenge-token",
        "in": "query",
        "description": "challenge-token",
        "required": true,

```

```

        "type": "string"
    }
  ],
  "responses": {
    "200": {
      "description": "OK",
      "schema": {
        "$ref": "#/definitions/SendSmsCodeResponseView"
      }
    }
  }
},
"/v1/sms-code-verifications": {
  "post": {
    "tags": [
      "customer-authentication-resource"
    ],
    "summary": "getSmsToken",
    "operationId": "getSmsTokenUsingPOST",
    "consumes": [
      "application/json"
    ],
    "produces": [
      "*/*"
    ],
    "parameters": [
      {
        "name": "challenge-token",
        "in": "query",
        "description": "challenge-token",
        "required": true,
        "type": "string"
      },
      {
        "name": "code",
        "in": "query",
        "description": "code",
        "required": true,
        "type": "string"
      }
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "$ref": "#/definitions/VerifySmsCodeResponseView"
        }
      }
    }
  }
}

```



```

    }
  }
}
},
"definitions": {
  "AuthenticationAppPayload": {
    "type": "object",
    "properties": {
      "type": {
        "type": "string",
        "enum": [
          "DSP2",
          "MOBILE",
          "PHONE",
          "WEB"
        ]
      },
      "version": {
        "type": "string"
      }
    },
    "title": "AuthenticationAppPayload"
  },
  "AuthenticationClientPayload": {
    "type": "object",
    "properties": {
      "app": {
        "$ref": "#/definitions/AuthenticationAppPayload"
      },
      "device": {
        "$ref": "#/definitions/AuthenticationDevicePayload"
      }
    },
    "title": "AuthenticationClientPayload"
  },
  "AuthenticationDevicePayload": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string"
      },
      "name": {
        "type": "string"
      },
      "type": {
        "type": "string",
        "enum": [
          "AMAZON",
          "ANDROID",

```

```

        "BLACKBERRY",
        "CHROME",
        "FIREFOX",
        "IE11",
        "IOS",
        "OSX",
        "SAFARI",
        "UNKNOWN",
        "WINDOWS_8",
        "WINDOWS_PHONE"
    ]
}
},
"title": "AuthenticationDevicePayload"
},
"AuthenticationPayload": {
    "type": "object",
    "properties": {
        "barcode": {
            "type": "string"
        },
        "client": {
            "$ref": "#/definitions/AuthenticationClientPayload"
        },
        "mfaToken": {
            "type": "string"
        },
        "password": {
            "type": "string"
        }
    },
    "title": "AuthenticationPayload"
},
"AuthenticationResponseView": {
    "type": "object",
    "properties": {
        "accessToken": {
            "type": "string"
        },
        "culture": {
            "type": "string"
        },
        "customerId": {
            "type": "integer",
            "format": "int32"
        },
        "customerType": {
            "type": "string"
        }
    },

```

```

        "registrationCountry": {
            "type": "string"
        }
    },
    "title": "AuthenticationResponseView"
},
"SendSmsCodeResponseView": {
    "type": "object",
    "properties": {
        "codeLength": {
            "type": "integer",
            "format": "int32"
        },
        "recipient": {
            "type": "string"
        }
    },
    "title": "SendSmsCodeResponseView"
},
"VerifySmsCodeResponseView": {
    "type": "object",
    "properties": {
        "mfaToken": {
            "type": "string"
        }
    },
    "title": "VerifySmsCodeResponseView"
}
}
}
}

```

2. customer-banking-api swagger

```

{
  "swagger": "2.0",
  "info": {
    "description": "Api Documentation for customer-banking-api",
    "version": "1.0",
    "title": "customer-banking-api",
    "termsOfService": "urn:tos",
    "contact": {},
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0"
    }
  },

```

