

[Documentation] TPP - Nickel APIs access

- I. Prerequisite
 - 1. Api data : customer-authentication-api-v2
 - 2. 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. Account
 - 4. Balance
 - 5. Transactions
 - 6. Beneficiaries
 - 7. Outgoing transfer
- III - Appendixes
 - 1. customer-authentication-api-v2 swagger
 - 2. personal-space-api swagger

I. Prerequisite [↗](#)

1. Api data : customer-authentication-api-v2 [↗](#)

URL	https://fallback.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 : personal-space-api [↗](#)

URL	https://fallback.nickel.eu/personal-space-api
Endpoints	POST /api/v3/authentications GET /api/v1/accounts GET /api/v1/accounts/{accountNumber}/operations GET /api/v1/accounts/{accountNumber}/overview GET /api/v2/accounts/{accountNumber}/beneficiaries POST /api/v3/accounts/{accountNumber}/outgoing-transfers

	POST /api/v2/challenges/password/validations POST /api/v2/challenges/sms/codes POST /api/v2/challenges/sms/validations
Swagger	Appendix 2

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

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 : <code>keyId="",algorithm="",headers="",signature=""</code> With : • <code>keyId</code> URL of certificate • <code>algorithm</code> algorithm used (at least RSA-SHA256) • <code>headers</code> list of headers signed • <code>signature</code> 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> 1 { 2 "customerId": <CUSTOMER_ID>, 3 "accessToken": "<JWT>", 4 "culture": "<CUSTOMER_CULTURE>", 5 "registrationCountry": "<CUSTOMER_REGISTRATION_COUNTRY>", 6 "customerType": "<CUSTOMER_TYPE>" 7 }</pre>
Description	Customer data is returned with the json web token (JWT)

401 Unauthorized	
Response body	<pre> 1 { 2 "path": "/v1/minimum-viable-authentications", 3 "message": "error.oldBarcode", 4 "status": 401 5 }</pre>
Description	The customer provided an old barcode.

401 Unauthorized	
Response body	<pre> 1 { 2 "challengeToken": "<CHALLENGE_TOKEN>", 3 "challengeType": "SMS", 4 "message": "error.authenticationChallenge", 5 "path": "/v1/minimum-viable-authentications", 6 "status": 401 7 }</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>1 { 2 "remainingAttemptCount": <REMAINING_ATTEMPT>, 3 "path": "/v1/minimum-viable-authentications", 4 "message": "error.badCredentials", 5 "status": 401 6 }</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>1 { 2 "path": "/v1/minimum-viable-authentications", 3 "message": "error.tooManyPasswordAttempts", 4 "status": 403 5 }</pre>
Description	Customer has to wait 24 hours due to many password attempts

403 Forbidden

Response body	<pre>1 { 2 "path": "/v1/minimum-viable-authentications", 3 "message": "error.customerDisabled", 4 "status": 403 5 }</pre>
Description	The customer has been disabled. He must contact the customer service

403 Forbidden

Response body	<pre>1 { 2 "path": "/v1/minimum-viable-authentications", 3 "message": "error.decisionRefused", 4 "status": 403 5 }</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>1 { 2 "mfaToken": "<MFA_TOKEN>" 3 }</pre>
Description	Return MFA token to provide on the next light authentication attempt

