

Version

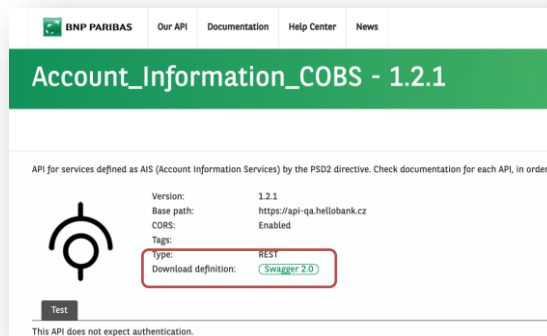
Version	Date	Author	Modifications
0.1	2018-19-21	Michal Bureš	Initialization
1.0	2018-12-17	Michal Bureš	Final version
1.1	2020-02-26	Michal Bureš	Updated with respect to COBS v3.1
1.2	2020-08-06	Sven Macke	New customer-wide transaction history

BNP Paribas PF PSD2 Services

BNP Paribas PF provides the following open-banking APIs:

- TPP management
- Authentication service (oAuth 2.0)
- Balance check for card-issuing providers (CIS)
- Account information service (AIS)
- Payment initiation service (PIS)
- Standing orders service

For SandBox setup and testing, we do recommend to download to use Swagger 2.0



TPP Management

TPP Management API can be used for registration of TPP to the Bank.

```
{
  "swagger": "2.0",
  "info": {
    "title": "TPP Management API",
    "description": "API for managing Third Party Payment Providers. Check documentation for each API, in order to find the security pre-requisites",
    "version": "1.7.0"
  },
  "schemes": [
    "http"
  ],
  "produces": [
    "application/json"
  ]
}
```

```

],
"basePath": "/tpp-management/api/v1/tpps",
"paths": {
  "/": {
    "get": {
      "summary": "Retrieves the list of active TPPs registered with the bank.",
      "description": "Retrieves the list of active TPPs registered with the bank \n",
      "consumes": [
        "application/json"
      ],
      "parameters": [
        {
          "$ref": "#/parameters/Accept-Language"
        }
      ],
      "tags": [
        "tpps"
      ],
      "responses": {
        "200": {
          "description": "Retrieved.",
          "schema": {
            "$ref": "#/definitions/TPPSummaries"
          }
        },
        "400": {
          "description": "Invalid request. A mandatory field is missing in the request, or
it is in an unsuitable / invalid format.",
          "schema": {
            "$ref": "#/definitions/Errors"
          }
        },
        "401": {
          "description": "The client is not unauthorized to execute the request.",
          "schema": {
            "$ref": "#/definitions/Errors"
          }
        },
        "403": {
          "description": "Insufficient authorization to use the required scope.",
          "schema": {
            "$ref": "#/definitions/Errors"
          }
        },
        "500": {
          "description": "Server error.",
          "schema": {
            "$ref": "#/definitions/Errors"
          }
        }
      }
    }
  },
 ("/{tppId}": {
    "get": {
      "summary": "Get TPP registration details.",
      "description": "Get the details of a TPP, optionally you can provide a valid
certificate and receive more information about this Tpp. \n",
      "consumes": [
        "application/json"
      ],
      "parameters": [
        {
          "$ref": "#/parameters/User-User-Agent"
        },
        {
          "$ref": "#/parameters/User-IP-Address"
        },
        {
          "$ref": "#/parameters/User-Geo-Location"
        },
        {
          "$ref": "#/parameters/Accept-Language"
        },
        {
          "$ref": "#/parameters/Action-ID"
        }
      ]
    }
  }
}

```

```

    },
    {
      "$ref": "#/parameters/X-Request-ID"
    },
    {
      "$ref": "#/parameters/Date"
    },
    {
      "$ref": "#/parameters/User-Involved"
    },
    {
      "in": "path",
      "name": "tppId",
      "type": "string",
      "required": true,
      "description": "The id of the tpp"
    }
  ],
  "tags": [
    "tpps"
  ],
  "responses": {
    "200": {
      "description": "Retrieved.",
      "schema": {
        "$ref": "#/definitions/TPPPublicProfile"
      }
    },
    "400": {
      "description": "Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "401": {
      "description": "The client is not unauthenticated to execute the request.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "403": {
      "description": "Insufficient authorization to use the required scope.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "500": {
      "description": "Server error.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    }
  }
}
},
"parameters": {
  "User-User-Agent": {
    "name": "User-User-Agent",
    "in": "header",
    "type": "string",
    "description": "Max 200 characters. Contains information about the end-user web browser. The format should be the same as the standard HTTP parameter User-Agent."
  },
  "User-IP-Address": {
    "name": "User-IP-Address",
    "in": "header",
    "type": "string",
    "format": "IPv4",
    "description": "Max 50 characters. Identifies the API's end-user IP address. The TPP should fill this value if the IP address can be detected. The address can be in IPv4 or IPv6 format."
  },
  "User-Geo-Location": {
    "name": "User-Geo-Location",

```

```

    "in": "header",
    "type": "string",
    "description": "Max 100 characters. End-user GPS coordinates (if the coordinates can
be reliably detected). Required format: GEO:Latitude;Longitude based on [RFC2426] in signed
degrees format (DDD.dddd) example: GEO:50.089354;14.400031"
  },
  "Accept-Language": {
    "name": "Accept-Language",
    "in": "header",
    "type": "string",
    "description": "The locale of the customer"
  },
  "Action-ID": {
    "name": "Action-ID",
    "in": "header",
    "type": "string",
    "format": "uuid",
    "description": "ID of the transaction as determined by the initiating party."
  },
  "X-Request-ID": {
    "name": "X-Request-ID",
    "in": "header",
    "type": "string",
    "description": "Max 60 characters. Unique identifier for each request specified by TPP.
It is recommended to use a UUID format with a length of 36 to 60 characters."
  },
  "Date": {
    "name": "Date",
    "in": "header",
    "type": "string",
    "description": "Each request of a transaction contains the date and time at which the
message was originated. In timestamp format."
  },
  "User-Involved": {
    "name": "User-Involved",
    "in": "header",
    "type": "boolean",
    "description": "The flag identifies if the end user is online and the request was made
based on his activity."
  }
},
"definitions": {
  "Role": {
    "type": "string",
    "description": "Role of the TPP.",
    "enum": [
      "PSP_AI",
      "PSP_IC",
      "PSP_PI",
      "PSP_AS"
    ]
  },
  "TPPPublicProfile": {
    "type": "object",
    "required": [
      "redirect_uris",
      "scopes"
    ],
    "properties": {
      "clientId": {
        "type": "string",
        "description": "The client_id assigned by the application. This ID starts the
authentication process and communication is decoded in the replacement of code and
refresh_token. (tppId in our case)"
      },
      "applicationType": {
        "type": "string",
        "description": "Type of application which uses this account (web or native). Will
not be used for CZ. This field will be ignored if coming on the request."
      },
      "redirectUris": {
        "type": "array",
        "description": "Enumeration of URL where the authentication flow is redirected at
the end. The authorization request must contain just one of these registered URIs in the
exact format.",
        "items": {

```

```

        "type": "string"
    }
},
"clientName": {
    "type": "string",
    "description": "Name of the client application. Will not be used in CZ. This field
will be ignored if comming on the request."
},
"contact": {
    "type": "string",
    "description": "E-mail as a contact to the responsible person on the part of the
client application."
},
"scopes": {
    "type": "array",
    "description": "The field of applications of required scopes. On the registration,
scopes are validated against the content of the certificate used.",
    "items": {
        "$ref": "#/definitions/Role"
    }
}
},
"TPPSummaries": {
    "type": "array",
    "description": "The list of TPPs.",
    "items": {
        "$ref": "#/definitions/TPPPublicSummary"
    }
},
"TPPPublicSummary": {
    "type": "object",
    "required": [
        "redirect_uris",
        "scopes"
    ],
    "properties": {
        "id": {
            "type": "string",
            "description": "The ID of the TPP."
        },
        "name": {
            "type": "string",
            "description": "The TPP name."
        },
        "nca": {
            "$ref": "#/definitions/NCAInformation"
        },
        "scopes": {
            "type": "array",
            "description": "The field of applications of required scopes. On the registration,
scopes are validated against the content of the certificate used.",
            "items": {
                "$ref": "#/definitions/Role"
            }
        }
    }
},
"NCAInformation": {
    "type": "object",
    "description": "General information about the TPP NCA details.",
    "required": [
        "name",
        "id"
    ],
    "properties": {
        "id": {
            "type": "string",
            "description": "The nca id of the TPP. To be retrieved from the TPP certificate."
        },
        "name": {
            "type": "string",
            "description": "The nca name of the TPP. To be retrieved from the TPP certificate."
        },
        "country": {
            "type": "string",

```

```

        "description": "The nca country of the TPP. To be retrieved from the TPP
certificate."
    },
    "authorizationId": {
        "type": "string",
        "description": "The nca authorization id of the TPP. To be retrieved from the TPP
certificate."
    }
},
"Links": {
    "type": "object",
    "properties": {
        "first": {
            "type": "string",
            "example": "api/v1/firstPage/",
            "description": "Navigation link for paginated account reports."
        },
        "last": {
            "type": "string",
            "example": "api/v1/lastPage/",
            "description": "Navigation link for paginated account reports."
        },
        "next": {
            "type": "string",
            "example": "api/v1/nextPage/",
            "description": "Navigation link for paginated account reports."
        },
        "previous": {
            "type": "string",
            "example": "api/v1/previousPage/",
            "description": "Navigation link for paginated account reports."
        },
        "self": {
            "type": "string",
            "example": "api/v1/payments/sepa-credit-transfers/1234-wertiq-983",
            "description": "Self: The link to the payment initiation resource created by the
request itself. This link can be used later to retrieve the transaction status of the payment
initiation."
        }
    },
    "title": "Links",
    "description": "Links "
},
"Errors": {
    "type": "array",
    "description": "The list of errors thrown.",
    "items": {
        "$ref": "#/definitions/Error"
    }
},
"Error": {
    "type": "object",
    "properties": {
        "error": {
            "type": "string",
            "description": "Contains the specific error code."
        },
        "scope": {
            "type": "string",
            "description": "It determines the JSON path of the request element that caused the
error status."
        },
        "message": {
            "type": "string",
            "description": "Optional text description. It is not intended for interpretation
to the end user but, for example, to enrich the error log. "
        },
        "parameters": {
            "description": "The field of additional elements specific for the given error code.
These parameters are always specified in the description of a specific error code.",
            "type": "object"
        }
    }
}
}
}

```

```
}
```

oAuth 2.0

oAuth2 is used to securely link the client with TPP.

```
{
  "swagger": "2.0",
  "info": {
    "description": "API operations for oauth2.",
    "version": "1.0",
    "title": "Oauth2 API"
  },
  "basePath": "/oauth2/api/v1/oauth2",
  "schemes": ["http"],
  "paths": {
    "/revoke": {
      "post": {
        "tags": ["API for managing OAUTH public operations"],
        "summary": "Oauth2 token revoking endpoint",
        "description": "This method also has a sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
        "operationId": "revokeTokenUsingPOST",
        "consumes": ["application/x-www-form-urlencoded"],
        "produces": ["application/json"],
        "parameters": [
          {
            "$ref": "#/parameters/User-User-Agent"
          },
          {
            "$ref": "#/parameters/User-IP-Address"
          },
          {
            "$ref": "#/parameters/User-Geo-Location"
          },
          {
            "$ref": "#/parameters/Accept-Language"
          },
          {
            "$ref": "#/parameters/Action-ID"
          },
          {
            "$ref": "#/parameters/X-Request-ID"
          },
          {
            "$ref": "#/parameters/Date"
          },
          {
            "$ref": "#/parameters/User-Involved"
          },
          {
            "in": "formData",
            "name": "token",
            "required": true,
            "type": "string"
          }
        ],
        "responses": {
          "200": {
            "description": "OK"
          },
          "201": {
            "description": "Created"
          },
          "400": {
            "description": "Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.",
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/ErrorMessage"
              }
            }
          }
        }
      }
    }
  }
}
```

```

    "401": {
      "description": "Unauthorized"
    },
    "403": {
      "description": "Access to the requested resource is not granted or is not possible
for the user.",
      "schema": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/ErrorMessage"
        }
      }
    },
    "404": {
      "description": "Not Found"
    },
    "500": {
      "description": "Server error.",
      "schema": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/ErrorMessage"
        }
      }
    }
  }
},
"/token": {
  "post": {
    "tags": ["API for managing OAUTH public operations"],
    "summary": "The Oauth2 refresh token endpoint.",
    "description": "You are expected to provide a valid certificate when retrieving a
token. This method also has a sandbox service which you can test. ![alt
text](images/logos/sandbox.png \"Sandbox API\")",
    "operationId": "refreshTokenUsingPOST",
    "consumes": ["application/x-www-form-urlencoded"],
    "produces": ["application/json"],
    "parameters": [
      {
        "$ref": "#/parameters/User-User-Agent"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "$ref": "#/parameters/Action-ID"
      },
      {
        "$ref": "#/parameters/X-Request-ID"
      },
      {
        "$ref": "#/parameters/Date"
      },
      {
        "$ref": "#/parameters/User-Involved"
      },
      {
        "name": "code",
        "in": "formData",
        "description": "The authorization code returned from the authentication flow.
example `zb328465af42c674d0b1c1128c37c2g'",
        "required": false,
        "type": "string"
      },
      {
        "name": "clientId",
        "in": "formData",
        "description": "Id of the TPP application. example `PSDES-BDE-3DFD2'",
        "required": false,

```



```

    "type": "string"
  },
  {
    "name": "clientSecret",
    "in": "formData",
    "description": "Client secret-password/token issued by the bank IDP for the TPP
application. for HelloBank this will be always NOT_PROVIDED. example 'NOT_PROVIDED'",
    "required": false,
    "type": "string"
  },
  {
    "name": "redirectUri",
    "in": "formData",
    "description": "Redirect URL identical to the URL deliveered in the authentication
request. example 'https://tppdomain'",
    "required": false,
    "type": "string"
  },
  {
    "name": "grantType",
    "in": "formData",
    "description": "According to the existing definition/practice, OAuth2 will be the
value of authorization_code if the code is replaces by the refresh_token, or refresh_token
in case tha access_token is replaces by the refresh_token. example 'authorization_code'",
    "required": true,
    "type": "string"
  },
  {
    "name": "refreshToken",
    "in": "formData",
    "description": "The refresh token used for retrieving a new access_token. example
'be9eef9b0af42c674d0b1c1128c37c2g'",
    "required": false,
    "type": "string"
  }
],
"responses": {
  "200": {
    "description": "The request executed successfully",
    "schema": {
      "$ref": "#/definitions/TokenRefreshResponse"
    }
  },
  "201": {
    "description": "Created"
  },
  "400": {
    "description": "Invalid request. A mandatory field is missing in the request, or
it is in an unsuitable / invalid format.",
    "schema": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/ErrorMessage"
      }
    }
  },
  "401": {
    "description": "Unauthorized"
  },
  "403": {
    "description": "Insufficient authorization to use the required scope.",
    "schema": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/ErrorMessage"
      }
    }
  },
  "404": {
    "description": "Not Found"
  },
  "500": {
    "description": "Server error.",
    "schema": {
      "type": "array",
      "items": {