```

"host": "api.nickel.eu",
"basePath": "/customer-banking-api",
"tags": [
  {
    "name": "account-resource",
    "description": "Account Resource"
  }
],
"paths": {
  "/accounts/{number}/operations": {
    "get": {
      "tags": [
        "account-resource"
      ],
      "summary": "getAccountOperations",
      "operationId": "getAccountOperationsUsingGET_1",
      "produces": [
        "*/*"
      ],
      "parameters": [
        {
          "name": "Authorization",
          "in": "header",
          "description": "Authorization",
          "required": true,
          "type": "string"
        },
        {
          "name": "categories",
          "in": "query",
          "description": "categories",
          "required": false,
          "type": "array",
          "items": {
            "type": "string",
            "enum": [
              "ADMINISTRATION",
              "ALIMENTATION",
              "CHEQUE",
              "CLOTHING",
              "DEPOSIT",
              "FEE",
              "GROCERIES",
              "LEISURE",
              "REAL_ESTATE",
              "TAX",
              "TRANSFER_IN",
              "TRANSFER_OUT",
              "TRANSPORTATION",
            ]
          }
        }
      ]
    }
  }
}

```

```

        "TRAVEL",
        "UNKNOWN",
        "WITHDRAWAL"
    ]
},
"collectionFormat": "multi",
"enum": [
    "ADMINISTRATION",
    "ALIMENTATION",
    "CHEQUE",
    "CLOTHING",
    "DEPOSIT",
    "FEE",
    "GROCERIES",
    "LEISURE",
    "REAL_ESTATE",
    "TAX",
    "TRANSFER_IN",
    "TRANSFER_OUT",
    "TRANSPORTATION",
    "TRAVEL",
    "UNKNOWN",
    "WITHDRAWAL"
]
},
{
    "name": "end",
    "in": "query",
    "description": "end",
    "required": true,
    "type": "string",
    "format": "date-time"
},
{
    "name": "label",
    "in": "query",
    "description": "label",
    "required": false,
    "type": "string"
},
{
    "name": "number",
    "in": "path",
    "description": "number",
    "required": true,
    "type": "string"
},
{
    "name": "start",

```

```

    "in": "query",
    "description": "start",
    "required": true,
    "type": "string",
    "format": "date-time"
  },
  {
    "name": "types",
    "in": "query",
    "description": "types",
    "required": false,
    "type": "array",
    "items": {
      "type": "string",
      "enum": [
        "ACCOUNT_REFILL",
        "ACTIVATION_COMMISSION",
        "ATM_WITHDRAWAL",
        "CANCEL_DEBIT",
        "CARD_MANUFACTURING_FEE",
        "CHECK_CASHING",
        "CHECK_FEE",
        "CHECK_FUNDING",
        "CHECK_REMITTANCE",
        "CHECK_UNPAID",
        "CHECK_UNPAID_EARLY",
        "CHECK_UNPAID_FEE",
        "DEFERRED_DEBIT",
        "DEPOSIT",
        "DEPOSIT_CANCELLATION",
        "DEPOSIT_COMMISSION",
        "DEPOSIT_COMMISSION_CANCELLATION",
        "DIRECT_DEBIT",
        "DIRECT_DEBIT_REJECTION",
        "FEE",
        "FIRST_RENEWAL_COMMISSION",
        "FRAUD_REFUND",
        "FRAUD_REFUND_CANCEL",
        "INITIAL_DEPOSIT",
        "INITIAL_DEPOSIT_COMMISSION",
        "INITIAL_DEPOSIT_COMMISSION_CANCELLATION",
        "INTERNET_PURCHASE",
        "MEMBERSHIP_FEE",
        "MONEY_TRANSFER",
        "MONEY_TRANSFER_CANCELLATION",
        "PACK_REFUND",
        "PACK_SALE",
        "PARTIAL_MEMBERSHIP_FEE",
        "PARTIAL_REJECT_SDD_FEE",

```