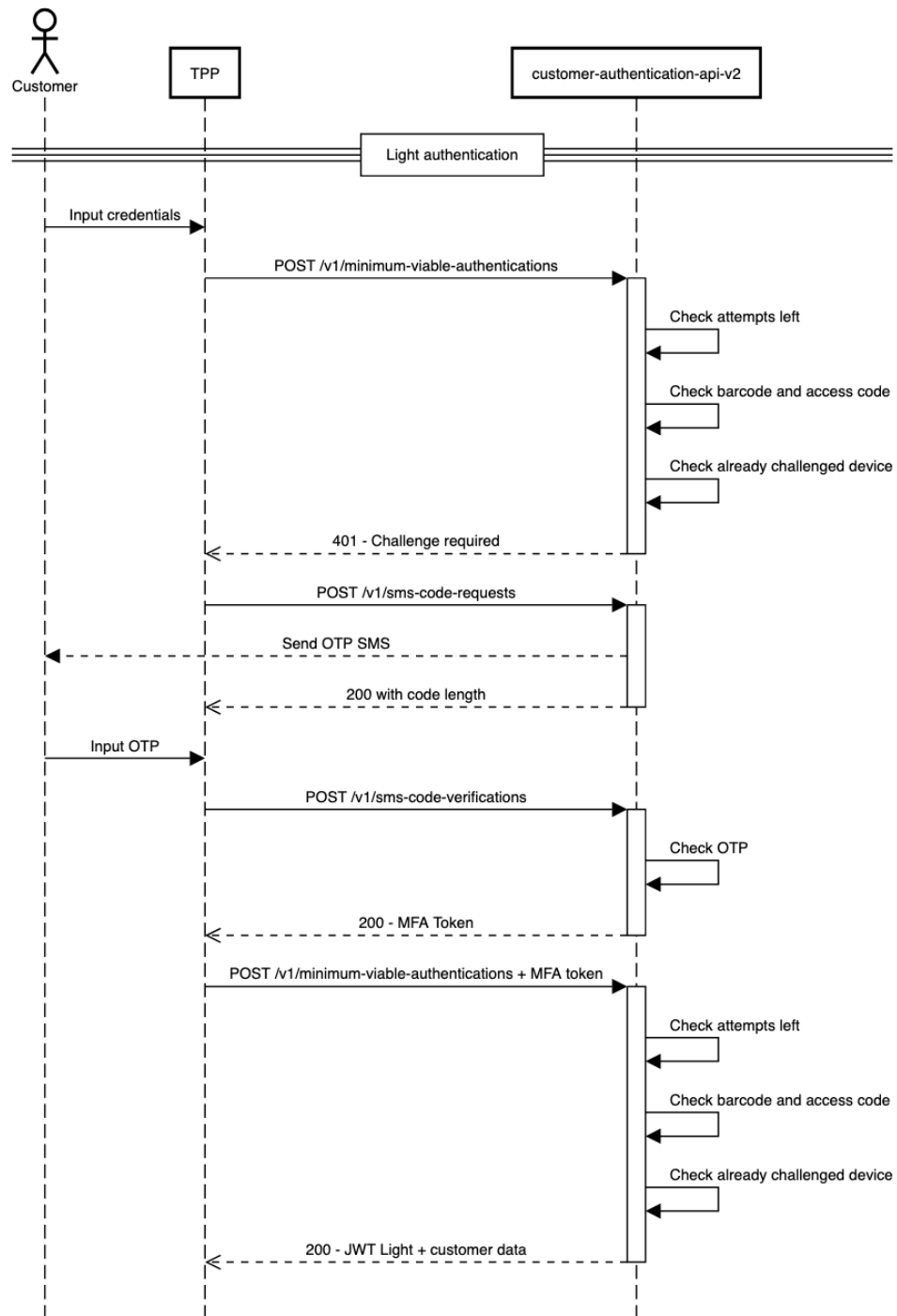
400 Bad request	
Response body	<pre>1 { 2 "path": "/v1/sms-code-verifications", 3 "message": "error.expiredChallenge", 4 "status": 400 5 }</pre>
Description	Challenge OTP expired



401 Unauthorized	
Response body	<pre>1 { 2 "path": "/v1/sms-code-verifications", 3 "message": "error.validation", 4 "remainingAttempts": <REMAINING_ATTEMPTS>, 5 "status": 401 6 }</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>1 { 2 "path": "/v1/sms-code-verifications", 3 "message": "error.tooManyChallenges", 4 "status": 400 5 }</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 [↗](#)

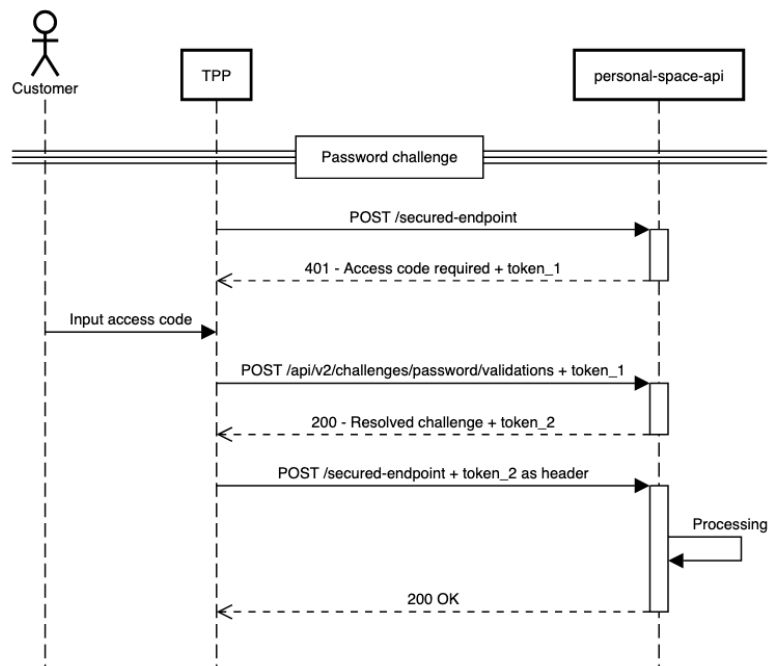
When a challenge type PASSWORD is required, customer is prompt to enter his access code to proceed. Coupled with the challenge token sent by the endpoint that returned 401, you will gain a new token to unlock this one. This last token must be sent in the header named *auth-challenge-token*.