```

```

        "$ref": "#/definitions/ErrorMessage"
    }
}
}
}
}
},
"parameters": {
  "User-User-Agent": {
    "name": "User-User-Agent",
    "in": "header",
    "type": "string",
    "description": "Max 200 characters. Contains information about the end-user web browser.
The format should be the same as the standard HTTP parameter User-Agent."
  },
  "User-IP-Address": {
    "name": "User-IP-Address",
    "in": "header",
    "type": "string",
    "format": "IPV4",
    "description": "Max 50 characters. Identifies the API's end-user IP address. The TPP
should fill this value if the IP address can be detected. The address can be in IPv4 or IPv6
format."
  },
  "User-Geo-Location": {
    "name": "User-Geo-Location",
    "in": "header",
    "type": "string",
    "description": "Max 100 characters. End-user GPS coordinates (if the coordinates can
be reliably detected). Required format: GEO:Latitude;Longitude based on [RFC2426] in signed
degrees format (DDD.dddd) example: GEO:50.089354;14.400031"
  },
  "Accept-Language": {
    "name": "Accept-Language",
    "in": "header",
    "type": "string",
    "description": "The locale of the customer"
  },
  "Action-ID": {
    "name": "Action-ID",
    "in": "header",
    "type": "string",
    "format": "uuid",
    "description": "ID of the transaction as determined by the initiating party."
  },
  "X-Request-ID": {
    "name": "X-Request-ID",
    "in": "header",
    "type": "string",
    "description": "Max 60 characters. Unique identifier for each request specified by TPP.
It is recommended to use a UUID format with a length of 36 to 60 characters."
  },
  "Date": {
    "name": "Date",
    "in": "header",
    "type": "string",
    "description": "Each request of a transaction contains the date and time at which the
message was originated. In timestamp format."
  },
  "User-Involved": {
    "name": "User-Involved",
    "in": "header",
    "type": "boolean",
    "description": "The flag identifies if the end user is online and the request was made
based on his activity."
  }
},
"definitions": {
  "ErrorMessage": {
    "type": "object",
    "properties": {
      "error": {
        "type": "string",
        "description": "Contains the specific error code"
      }
    }
  }
},

```

```

"message": {
  "type": "string",
  "description": "Optional text description. It is not intended for interpretation
to the end user but, for example, to enrich the error log."
},
"parameters": {
  "type": "array",
  "description": "Optional key value pairs with further description",
  "items": {
    "type": "object"
  }
},
"scope": {
  "type": "string",
  "description": "It determines the JSON path of the request element that caused the
error status."
},
"description": "Custom error message"
},
"TokenCreationResponse": {
  "type": "object",
  "required": ["expiresIn", "tokenScope", "tokenType"],
  "properties": {
    "access_token": {
      "type": "string",
      "description": "The client access token."
    },
    "code": {
      "type": "string",
      "description": "The authorization code to be used by TPP to retrieve access and
refresh tokens."
    },
    "expiresIn": {
      "type": "integer",
      "format": "int64",
      "description": "Token expiration time in seconds."
    },
    "tokenScope": {
      "type": "string",
      "description": "the scope of the token."
    },
    "tokenType": {
      "type": "string",
      "description": "The token type."
    }
  },
  "description": "\tGeneral information about the token and the holder."
},
"TokenRefreshResponse": {
  "type": "object",
  "properties": {
    "accessToken": {
      "type": "string",
      "description": "A short-term (in some cases, a one-time) token that may be generated
again using the refresh_token. The token serves for the authorization of the request at API."
    },
    "acr": {
      "type": "string",
      "description": "Verification level. It may have the values between 0 and 4. Default
3. The value „0“ corresponds to nonSCA. example: '0'"
    },
    "expiresIn": {
      "type": "integer",
      "format": "int32",
      "description": "A time left before the expiration of access_token - in seconds.
example: 1000"
    },
    "refreshToken": {
      "type": "string",
      "description": "A long-term token issued based on replacement by the one-time code."
    },
    "tokenType": {
      "type": "string",
      "description": "The token type. example: 'Bearer'"
    }
  }
}

```

```

    },
    "description": "General information about the token and the holder"
  }
}

```

Balance Check for Card-Issuing Payment Service Providers (CIS)

The service allows card-issuing payment service providers to check disposable balance of a linked account.

```

{
  "swagger": "2.0",
  "info": {
    "title": "CIS API",
    "description": "API for the acquisition of information on sufficient funds of the payer (client) at a specific financial institution.",
    "version": "1.1.0"
  },
  "host": "svcs-psd2-ops-qa-cz-pf-cortalconsors.is.echonet:443",
  "schemes": [
    "https"
  ],
  "basePath": "/api/v1/accounts/balanceCheck",
  "produces": [
    "application/json"
  ],
  "paths": {
    "/": {
      "parameters": [
        {
          "$ref": "#/parameters/User-User-Agent"
        },
        {
          "$ref": "#/parameters/User-IP-Address"
        },
        {
          "$ref": "#/parameters/User-Geo-Location"
        },
        {
          "$ref": "#/parameters/Accept-Language"
        },
        {
          "$ref": "#/parameters/Action-ID"
        },
        {
          "$ref": "#/parameters/X-Request-ID"
        },
        {
          "$ref": "#/parameters/Date"
        },
        {
          "$ref": "#/parameters/User-Involved"
        }
      ],
      "post": {
        "summary": "Checking the balance for a specific payment account of the account payer.",
        "description": "Checking the balance for a specific payment account of the account payer. You are expected to provide a valid certificate. This method also has a sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")\n",
        "consumes": [
          "application/json"
        ],
        "parameters": [
          {
            "in": "body",
            "name": "balanceInquiry",
            "required": true,
            "schema": {
              "$ref": "#/definitions/CardPaymentDetails"
            }
          }
        ]
      }
    }
  }
}

```

```

    ],
    "tags": [
      "accounts/balanceCheck"
    ],
    "responses": {
      "200": {
        "description": "Success",
        "schema": {
          "$ref": "#/definitions/CISSummary"
        }
      },
      "400": {
        "description": "Invalid request. A mandatory field is missing in the request, or
it is in an unsuitable / invalid format.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "401": {
        "description": "The client is not unauthorized to execute the request.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "403": {
        "description": "Insufficient authorization to use the required scope.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "500": {
        "description": "Server error.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      }
    }
  },
  "parameters": {
    "User-User-Agent": {
      "name": "User-User-Agent",
      "in": "header",
      "type": "string",
      "description": "Max 200 characters. Contains information about the end-user web browser.
The format should be the same as the standard HTTP parameter User-Agent."
    },
    "User-IP-Address": {
      "name": "User-IP-Address",
      "in": "header",
      "type": "string",
      "format": "IPv4",
      "description": "Max 50 characters. Identifies the API's end-user IP address. The TPP
should fill this value if the IP address can be detected. The address can be in IPv4 or IPv6
format."
    },
    "User-Geo-Location": {
      "name": "User-Geo-Location",
      "in": "header",
      "type": "string",
      "description": "Max 100 characters. End-user GPS coordinates (if the coordinates can
be reliably detected). Required format: GEO:Latitude;Longitude based on [RFC2426] in signed
degrees format (DDD.dddd) example: GEO:50.089354;14.400031"
    },
    "Accept-Language": {
      "name": "Accept-Language",
      "in": "header",
      "type": "string",
      "description": "The locale of the customer"
    },
    "Action-ID": {
      "name": "Action-ID",
      "in": "header",
      "type": "string",
      "format": "uuid",

```

```

    "description": "ID of the transaction as determined by the initiating party."
  },
  "X-Request-ID": {
    "name": "X-Request-ID",
    "in": "header",
    "type": "string",
    "description": "Max 60 characters. Unique identifier for each request specified by TPP.
It is recommended to use a UUID format with a length of 36 to 60 characters."
  },
  "Date": {
    "name": "Date",
    "in": "header",
    "type": "string",
    "description": "Each request of a transaction contains the date and time at which the
message was originated. In timestamp format."
  },
  "User-Involved": {
    "name": "User-Involved",
    "in": "header",
    "type": "boolean",
    "description": "The flag identifies if the end user is online and the request was made
based on his activity."
  }
},
"definitions": {
  "CISSummary": {
    "type": "object",
    "required": [
      "responseIdentification",
      "exchangeIdentification",
      "responseCode"
    ],
    "properties": {
      "responseIdentification": {
        "type": "string",
        "description": "Unique identification of response to query for Balance Check (from
ASPSP)."
      },
      "exchangeIdentification": {
        "type": "string",
        "description": "Repeated identification of a payment transaction (query for Balance
Check) from the issuer of the card to which the request for Balance Check linked to the
account."
      },
      "responseCode": {
        "$ref": "#/definitions/ResponseCode"
      },
      "links": {
        "$ref": "#/definitions/Links"
      }
    }
  },
  "CardPaymentDetails": {
    "type": "object",
    "required": [
      "exchangeIdentification",
      "debtorAccount",
      "transactionDetails"
    ],
    "properties": {
      "exchangeIdentification": {
        "type": "string",
        "description": "Clear query identification."
      },
      "card": {
        "$ref": "#/definitions/CardDetails"
      },
      "debtorAccount": {
        "$ref": "#/definitions/DebtorAccount"
      },
      "authenticationMethod": {
        "$ref": "#/definitions/AuthenticationCode"
      },
      "merchant": {
        "$ref": "#/definitions/MerchantDetails"
      }
    }
  },

```

```

    "transactionDetails": {
      "$ref": "#/definitions/TransactionDetails"
    }
  },
  "CardDetails": {
    "type": "object",
    "required": [
      "maskedPan"
    ],
    "properties": {
      "cardholderName": {
        "type": "string",
        "description": "Card holder name."
      },
      "maskedPan": {
        "type": "string",
        "description": "Masked card number."
      }
    }
  },
  "DebtorAccount": {
    "type": "object",
    "required": [
      "identification"
    ],
    "properties": {
      "identification": {
        "$ref": "#/definitions/AccountIdentification"
      }
    }
  },
  "AccountIdentification": {
    "type": "object",
    "description": "Payer account identification.",
    "properties": {
      "iban": {
        "type": "string",
        "pattern": "[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}"
      },
      "other": {
        "$ref": "#/definitions/OtherIdentification"
      }
    }
  },
  "OtherIdentification": {
    "type": "object",
    "description": "Other identification for account (BBAN, contract number).",
    "properties": {
      "identification": {
        "type": "string"
      }
    }
  },
  "MerchantDetails": {
    "type": "object",
    "required": [
      "identification",
      "shortName",
      "commonName",
      "merchantCategoryCode"
    ],
    "properties": {
      "identification": {
        "type": "string",
        "description": "Merchant identification"
      },
      "type": {
        "$ref": "#/definitions/MerchantType"
      },
      "shortName": {
        "type": "string",
        "description": "Merchant name."
      },
      "commonName": {
        "type": "string",

```

```

    "description": "Merchant name as stated in the payment receipt."
  },
  "address": {
    "type": "string",
    "description": "Merchant address."
  },
  "countryCode": {
    "type": "string",
    "description": "ISO 3166 - Merchant country code."
  },
  "merchantCategoryCode": {
    "type": "string",
    "description": "ISO 18245 - Merchant code following the transaction type."
  }
},
"TransactionDetails": {
  "type": "object",
  "required": [
    "currency",
    "totalAmount"
  ],
  "properties": {
    "currency": {
      "type": "string",
      "description": "ISO 4217 - Balance query currency."
    },
    "totalAmount": {
      "type": "number",
      "description": "Balance query amount."
    }
  }
},
"ResponseCode": {
  "type": "string",
  "enum": [
    "APPR",
    "DECL"
  ],
  "description": "* APPR - Enough funds on this account.\n* DECL - Uninsufficient funds on
this account."
},
"AuthenticationCode": {
  "type": "string",
  "enum": [
    "NPIN",
    "PPSG",
    "PSWD",
    "SCRT",
    "SCNL",
    "SNCT",
    "CPSG",
    "ADDB",
    "BIOM",
    "CDHI",
    "CRYP",
    "CSCV",
    "PSVE",
    "CSEC",
    "ADDS",
    "TOKP"
  ]
},
"MerchantType": {
  "type": "string",
  "enum": [
    "OPOI",
    "MERC",
    "ACCP",
    "ITAG"
  ],
  "description": "* OPOI - OriginatingPOI - Point Of Interaction initiating the card
payment transaction.\n* MERC - Merchant - Merchant providing goods and service in the card
payment transaction\n* ACCP - Acceptor - Card acceptor, party accepting the card and
presenting transaction data to the acquirer.\n* ITAG - IntermediaryAgent - Party acting on
behalf of other parties to process or forward data to other parties."
}

```



```

    },
    "Errors": {
      "type": "object",
      "properties": {
        "errors": {
          "type": "array",
          "description": "The list of errors thrown.",
          "items": {
            "$ref": "#/definitions/Error"
          }
        }
      }
    },
    "required": [
      "errors"
    ]
  },
  "Error": {
    "type": "object",
    "properties": {
      "error": {
        "type": "string",
        "description": "Contains the specific error code."
      },
      "scope": {
        "type": "string",
        "description": "It determines the JSON path of the request element that caused the error status."
      },
      "message": {
        "type": "string",
        "description": "Optional text description. It is not intended for interpretation to the end user but, for example, to enrich the error log. "
      },
      "parameters": {
        "description": "The field of additional elements specific for the given error code. These parameters are always specified in the description of a specific error code.",
        "type": "object"
      }
    }
  },
  "Links": {
    "type": "object",
    "properties": {
      "payment": {
        "type": "string",
        "description": "POST a new payment. Requires the authorization of user/client as part of the API call"
      }
    }
  }
}

```

Account Information Service (AIS)

This service provides basic information about client's accounts.

```

{
  "swagger": "2.0",
  "info": {
    "title": "AIS API",
    "description": "API for services defined as AIS (Account Information Services) by the PSD2 directive.",
    "version": "1.2.8"
  },
  "host": "svcs-psd2-ops-qa-cz-pf-cortalconsors.is.echonet:443",
  "schemes": [
    "https"
  ],
  "basePath": "/api/v1/my/accounts",
  "produces": [
    "application/json"
  ],
  "paths": {

```

```

"/": {
  "get": {
    "summary": "Obtain the list of the client's accounts",
    "description": "List of client accounts. Each account contains a unique id usable for
URI referencing, e.g. for the account detail. The pagination is not applied. You are expected
to provide a valid certificate and optional an OAuth2 token. This method also has a sandbox
service which you can test. ![alt text] (images/logos/sandbox.png \"Sandbox API\")",
    "tags": [
      "accounts"
    ],
    "parameters": [
      {
        "$ref": "#/parameters/User-User-Agent"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "name": "Authorization",
        "in": "header",
        "type": "string",
        "description": "The parameter is used to pass an access token of the
authenticated user together with its type."
      },
      {
        "$ref": "#/parameters/Action-ID"
      },
      {
        "$ref": "#/parameters/X-Request-ID"
      },
      {
        "$ref": "#/parameters/Date"
      },
      {
        "$ref": "#/parameters/User-Involved"
      },
      {
        "$ref": "#/parameters/sort"
      },
      {
        "$ref": "#/parameters/order"
      }
    ],
    "responses": {
      "200": {
        "description": "A list of accounts (sorted if so requested). The pagination is
not used.",
        "schema": {
          "allOf": [
            {
              "$ref": "#/definitions/PageMetadata"
            },
            {
              "$ref": "#/definitions/Accounts"
            }
          ]
        }
      },
      "400": {
        "description": "Invalid request. A mandatory parameter is missing or it is in an
unsuitable / invalid format.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "401": {
        "description": "The client is not authorized to execute the request.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      }
    }
  }
}