```

    "PENDING_FEE" ,
    "PURCHASE" ,
    "REFUND" ,
    "REGULARIZATION" ,
    "REJECT_SDD_FEE" ,
    "SEIZURE_FEES" ,
    "SEIZURE_LIFTING" ,
    "SEIZURE_SEQUESTRATION" ,
    "TIP" ,
    "TRANSFER_RECEIVED" ,
    "TRANSFER_REJECTION" ,
    "TRANSFER_SENT" ,
    "UNKNOWN" ,
    "VARIOUS" ,
    "WITHDRAWAL" ,
    "WITHDRAWAL_CANCELLATION" ,
    "WITHDRAWAL_COMMISSION" ,
    "WITHDRAWAL_COMMISSION_CANCELLATION"
  ]
},
"collectionFormat": "multi",
"enum": [
  "ACCOUNT_REFILL" ,
  "ACTIVATION_COMMISSION" ,
  "ATM_WITHDRAWAL" ,
  "CANCEL_DEBIT" ,
  "CARD_MANUFACTURING_FEE" ,
  "CHECK_CASHING" ,
  "CHECK_FEE" ,
  "CHECK_FUNDING" ,
  "CHECK_REMITTANCE" ,
  "CHECK_UNPAID" ,
  "CHECK_UNPAID_EARLY" ,
  "CHECK_UNPAID_FEE" ,
  "DEFERRED_DEBIT" ,
  "DEPOSIT" ,
  "DEPOSIT_CANCELLATION" ,
  "DEPOSIT_COMMISSION" ,
  "DEPOSIT_COMMISSION_CANCELLATION" ,
  "DIRECT_DEBIT" ,
  "DIRECT_DEBIT_REJECTION" ,
  "FEE" ,
  "FIRST_RENEWAL_COMMISSION" ,
  "FRAUD_REFUND" ,
  "FRAUD_REFUND_CANCEL" ,
  "INITIAL_DEPOSIT" ,
  "INITIAL_DEPOSIT_COMMISSION" ,
  "INITIAL_DEPOSIT_COMMISSION_CANCELLATION" ,
  "INTERNET_PURCHASE" ,

```

```

        "MEMBERSHIP_FEE" ,
        "MONEY_TRANSFER" ,
        "MONEY_TRANSFER_CANCELLATION" ,
        "PACK_REFUND" ,
        "PACK_SALE" ,
        "PARTIAL_MEMBERSHIP_FEE" ,
        "PARTIAL_REJECT_SDD_FEE" ,
        "PENDING_FEE" ,
        "PURCHASE" ,
        "REFUND" ,
        "REGULARIZATION" ,
        "REJECT_SDD_FEE" ,
        "SEIZURE_FEES" ,
        "SEIZURE_LIFTING" ,
        "SEIZURE_SEQUESTRATION" ,
        "TIP" ,
        "TRANSFER_RECEIVED" ,
        "TRANSFER_REJECTION" ,
        "TRANSFER_SENT" ,
        "UNKNOWN" ,
        "VARIOUS" ,
        "WITHDRAWAL" ,
        "WITHDRAWAL_CANCELLATION" ,
        "WITHDRAWAL_COMMISSION" ,
        "WITHDRAWAL_COMMISSION_CANCELLATION"
    ]
}
],
"responses": {
    "200": {
        "description": "OK" ,
        "schema": {
            "$ref": "#/definitions/AccountOperationListView"
        }
    }
}
},
"/accounts/{number}/overview": {
    "get": {
        "tags": [
            "account-resource"
        ],
        "summary": "getAccountOverview" ,
        "operationId": "getAccountOverviewUsingGET" ,
        "produces": [
            "*"/*"
        ],
        "parameters": [

```