Responses [↗](#)

200 OK	
Response body	<pre>1 { 2 "token": "<CHALLENGE_RESOLVED_TOKEN>" 3 }</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>1 { 2 "status": 400, 3 "path": "/api/v2/challenges/password/validations", 4 "message": "error.validation", 5 "token": "<TOKEN>", 6 "remainingAttempts": <REMAINING_ATTEMPT> 7 }</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 [↗](#)

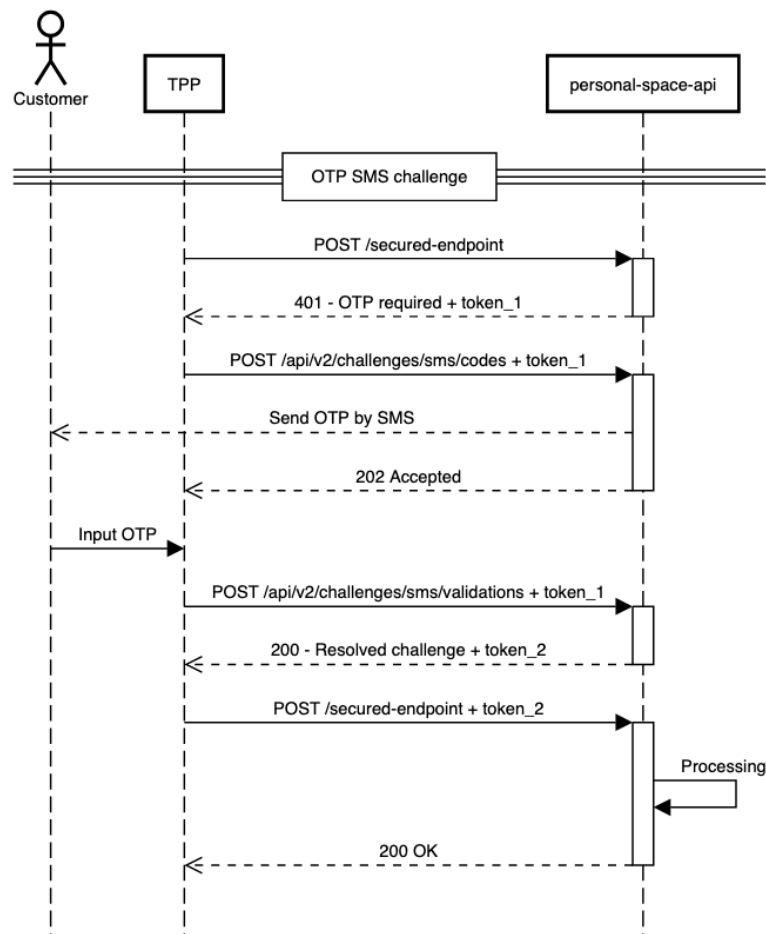
While requesting on some secured endpoint, a 401 Unauthorized error can be obtained if a challenge SMS is required. You must request the challenge by calling the endpoint /api/v2/challenges/sms/codes . The customer therefore receives an OTP by SMS, this OTP must be transmitted in the endpoint /api/v2/challenges/sms/validations . 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	<pre>1 { 2 "token": "<CHALLENGE_RESOLVED_TOKEN>" 3 }</pre>
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>1 { 2 "path": "/api/v2/challenges/sms/validations", 3 "message": "error.validation", 4 "remainingAttempts": <REMAINING_ATTEMPTS>, 5 "token": "<TOKEN>", 6 "status": 400 7 }</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. Account [↗](#)

This endpoint is secured by **LIGHT** JWT

Endpoint [↗](#)

GET /api/v1/accounts

Description [↗](#)

Endpoint to retrieve the `accountNumber` of the customer's bank account. Use `primaryAccount: true` to ignore linked bank accounts.

4. Balance [↗](#)

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

Endpoint [↗](#)

GET /api/v1/accounts/{accountNumber}/overview

Description [↗](#)

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

5. Transactions [↗](#)

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

Endpoint [↗](#)

GET /api/v1/accounts/{accountNumber}/operations

Description [↗](#)

Search method on customer's transactions.

6. Beneficiaries [↗](#)

i 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.

7. 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 [Beneficiaries](#) above).