```

```

    },
    "404": {
      "description": "The requested page does not exist",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "500": {
      "description": "Server error.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    }
  }
},
"/{id}/balance": {
  "get": {
    "summary": "Obtain the balance of the specified client's account",
    "description": "Obtain the balance of the specific client account according to the reference id account. You are expected to provide a valid certificate and a valid oAuth2 token. This method also has a sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
    "tags": [
      "accounts"
    ],
    "parameters": [
      {
        "$ref": "#/parameters/User-User-Agent"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "name": "Authorization",
        "in": "header",
        "required": true,
        "type": "string",
        "description": "The parameter is used to pass an access token of the authenticated user together with its type."
      },
      {
        "$ref": "#/parameters/Action-ID"
      },
      {
        "$ref": "#/parameters/X-Request-ID"
      },
      {
        "$ref": "#/parameters/Date"
      },
      {
        "$ref": "#/parameters/User-Involved"
      },
      {
        "in": "path",
        "name": "id",
        "type": "string",
        "required": true,
        "description": "System identifier of client account"
      },
      {
        "in": "query",
        "name": "currency",
        "type": "string",
        "description": "Required account currency (ISO 4217 currency code) for multi-currency accounts"
      }
    ],
    "responses": {
      "200": {

```

```

        "description": "The current balance(s) of the client's account",
        "schema": {
            "$ref": "#/definitions/Balances"
        }
    },
    "400": {
        "description": "Invalid request. A mandatory parameter is missing or it is in an
unsuitable / invalid format.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "401": {
        "description": "The client is not authorized to execute the request.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "404": {
        "description": "The requested account does not exist",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "500": {
        "description": "Server error.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    }
}
},
"/{id}/transactions": {
    "get": {
        "summary": "Obtain the transactions on the specified client's account",
        "description": "Obtain a paged list of transactions of a selected client account. You
are expected to provide a valid certificate and a valid oAuth2 token. This method also has a
sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
        "tags": [
            "accounts"
        ],
        "parameters": [
            {
                "$ref": "#/parameters/User-User-Agent"
            },
            {
                "$ref": "#/parameters/User-IP-Address"
            },
            {
                "$ref": "#/parameters/User-Geo-Location"
            },
            {
                "$ref": "#/parameters/Accept-Language"
            },
            {
                "name": "Authorization",
                "in": "header",
                "required": true,
                "type": "string",
                "description": "The parameter is used to pass an access token of the
authenticated user together with its type."
            },
            {
                "$ref": "#/parameters/Action-ID"
            },
            {
                "$ref": "#/parameters/X-Request-ID"
            },
            {
                "$ref": "#/parameters/Date"
            },
            {
                "$ref": "#/parameters/User-Involved"
            },
            {

```

```

        "$ref": "#/parameters/Accept"
    },
    {
        "in": "path",
        "name": "id",
        "type": "string",
        "required": true,
        "description": "System identifier of client account"
    },
    {
        "in": "query",
        "name": "fromDate",
        "type": "string",
        "description": "Date and time of the start of required transaction history"
    },
    {
        "in": "query",
        "name": "toDate",
        "type": "string",
        "description": "Date and time of the end of required transaction history
(inclusive)"
    },
    {
        "$ref": "#/parameters/size"
    },
    {
        "$ref": "#/parameters/page"
    },
    {
        "$ref": "#/parameters/sort"
    },
    {
        "$ref": "#/parameters/order"
    }
],
"responses": {
    "200": {
        "description": "Paged list of transactions of the specified client account",
        "schema": {
            "allOf": [
                {
                    "$ref": "#/definitions/PageMetadata"
                },
                {
                    "$ref": "#/definitions/Transactions"
                }
            ]
        }
    },
    "400": {
        "description": "Invalid request. A mandatory parameter is missing or it is in an
unsuitable / invalid format.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "401": {
        "description": "The client is not authorized to execute the request.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "404": {
        "description": "The requested account or page does not exist",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "500": {
        "description": "Server error.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    }
}
}
}

```

```

    },
    "/transactions": {
      "get": {
        "summary": "Obtain the list of the client's transactions for all accounts",
        "description": "Obtain the list of the client's transactions for all accounts. You are expected to provide a valid certificate and a valid oAuth2 token. This method also has a sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
        "tags": [
          "accounts"
        ],
        "parameters": [
          {
            "$ref": "#/parameters/User-User-Agent"
          },
          {
            "$ref": "#/parameters/User-IP-Address"
          },
          {
            "$ref": "#/parameters/User-Geo-Location"
          },
          {
            "$ref": "#/parameters/Accept-Language"
          },
          {
            "name": "Authorization",
            "in": "header",
            "required": true,
            "type": "string",
            "description": "The parameter is used to pass an access token of the authenticated user together with its type."
          },
          {
            "$ref": "#/parameters/Action-ID"
          },
          {
            "$ref": "#/parameters/X-Request-ID"
          },
          {
            "$ref": "#/parameters/Date"
          },
          {
            "$ref": "#/parameters/User-Involved"
          },
          {
            "$ref": "#/parameters/Accept"
          },
          {
            "in": "query",
            "name": "fromDate",
            "type": "string",
            "description": "Date and time of the start of required transaction history"
          },
          {
            "in": "query",
            "name": "toDate",
            "type": "string",
            "description": "Date and time of the end of required transaction history (inclusive)"
          },
          {
            "in": "query",
            "name": "status",
            "required": false,
            "type": "array",
            "uniqueItems": true,
            "items": {
              "type": "string",
              "enum": [
                "BOOK",
                "PDNG",
                "RJTD"
              ]
            }
          },
          {
            "description": "A comma separated list of transaction statuses for which transaction history should be returned"
          }
        ]
      }
    }
  },

```

```

    {
      "in": "query",
      "name": "paymentType",
      "required": false,
      "type": "string",
      "enum": [
        "ALL",
        "INCOMING",
        "OUTGOING"
      ],
      "description": "The payment type of required transaction history"
    },
    {
      "in": "query",
      "name": "accountNumber",
      "required": false,
      "type": "array",
      "uniqueItems": true,
      "items": {
        "type": "string"
      },
      "description": "A comma separated list of account numbers for which transaction
history should be returned"
    },
    {
      "in": "query",
      "name": "transactionType",
      "type": "array",
      "uniqueItems": true,
      "items": {
        "type": "string",
        "enum": [
          "DOMESTIC",
          "STANDING_ORDER",
          "SEPA",
          "DIRECT_DEBIT",
          "SIPO",
          "DEPOSITS_AND WITHDRAWALS",
          "CARD_PAYMENT",
          "STICKER_PAYMENT",
          "REPAYMENTS",
          "OTHER"
        ]
      },
      "description": "A comma separated list of transaction types for which transaction
history should be returned"
    },
    {
      "in": "query",
      "name": "fromAmount",
      "type": "number",
      "description": "The minimum amount of required transaction history (inclusive)"
    },
    {
      "in": "query",
      "name": "toAmount",
      "type": "number",
      "description": "The maximum amount of required transaction history (inclusive)"
    },
    {
      "in": "query",
      "name": "cardNumber",
      "required": false,
      "type": "array",
      "uniqueItems": true,
      "items": {
        "type": "string"
      },
      "description": "A comma separated list of masked pan numbers for which transaction
history should be returned"
    },
    {
      "$ref": "#/parameters/size"
    },
    {
      "$ref": "#/parameters/page"
    }
  ]
}

```

```

    },
    {
      "$ref": "#/parameters/sort"
    },
    {
      "$ref": "#/parameters/order"
    }
  ],
  "responses": {
    "200": {
      "description": "Paged list of transactions of the specified client",
      "schema": {
        "allOf": [
          {
            "$ref": "#/definitions/PageMetadata"
          },
          {
            "$ref": "#/definitions/ExtendedTransactions"
          }
        ]
      }
    },
    "400": {
      "description": "Invalid request. A mandatory parameter is missing or it is in an
unsuitable / invalid format.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "401": {
      "description": "The client is not authorized to execute the request.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "404": {
      "description": "The requested account or page does not exist",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "500": {
      "description": "Server error.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    }
  }
},
"/{id}": {
  "get": {
    "summary": "Obtain the details of a specific client's account",
    "description": "Obtain the details of the account. You are expected to provide a
valid certificate and a valid oAuth2 token. This method also has a sandbox service which you
can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
    "tags": [
      "accounts"
    ],
    "parameters": [
      {
        "$ref": "#/parameters/User-User-Agent"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "name": "Authorization",
        "in": "header",
        "required": true,

```



```

        "type": "string",
        "description": "The parameter is used to pass an access token of the
authenticated user together with its type."
    },
    {
        "$ref": "#/parameters/Action-ID"
    },
    {
        "$ref": "#/parameters/X-Request-ID"
    },
    {
        "$ref": "#/parameters/Date"
    },
    {
        "$ref": "#/parameters/User-Involved"
    },
    {
        "$ref": "#/parameters/Accept"
    },
    {
        "in": "path",
        "name": "id",
        "type": "string",
        "required": true,
        "description": "System identifier of client account"
    }
],
"responses": {
    "200": {
        "description": "The account details",
        "schema": {
            "allOf": [
                {
                    "$ref": "#/definitions/AccountDetails"
                }
            ]
        }
    },
    "400": {
        "description": "Invalid request. A mandatory parameter is missing or it is in an
unsuitable / invalid format.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "401": {
        "description": "The client is not authorized to execute the request.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "404": {
        "description": "The requested page does not exist",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "500": {
        "description": "Server error.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    }
}
},
"parameters": {
    "User-User-Agent": {
        "name": "User-User-Agent",
        "in": "header",
        "type": "string",
        "description": "Max 200 characters. Contains information about the end-user web
browser. The format should be the same as the standard HTTP parameter User-Agent."
    },
    "User-IP-Address": {

```

```

    "name": "User-IP-Address",
    "in": "header",
    "type": "string",
    "format": "IPv4",
    "description": "Max 50 characters. Identifies the API's end-user IP address. The TPP
should fill this value if the IP address can be detected. The address can be in IPv4 or IPv6
format."
  },
  "User-Geo-Location": {
    "name": "User-Geo-Location",
    "in": "header",
    "type": "string",
    "description": "Max 100 characters. End-user GPS coordinates (if the coordinates can
be reliably detected). Required format: GEO:Latitude;Longitude based on [RFC2426] in signed
degrees format (DDD.dddd) example: GEO:50.089354;14.400031"
  },
  "Accept-Language": {
    "name": "Accept-Language",
    "in": "header",
    "type": "string",
    "description": "The locale of the customer"
  },
  "Action-ID": {
    "name": "Action-ID",
    "in": "header",
    "type": "string",
    "format": "uuid",
    "description": "ID of the transaction as determined by the initiating party."
  },
  "X-Request-ID": {
    "name": "X-Request-ID",
    "in": "header",
    "type": "string",
    "description": "Max 60 characters. Unique identifier for each request specified by
TPP. It is recommended to use a UUID format with a length of 36 to 60 characters."
  },
  "Date": {
    "name": "Date",
    "in": "header",
    "type": "string",
    "description": "Each request of a transaction contains the date and time at which the
message was originated. In timestamp format."
  },
  "User-Involved": {
    "name": "User-Involved",
    "in": "header",
    "type": "boolean",
    "description": "The flag identifies if the end user is online and the request was
made based on his activity."
  },
  "Accept": {
    "name": "Accept",
    "in": "header",
    "type": "string",
    "description": "The TPP can indicate the formats of account reports supported together
with a prioritisation following the http header definition.."
  },
  "size": {
    "name": "size",
    "in": "query",
    "type": "integer",
    "description": "It defines the required number of records on the page. If the parameter
is not specified, the API returns the entire set."
  },
  "page": {
    "name": "page",
    "in": "query",
    "type": "integer",
    "description": "It defines the required page number. Pages are numbered from 0. If the
parameter is not specified, the API returns the first (zero) page.",
    "default": 0
  },
  "sort": {
    "name": "sort",
    "in": "query",
    "type": "array",

```

```

    "items": {
      "type": "string"
    },
    "description": "It defines a list of fields for sorting. The individual fields in the
list are separated by a comma "
  },
  "order": {
    "name": "order",
    "in": "query",
    "type": "array",
    "items": {
      "type": "string",
      "enum": [
        "asc",
        "desc"
      ]
    }
  },
  "description": "It defines the sorting direction for each field listed in the sort
parameter. The individual sorting directions are separated by commas precisely as the field
names in the sort parameter and are listed in the same order as the fields listed in the sort
parameter."
}
},
"definitions": {
  "PageMetadata": {
    "type": "object",
    "properties": {
      "pageNumber": {
        "type": "integer",
        "description": "Number of current page"
      },
      "pageCount": {
        "type": "integer",
        "description": "Total number of query pages"
      },
      "nextPage": {
        "type": "integer",
        "description": "The number of the next page. If the current page is at the same
time the last page of the list, then this parameter is not listed or is given with a null
value"
      },
      "pageSize": {
        "type": "integer",
        "description": "Number of entries per page. This parameter can match the required
size value from the query, except for the cases when it is the last page, or when the requested
page range exceeded the maximum limit defined for the particular API resource"
      },
      "totalCount": {
        "type": "integer",
        "description": "Optional data about the total number of entries for all pages. If
this value cannot be precisely determined, it is not stated"
      }
    }
  },
  "Accounts": {
    "type": "object",
    "properties": {
      "accounts": {
        "type": "array",
        "description": "Set of client accounts",
        "items": {
          "$ref": "#/definitions/Account"
        }
      },
      "_links": {
        "$ref": "#/definitions/AccountsListLinks"
      }
    }
  },
  "required": [
    "accounts"
  ]
},
"Account": {
  "type": "object",
  "properties": {
    "id": {

```

```

        "type": "string",
        "description": "API Payment account identifier"
    },
    "identification": {
        "$ref": "#/definitions/Identification"
    },
    "currency": {
        "$ref": "#/definitions/ISOCurrencyCode"
    },
    "servicer": {
        "$ref": "#/definitions/ServiceProviderIdentification"
    },
    "nameI18N": {
        "type": "string",
        "description": "Account name / label"
    },
    "productI18N": {
        "type": "string",
        "description": "Product name"
    },
    "_links": {
        "$ref": "#/definitions/AccountLinks"
    },
    "ownersNames": {
        "type": "array",
        "description": "Account holder name",
        "items": {
            "type": "string"
        }
    }
},
"required": [
    "id",
    "identification",
    "servicer"
]
},
"AccountInformation": {
    "type": "object",
    "properties": {
        "id": {
            "type": "string",
            "description": "API Payment account identifier"
        },
        "identification": {
            "$ref": "#/definitions/Identification"
        }
    }
},
"ExtendedAccountInformation": {
    "type": "object",
    "properties": {
        "id": {
            "type": "string",
            "description": "API Payment account identifier"
        },
        "identification": {
            "$ref": "#/definitions/Identification"
        },
        "accountState": {
            "type": "string",
            "enum": [
                "ACTIVE",
                "HISTORICAL"
            ],
            "description": "The status of the account."
        },
        "servicer": {
            "$ref": "#/definitions/ServiceProviderIdentification"
        },
        "nameI18N": {
            "type": "string",
            "description": "Account name / label"
        },
        "productI18N": {
            "type": "string",