```

    {
      "name": "Authorization",
      "in": "header",
      "description": "Authorization",
      "required": true,
      "type": "string"
    },
    {
      "name": "number",
      "in": "path",
      "description": "number",
      "required": true,
      "type": "string"
    }
  ],
  "responses": {
    "200": {
      "description": "OK",
      "schema": {
        "$ref": "#/definitions/AccountOverviewView"
      }
    }
  }
},
},
"definitions": {
  "AccountOperationCategoryView": {
    "type": "object",
    "properties": {
      "i18nKey": {
        "type": "string"
      },
      "id": {
        "type": "string"
      },
      "label": {
        "type": "string"
      }
    },
    "title": "AccountOperationCategoryView"
  },
  "AccountOperationListView": {
    "type": "object",
    "properties": {
      "operations": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/AccountOperationView"
        }
      }
    }
  }
}

```

```

    }
  }
},
"title": "AccountOperationListView"
},
"AccountOperationView": {
  "type": "object",
  "properties": {
    "accountingDate": {
      "type": "string",
      "format": "date-time"
    },
    "amount": {
      "type": "integer",
      "format": "int64"
    },
    "authorizationNumber": {
      "type": "string"
    },
    "bucket": {
      "type": "string",
      "format": "date-time"
    },
    "category": {
      "$ref": "#/definitions/AccountOperationCategoryView"
    },
    "counterpartBic": {
      "type": "string"
    },
    "counterpartIban": {
      "type": "string"
    },
    "counterpartName": {
      "type": "string"
    },
    "currency": {
      "type": "string"
    },
    "date": {
      "type": "string",
      "format": "date-time"
    },
    "expectedAccountingDate": {
      "type": "string",
      "format": "date-time"
    },
    "fee": {
      "$ref": "#/definitions/Fee"
    }
  }
},

```

```
"id": {
  "type": "string"
},
"label": {
  "type": "string"
},
"mccCode": {
  "type": "string"
},
"merchantAddress": {
  "$ref": "#/definitions/AddressView"
},
"merchantName": {
  "type": "string"
},
"newBalance": {
  "type": "integer",
  "format": "int64"
},
"pinMode": {
  "type": "string",
  "enum": [
    "00",
    "01",
    "02",
    "99"
  ]
},
"provisionDate": {
  "type": "string",
  "format": "date-time"
},
"readMode": {
  "type": "string",
  "enum": [
    "00",
    "01",
    "02",
    "03",
    "04",
    "05",
    "06",
    "10",
    "81",
    "99"
  ]
},
"status": {
  "type": "string",
```

```

        "enum": [
            "EXECUTED",
            "PENDING"
        ]
    },
    "type": {
        "type": "string"
    },
    "typeI18n": {
        "type": "string"
    }
},
"title": "AccountOperationView"
},
"AccountOverviewView": {
    "type": "object",
    "properties": {
        "balance": {
            "type": "integer",
            "format": "int64"
        },
        "creationDate": {
            "type": "string",
            "format": "date-time"
        }
    },
    "title": "AccountOverviewView"
},
"AddressView": {
    "type": "object",
    "properties": {
        "address": {
            "type": "string"
        },
        "address2": {
            "type": "string"
        },
        "address3": {
            "type": "string"
        },
        "address4": {
            "type": "string"
        },
        "at": {
            "type": "string"
        },
        "city": {
            "type": "string"
        }
    },

```

```

    "country": {
      "type": "string"
    },
    "zipCode": {
      "type": "string"
    }
  },
  "title": "AddressView"
}
}
}

```

3. personal-space-api swagger

```

{
  "swagger": "2.0",
  "info": {
    "description": "Api Documentation for personal space",
    "version": "1.0",
    "title": "personal-space-api",
    "termsOfService": "urn:tos",
    "contact": {},
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0"
    }
  },
  "host": "api.nickel.eu",
  "basePath": "/personal-space-api",
  "tags": [
    {
      "name": "authentication-challenge-resource",
      "description": "Authentication Challenge Resource"
    },
    {
      "name": "authentication-resource",
      "description": "Authentication Resource"
    },
    {
      "name": "bank-account-resource",
      "description": "Bank Account Resource"
    }
  ],
  "paths": {
    "/api/v3/accounts/{accountNumber}/outgoing-transfers": {
      "post": {

```