III - Appendixes [↗](#)

1. customer-authentication-api-v2 swagger [↗](#)

```
1 {
2   "swagger": "2.0",
3   "info": {
4     "description": "Api documentation about the customer's authentication",
5     "version": "1.0",
6     "title": "customer-authentication-api",
7     "termsOfService": "urn:tos",
8     "contact": {},
9     "license": {
10      "name": "Apache 2.0",
11      "url": "http://www.apache.org/licenses/LICENSE-2.0"
12    }
13  },
14  "host": "api.nickel.eu",
```

```

15  "basePath": "/customer-authentication-api-v2",
16  "tags": [
17    {
18      "name": "customer-authentication-resource",
19      "description": "Customer Authentication Resource"
20    }
21  ],
22  "paths": {
23    "/v1/minimum-viable-authentications": {
24      "post": {
25        "tags": [
26          "customer-authentication-resource"
27        ],
28        "summary": "getMinimumViableToken",
29        "operationId": "getMinimumViableTokenUsingPOST",
30        "consumes": [
31          "application/json"
32        ],
33        "produces": [
34          "*/*"
35        ],
36        "parameters": [
37          {
38            "in": "body",
39            "name": "payload",
40            "description": "payload",
41            "required": true,
42            "schema": {
43              "$ref": "#/definitions/AuthenticationPayload"
44            }
45          }
46        ],
47        "responses": {
48          "200": {
49            "description": "OK",
50            "schema": {
51              "$ref": "#/definitions/AuthenticationResponseView"
52            }
53          }
54        }
55      }
56    },
57    "/v1/sms-code-requests": {
58      "post": {
59        "tags": [
60          "customer-authentication-resource"
61        ],
62        "summary": "sendSmsCode",
63        "operationId": "sendSmsCodeUsingPOST",
64        "consumes": [
65          "application/json"
66        ],
67        "produces": [
68          "*/*"
69        ],
70        "parameters": [
71          {

```

```

72         "name": "challenge-token",
73         "in": "query",
74         "description": "challenge-token",
75         "required": true,
76         "type": "string"
77     }
78 },
79     "responses": {
80         "200": {
81             "description": "OK",
82             "schema": {
83                 "$ref": "#/definitions/SendSmsCodeResponseView"
84             }
85         }
86     }
87 },
88 },
89 "/v1/sms-code-verifications": {
90     "post": {
91         "tags": [
92             "customer-authentication-resource"
93         ],
94         "summary": "getSmsToken",
95         "operationId": "getSmsTokenUsingPOST",
96         "consumes": [
97             "application/json"
98         ],
99         "produces": [
100             "*/*"
101         ],
102         "parameters": [
103             {
104                 "name": "challenge-token",
105                 "in": "query",
106                 "description": "challenge-token",
107                 "required": true,
108                 "type": "string"
109             },
110             {
111                 "name": "code",
112                 "in": "query",
113                 "description": "code",
114                 "required": true,
115                 "type": "string"
116             }
117         ],
118         "responses": {
119             "200": {
120                 "description": "OK",
121                 "schema": {
122                     "$ref": "#/definitions/VerifySmsCodeResponseView"
123                 }
124             }
125         }
126     }
127 },
128 },

```

```

129 "definitions": {
130   "AuthenticationAppPayload": {
131     "type": "object",
132     "properties": {
133       "type": {
134         "type": "string",
135         "enum": [
136           "DSP2",
137           "MOBILE",
138           "PHONE",
139           "WEB"
140         ]
141       },
142       "version": {
143         "type": "string"
144       }
145     },
146     "title": "AuthenticationAppPayload"
147   },
148   "AuthenticationClientPayload": {
149     "type": "object",
150     "properties": {
151       "app": {
152         "$ref": "#/definitions/AuthenticationAppPayload"
153       },
154       "device": {
155         "$ref": "#/definitions/AuthenticationDevicePayload"
156       }
157     },
158     "title": "AuthenticationClientPayload"
159   },
160   "AuthenticationDevicePayload": {
161     "type": "object",
162     "properties": {
163       "id": {
164         "type": "string"
165       },
166       "name": {
167         "type": "string"
168       },
169       "type": {
170         "type": "string",
171         "enum": [
172           "AMAZON",
173           "ANDROID",
174           "BLACKBERRY",
175           "CHROME",
176           "FIREFOX",
177           "IE11",
178           "IOS",
179           "OSX",
180           "SAFARI",
181           "UNKNOWN",
182           "WINDOWS_8",
183           "WINDOWS_PHONE"
184         ]
185       }
186     }
187   }
188 }

```

```

186     },
187     "title": "AuthenticationDevicePayload"
188 },
189 "AuthenticationPayload": {
190     "type": "object",
191     "properties": {
192         "barcode": {
193             "type": "string"
194         },
195         "client": {
196             "$ref": "#/definitions/AuthenticationClientPayload"
197         },
198         "mfaToken": {
199             "type": "string"
200         },
201         "password": {
202             "type": "string"
203         }
204     },
205     "title": "AuthenticationPayload"
206 },
207 "AuthenticationResponseView": {
208     "type": "object",
209     "properties": {
210         "accessToken": {
211             "type": "string"
212         },
213         "culture": {
214             "type": "string"
215         },
216         "customerId": {
217             "type": "integer",
218             "format": "int32"
219         },
220         "customerType": {
221             "type": "string"
222         },
223         "registrationCountry": {
224             "type": "string"
225         }
226     },
227     "title": "AuthenticationResponseView"
228 },
229 "SendSmsCodeResponseView": {
230     "type": "object",
231     "properties": {
232         "codeLength": {
233             "type": "integer",
234             "format": "int32"
235         },
236         "recipient": {
237             "type": "string"
238         }
239     },
240     "title": "SendSmsCodeResponseView"
241 },
242 "VerifySmsCodeResponseView": {