```

```

    "description": "Product name"
  }
},
"AccountDetails": {
  "type": "object",
  "properties": {
    "account": {
      "$ref": "#/definitions/Account"
    },
    "interest": {
      "description": "The interest rate of the account",
      "$ref": "#/definitions/Interest"
    },
    "installmentAmount": {
      "type": "object",
      "description": "Value/amount of account balance according to balance type. Currency corresponds to the account currency to which the statement is generated",
      "properties": {
        "value": {
          "type": "number",
          "description": "Amount of account balance"
        },
        "currency": {
          "$ref": "#/definitions/ISOCurrencyCode"
        }
      },
      "required": [
        "value",
        "currency"
      ]
    },
    "installmentDueDate": {
      "type": "object",
      "description": "Date (and time) of balance of the account for which the statement is generated",
      "properties": {
        "dateTime": {
          "$ref": "#/definitions/ISODateTime"
        }
      },
      "required": [
        "dateTime"
      ]
    },
    "variableSymbol": {
      "type": "string"
    },
    "insuranceType": {
      "type": "string"
    },
    "rewardType": {
      "type": "string"
    },
    "wayOfRepayment": {
      "type": "string",
      "enum": [
        "PAYMENTORDER",
        "DIRECTDEBIT",
        "POSTORDER",
        "SIPO",
        "POSTALMANDATE",
        "CASH",
        "POSTALDEPOSIT"
      ]
    },
    "repaymentAccount": {
      "type": "string"
    },
    "gracePeriod": {
      "type": "string"
    },
    "accountState": {
      "type": "string",
      "enum": [
        "ACTIVE",

```

```

    "HISTORICAL"
  ],
  "description": "The status of the account."
},
"productState": {
  "type": "string",
  "description": "The state in which the product is."
},
"installmentGraceAmount": {
  "type": "number",
  "description": "the grace amount for the product."
},
"statementDistribution": {
  "type": "string",
  "enum": [
    "ELECTRONIC",
    "PAPER"
  ],
  "description": "The distribution method for the account statements."
}
},
"Balances": {
  "type": "object",
  "properties": {
    "balances": {
      "type": "array",
      "description": "Set of balances of client payment account",
      "items": {
        "$ref": "#/definitions/Balance"
      }
    },
    "account": {
      "$ref": "#/definitions/AccountInformation"
    },
    "_links": {
      "$ref": "#/definitions/BalancesListLinks"
    }
  },
  "required": [
    "balances"
  ]
},
"Balance": {
  "type": "object",
  "description": "Account balance information",
  "properties": {
    "type": {
      "$ref": "#/definitions/BalanceType"
    },
    "creditLine": {
      "$ref": "#/definitions/OverdraftInformation"
    },
    "amount": {
      "type": "object",
      "description": "Value/amount of account balance according to balance type. Currency corresponds to the account currency to which the statement is generated",
      "properties": {
        "value": {
          "type": "number",
          "description": "Amount of account balance"
        },
        "currency": {
          "$ref": "#/definitions/ISOCurrencyCode"
        }
      }
    },
    "required": [
      "value",
      "currency"
    ]
  },
  "creditDebitIndicator": {
    "type": "string",
    "enum": [
      "DBIT",
      "CRDT"
    ]
  }
}

```

```

    ],
    "description": "Indication of whether the balance in the account for which the
statement is generated is positive or negative"
  },
  "date": {
    "type": "object",
    "description": "Date (and time) of balance of the account for which the statement
is generated",
    "properties": {
      "dateTime": {
        "$ref": "#/definitions/ISODatetime"
      }
    }
  },
  "required": [
    "dateTime"
  ]
},
"required": [
  "type",
  "amount",
  "creditDebitIndicator",
  "date"
]
},
"Transactions": {
  "type": "object",
  "properties": {
    "transactions": {
      "type": "array",
      "description": "Set of transactions of client payment account",
      "items": {
        "$ref": "#/definitions/Transaction"
      }
    },
    "account": {
      "$ref": "#/definitions/AccountInformation"
    },
    "_links": {
      "$ref": "#/definitions/TransactionsListLinks"
    }
  },
  "required": [
    "transactions"
  ]
},
"ExtendedTransactions": {
  "type": "object",
  "properties": {
    "extendedTransactions": {
      "type": "array",
      "description": "Set of transactions of client",
      "items": {
        "$ref": "#/definitions/ExtendedTransaction"
      }
    },
    "_links": {
      "$ref": "#/definitions/ExtendedTransactionsListLinks"
    }
  },
  "required": [
    "extendedTransactions"
  ]
},
"ExtendedTransaction": {
  "type": "object",
  "properties": {
    "transaction": {
      "$ref": "#/definitions/Transaction"
    },
    "account": {
      "$ref": "#/definitions/ExtendedAccountInformation"
    },
    "ownAccount": {
      "type": "boolean",

```

```

    "description": "Indicates whether the transaction is performed between client's own
accounts"
  },
  "required": [
    "transaction",
    "account"
  ]
},
"Transaction": {
  "type": "object",
  "properties": {
    "entryReference": {
      "type": "string",
      "maxLength": 200,
      "description": "Identification of the payment assigned by the bank"
    },
    "amount": {
      "type": "object",
      "description": "Payment amount in the account currency for which payment history
is generated",
      "properties": {
        "value": {
          "type": "number",
          "description": "Payment amount"
        },
        "currency": {
          "$ref": "#/definitions/ISOCurrencyCode"
        }
      }
    },
    "required": [
      "value",
      "currency"
    ]
  },
  "creditDebitIndicator": {
    "type": "string",
    "enum": [
      "DBIT",
      "CRDT"
    ],
    "description": "Indication of whether it is a debit payment or credit payment of
the account"
  },
  "reversalIndicator": {
    "type": "boolean",
    "description": "Indication of whether it is a cancellation"
  },
  "status": {
    "type": "string",
    "enum": [
      "BOOK",
      "PDNG",
      "RJTD"
    ],
    "description": "Item status (debited or credited payments) of the account from the
point of view of the bank. The statement will present only posted items, with the constant
BOOK, or blocked items, with the constant PDNG"
  },
  "bookingDate": {
    "type": "object",
    "description": "Date (or date and time) of processing/posting of payment by bank
in the ISO 8601 format",
    "properties": {
      "date": {
        "$ref": "#/definitions/ISODateTime"
      }
    }
  },
  "valueDate": {
    "type": "object",
    "description": "Due date (or date and time) of payment requested by client in the
ISO 8601 format",
    "properties": {
      "date": {
        "$ref": "#/definitions/ISODateTime"
      }
    }
  }
}

```



```

    }
  },
  "bankTransactionCode": {
    "$ref": "#/definitions/BankTransactionCode"
  },
  "cancellationReason": {
    "type": "string",
    "enum": [
      "NO_COVER",
      "NO_MANDATE",
      "CANCELED_BY_BANK",
      "CANCELED_BY_CLIENT"
    ],
    "description": "The reason for failure when the payment has failed"
  },
  "entryDetails": {
    "type": "object",
    "description": "Turnover details. The level repeats only for the given item",
    "properties": {
      "transactionDetails": {
        "$ref": "#/definitions/TransactionDetails"
      }
    }
  },
  "required": [
    "amount",
    "creditDebitIndicator",
    "status",
    "bookingDate",
    "valueDate",
    "bankTransactionCode"
  ]
},
"TransactionDetails": {
  "type": "object",
  "description": "Payment details. The level repeats only for the given item",
  "properties": {
    "references": {
      "$ref": "#/definitions/PaymentReferences"
    },
    "amountDetails": {
      "$ref": "#/definitions/PaymentAmountDetails"
    },
    "charges": {
      "type": "object",
      "description": "Information on fees (only for foreign payments)",
      "properties": {
        "bearer": {
          "type": "string",
          "enum": [
            "OUR",
            "SHA",
            "BEN"
          ],
          "description": "Indicates whose debit the payments will be charged to"
        }
      }
    },
    "relatedParties": {
      "allOf": [
        {
          "$ref": "#/definitions/TransactionParties"
        },
        {
          "type": "object",
          "description": "Information on the payer, the payer's account and the original payer and payee, the payee's account and the final payee in the payment"
        }
      ]
    },
    "relatedAgents": {
      "allOf": [
        {
          "$ref": "#/definitions/TransactionAgents"
        }
      ]
    }
  }
}

```

```

    },
    {
      "type": "object",
      "description": "Information on the payer bank and the payee bank in the payment"
    }
  ]
},
"purpose": {
  "type": "object",
  "description": "Payment purpose",
  "properties": {
    "code": {
      "type": "string",
      "description": "Payment purpose as stated in the external list of payment
purpose codes"
    },
    "proprietary": {
      "type": "string",
      "maxLength": 35,
      "description": "Payment purpose in free text format, as entered by payer"
    }
  }
},
"remittanceInformation": {
  "type": "object",
  "description": "Information that allows match (i.e. pairing, reconciliation) of
payments with which the payment should be settled, such as commercial invoices in the
receivables system",
  "properties": {
    "unstructured": {
      "type": "string",
      "maxLength": 140,
      "description": "Additional payment information filled in in the payment as
unstructured information. For example note for the recipient or payer's reference."
    },
    "structured": {
      "$ref": "#/definitions/StructuredRemittanceInformation",
      "example": "\"reference\": [ \"VS:123456\", \"SS:879213546\", \"KS:456789\"]"
    }
  }
},
"additionalTransactionInformation": {
  "type": "string",
  "maxLength": 500,
  "description": "Additional information provided by the bank. Only stated if it is
included in the record. Additional info for SEPA DD is entered here (e.g., Creditor Identifier,
Payment scheme, SEPA collection order, etc.)"
},
"noteForMe": {
  "type": "string",
  "maxLength": 140,
  "description": "Note for me"
}
},
"BalanceType": {
  "type": "object",
  "description": "Determines the balance type to which information on account balance
applies",
  "properties": {
    "codeOrProprietary": {
      "type": "object",
      "properties": {
        "code": {
          "type": "string",
          "enum": [
            "PRCD",
            "OPBD",
            "CLBD",
            "ITBD",
            "CLAV",
            "FWAV"
          ],
          "description": "Type of balance\n * PRCD - Balance of the account at the
previously closed account reporting period. The opening booked balance for the new period has
to be equal to this balance.\n * OPBD - Book balance of the account at the beginning of the

```

```

account reporting period. It always equals the closing book balance from the previous
report.\n * CLBD - Balance of the account at the end of the pre-agreed account reporting
period. It is the sum of the opening booked balance at the beginning of the period and all
entries booked to the account during the pre-agreed account reporting period.\n * ITBD -
Balance calculated in the course of the account service's business day, at the time
specified, and subject to further changes during the business day. The interim balance is
calculated on the basis of booked credit and debit items during the calculation time/period
specified. This balance type is only present in CAMT.052.\n * CLAV - Closing balance of amount
of money that is at the disposal of the account owner on the date specified.\n * FWAV -
Forward available balance of money that is at the disposal of the account owner on the date
specified.\n"
    },
    "proprietary": {
      "type": "string",
      "enum": [
        "SAVG",
        "CRDT"
      ]
    },
    "description": "Type of balance\n * CRDT - Credit framework of revolving
account\n * SAVG - Savings goal\n"
  }
},
"required": [
  "codeOrProprietary"
]
},
"OverdraftInformation": {
  "type": "object",
  "description": "Information on agreed permitted debit/overdraft",
  "properties": {
    "included": {
      "type": "boolean",
      "description": "Indicates if overdraft is possible"
    },
    "amount": {
      "type": "object",
      "description": "Amount of agreed permitted debit",
      "properties": {
        "value": {
          "type": "number",
          "description": "Amount of agreed permitted debit"
        },
        "currency": {
          "$ref": "#/definitions/ISOCurrencyCode"
        }
      }
    }
  }
},
"BankTransactionCode": {
  "type": "object",
  "description": "The code of banking transaction according to the code list of the Czech
Banking Association assigned to a specific payment",
  "properties": {
    "proprietary": {
      "type": "object",
      "description": "Each bank uses an own code list to identify payments which is,
however, based on the 1st to 3rd level of transaction code list according to the CBA Standard
for camt.053",
      "properties": {
        "code": {
          "type": "string",
          "maxLength": 35,
          "description": "Proprietary code based on the CBA Standard for camt.053"
        },
        "issuer": {
          "type": "string",
          "maxLength": 35,
          "description": "Identification of the code list issuer for banking transactions
which acquires the value of the Czech Banking Association"
        }
      }
    }
  }
}
}
}

```

```

    },
    "PaymentReferences": {
      "type": "object",
      "description": "A set of references unambiguously identifying the payment",
      "properties": {
        "messageIdentification": {
          "type": "string",
          "maxLength": 35,
          "description": "Assumed payment identification entered by the client on its
initiation or order of the payment in the payment history"
        },
        "accountServicerReference": {
          "type": "string",
          "maxLength": 35,
          "description": "A banking reference assigned to the payment, e.g., on initiation
through direct banking services"
        },
        "paymentInformationIdentification": {
          "type": "string",
          "maxLength": 35,
          "description": "Another/next banking reference assigned to the payment assigned by
the bank, the card sequence number or the specific symbol"
        },
        "instructionIdentification": {
          "type": "string",
          "maxLength": 35,
          "description": "Payment identification entered by a third party, or a constant
symbol "
        },
        "endToEndIdentification": {
          "type": "string",
          "maxLength": 35,
          "description": "A unique identifier specified by the client initiating the payment
that serves for unambiguous payment identification and is passed in the unchanged state
throughout the payment chain, or a variable symbol. If the value is NOTPROVIDED, then is not
displayed to customer in transaction history and therefore it will be not send as part of
response to AIS Transaction history request."
        },
        "mandateIdentification": {
          "type": "string",
          "maxLength": 35,
          "description": "Unique Mandate Reference stated for the given SEPA collection
(mandatory for SEPA collections)"
        },
        "chequeNumber": {
          "type": "string",
          "maxLength": 35,
          "description": "For cheque transactions the cheque number; for card transactions a
masked or incomplete card number."
        },
        "clearingSystemReference": {
          "type": "string",
          "maxLength": 35,
          "description": "A bank-defined code list value identifying the type of payment or
payment type used. For card transactions, card association identification may be indicated."
        },
        "connectionNumber": {
          "type": "string",
          "maxLength": 35,
          "description": "Unique identification number for the SIPO payer."
        },
        "ownCardName": {
          "type": "string",
          "maxLength": 35,
          "description": "The name of the card."
        }
      }
    },
    "PaymentAmountDetails": {
      "type": "object",
      "description": "Details of the payment amount, especially if it is a conversion payment
or cashback",
      "properties": {
        "instructedAmount": {
          "type": "object",