```

"tags": [
  "bank-account-resource"
],
"summary": "submitTransfer",
"operationId": "submitTransferUsingPOST",
"consumes": [
  "application/json"
],
"produces": [
  "**/*"
],
"parameters": [
  {
    "name": "accountNumber",
    "in": "path",
    "description": "accountNumber",
    "required": true,
    "type": "string"
  },
  {
    "name": "auth-challenge-token",
    "in": "header",
    "description": "auth-challenge-token",
    "required": false,
    "type": "string"
  },
  {
    "name": "nudetect-payload",
    "in": "header",
    "required": false,
    "type": "string"
  },
  {
    "in": "body",
    "name": "transferVM",
    "description": "transferVM",
    "required": true,
    "schema": {
      "$ref": "#/definitions/TransferVM"
    }
  }
],
"responses": {
  "200": {
    "description": "OK",
    "schema": {
      "$ref": "#/definitions
/DeferredResultOfTransferSubmissionResultVM"
    }
  }
}

```

```

        }
    },
    "deprecated": false
}
},
"/api/v3/authentications": {
    "post": {
        "tags": [
            "authentication-resource"
        ],
        "summary": "authenticate",
        "operationId": "authenticateUsingPOST",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "**/*"
        ],
        "parameters": [
            {
                "name": "auth-challenge-token",
                "in": "header",
                "description": "auth-challenge-token",
                "required": false,
                "type": "string"
            },
            {
                "in": "body",
                "name": "body",
                "description": "body",
                "required": true,
                "schema": {
                    "$ref": "#/definitions/LoginVM"
                }
            }
        ],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "$ref": "#/definitions/DeferredResultOfResponseEntityOfAuthenticationVM"
                }
            }
        },
        "deprecated": false
    }
},
"/api/v2/accounts/{accountNumber}/beneficiaries": {

```

```

"get": {
  "tags": [
    "bank-account-resource"
  ],
  "summary": "getAllBeneficiaries",
  "operationId": "getAllBeneficiariesUsingGET",
  "produces": [
    "**/*"
  ],
  "parameters": [
    {
      "name": "accountNumber",
      "in": "path",
      "description": "accountNumber",
      "required": true,
      "type": "string"
    }
  ],
  "responses": {
    "200": {
      "description": "OK",
      "schema": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/Beneficiary"
        }
      }
    }
  },
  "deprecated": false
},
"post": {
  "tags": [
    "bank-account-resource"
  ],
  "summary": "addBeneficiary",
  "operationId": "addBeneficiaryUsingPOST",
  "consumes": [
    "application/json"
  ],
  "produces": [
    "**/*"
  ],
  "parameters": [
    {
      "name": "accountNumber",
      "in": "path",
      "description": "accountNumber",
      "required": true,

```



```

        "type": "string"
    },
    {
        "name": "auth-challenge-token",
        "in": "header",
        "description": "auth-challenge-token",
        "required": false,
        "type": "string"
    },
    {
        "in": "body",
        "name": "beneficiary",
        "description": "beneficiary",
        "required": true,
        "schema": {
            "$ref": "#/definitions/BeneficiaryVM"
        }
    },
    {
        "name": "nudetect-payload",
        "in": "header",
        "required": false,
        "type": "string"
    }
],
"responses": {
    "200": {
        "description": "OK",
        "schema": {
            "$ref": "#/definitions/Beneficiary"
        }
    }
},
"deprecated": false
}
},
"/api/v2/accounts/{accountNumber}/beneficiaries/
{beneficiaryId}": {
    "delete": {
        "tags": [
            "bank-account-resource"
        ],
        "summary": "deleteBeneficiary",
        "operationId": "deleteBeneficiaryUsingDELETE",
        "produces": [
            "**/*"
        ],
        "parameters": [
            {

```