```

```

243     "type": "object",
244     "properties": {
245         "mfaToken": {
246             "type": "string"
247         }
248     },
249     "title": "VerifySmsCodeResponseView"
250 }
251 }
252 }

```

2. personal-space-api swagger [↗](#)

```

1  {
2      "swagger": "2.0",
3      "info": {
4          "description": "Api Documentation for personal space",
5          "version": "1.0",
6          "title": "personal-space-api",
7          "termsOfService": "urn:tos",
8          "contact": {},
9          "license": {
10             "name": "Apache 2.0",
11             "url": "http://www.apache.org/licenses/LICENSE-2.0"
12         }
13     },
14     "host": "api.nickel.eu",
15     "basePath": "/personal-space-api",
16     "tags": [
17         {
18             "name": "authentication-challenge-resource",
19             "description": "Authentication Challenge Resource"
20         },
21         {
22             "name": "authentication-resource",
23             "description": "Authentication Resource"
24         },
25         {
26             "name": "bank-account-resource",
27             "description": "Bank Account Resource"
28         }
29     ],
30     "paths": {
31         "/api/v1/accounts": {
32             "get": {
33                 "operationId": "getAccounts",
34                 "parameters": [
35                     {
36                         "in": "header",
37                         "name": "x-light-token",
38                         "required": true,
39                         "type": "string"
40                     }
41                 ],
42                 "produces": [
43                     "**/*"

```

```

44         ],
45         "responses": {
46             "200": {
47                 "schema": {
48                     "$ref": "#/definitions/AccountList"
49                 },
50                 "description": "OK"
51             }
52         },
53         "security": [],
54         "tags": [
55             "account-resource"
56         ]
57     }
58 },
59 "/api/v1/accounts/{accountNumber}/operations": {
60     "get": {
61         "operationId": "getOperations",
62         "parameters": [
63             {
64                 "in": "path",
65                 "name": "accountNumber",
66                 "required": true,
67                 "type": "string"
68             },
69             {
70                 "in": "query",
71                 "name": "start",
72                 "required": true,
73                 "type": "string",
74                 "format": "date-time"
75             },
76             {
77                 "in": "query",
78                 "name": "end",
79                 "required": true,
80                 "type": "string",
81                 "format": "date-time"
82             }
83         ],
84         "produces": [
85             "*/*"
86         ],
87         "responses": {
88             "200": {
89                 "schema": {
90                     "$ref": "#/definitions/AccountOperationList"
91                 },
92                 "description": "OK"
93             }
94         },
95         "tags": [
96             "account-resource"
97         ]
98     }
99 },
100 "/api/v1/accounts/{accountNumber}/overview": {

```

```

101     "get": {
102         "operationId": "getAccountOverview",
103         "parameters": [
104             {
105                 "in": "path",
106                 "name": "accountNumber",
107                 "required": true,
108                 "type": "string"
109             }
110         ],
111         "produces": [
112             "*/*"
113         ],
114         "responses": {
115             "200": {
116                 "schema": {
117                     "$ref": "#/definitions/AccountOverview"
118                 },
119                 "description": "OK"
120             }
121         },
122         "tags": [
123             "account-resource"
124         ]
125     }
126 },
127 "/api/v3/accounts/{accountNumber}/outgoing-transfers": {
128     "post": {
129         "tags": [
130             "bank-account-resource"
131         ],
132         "summary": "submitTransfer",
133         "operationId": "submitTransferUsingPOST",
134         "consumes": [
135             "application/json"
136         ],
137         "produces": [
138             "*/*"
139         ],
140         "parameters": [
141             {
142                 "name": "accountNumber",
143                 "in": "path",
144                 "description": "accountNumber",
145                 "required": true,
146                 "type": "string"
147             },
148             {
149                 "name": "auth-challenge-token",
150                 "in": "header",
151                 "description": "auth-challenge-token",
152                 "required": false,
153                 "type": "string"
154             },
155             {
156                 "in": "body",
157                 "name": "transferVM",