```

```

    "description": "Amount and currency of payment in the currency that was required
by the client (e.g. for intra-bank payments, the payer account currency and the payment
currency if the client required the payment in the payer account currency)",
    "properties": {
      "amount": {
        "$ref": "#/definitions/AmountWithCurrency"
      }
    },
    "required": [
      "amount"
    ]
  },
  "transactionAmount": {
    "type": "object",
    "description": "The total amount and currency of payment for cumulated payments and
Cashback",
    "properties": {
      "amount": {
        "$ref": "#/definitions/AmountWithCurrency"
      }
    },
    "required": [
      "amount"
    ]
  },
  "counterValueAmount": {
    "type": "object",
    "description": "The amount and currency of payment in the client account currency
after conversion of amount that was required by the client for transfer",
    "properties": {
      "amount": {
        "$ref": "#/definitions/AmountWithCurrency"
      },
      "currencyExchange": {
        "$ref": "#/definitions/CurrencyExchange"
      }
    },
    "required": [
      "amount"
    ]
  },
  "proprietaryAmount": {
    "type": "object",
    "description": "Amount of cash withdrawal through the Cashback service (only the
part of total payment, to which the cash amount withdrawn falls within the Cashback service)",
    "properties": {
      "type": {
        "type": "string",
        "description": "The constant \"CASHBACK\""
      },
      "amount": {
        "$ref": "#/definitions/AmountWithCurrency"
      }
    },
    "required": [
      "amount"
    ]
  }
}
},
"TransactionParties": {
  "type": "object",
  "description": "Information on the parties involved in a transaction",
  "properties": {
    "debtor": {
      "type": "object",
      "description": "Information on the payer. It is according to the payment direction,
and filled in in the case of counterparty",
      "properties": {
        "name": {
          "type": "string",
          "maxLength": 140,
          "description": "Payer name"
        },
        "postalAddress": {
          "allOf": [

```

```

        {
            "$ref": "#/definitions/PostalAddress"
        },
        {
            "type": "object",
            "description": "Payer postal address"
        }
    ],
    "identification": {
        "type": "object",
        "description": "Payer identification (for SEPA either
organisationIdentification or privateIdentification must be present)",
        "properties": {
            "organisationIdentification": {
                "$ref": "#/definitions/OrganizationIdentification"
            },
            "privateIdentification": {
                "$ref": "#/definitions/NaturalPersonIdentification"
            }
        }
    }
},
"debtorAccount": {
    "type": "object",
    "description": "Information on the account payer. It is according to the payment
direction, and filled in in the case of counterparty. Revolving account uses non-standard
format of account number which is not transformable to IBAN format. Therefore Debtor account
will be stated in other identification message element.Iban message element will be filled
with IBAN format of account. Other identification message element will be used for account
number in BBAN format in case of savings, current and basic accounts. If the value of BBAN
is not available message element other identification will be filled with no value",
    "properties": {
        "identification": {
            "$ref": "#/definitions/AccountIdentification"
        },
        "currency": {
            "$ref": "#/definitions/ISOCurrencyCode"
        },
        "name": {
            "type": "string",
            "maxLength": 70,
            "description": "Payer account name"
        }
    },
    "required": [
        "identification"
    ]
},
"ultimateDebtor": {
    "type": "object",
    "description": "Information on the original/actual payer (only for SEPA). It is
according to the payment direction, and filled in in the case of counterparty",
    "properties": {
        "name": {
            "type": "string",
            "maxLength": 140,
            "description": "Name of the original/actual payer"
        },
        "postalAddress": {
            "allOf": [
                {
                    "$ref": "#/definitions/PostalAddress"
                },
                {
                    "type": "object",
                    "description": "Postal address of the original payer"
                }
            ]
        },
        "identification": {
            "type": "object",
            "description": "Original payer identification (for SEPA either
organisationIdentification or privateIdentification must be present)",
            "properties": {

```

```

        "organisationIdentification": {
            "$ref": "#/definitions/OrganizationIdentification"
        },
        "privateIdentification": {
            "$ref": "#/definitions/NaturalPersonIdentification"
        }
    }
}
},
"creditor": {
    "type": "object",
    "description": "Information on the payee. It is according to the payment direction,
and filled in in the case of counterparty",
    "properties": {
        "name": {
            "type": "string",
            "maxLength": 140,
            "description": "Payee name"
        },
        "postalAddress": {
            "allOf": [
                {
                    "$ref": "#/definitions/PostalAddress"
                },
                {
                    "type": "object",
                    "description": "Payee postal address"
                }
            ]
        },
        "identification": {
            "type": "object",
            "description": "Payee identification",
            "properties": {
                "organisationIdentification": {
                    "$ref": "#/definitions/OrganizationIdentification"
                },
                "privateIdentification": {
                    "$ref": "#/definitions/NaturalPersonIdentification"
                }
            }
        }
    }
},
"creditorAccount": {
    "type": "object",
    "description": "Information on the payee account. It is according to the payment
direction, and filled in in the case of counterparty",
    "properties": {
        "identification": {
            "$ref": "#/definitions/AccountIdentification"
        },
        "currency": {
            "$ref": "#/definitions/ISOCurrencyCode"
        },
        "name": {
            "type": "string",
            "maxLength": 70,
            "description": "Payee account name"
        }
    }
},
"required": [
    "identification"
]
},
"ultimateCreditor": {
    "type": "object",
    "description": "Information on the final payee (only for SEPA). It is according to
the payment direction, and filled in in the case of counterparty",
    "properties": {
        "name": {
            "type": "string",
            "maxLength": 140,
            "description": "Final payee name"
        }
    }
},

```

```

        "postalAddress": {
            "allOf": [
                {
                    "$ref": "#/definitions/PostalAddress"
                },
                {
                    "type": "object",
                    "description": "Final payee postal address"
                }
            ]
        },
        "identification": {
            "type": "object",
            "description": "Final payee identification",
            "properties": {
                "organisationIdentification": {
                    "$ref": "#/definitions/OrganizationIdentification"
                },
                "privateIdentification": {
                    "$ref": "#/definitions/NaturalPersonIdentification"
                }
            }
        }
    },
    "proprietary": {
        "type": "object",
        "description": "For more detailed identification of which ATM a card transaction
occurred on",
        "properties": {
            "type": {
                "type": "string",
                "maxLength": 35,
                "description": "Indicates if own or external ATM was used"
            },
            "party": {
                "type": "object",
                "description": "For identification of the ATM name/owner",
                "properties": {
                    "name": {
                        "type": "string",
                        "maxLength": 140,
                        "description": "For identification of ATM name"
                    },
                    "postalAddress": {
                        "allOf": [
                            {
                                "$ref": "#/definitions/PostalAddress"
                            },
                            {
                                "type": "object",
                                "description": "For identification of ATM location"
                            }
                        ]
                    }
                }
            }
        }
    },
    "transactionSource": {
        "type": "object",
        "properties": {
            "name": {
                "type": "string",
                "maxLength": 70,
                "description": "Name of TPP via the client initiated the payment transaction"
            }
        }
    }
},
"TransactionAgents": {
    "type": "object",
    "description": "Information on the agents facilitating a transaction",
    "properties": {
        "debtorAgent": {

```



```

    "type": "object",
    "description": "Information on the payer bank. It is according to the payment
direction, and filled in in the case of counterparty",
    "properties": {
      "financialInstitutionIdentification": {
        "$ref": "#/definitions/FinancialInstitutionIdentification"
      }
    },
    "required": [
      "financialInstitutionIdentification"
    ]
  },
  "creditorAgent": {
    "type": "object",
    "description": "Information on the payee bank. It is according to the payment
direction, and filled in in the case of counterparty",
    "properties": {
      "financialInstitutionIdentification": {
        "$ref": "#/definitions/FinancialInstitutionIdentification"
      }
    },
    "required": [
      "financialInstitutionIdentification"
    ]
  }
},
"StructuredRemittanceInformation": {
  "type": "object",
  "description": "Structured message for payee ÃÃ variable, specific, and constant
symbol. If the payment does not have a variable, specific or constant symbol, then the entire
Structured Remittance Information will be left empty",
  "properties": {
    "creditorReferenceInformation": {
      "type": "object",
      "description": "Reference information provided by the payee that allows
identification of underlying documents (e.g. invoice)",
      "properties": {
        "reference": {
          "type": "array",
          "description": "Information allowing to pass on information about the value of
a variable symbol, a specific symbol, or a constant symbol.",
          "items": {
            "type": "string"
          }
        }
      }
    }
  }
},
"OrganizationIdentification": {
  "type": "object",
  "description": "Identification information for an organization/legal entity",
  "properties": {
    "bicOrBei": {
      "$ref": "#/definitions/BIC"
    },
    "other": {
      "type": "object",
      "description": "Other identification as the organization/legal entity",
      "properties": {
        "identification": {
          "type": "string",
          "maxLength": 35,
          "description": "Other identification as the organization/legal entity in
unstructured form"
        },
        "schemeName": {
          "$ref": "#/definitions/IdentificationScheme"
        }
      },
      "issuer": {
        "type": "string",
        "maxLength": 35,
        "description": "Name of document issuer, i.e. the entity assigning the
identification"
      }
    }
  }
}

```

```

    },
    "required": [
      "identification"
    ]
  }
},
"NaturalPersonIdentification": {
  "type": "object",
  "description": "Identification information for a natural person",
  "properties": {
    "other": {
      "type": "object",
      "description": "Other identification as the natural person",
      "properties": {
        "identification": {
          "type": "string",
          "maxLength": 35,
          "description": "Other identification as the natural person in unstructured
form"
        },
        "schemeName": {
          "$ref": "#/definitions/IdentificationScheme"
        },
        "issuer": {
          "type": "string",
          "maxLength": 35,
          "description": "Name of document issuer, i.e. the entity assigning the
identification"
        }
      },
      "required": [
        "identification"
      ]
    }
  }
},
"FinancialInstitutionIdentification": {
  "type": "object",
  "description": "A unique and unambiguous financial institution identification assigned
by the international standard",
  "properties": {
    "bic": {
      "$ref": "#/definitions/BIC"
    },
    "clearingSystemMemberIdentification": {
      "type": "object",
      "description": "Identification of participant in the local clearing system (only
for foreign payments outside EEA)",
      "properties": {
        "clearingSystemIdentification": {
          "type": "object",
          "description": "Clearing system identification",
          "properties": {
            "code": {
              "type": "string",
              "description": "A code identifying the local clearing system as stated in
the external code list"
            },
            "proprietary": {
              "type": "string",
              "maxLength": 35,
              "description": "Identification of the local clearing system, in free text
format"
            }
          }
        },
        "memberIdentification": {
          "type": "string",
          "maxLength": 35,
          "description": "Participant's identification code or bank's clearing code"
        }
      }
    },
    "name": {
      "type": "string",

```

```

        "maxLength": 140,
        "description": "Name under which the party is known, and which is normally used to
identify that party"
    },
    "postalAddress": {
        "allOf": [
            {
                "$ref": "#/definitions/PostalAddress"
            },
            {
                "type": "object",
                "description": "Postal address of the financial institution"
            }
        ]
    },
    "other": {
        "type": "object",
        "description": "Unique agent identification assigned to the institution using the
identification scheme (not used for SEPA)",
        "properties": {
            "identification": {
                "type": "string",
                "maxLength": 35,
                "description": "Other bank identification, mainly in the form of a local bank
code"
            }
        },
        "required": [
            "identification"
        ]
    }
},
"IdentificationScheme": {
    "type": "object",
    "description": "Code type for identification as an organization/legal entity or natural
person",
    "properties": {
        "code": {
            "type": "string",
            "description": "Code type for identification in the form of code according to the
ISO code list"
        },
        "proprietary": {
            "type": "string",
            "maxLength": 35,
            "description": "Code type for identification in free text format"
        }
    },
    "required": [
        "proprietary"
    ]
},
"PostalAddress": {
    "type": "object",
    "description": "A postal address",
    "properties": {
        "streetName": {
            "type": "string",
            "maxLength": 70,
            "description": "Street name"
        },
        "buildingNumber": {
            "type": "string",
            "maxLength": 16,
            "description": "Land registry no."
        },
        "postCode": {
            "type": "string",
            "maxLength": 16,
            "description": "Postcode"
        },
        "townName": {
            "type": "string",
            "maxLength": 35,
            "description": "Town name"
        }
    }
}

```

```

    },
    "country": {
      "$ref": "#/definitions/ISOCountryCode"
    },
    "addressLine": {
      "type": "string",
      "maxLength": 70,
      "description": "Information that localizes and identifies the specific address as
defined by postal services, presented in the free text format"
    }
  },
  "AccountIdentification": {
    "type": "object",
    "description": "Account identification (either IBAN or other, except for SEPA where
IBAN is mandatory).If the client use the Proxy_ID instead of IBAN as creditor account
identifier, the payment transaction will have in transaction history used Proxy_ID instead
of IBAN. Information about creditor's IBAN is not revealed to client.",
    "properties": {
      "iban": {
        "$ref": "#/definitions/IBAN"
      },
      "other": {
        "$ref": "#/definitions/OtherIdentification"
      },
      "contactDetails": {
        "$ref": "#/definitions/ContactDetails"
      }
    }
  },
  "Identification": {
    "type": "object",
    "description": "Account identification (either IBAN or other, except for SEPA where
IBAN is mandatory).If the client use the Proxy_ID instead of IBAN as creditor account
identifier, the payment transaction will have in transaction history used Proxy_ID instead
of IBAN.",
    "properties": {
      "iban": {
        "$ref": "#/definitions/IBAN"
      },
      "other": {
        "$ref": "#/definitions/OtherIdentification"
      }
    }
  },
  "IBAN": {
    "type": "string",
    "description": "Account number in the IBAN format which is defined by the international
standard ISO 13616.",
    "pattern": "[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}"
  },
  "OtherIdentification": {
    "type": "object",
    "description": "Other identification for account (BBAN, contract number).",
    "properties": {
      "identification": {
        "type": "string"
      }
    }
  },
  "ServiceProviderIdentification": {
    "type": "object",
    "description": "Bank / service provider identification.",
    "properties": {
      "bankCode": {
        "type": "string",
        "description": "National bank code (if any)"
      },
      "countryCode": {
        "$ref": "#/definitions/ISOCountryCode"
      },
      "bic": {
        "$ref": "#/definitions/BIC"
      }
    }
  }
},

```

```

"BIC": {
  "type": "string",
  "pattern": "[A-Z]{6}[A-Z0-9]{2}|[A-Z]{6}[A-Z0-9]{5}",
  "description": "ISO 9362 BIC/BEI"
},
"AmountWithCurrency": {
  "type": "object",
  "description": "Amount and currency of payment",
  "properties": {
    "value": {
      "type": "number",
      "description": "Payment amount"
    },
    "currency": {
      "$ref": "#/definitions/ISOCurrencyCode"
    }
  },
  "required": [
    "value",
    "currency"
  ]
},
"CurrencyExchange": {
  "type": "object",
  "description": "Information on currencies and exchange rates",
  "properties": {
    "sourceCurrency": {
      "$ref": "#/definitions/ISOCurrencyCode"
    },
    "targetCurrency": {
      "$ref": "#/definitions/ISOCurrencyCode"
    },
    "exchangeRate": {
      "type": "number",
      "format": "float",
      "description": "The exchange rate. In the case of cross conversion, only one rate
is filled in"
    }
  },
  "required": [
    "sourceCurrency",
    "exchangeRate"
  ]
},
"ContactDetails": {
  "type": "object",
  "description": "Information of the contact details",
  "properties": {
    "emailAddress": {
      "type": "string",
      "maxLength": 2048,
      "description": "Email address"
    },
    "mobileNumber": {
      "type": "string",
      "description": "European economic area mobile number"
    },
    "faxNumber": {
      "type": "string",
      "description": "Hungarian tax ID number"
    }
  }
},
"ISOCountryCode": {
  "type": "string",
  "description": "ISO 3166 alphanumeric 3 country code",
  "pattern": "[A-Z]{3}|[0-9]{3}"
},
"ISOCurrencyCode": {
  "type": "string",
  "description": "ISO 4217 currency code",
  "pattern": "[A-Z]{3}",
  "example": "CZK"
},
"ISODateTime": {
  "type": "string",