```

        "name": "accountNumber",
        "in": "path",
        "description": "accountNumber",
        "required": true,
        "type": "string"
    },
    {
        "name": "beneficiaryId",
        "in": "path",
        "description": "beneficiaryId",
        "required": true,
        "type": "integer",
        "format": "int64"
    }
],
"responses": {
    "200": {
        "description": "OK"
    }
},
"deprecated": false
}
},
"/api/v2/challenges/password/validations": {
    "post": {
        "tags": [
            "authentication-challenge-resource"
        ],
        "summary": "resolvePasswordChallenge",
        "operationId": "resolvePasswordChallengeUsingPOST",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "**/*"
        ],
        "parameters": [
            {
                "in": "body",
                "name": "vm",
                "description": "vm",
                "required": true,
                "schema": {
                    "$ref": "#/definitions
/ResolvePasswordChallengeVM"
                }
            }
        ],
        "responses": {

```

```

        "200": {
            "description": "OK",
            "schema": {
                "$ref": "#/definitions
/ResolvedAuthenticationChallengeVM"
            }
        },
        "deprecated": false
    }
},
"/api/v2/challenges/sms/codes": {
    "post": {
        "tags": [
            "authentication-challenge-resource"
        ],
        "summary": "sendSmsCode",
        "operationId": "sendSmsCodeUsingPOST",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "*/*"
        ],
        "parameters": [
            {
                "in": "body",
                "name": "vm",
                "description": "vm",
                "required": true,
                "schema": {
                    "$ref": "#/definitions/SendSmsCodeVM"
                }
            }
        ],
        "responses": {
            "202": {
                "description": "Accepted",
                "schema": {
                    "$ref": "#/definitions
/SendSmsCodeResponseVM"
                }
            }
        },
        "deprecated": false
    }
},
"/api/v2/challenges/sms/validations": {
    "post": {

```

```

    "tags": [
      "authentication-challenge-resource"
    ],
    "summary": "resolveSmsChallenge",
    "operationId": "resolveSmsChallengeUsingPOST",
    "consumes": [
      "application/json"
    ],
    "produces": [
      "**/*"
    ],
    "parameters": [
      {
        "in": "body",
        "name": "vm",
        "description": "vm",
        "required": true,
        "schema": {
          "$ref": "#/definitions
/ResolveVerificationCodeChallengeVM"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "$ref": "#/definitions
/ResolvedAuthenticationChallengeVM"
        }
      }
    },
    "deprecated": false
  }
},
"definitions": {
  "AuthenticationVM": {
    "type": "object",
    "properties": {
      "accessToken": {
        "type": "string"
      },
      "features": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/FeatureVM"
        }
      }
    }
  },

```

```

        "id_token": {
            "type": "string"
        }
    },
    "title": "AuthenticationVM"
},
"Beneficiary": {
    "type": "object",
    "required": [
        "bic",
        "email",
        "iban",
        "id",
        "name"
    ],
    "properties": {
        "bankName": {
            "type": "string"
        },
        "bic": {
            "type": "string"
        },
        "creationDate": {
            "type": "string",
            "format": "date-time"
        },
        "email": {
            "type": "string"
        },
        "iban": {
            "type": "string"
        },
        "id": {
            "type": "integer",
            "format": "int64"
        },
        "name": {
            "type": "string"
        }
    }
},
"title": "Beneficiary"
},
"BeneficiaryVM": {
    "type": "object",
    "required": [
        "iban",
        "name"
    ],
    "properties": {

```