```

```

158         "description": "transferVM",
159         "required": true,
160         "schema": {
161             "$ref": "#/definitions/TransferVM"
162         }
163     },
164 ],
165 "responses": {
166     "200": {
167         "description": "OK",
168         "schema": {
169             "$ref": "#/definitions/TransferSubmissionResultVM"
170         }
171     }
172 },
173 "deprecated": false
174 }
175 },
176 "/api/v3/authentications": {
177     "post": {
178         "tags": [
179             "authentication-resource"
180         ],
181         "summary": "authenticate",
182         "operationId": "authenticateUsingPOST",
183         "consumes": [
184             "application/json"
185         ],
186         "produces": [
187             "*/*"
188         ],
189         "parameters": [
190             {
191                 "name": "auth-challenge-token",
192                 "in": "header",
193                 "description": "auth-challenge-token",
194                 "required": false,
195                 "type": "string"
196             },
197             {
198                 "in": "body",
199                 "name": "body",
200                 "description": "body",
201                 "required": true,
202                 "schema": {
203                     "$ref": "#/definitions/LoginVM"
204                 }
205             }
206         ],
207         "responses": {
208             "200": {
209                 "description": "OK",
210                 "schema": {
211                     "$ref": "#/definitions/AuthenticationVM"
212                 }
213             }
214         },

```

```

215         "deprecated": false
216     }
217 },
218 "/api/v2/accounts/{accountNumber}/beneficiaries": {
219     "get": {
220         "tags": [
221             "bank-account-resource"
222         ],
223         "summary": "getAllBeneficiaries",
224         "operationId": "getAllBeneficiariesUsingGET",
225         "produces": [
226             "**/*"
227         ],
228         "parameters": [
229             {
230                 "name": "accountNumber",
231                 "in": "path",
232                 "description": "accountNumber",
233                 "required": true,
234                 "type": "string"
235             }
236         ],
237         "responses": {
238             "200": {
239                 "description": "OK",
240                 "schema": {
241                     "type": "array",
242                     "items": {
243                         "$ref": "#/definitions/Beneficiary"
244                     }
245                 }
246             }
247         },
248         "deprecated": false
249     }
250 },
251 "/api/v2/challenges/password/validations": {
252     "post": {
253         "tags": [
254             "authentication-challenge-resource"
255         ],
256         "summary": "resolvePasswordChallenge",
257         "operationId": "resolvePasswordChallengeUsingPOST",
258         "consumes": [
259             "application/json"
260         ],
261         "produces": [
262             "**/*"
263         ],
264         "parameters": [
265             {
266                 "in": "body",
267                 "name": "vm",
268                 "description": "vm",
269                 "required": true,
270                 "schema": {
271                     "$ref": "#/definitions/ResolvePasswordChallengeVM"

```

```

272     }
273   }
274   ],
275   "responses": {
276     "200": {
277       "description": "OK",
278       "schema": {
279         "$ref": "#/definitions/ResolvedAuthenticationChallengeVM"
280       }
281     }
282   },
283   "deprecated": false
284 }
285 },
286 "/api/v2/challenges/sms/codes": {
287   "post": {
288     "tags": [
289       "authentication-challenge-resource"
290     ],
291     "summary": "sendSmsCode",
292     "operationId": "sendSmsCodeUsingPOST",
293     "consumes": [
294       "application/json"
295     ],
296     "produces": [
297       "**/*"
298     ],
299     "parameters": [
300       {
301         "in": "body",
302         "name": "vm",
303         "description": "vm",
304         "required": true,
305         "schema": {
306           "$ref": "#/definitions/SendSmsCodeVM"
307         }
308       }
309     ],
310     "responses": {
311       "202": {
312         "description": "Accepted",
313         "schema": {
314           "$ref": "#/definitions/SendSmsCodeResponseVM"
315         }
316       }
317     },
318     "deprecated": false
319   }
320 },
321 "/api/v2/challenges/sms/validations": {
322   "post": {
323     "tags": [
324       "authentication-challenge-resource"
325     ],
326     "summary": "resolveSmsChallenge",
327     "operationId": "resolveSmsChallengeUsingPOST",
328     "consumes": [

```

```

329         "application/json"
330     ],
331     "produces": [
332         "*"
333     ],
334     "parameters": [
335         {
336             "in": "body",
337             "name": "vm",
338             "description": "vm",
339             "required": true,
340             "schema": {
341                 "$ref": "#/definitions/ResolveVerificationCodeChallengeVM"
342             }
343         }
344     ],
345     "responses": {
346         "200": {
347             "description": "OK",
348             "schema": {
349                 "$ref": "#/definitions/ResolvedAuthenticationChallengeVM"
350             }
351         }
352     },
353     "deprecated": false
354 }
355 },
356 {
357     "definitions": {
358         "Account": {
359             "type": "object",
360             "properties": {
361                 "customerId": {
362                     "type": "integer",
363                     "format": "int32"
364                 },
365                 "firstName": {
366                     "type": "string"
367                 },
368                 "lastName": {
369                     "type": "string"
370                 },
371                 "number": {
372                     "type": "string"
373                 },
374                 "primaryAccount": {
375                     "type": "boolean"
376                 }
377             }
378         },
379         "AccountList": {
380             "type": "object",
381             "properties": {
382                 "accounts": {
383                     "type": "array",
384                     "items": {
385                         "$ref": "#/definitions/Account"