```

```

    "description": "Date (or date and time) in the ISO 8601 format"
  },
  "Errors": {
    "type": "object",
    "properties": {
      "errors": {
        "type": "array",
        "description": "The list of errors thrown.",
        "items": {
          "$ref": "#/definitions/Error"
        }
      }
    }
  },
  "required": [
    "errors"
  ]
},
"Error": {
  "type": "object",
  "properties": {
    "error": {
      "type": "string",
      "description": "Contains the specific error code."
    },
    "scope": {
      "type": "string",
      "description": "It determines the JSON path of the request element that caused the error status."
    },
    "message": {
      "type": "string",
      "description": "Optional text description. It is not intended for interpretation to the end user but, for example, to enrich the error log. "
    },
    "parameters": {
      "description": "The field of additional elements specific for the given error code. These parameters are always specified in the description of a specific error code.",
      "type": "object"
    }
  }
},
"AccountsListLinks": {
  "type": "object",
  "properties": {
    "self": {
      "type": "string",
      "example": "api/v1/my/accounts/",
      "description": "The link to the resource itself"
    },
    "first": {
      "type": "string",
      "example": "api/v1/my/accounts/",
      "description": "GET the first page of the account list"
    },
    "last": {
      "type": "string",
      "example": "api/v1/my/accounts/?page=3&size=10",
      "description": "GET the last page of the account list"
    },
    "next": {
      "type": "string",
      "example": "api/v1/my/accounts/?page=2&size=10",
      "description": "GET the next page of the account list"
    },
    "previous": {
      "type": "string",
      "example": "api/v1/my/accounts/?page=1&size=10",
      "description": "GET the previous page of the account list"
    }
  }
},
"AccountLinks": {
  "type": "object",
  "properties": {
    "balance": {
      "type": "string",

```

```

    "example": "api/v1/my/accounts/0123456789/balance",
    "description": "GET the account's balance(s) "
  },
  "transactions": {
    "type": "string",
    "example": "api/v1/my/accounts/0123456789/transactions",
    "description": "GET the list of transactions on the account"
  },
  "balanceCheck": {
    "type": "string",
    "example": "api/v1/my/payments/balanceCheck",
    "description": "POST an inquiry about availability of funds. Requires the
authorization of user/client as part of the API call"
  },
  "thirdPartyBalanceCheck": {
    "type": "string",
    "example": "api/v1/accounts/balanceCheck",
    "description": "POST an inquiry about availability of funds. Does not require the
authorization of user/client as part of the API call"
  },
  "payment": {
    "type": "string",
    "example": "api/v1/my/payments/",
    "description": "POST a new payment. Requires the authorization of user/client as
part of the API call"
  }
},
"BalancesListLinks": {
  "type": "object",
  "properties": {
    "self": {
      "type": "string",
      "example": "api/v1/my/accounts/0123456789/balance",
      "description": "The link to the resource itself"
    },
    "transactions": {
      "type": "string",
      "example": "api/v1/my/accounts/0123456789/transactions",
      "description": "GET the list of transactions on the account"
    },
    "accounts": {
      "type": "string",
      "example": "api/v1/my/accounts/",
      "description": "GET the list of accounts"
    },
    "payment": {
      "type": "string",
      "example": "api/v1/my/payments/",
      "description": "POST a new payment. Requires the authorization of user/client as
part of the API call"
    }
  }
},
"TransactionsListLinks": {
  "type": "object",
  "properties": {
    "self": {
      "type": "string",
      "example": "api/v1/my/accounts/0123456789/transactions",
      "description": "The link to the resource itself"
    },
    "first": {
      "type": "string",
      "example": "api/v1/my/accounts/0123456789/transactions",
      "description": "GET the first page of the account list"
    },
    "last": {
      "type": "string",
      "example": "api/v1/my/accounts/0123456789/transactions?page=3&size=10",
      "description": "GET the last page of the account list"
    },
    "next": {
      "type": "string",
      "example": "api/v1/my/accounts/0123456789/transactions?page=2&size=10",
      "description": "GET the next page of the account list"
    }
  }
}

```

```

    },
    "previous": {
      "type": "string",
      "example": "api/v1/my/accounts/0123456789/transactions?page=1&size=10",
      "description": "GET the previous page of the account list"
    },
    "balance": {
      "type": "string",
      "example": "api/v1/my/accounts/0123456789/balance",
      "description": "GET the account's balance(s)"
    },
    "balanceCheck": {
      "type": "string",
      "example": "api/v1/my/payments/balanceCheck",
      "description": "POST an inquiry about availability of funds. Requires the
authorization of user/client as part of the API call"
    },
    "thirdPartyBalanceCheck": {
      "type": "string",
      "example": "api/v1/accounts/balanceCheck",
      "description": "POST an inquiry about availability of funds. Does not require the
authorization of user/client as part of the API call"
    },
    "payment": {
      "type": "string",
      "example": "api/v1/my/payments/",
      "description": "POST a new payment. Requires the authorization of user/client as
part of the API call"
    },
    "accounts": {
      "type": "string",
      "example": "api/v1/my/accounts/",
      "description": "GET the list of accounts"
    }
  }
},
"ExtendedTransactionsListLinks": {
  "type": "object",
  "properties": {
    "self": {
      "type": "string",
      "example": "api/v1/my/accounts/transactions/",
      "description": "The link to the resource itself"
    },
    "first": {
      "type": "string",
      "example": "api/v1/my/accounts/transactions/",
      "description": "GET the first page of the transactions list"
    },
    "last": {
      "type": "string",
      "example": "api/v1/my/accounts/transactions/?page=3&size=10",
      "description": "GET the last page of the transactions list"
    },
    "next": {
      "type": "string",
      "example": "api/v1/my/accounts/transactions/?page=2&size=10",
      "description": "GET the next page of the transactions list"
    },
    "previous": {
      "type": "string",
      "example": "api/v1/my/accounts/transactions/?page=1&size=10",
      "description": "GET the previous page of the transactions list"
    }
  }
},
"Interest": {
  "type": "object",
  "properties": {
    "rate": {
      "$ref": "#/definitions/Rate"
    }
  }
},
"Rate": {
  "type": "object",

```



```

    "properties": {
      "type": {
        "$ref": "#/definitions/RateType"
      }
    },
    "RateType": {
      "type": "object",
      "properties": {
        "percentage": {
          "type": "number"
        }
      }
    }
  }
}

```

Payment Initiation Service (PIS)

Payment initiation service allows clients to initiate various types of payments using TPPs' applications.

```

{
  "swagger": "2.0",
  "info": {
    "title": "PISP API",
    "description": "Public API for PISP.",
    "version": "1.15"
  },
  "schemes": [
    "https"
  ],
  "produces": [
    "application/json"
  ],
  "basePath": "/api/v1/my/payments",
  "paths": {
    "/": {
      "post": {
        "summary": "Initiate a new payment.",
        "description": "Resource for entering a new payment. You are expected to provide a valid certificate and a valid OAuth2 token and optional an electronic seal. This method also has a sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
        "parameters": [
          {
            "$ref": "#/parameters/Accept-Language"
          },
          {
            "$ref": "#/parameters/User-Geo-Location"
          },
          {
            "$ref": "#/parameters/User-IP-Address"
          },
          {
            "$ref": "#/parameters/User-User-Agent"
          },
          {
            "$ref": "#/parameters/X-Request-ID"
          },
          {
            "$ref": "#/parameters/Date"
          },
          {
            "$ref": "#/parameters/User-Involved"
          },
          {
            "$ref": "#/parameters/Action-ID"
          },
          {
            "$ref": "#/parameters/Authorization"
          },
          {
            "$ref": "#/parameters/TPP-Signature-Certificate"
          }
        ]
      }
    }
  }
}

```

```

    {
      "$ref": "#/parameters/Digest"
    },
    {
      "$ref": "#/parameters/Signature"
    },
  ],
  {
    "in": "body",
    "name": "PaymentInitiationRequest",
    "schema": {
      "$ref": "#/definitions/PaymentInitiationRequest"
    }
  }
],
"tags": [
  "payments"
],
"responses": {
  "200": {
    "description": "OK",
    "schema": {
      "$ref": "#/definitions/PaymentInitiationResponse"
    }
  },
  "400": {
    "description": "Missing mandatory field in the request or field value is not
valid.",
    "schema": {
      "$ref": "#/definitions/Errors"
    }
  },
  "401": {
    "description": "Missing certificate.",
    "schema": {
      "$ref": "#/definitions/Errors"
    }
  },
  "403": {
    "description": "Calling of the method which does not correspond to the licence,
or invalid certificate",
    "schema": {
      "$ref": "#/definitions/Errors"
    }
  },
  "500": {
    "description": "Server error.",
    "schema": {
      "$ref": "#/definitions/Errors"
    }
  }
}
},
"/{paymentId}/status": {
  "get": {
    "summary": "Display the status of the payment.",
    "description": "The resource returns the information only for transactions entered
through a specific provider. You are expected to provide a valid certificate. This method
also has a sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox
API\")",
    "parameters": [
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      },
      {
        "$ref": "#/parameters/User-User-Agent"
      },
      {
        "$ref": "#/parameters/X-Request-ID"
      }
    ],

```

```

        {
            "$ref": "#/parameters/Date"
        },
        {
            "$ref": "#/parameters/User-Involved"
        },
        {
            "$ref": "#/parameters/Action-ID"
        },
        {
            "in": "path",
            "name": "paymentId",
            "type": "string",
            "required": true,
            "description": "The id of the payment"
        }
    ],
    "tags": [
        "payments"
    ],
    "responses": {
        "200": {
            "description": "OK",
            "schema": {
                "$ref": "#/definitions/PaymentStatusResponse"
            }
        },
        "400": {
            "description": "Missing mandatory field in the request or field value is not
valid.",
            "schema": {
                "$ref": "#/definitions/Errors"
            }
        },
        "401": {
            "description": "Invalid/missing certificate",
            "schema": {
                "$ref": "#/definitions/Errors"
            }
        },
        "404": {
            "description": "Calling of the method which does not correspond to the licence,
or invalid certificate",
            "schema": {
                "$ref": "#/definitions/Errors"
            }
        },
        "500": {
            "description": "Server error.",
            "schema": {
                "$ref": "#/definitions/Errors"
            }
        }
    }
}
},
"/{paymentId}/sign": {
    "post": {
        "summary": "Sing transaction",
        "description": "This is the resource for signing a particular transaction. You are
expected to provide a valid certificate and a valid oAuth2 token. This method also has a
sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
        "parameters": [
            {
                "$ref": "#/parameters/Accept-Language"
            },
            {
                "$ref": "#/parameters/User-Geo-Location"
            },
            {
                "$ref": "#/parameters/User-IP-Address"
            },
            {
                "$ref": "#/parameters/User-User-Agent"
            }
        ]
    }
}
}

```

```

        "$ref": "#/parameters/X-Request-ID"
      },
      {
        "$ref": "#/parameters/Date"
      },
      {
        "$ref": "#/parameters/User-Involved"
      },
      {
        "$ref": "#/parameters/Action-ID"
      },
      {
        "$ref": "#/parameters/Authorization"
      },
      {
        "in": "path",
        "name": "paymentId",
        "type": "string",
        "required": true,
        "description": "The id of the payment"
      }
    ],
    "tags": [
      "payments"
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "$ref": "#/definitions/PaymentSigningResponse"
        }
      },
      "400": {
        "description": "Missing mandatory field in the request or field value is not
valid.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "401": {
        "description": "Missing certificate.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "403": {
        "description": "Calling of the method which does not correspond to the licence,
or invalid certificate",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "500": {
        "description": "Server error.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      }
    }
  }
},
"/{paymentId}": {
  "put": {
    "summary": "Update a payment.",
    "description": "Resource updating a payment. You are expected to provide a valid
certificate and a valid oAuth2 token and optional an electronic seal.",
    "parameters": [
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      }
    ],

```

```

        {
            "$ref": "#/parameters/User-User-Agent"
        },
        {
            "$ref": "#/parameters/X-Request-ID"
        },
        {
            "$ref": "#/parameters/Date"
        },
        {
            "$ref": "#/parameters/User-Involved"
        },
        {
            "$ref": "#/parameters/Action-ID"
        },
        {
            "$ref": "#/parameters/Authorization"
        },
        {
            "$ref": "#/parameters/TPP-Signature-Certificate"
        },
        {
            "$ref": "#/parameters/Digest"
        },
        {
            "$ref": "#/parameters/Signature"
        },
    ],
    {
        "in": "body",
        "name": "PaymentUpdateRequest",
        "schema": {
            "$ref": "#/definitions/PaymentInitiationRequest"
        }
    },
    {
        "in": "path",
        "name": "paymentId",
        "type": "string",
        "required": true,
        "description": "The id of the payment"
    }
],
"tags": [
    "payments"
],
"responses": {
    "200": {
        "description": "OK",
        "schema": {
            "$ref": "#/definitions/PaymentInitiationResponse"
        }
    },
    "400": {
        "description": "Missing mandatory field in the request or field value is not
valid.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "401": {
        "description": "Missing certificate.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "403": {
        "description": "Calling of the method which does not correspond to the licence,
or invalid certificate",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "500": {
        "description": "Server error.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    }
}

```

```

    }
  }
},
"delete": {
  "summary": "Delete a payment.",
  "description": "Resource for deleting a payment. You are expected to provide a valid
certificate and a valid OAuth2 token. This method also has a sandbox service which you can
test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
  "parameters": [
    {
      "$ref": "#/parameters/Accept-Language"
    },
    {
      "$ref": "#/parameters/User-Geo-Location"
    },
    {
      "$ref": "#/parameters/User-IP-Address"
    },
    {
      "$ref": "#/parameters/User-User-Agent"
    },
    {
      "$ref": "#/parameters/X-Request-ID"
    },
    {
      "$ref": "#/parameters/Date"
    },
    {
      "$ref": "#/parameters/User-Involved"
    },
    {
      "$ref": "#/parameters/Action-ID"
    },
    {
      "$ref": "#/parameters/Authorization"
    },
    {
      "in": "path",
      "name": "paymentId",
      "type": "string",
      "required": true,
      "description": "The id of the payment"
    }
  ],
  "tags": [
    "payments"
  ],
  "responses": {
    "204": {
      "description": "The resource was deleted successfully."
    },
    "400": {
      "description": "Missing mandatory field in the request or field value is not
valid.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "401": {
      "description": "Missing certificate.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "403": {
      "description": "Calling of the method which does not correspond to the licence,
or invalid certificate",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "500": {
      "description": "Server error.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    }
  }
}