```

        "bankName": {
            "type": "string"
        },
        "bic": {
            "type": "string"
        },
        "email": {
            "type": "string"
        },
        "iban": {
            "type": "string"
        },
        "name": {
            "type": "string",
            "minLength": 0,
            "maxLength": 70
        }
    },
    "title": "BeneficiaryVM"
},
"DeferredResultOfResponseEntityOfAuthenticationVM": {
    "type": "object",
    "properties": {
        "result": {
            "type": "object"
        },
        "setOrExpired": {
            "type": "boolean"
        }
    }
},
"title":
"DeferredResultOfResponseEntityOfAuthenticationVM"
},
"DeferredResultOfTransferSubmissionResultVM": {
    "type": "object",
    "properties": {
        "result": {
            "type": "object"
        },
        "setOrExpired": {
            "type": "boolean"
        }
    }
},
"title": "DeferredResultOfTransferSubmissionResultVM"
},
"FeatureVM": {
    "type": "object",
    "properties": {
        "id": {

```

```

        "type": "string"
    },
    "status": {
        "type": "string",
        "enum": [
            "ENABLED",
            "DISABLED",
            "DISABLED_MOMENTARILY",
            "DISABLED_FOR_USER"
        ]
    }
},
"title": "FeatureVM"
},
>LoginVM": {
    "type": "object",
    "required": [
        "barcode",
        "deviceToken",
        "deviceType",
        "minimumViableToken",
        "version"
    ],
    "properties": {
        "barcode": {
            "type": "string",
            "minLength": 10,
            "maxLength": 10
        },
        "deviceName": {
            "type": "string"
        },
        "deviceToken": {
            "type": "string"
        },
        "deviceType": {
            "type": "string",
            "enum": [
                "IOS",
                "BLACKBERRY",
                "ANDROID",
                "WINDOWS_PHONE",
                "OSX",
                "WINDOWS_8",
                "AMAZON",
                "SAFARI",
                "CHROME",
                "FIREFOX",
                "IE11"
            ]
        }
    }
}

```

```

        ],
        },
        "minimumViableToken": {
            "type": "string"
        },
        "version": {
            "type": "string"
        }
    },
    "title": "LoginVM"
},
"ResolvedAuthenticationChallengeVM": {
    "type": "object",
    "properties": {
        "token": {
            "type": "string"
        }
    },
    "title": "ResolvedAuthenticationChallengeVM"
},
"ResolvePasswordChallengeVM": {
    "type": "object",
    "properties": {
        "password": {
            "type": "string"
        },
        "token": {
            "type": "string"
        }
    },
    "title": "ResolvePasswordChallengeVM"
},
"ResolveVerificationCodeChallengeVM": {
    "type": "object",
    "properties": {
        "code": {
            "type": "string"
        },
        "token": {
            "type": "string"
        }
    },
    "title": "ResolveVerificationCodeChallengeVM"
},
"SendSmsCodeVM": {
    "type": "object",
    "properties": {
        "resendChallenge": {
            "type": "boolean"
        }
    }
}

```



```

        },
        "token": {
            "type": "string"
        }
    },
    "title": "SendSmsCodeVM"
},
"SendSmsCodeResponseVM": {
    "type": "object",
    "properties": {
        "codeLength": {
            "type": "integer",
            "format": "int32"
        },
        "recipient": {
            "type": "string"
        },
        "token": {
            "type": "string"
        }
    },
    "title": "SendSmsCodeResponseVM"
},
"TransferSubmissionResultVM": {
    "type": "object",
    "required": [
        "status"
    ],
    "properties": {
        "status": {
            "type": "string",
            "enum": [
                "SENT",
                "INTERNAL_VALIDATION_NEEDED"
            ]
        }
    },
    "title": "TransferSubmissionResultVM"
},
"TransferVM": {
    "type": "object",
    "required": [
        "amount"
    ],
    "properties": {
        "amount": {
            "type": "number"
        },
        "beneficiaryId": {

```

```
        "type": "string"
    },
    "beneficiaryToken": {
        "type": "string"
    },
    "notifyBeneficiary": {
        "type": "boolean"
    },
    "reason": {
        "type": "string"
    }
},
"title": "TransferVM"
}
}
```