```

```

386     }
387   }
388 }
389 },
390 "AccountOperation": {
391   "type": "object",
392   "properties": {
393     "additionalData": {
394       "$ref": "#/definitions/AccountOperationAdditionalData"
395     },
396     "amount": {
397       "type": "integer",
398       "format": "int64"
399     },
400     "atmFeeAmount": {
401       "type": "integer",
402       "format": "int64"
403     },
404     "bucket": {
405       "type": "string",
406       "format": "date-time"
407     },
408     "category": {
409       "$ref": "#/definitions/AccountOperationCategory"
410     },
411     "counterpartBankName": {
412       "type": "string"
413     },
414     "counterpartName": {
415       "type": "string"
416     },
417     "currency": {
418       "type": "string"
419     },
420     "date": {
421       "type": "string",
422       "format": "date-time"
423     },
424     "exchangeRate": {
425       "type": "number"
426     },
427     "expectedAccountingDate": {
428       "type": "string",
429       "format": "date-time"
430     },
431     "fee": {
432       "$ref": "#/definitions/AccountOperationFee"
433     },
434     "id": {
435       "type": "string"
436     },
437     "label": {
438       "type": "string"
439     },
440     "localAmount": {
441       "type": "integer",
442       "format": "int64"

```

```

443     },
444     "localCurrency": {
445         "type": "string"
446     },
447     "mccCode": {
448         "type": "string"
449     },
450     "merchantAddress": {
451         "$ref": "#/definitions/AccountOperationMerchantAddress"
452     },
453     "merchantName": {
454         "type": "string"
455     },
456     "mobilePaymentWallet": {
457         "$ref": "#/definitions/AccountOperationMobilePaymentWallet"
458     },
459     "paymentMode": {
460         "$ref": "#/definitions/AccountOperationPaymentMode"
461     },
462     "pinMode": {
463         "$ref": "#/definitions/AccountOperationPinMode"
464     },
465     "ratePercentDifferential": {
466         "type": "number",
467         "format": "double"
468     },
469     "readMode": {
470         "$ref": "#/definitions/AccountOperationReadMode"
471     },
472     "referenceConversionRate": {
473         "type": "number"
474     },
475     "status": {
476         "$ref": "#/definitions/AccountOperationStatus"
477     },
478     "typeI18n": {
479         "type": "string"
480     }
481 }
482 },
483 "AccountOperationAdditionalData": {
484     "type": "object",
485     "properties": {
486         "cmc7": {
487             "type": "string"
488         },
489         "description": {
490             "type": "string"
491         },
492         "label": {
493             "type": "string"
494         },
495         "operationDate": {
496             "type": "string",
497             "format": "date-time"
498         },
499         "reference": {

```

```

500         "type": "string"
501     }
502 }
503 },
504 "AccountOperationCategory": {
505     "type": "object",
506     "properties": {
507         "il8nKey": {
508             "type": "string"
509         },
510         "id": {
511             "type": "string"
512         },
513         "label": {
514             "type": "string"
515         }
516     }
517 },
518 "AccountOperationFee": {
519     "type": "object",
520     "properties": {
521         "amount": {
522             "type": "integer",
523             "format": "int64"
524         }
525     }
526 },
527 "AccountOperationList": {
528     "type": "object",
529     "properties": {
530         "operations": {
531             "type": "array",
532             "items": {
533                 "$ref": "#/definitions/AccountOperation"
534             }
535         }
536     }
537 },
538 "AccountOperationMerchantAddress": {
539     "type": "object",
540     "properties": {
541         "address": {
542             "type": "string"
543         },
544         "address2": {
545             "type": "string"
546         },
547         "address3": {
548             "type": "string"
549         },
550         "address4": {
551             "type": "string"
552         },
553         "at": {
554             "type": "string"
555         },
556         "city": {

```

```

557         "type": "string"
558     },
559     "country": {
560         "type": "string"
561     },
562     "zipCode": {
563         "type": "string"
564     }
565 },
566 },
567 "AccountOperationMobilePaymentWallet": {
568     "type": "string",
569     "enum": [
570         "APPLE_PAY",
571         "GOOGLE_PAY",
572         "UNKNOWN"
573     ]
574 },
575 "AccountOperationPaymentMode": {
576     "type": "string",
577     "enum": [
578         "CLICK_TO_PAY",
579         "MOBILE",
580         "NFC_WITH_PIN",
581         "NFC_WITHOUT_PIN",
582         "PAYMENT_3DS",
583         "WITH_PIN",
584         "WITHOUT_PIN",
585         "UNKNOWN"
586     ]
587 },
588 "AccountOperationPinMode": {
589     "type": "string",
590     "enum": [
591         "00",
592         "01",
593         "02",
594         "99"
595     ]
596 },
597 "AccountOperationReadMode": {
598     "type": "string",
599     "enum": [
600         "00",
601         "01",
602         "02",
603         "03",
604         "04",
605         "05",
606         "06",
607         "10",
608         "81",
609         "99"
610     ]
611 },
612 "AccountOperationStatus": {
613     "type": "string",