```

```

    }
  }
}
},
"/balanceCheck": {
  "post": {
    "summary": "Check the balance",
    "description": "This is the resource for sending a request for balance check in a
particular payer's payment account. You are expected to provide a valid certificate and a
valid OAuth2 token. This method also has a sandbox service which you can test. ![alt
text](images/logos/sandbox.png \"Sandbox API\")",
    "parameters": [
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      },
      {
        "$ref": "#/parameters/User-User-Agent"
      },
      {
        "$ref": "#/parameters/X-Request-ID"
      },
      {
        "$ref": "#/parameters/Date"
      },
      {
        "$ref": "#/parameters/User-Involved"
      },
      {
        "$ref": "#/parameters/Action-ID"
      },
      {
        "$ref": "#/parameters/Authorization"
      },
      {
        "in": "body",
        "name": "BalanceCheckRequest",
        "schema": {
          "$ref": "#/definitions/BalanceCheckRequest"
        }
      }
    ],
    "tags": [
      "payments"
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "$ref": "#/definitions/BalanceCheckResponse"
        }
      },
      "400": {
        "description": "Missing mandatory field in the request or field value is not
valid.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "401": {
        "description": "Missing certificate.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "403": {
        "description": "Calling of the method which does not correspond to the licence,
or invalid certificate",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      }
    }
  }
}
}

```

```

    }
    },
    "500": {
      "description": "Server error.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    }
  }
}
},
"parameters": {
  "Accept-Language": {
    "name": "Accept-Language",
    "in": "header",
    "type": "string",
    "description": "The locale of the customer"
  },
  "User-Agent": {
    "name": "User-Agent",
    "in": "header",
    "type": "string",
    "description": "Max 200 characters. Contains information about the end-user web browser. The format should be the same as the standard HTTP parameter User-Agent."
  },
  "User-IP-Address": {
    "name": "User-IP-Address",
    "in": "header",
    "type": "string",
    "format": "IPv4",
    "description": "Max 50 characters. Identifies the API's end-user IP address. The TPP should fill this value if the IP address can be detected. The address can be in IPv4 or IPv6 format."
  },
  "User-Geo-Location": {
    "name": "User-Geo-Location",
    "in": "header",
    "type": "string",
    "description": "Max 100 characters. End-user GPS coordinates (if the coordinates can be reliably detected). Required format: GEO:Latitude;Longitude based on [RFC2426] in signed degrees format (DDD.dddd) example: GEO:50.089354;14.400031"
  },
  "X-Request-ID": {
    "name": "X-Request-ID",
    "in": "header",
    "type": "string",
    "description": "Max 60 characters. Unique identifier for each request specified by TPP. It is recommended to use a UUID format with a length of 36 to 60 characters."
  },
  "Date": {
    "name": "Date",
    "in": "header",
    "type": "string",
    "description": "Each request of a transaction contains the date and time at which the message was originated. In timestamp format."
  },
  "User-Involved": {
    "name": "User-Involved",
    "in": "header",
    "type": "boolean",
    "description": "The flag identifies if the end user is online and the request was made based on his activity."
  },
  "Action-ID": {
    "name": "Action-ID",
    "in": "header",
    "type": "string",
    "format": "uuid",
    "description": "ID of the transaction as determined by the initiating party."
  },
  "Authorization": {
    "name": "Authorization",
    "in": "header",
    "type": "string",

```



```

    "description": "The parameter is used to pass an access token of the authenticated
user together with its type."
  },
  "TPP-Signature-Certificate": {
    "name": "TPP-Certificate",
    "in": "header",
    "type": "string",
    "description": "The parameter is used to pass a QSeal Certificate."
  },
  "Digest": {
    "name": "Digest",
    "in": "header",
    "type": "string",
    "description": "The Digest Header contains a Hash of the message body."
  },
  "Signature": {
    "name": "Signature",
    "in": "header",
    "type": "string",
    "description": "The Signature Header contains information about specific requirements
of PSD2 for electronic signatures."
  }
},
"definitions": {
  "PaymentInitiationRequest": {
    "type": "object",
    "description": "Request used when initiating a new payment.",
    "required": [
      "paymentIdentification",
      "amount",
      "debtorAccount"
    ],
    "properties": {
      "paymentIdentification": {
        "$ref": "#/definitions/PaymentIdentificationRequest"
      },
      "paymentTypeInformation": {
        "$ref": "#/definitions/PaymentType"
      },
      "amount": {
        "$ref": "#/definitions/Amount"
      },
      "requestedExecutionDate": {
        "$ref": "#/definitions/RequestedExecutionDate"
      },
      "startDate": {
        "$ref": "#/definitions/StartDate"
      },
      "endDate": {
        "$ref": "#/definitions/EndDate"
      },
      "frequency": {
        "$ref": "#/definitions/Frequency"
      },
      "debtor": {
        "$ref": "#/definitions/Debtor"
      },
      "debtorAccount": {
        "$ref": "#/definitions/DebtorAccount"
      },
      "creditor": {
        "$ref": "#/definitions/Creditor"
      },
      "creditorAccount": {
        "$ref": "#/definitions/CreditorAccount"
      },
      "remittanceInformation": {
        "$ref": "#/definitions/RemittanceInformation"
      },
      "purpose": {
        "$ref": "#/definitions/PaymentPurpose"
      },
      "paymentDescription": {
        "type": "string",
        "description": "The payment description"
      }
    }
  },

```

```

        "noteForMe": {
            "type": "string",
            "description": "The payment description for the debtor."
        }
    },
    "PaymentIdentificationRequest": {
        "type": "object",
        "description": "Set of elements used to identify the payment instruction",
        "required": [
            "instructionIdentification"
        ],
        "properties": {
            "instructionIdentification": {
                "type": "string",
                "maxLength": 35,
                "description": "Unique identification assigned by PISP that identifies the instruction."
            },
            "endToEndIdentification": {
                "type": "string",
                "maxLength": 35,
                "description": "Identification agreed between the payer and the payee. If not filled in by the payer, then PISP will fill in „NOTPROVIDED“."
            }
        }
    },
    "PaymentType": {
        "type": "object",
        "description": "A set of elements used to further determine the transaction type.",
        "properties": {
            "instructionPriority": {
                "type": "string",
                "description": "If the element is not filled in, the payer's provider assigns a NORM value and makes a payment by default. If a HIGH value is indicated, the payer's provider will make the payment as priority"
            },
            "serviceLevel": {
                "type": "object",
                "required": [
                    "code"
                ],
                "description": "Service placement(within type of payment)",
                "properties": {
                    "code": {
                        "type": "string"
                    }
                }
            }
        }
    },
    "Amount": {
        "type": "object",
        "description": "Amount information",
        "required": [
            "instructedAmount"
        ],
        "properties": {
            "instructedAmount": {
                "type": "object",
                "description": "The amount to be transferred between the payer and the recipient before charges.",
                "required": [
                    "value",
                    "currency"
                ],
                "properties": {
                    "value": {
                        "type": "number"
                    },
                    "currency": {
                        "$ref": "#/definitions/CurrencyCode"
                    }
                }
            }
        }
    },
    "equivalentAmount": {

```

```

        "type": "object",
        "properties": {
            "value": {
                "type": "number"
            },
            "margin": {
                "type": "number"
            },
            "currency": {
                "$ref": "#/definitions/CurrencyCode"
            }
        }
    },
    "ExchangeRateInformation": {
        "type": "object",
        "description": "The element provides details about the exchange rate and Contract",
        "properties": {
            "exchangeRate": {
                "type": "number",
                "format": "float",
                "description": "Detail of Agreed Exchange rate."
            },
            "rateType": {
                "type": "string",
                "enum": [
                    "AGREED",
                    "SALE",
                    "SPOT"
                ]
            },
            "contractIdentification": {
                "type": "string",
                "maxLength": 35,
                "description": "Detail of Contract agreed with client for the execution of payments
at the Agreed Exchange rate"
            }
        }
    },
    "Debtor": {
        "type": "object",
        "description": "A party that owes money to the payee.",
        "properties": {
            "name": {
                "type": "string",
                "maxLength": 70,
                "description": "Name under which the party is known."
            },
            "postalAddress": {
                "$ref": "#/definitions/DebtorPostalAddress"
            }
        }
    },
    "DebtorAccount": {
        "type": "object",
        "description": "Clear identification of the payer account in which a debit item will
be executed as the transaction result.",
        "required": [
            "identification"
        ],
        "properties": {
            "identification": {
                "type": "object",
                "properties": {
                    "iban": {
                        "$ref": "#/definitions/IBAN2007Identifier"
                    },
                    "other": {
                        "type": "object",
                        "description": "Unique account identification assigned by the party
administering the account.",
                        "properties": {
                            "identification": {
                                "type": "string",
                                "maxLength": 34,

```

```

        "description": "An account number used for identification of the client
account."
    }
}
},
    "currency": {
        "$ref": "#/definitions/CurrencyCode"
    }
},
    "RequestedExecutionDate": {
        "type": "string",
        "description": "The date on which the initiating party requests payment processing by
a settlement agent. Date (or date and time) in the ISO 8601 format"
    },
    "StartDate": {
        "type": "string",
        "description": "The first applicable day of execution starting from this date is the
first payment. Date (or date and time) in the ISO 8601 format"
    },
    "EndDate": {
        "type": "string",
        "description": "The last applicable day of execution If not given, it is an infinite
standing order. Date (or date and time) in the ISO 8601 format"
    },
    "RemittanceInformation": {
        "type": "object",
        "description": "Information that allows match (i.e. pairing, reconciliation) of
payments with which the payment should be settled.",
        "properties": {
            "unstructured": {
                "type": "string",
                "maxLength": 140,
                "description": "Unstructured message for payee."
            }
        }
    },
    "Creditor": {
        "type": "object",
        "properties": {
            "name": {
                "type": "string",
                "maxLength": 105,
                "description": "The authentication method needed for this enrollment."
            },
            "postalAddress": {
                "$ref": "#/definitions/CreditorPostalAddress"
            }
        }
    },
    "CreditorAccount": {
        "type": "object",
        "description": "Clear identification of the payee account to which the credit item will
be transferred as the transaction result.",
        "properties": {
            "identification": {
                "type": "object",
                "properties": {
                    "iban": {
                        "$ref": "#/definitions/IBAN2007Identifier"
                    },
                    "other": {
                        "type": "object",
                        "description": "Unique account identification assigned by the party
administering the account,",
                        "properties": {
                            "identification": {
                                "type": "string",
                                "maxLength": 34,
                                "description": "An account number used by financial institutions"
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    },
    "currency": {
      "$ref": "#/definitions/CurrencyCode"
    }
  },
  "IBAN2007Identifier": {
    "type": "string",
    "description": "Account number in the IBAN format which is defined by the international standard ISO 13616",
    "pattern": "[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}"
  },
  "PaymentStatusResponse": {
    "type": "object",
    "required": [
      "instructionStatus"
    ],
    "description": "Status of entered payment",
    "properties": {
      "instructionStatus": {
        "$ref": "#/definitions/StatusCode"
      }
    }
  },
  "StatusCode": {
    "type": "string",
    "enum": [
      "ACTC",
      "RJCT",
      "ACSP",
      "ACSC",
      "ACWC"
    ]
  },
  "Frequency": {
    "type": "string",
    "description": "The frequency of the recurring payment resulting from this standing order",
    "enum": [
      "Daily",
      "Weekly",
      "EveryTwoWeeks",
      "Monthly",
      "EveryTwoMonths",
      "Quarterly",
      "SemiAnnual",
      "Annual"
    ]
  },
  "PaymentInitiationResponse": {
    "type": "object",
    "required": [
      "transactionIdentification",
      "signInfo"
    ],
    "properties": {
      "paymentIdentification": {
        "$ref": "#/definitions/PaymentIdentificationResponse"
      },
      "paymentTypeInformation": {
        "$ref": "#/definitions/PaymentType"
      },
      "amount": {
        "$ref": "#/definitions/Amount"
      },
      "requestedExecutionDate": {
        "$ref": "#/definitions/RequestedExecutionDate"
      },
      "startDate": {
        "$ref": "#/definitions/StartDate"
      },
      "endDate": {
        "$ref": "#/definitions/EndDate"
      },
      "frequency": {
        "$ref": "#/definitions/Frequency"
      }
    }
  }
}

```

```

    },
    "debtor": {
      "$ref": "#/definitions/Debtor"
    },
    "debtorAccount": {
      "$ref": "#/definitions/DebtorAccount"
    },
    "creditor": {
      "$ref": "#/definitions/Creditor"
    },
    "creditorAccount": {
      "$ref": "#/definitions/CreditorAccount"
    },
    "remittanceInformation": {
      "$ref": "#/definitions/RemittanceInformation"
    },
    "exchangeRateInformation": {
      "$ref": "#/definitions/ExchangeRateInformation"
    },
    "purpose": {
      "$ref": "#/definitions/PaymentPurpose"
    },
    "paymentDescription": {
      "type": "string",
      "description": "The payment description"
    },
    "noteForMe": {
      "type": "string",
      "description": "The payment description for the debtor."
    },
    "margin": {
      "type": "object",
      "required": [
        "value",
        "currency"
      ],
      "properties": {
        "value": {
          "type": "number"
        },
        "currency": {
          "$ref": "#/definitions/CurrencyCode"
        }
      },
      "description": "The margin of the transaction for SEPA payments."
    },
    "signInfo": {
      "$ref": "#/definitions/SignInfo"
    },
    "links": {
      "$ref": "#/definitions/PaymentsListLinks"
    }
  }
},
"PaymentIdentificationResponse": {
  "type": "object",
  "description": "Set of elements used to identify the payment instruction",
  "required": [
    "instructionIdentification",
    "transactionIdentification"
  ],
  "properties": {
    "instructionIdentification": {
      "type": "string",
      "maxLength": 35,
      "description": "Unique identification assigned by PISP that identifies the instruction."
    },
    "endToEndIdentification": {
      "type": "string",
      "maxLength": 35,
      "description": "Identification agreed between the payer and the payee. If not filled in by the payer, then PISP will fill in „NOTPROVIDED“."
    },
    "transactionIdentification": {
      "type": "string",

```

```

        "maxLength": 200,
        "description": "The unique reference of the payer's bank"
    }
},
"PaymentsListLinks": {
    "type": "object",
    "properties": {
        "self": {
            "type": "string",
            "description": "The link to the resource itself"
        }
    }
},
"PaymentSigningResponse": {
    "type": "object",
    "required": [
        "scenarios",
        "signInfo"
    ],
    "properties": {
        "scenarios": {
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "description": "A set of possible authorization scenarios"
    },
    "signInfo": {
        "$ref": "#/definitions/SignInfo"
    }
},
"PaymentDescriptionSetRequest": {
    "type": "object",
    "properties": {
        "paymentDescription": {
            "type": "string",
            "description": "The payment description"
        }
    }
},
"SignInfo": {
    "type": "object",
    "required": [
        "state",
        "signId"
    ],
    "properties": {
        "state": {
            "$ref": "#/definitions/State"
        },
        "signId": {
            "type": "string"
        }
    }
},
"description": "Status information and id of unauthorized transactions"
},
"BalanceCheckRequest": {
    "type": "object",
    "required": [
        "exchangeIdentification",
        "debtorAccount"
    ],
    "description": "Request used to check the balance.",
    "properties": {
        "exchangeIdentification": {
            "type": "string",
            "maxLength": 18,
            "description": "Identification is assigned by the third party; the value will be contained in the response to request."
        },
        "card": {
            "type": "object",
            "required": [
                "maskedPan"
            ]
        }
    }
}