```

```

614         "enum": [
615             "EXECUTED",
616             "PENDING",
617             "UNKNOWN"
618         ]
619     },
620     "AccountOverview": {
621         "type": "object",
622         "properties": {
623             "balance": {
624                 "type": "integer",
625                 "format": "int64"
626             },
627             "creationDate": {
628                 "type": "string",
629                 "format": "date-time"
630             }
631         }
632     },
633     "AuthenticationVM": {
634         "type": "object",
635         "properties": {
636             "accessToken": {
637                 "type": "string"
638             }
639         },
640         "title": "AuthenticationVM"
641     },
642     "Beneficiary": {
643         "type": "object",
644         "required": [
645             "bic",
646             "email",
647             "iban",
648             "id",
649             "name"
650         ],
651         "properties": {
652             "bankName": {
653                 "type": "string"
654             },
655             "bic": {
656                 "type": "string"
657             },
658             "creationDate": {
659                 "type": "string",
660                 "format": "date-time"
661             },
662             "email": {
663                 "type": "string"
664             },
665             "iban": {
666                 "type": "string"
667             },
668             "id": {
669                 "type": "integer",
670                 "format": "int64"

```

```

671         },
672         "name": {
673             "type": "string"
674         }
675     },
676     "title": "Beneficiary"
677 },
678 "LoginVM": {
679     "type": "object",
680     "required": [
681         "barcode",
682         "deviceToken",
683         "deviceType",
684         "minimumViableToken",
685         "version"
686     ],
687     "properties": {
688         "barcode": {
689             "type": "string",
690             "minLength": 10,
691             "maxLength": 10
692         },
693         "deviceName": {
694             "type": "string"
695         },
696         "deviceToken": {
697             "type": "string"
698         },
699         "deviceType": {
700             "type": "string",
701             "enum": [
702                 "IOS",
703                 "BLACKBERRY",
704                 "ANDROID",
705                 "WINDOWS_PHONE",
706                 "OSX",
707                 "WINDOWS_8",
708                 "AMAZON",
709                 "SAFARI",
710                 "CHROME",
711                 "FIREFOX",
712                 "IE11"
713             ]
714         },
715         "minimumViableToken": {
716             "type": "string"
717         },
718         "version": {
719             "type": "string"
720         }
721     },
722     "title": "LoginVM"
723 },
724 "ResolvedAuthenticationChallengeVM": {
725     "type": "object",
726     "properties": {
727         "token": {

```

```

728         "type": "string"
729     },
730 },
731     "title": "ResolvedAuthenticationChallengeVM"
732 },
733     "ResolvePasswordChallengeVM": {
734         "type": "object",
735         "properties": {
736             "password": {
737                 "type": "string"
738             },
739             "token": {
740                 "type": "string"
741             }
742         },
743         "title": "ResolvePasswordChallengeVM"
744     },
745     "ResolveVerificationCodeChallengeVM": {
746         "type": "object",
747         "properties": {
748             "code": {
749                 "type": "string"
750             },
751             "token": {
752                 "type": "string"
753             }
754         },
755         "title": "ResolveVerificationCodeChallengeVM"
756     },
757     "SendSmsCodeVM": {
758         "type": "object",
759         "properties": {
760             "resendChallenge": {
761                 "type": "boolean"
762             },
763             "token": {
764                 "type": "string"
765             }
766         },
767         "title": "SendSmsCodeVM"
768     },
769     "SendSmsCodeResponseVM": {
770         "type": "object",
771         "properties": {
772             "codeLength": {
773                 "type": "integer",
774                 "format": "int32"
775             },
776             "recipient": {
777                 "type": "string"
778             },
779             "token": {
780                 "type": "string"
781             }
782         },
783         "title": "SendSmsCodeResponseVM"
784     },

```

```
785     "TransferSubmissionResultVM": {
786         "type": "object",
787         "required": [
788             "status"
789         ],
790         "properties": {
791             "status": {
792                 "type": "string",
793                 "enum": [
794                     "SENT",
795                     "INTERNAL_VALIDATION_NEEDED"
796                 ]
797             }
798         },
799         "title": "TransferSubmissionResultVM"
800     },
801     "TransferVM": {
802         "type": "object",
803         "required": [
804             "amount"
805         ],
806         "properties": {
807             "amount": {
808                 "type": "number"
809             },
810             "beneficiaryId": {
811                 "type": "string"
812             },
813             "reason": {
814                 "type": "string"
815             }
816         },
817         "title": "TransferVM"
818     }
819 }
820 }
```