```

```

    ],
    "description": "Payment card linked to the transaction, if the card was used for
initiation.",
    "properties": {
      "cardholderName": {
        "type": "string",
        "maxLength": 45,
        "description": "The field contains the name of card holder stated on the card."
      },
      "maskedPan": {
        "type": "string",
        "maxLength": 30,
        "description": "Masked card number."
      }
    }
  },
  "debtorAccount": {
    "$ref": "#/definitions/DebtorAccount"
  },
  "authenticationMethod": {
    "$ref": "#/definitions/CodeSet"
  },
  "merchant": {
    "$ref": "#/definitions/Merchant"
  },
  "transactionDetails": {
    "$ref": "#/definitions/TransactionDetails"
  }
}
},
"Merchant": {
  "type": "object",
  "description": "A merchant executing a balance query",
  "required": [
    "identification",
    "shortName",
    "commonName",
    "merchantCategoryCode"
  ],
  "properties": {
    "identification": {
      "type": "string",
      "maxLength": 35,
      "description": "Identification of an entity requesting a disposable balance"
    },
    "type": {
      "$ref": "#/definitions/Code"
    },
    "shortName": {
      "type": "string",
      "maxLength": 35,
      "description": "Name of entity executing a query for disposable balance."
    },
    "commonName": {
      "type": "string",
      "maxLength": 70,
      "description": "Full name of entity as it will be stated in the payment receipt."
    },
    "address": {
      "type": "string",
      "maxLength": 140,
      "description": "Address of entity executing a query for disposable balance."
    },
    "countryCode": {
      "$ref": "#/definitions/CountryCode"
    },
    "merchantCategoryCode": {
      "type": "string",
      "minLength": 3,
      "maxLength": 4,
      "description": "Merchant code following the transaction type"
    }
  }
}
},
"Code": {
  "type": "string",

```



```

    "enum": [
      "OPOI",
      "MERC",
      "ACCP",
      "ITAG",
      "ACQR",
      "CISS",
      "DLIS"
    ]
  },
  "State": {
    "type": "string",
    "enum": [
      "OPEN",
      "REJECTED",
      "DONE",
      "NONE"
    ],
    "description": "Transaction authorisation status"
  },
  "DebtorPostalAddress": {
    "type": "object",
    "description": "Postal address information",
    "properties": {
      "country": {
        "$ref": "#/definitions/CountryCode"
      }
    }
  },
  "CreditorPostalAddress": {
    "type": "object",
    "description": "Postal address information",
    "properties": {
      "country": {
        "$ref": "#/definitions/CountryCode"
      },
      "addressLine": {
        "type": "string",
        "maxLength": 70
      }
    }
  },
  "TransactionDetails": {
    "type": "object",
    "required": [
      "currency",
      "totalAmount"
    ],
    "description": "Transaction details",
    "properties": {
      "currency": {
        "$ref": "#/definitions/CurrencyCode"
      },
      "totalAmount": {
        "type": "number",
        "format": "float"
      }
    }
  },
  "BalanceCheckResponse": {
    "type": "object",
    "required": [
      "responseIdentification",
      "exchangeIdentification",
      "response"
    ],
    "properties": {
      "responseIdentification": {
        "type": "number",
        "description": "Unique identification of response to query for Balance Check(from ASPSP)"
      },
      "exchangeIdentification": {
        "type": "number",

```

```

        "description": "Repeated identification of a payment transaction (query for Balance
        Check) from the issuer of the card to which the request for Balance Check linked to the
        account"
    },
    "response": {
        "$ref": "#/definitions/ResponseCodeSet"
    }
},
"CurrencyCode": {
    "type": "string",
    "description": "ISO 4217 currency code",
    "pattern": "[A-Z]{3}"
},
"CountryCode": {
    "type": "string",
    "pattern": "[A-Z]{2,2}"
},
"CodeSet": {
    "type": "string",
    "enum": [
        "NPIN",
        "PPSG",
        "PSWD",
        "SCRT",
        "SCNL",
        "SNCT",
        "CPSG",
        "ADDB",
        "BIOM",
        "CDHI",
        "CRYP",
        "CSCV",
        "PSVE",
        "CSEC",
        "ADDS",
        "TOKP"
    ]
},
"ResponseCodeSet": {
    "type": "string",
    "enum": [
        "APPR",
        "DECL"
    ]
},
"PaymentPurpose": {
    "description": "Description of the payment purpose\n",
    "type": "object",
    "properties": {
        "code": {
            "description": "Purpose of the payment",
            "type": "string",
            "maxLength": 70
        },
        "proprietary": {
            "description": "Free form of payment purpose",
            "type": "string",
            "maxLength": 35
        }
    }
},
"Errors": {
    "type": "array",
    "description": "The list of errors thrown.",
    "items": {
        "$ref": "#/definitions/Error"
    }
},
"Error": {
    "type": "object",
    "properties": {
        "error": {
            "type": "string",
            "description": "Contains the specific error code."
        }
    }
},

```



```

    },
    {
      "$ref": "#/parameters/User-Involved"
    },
    {
      "in": "query",
      "name": "fromDate",
      "type": "string",
      "description": "Date and time of the start of required transaction history"
    },
    {
      "in": "query",
      "name": "toDate",
      "type": "string",
      "description": "Date and time of the end of required transaction history
(inclusive)"
    },
    {
      "$ref": "#/parameters/size"
    },
    {
      "$ref": "#/parameters/page"
    },
    {
      "$ref": "#/parameters/sort"
    },
    {
      "$ref": "#/parameters/order"
    }
  ],
  "responses": {
    "200": {
      "description": "Paged list of transactions of the specified client account",
      "schema": {
        "allOf": [
          {
            "$ref": "#/definitions/PageMetadata"
          },
          {
            "$ref": "#/definitions/StandingOrderList"
          }
        ]
      }
    },
    "400": {
      "description": "Invalid request. A mandatory parameter is missing or it is in an
unsuitable / invalid format.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "401": {
      "description": "The client is not authorized to execute the request.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "404": {
      "description": "The requested account or page does not exist",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "500": {
      "description": "Server error.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "501": {
      "description": "Method not implemented.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    }
  }
}

```

```

    },
    "post": {
      "summary": "Initiate a new standing order.",
      "description": "Resource for entering a new standing order. You are expected to provide a valid certificate and a valid OAuth2 token. This method also has a sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
      "parameters": [
        {
          "$ref": "#/parameters/Accept-Language"
        },
        {
          "$ref": "#/parameters/User-Geo-Location"
        },
        {
          "$ref": "#/parameters/User-IP-Address"
        },
        {
          "$ref": "#/parameters/User-User-Agent"
        },
        {
          "$ref": "#/parameters/Action-ID"
        },
        {
          "$ref": "#/parameters/Authorization"
        },
        {
          "$ref": "#/parameters/Digest"
        },
        {
          "$ref": "#/parameters/Signature"
        },
        {
          "$ref": "#/parameters/X-Request-ID"
        },
        {
          "$ref": "#/parameters/Date"
        },
        {
          "$ref": "#/parameters/User-Involved"
        }
      ],
      {
        "in": "body",
        "name": "StandingOrderInitiationRequest",
        "schema": {
          "$ref": "#/definitions/StandingOrderInitiationRequest"
        }
      }
    ],
    "tags": [
      "Standing Orders"
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "$ref": "#/definitions/StandingOrderInitiationResponse"
        }
      },
      "400": {
        "description": "Missing mandatory field in the request or field value is not valid.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "401": {
        "description": "Missing certificate.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "403": {
        "description": "Calling of the method which does not correspond to the licence, or invalid certificate",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      }
    }
  }
}

```

```

    }
  },
  "500": {
    "description": "Server error.",
    "schema": {
      "$ref": "#/definitions/Errors"
    }
  }
}
},
"/{transactionIdentification}": {
  "get": {
    "summary": "Obtain the details of a specific client's account",
    "description": "Obtain the details of the account. You are expected to provide a valid certificate and a valid OAuth2 token. This method also has a sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
    "tags": [
      "Standing Orders"
    ],
    "parameters": [
      {
        "$ref": "#/parameters/User-User-Agent"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "$ref": "#/parameters/Authorization"
      },
      {
        "$ref": "#/parameters/Action-ID"
      },
      {
        "$ref": "#/parameters/X-Request-ID"
      },
      {
        "$ref": "#/parameters/Date"
      },
      {
        "$ref": "#/parameters/User-Involved"
      }
    ],
    {
      "in": "path",
      "name": "transactionIdentification",
      "type": "string",
      "required": true,
      "description": "System identifier of the standing order."
    }
  ],
  "responses": {
    "200": {
      "description": "The standing order details",
      "schema": {
        "allOf": [
          {
            "$ref": "#/definitions/StandingOrderDetails"
          }
        ]
      }
    },
    "400": {
      "description": "Invalid request. A mandatory parameter is missing or it is in an unsuitable / invalid format.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "401": {
      "description": "The client is not authorized to execute the request.",

```

```

    "schema": {
      "$ref": "#/definitions/Errors"
    }
  },
  "404": {
    "description": "The requested page does not exist",
    "schema": {
      "$ref": "#/definitions/Errors"
    }
  },
  "500": {
    "description": "Server error.",
    "schema": {
      "$ref": "#/definitions/Errors"
    }
  },
  "501": {
    "description": "Method not implemented.",
    "schema": {
      "$ref": "#/definitions/Errors"
    }
  }
},
"put": {
  "summary": "Update a standing order.",
  "description": "Resource updating a standing order. You are expected to provide a valid certificate and a valid oAuth2 token. This method also has a sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
  "parameters": [
    {
      "$ref": "#/parameters/Accept-Language"
    },
    {
      "$ref": "#/parameters/User-Geo-Location"
    },
    {
      "$ref": "#/parameters/User-IP-Address"
    },
    {
      "$ref": "#/parameters/User-User-Agent"
    },
    {
      "$ref": "#/parameters/Action-ID"
    },
    {
      "$ref": "#/parameters/Authorization"
    },
    {
      "$ref": "#/parameters/Digest"
    },
    {
      "$ref": "#/parameters/Signature"
    },
    {
      "$ref": "#/parameters/X-Request-ID"
    },
    {
      "$ref": "#/parameters/Date"
    },
    {
      "$ref": "#/parameters/User-Involved"
    }
  ],
  {
    "in": "body",
    "name": "PaymentUpdateRequest",
    "schema": {
      "$ref": "#/definitions/StandingOrderInitiationRequest"
    }
  },
  {
    "in": "path",
    "name": "transactionIdentification",
    "type": "string",
    "required": true,
    "description": "The id of the payment"
  }
}

```

```

    },
    "tags": [
      "Standing Orders"
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "$ref": "#/definitions/StandingOrderInitiationResponse"
        }
      },
      "400": {
        "description": "Missing mandatory field in the request or field value is not
valid.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "401": {
        "description": "Missing certificate.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "403": {
        "description": "Calling of the method which does not correspond to the licence,
or invalid certificate",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      },
      "500": {
        "description": "Server error.",
        "schema": {
          "$ref": "#/definitions/Errors"
        }
      }
    }
  },
  "delete": {
    "summary": "Delete a standing order.",
    "description": "Resource for deleting a standing order. You are expected to provide
a valid certificate and a valid oAuth2 token. This method also has a sandbox service which
you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
    "parameters": [
      {
        "$ref": "#/parameters/User-User-Agent"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "$ref": "#/parameters/Authorization"
      },
      {
        "$ref": "#/parameters/Action-ID"
      },
      {
        "$ref": "#/parameters/X-Request-ID"
      },
      {
        "$ref": "#/parameters/Date"
      },
      {
        "$ref": "#/parameters/User-Involved"
      },
      {
        "in": "path",
        "name": "transactionIdentification",

```



```

        "type": "string",
        "required": true,
        "description": "The id of the payment"
    }
  ],
  "tags": [
    "Standing Orders"
  ],
  "responses": {
    "200": {
      "description": "OK",
      "schema": {
        "$ref": "#/definitions/StandingOrderStatusResponse"
      }
    },
    "400": {
      "description": "Missing mandatory field in the request or field value is not
valid.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "401": {
      "description": "Invalid/missing certificate",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "404": {
      "description": "Calling of the method which does not correspond to the licence,
or invalid certificate",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    },
    "500": {
      "description": "Server error.",
      "schema": {
        "$ref": "#/definitions/Errors"
      }
    }
  }
}
},
"/{paymentId}/sign": {
  "post": {
    "summary": "Sing transaction",
    "description": "This is the resource for signing a particular transaction. You are
expected to provide a valid certificate and a valid oAuth2 token. This method also has a
sandbox service which you can test. ![alt text](images/logos/sandbox.png \"Sandbox API\")",
    "parameters": [
      {
        "$ref": "#/parameters/User-User-Agent"
      },
      {
        "$ref": "#/parameters/User-IP-Address"
      },
      {
        "$ref": "#/parameters/User-Geo-Location"
      },
      {
        "$ref": "#/parameters/Accept-Language"
      },
      {
        "$ref": "#/parameters/Authorization"
      },
      {
        "$ref": "#/parameters/Action-ID"
      },
      {
        "$ref": "#/parameters/X-Request-ID"
      },
      {
        "$ref": "#/parameters/Date"
      }
    ]
  }
}

```

```

        "$ref": "#/parameters/User-Involved"
    },
    {
        "in": "path",
        "name": "paymentId",
        "type": "string",
        "required": true,
        "description": "The id of the payment"
    }
],
"tags": [
    "Standing Orders"
],
"responses": {
    "200": {
        "description": "OK",
        "schema": {
            "$ref": "#/definitions/PaymentSigningResponse"
        }
    },
    "400": {
        "description": "Missing mandatory field in the request or field value is not
valid.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "401": {
        "description": "Missing certificate.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "403": {
        "description": "Calling of the method which does not correspond to the licence,
or invalid certificate",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    },
    "500": {
        "description": "Server error.",
        "schema": {
            "$ref": "#/definitions/Errors"
        }
    }
}
}
},
"parameters": {
    "Accept-Language": {
        "name": "Accept-Language",
        "in": "header",
        "type": "string",
        "description": "The locale of the customer"
    },
    "User-User-Agent": {
        "name": "User-User-Agent",
        "in": "header",
        "type": "string",
        "description": "Max 200 characters. Contains information about the end-user web
browser. The format should be the same as the standard HTTP parameter User-Agent."
    },
    "User-IP-Address": {
        "name": "User-IP-Address",
        "in": "header",
        "type": "string",
        "format": "IPV4",
        "description": "Max 50 characters. Identifies the API's end-user IP address. The TPP
should fill this value if the IP address can be detected. The address can be in IPv4 or IPv6
format."
    },
    "User-Geo-Location": {
        "name": "User-Geo-Location",
        "in": "header",

```

```

        "type": "string",
        "description": "Max 100 characters. End-user GPS coordinates (if the coordinates can
be reliably detected). Required format: GEO:Latitude;Longitude based on [RFC2426] in signed
degrees format (DDD.dddd) example: GEO:50.089354;14.400031"
    },
    "X-Request-ID": {
        "name": "X-Request-ID",
        "in": "header",
        "type": "string",
        "description": "Max 60 characters. Unique identifier for each request specified by
TPP. It is recommended to use a UUID format with a length of 36 to 60 characters."
    },
    "Date": {
        "name": "Date",
        "in": "header",
        "type": "string",
        "description": "Each request of a transaction contains the date and time at which the
message was originated. In timestamp format."
    },
    "User-Involved": {
        "name": "User-Involved",
        "in": "header",
        "type": "boolean",
        "description": "The flag identifies if the end user is online and the request was
made based on his activity."
    },
    "Action-ID": {
        "name": "Action-ID",
        "in": "header",
        "type": "string",
        "format": "uuid",
        "description": "ID of the transaction as determined by the initiating party."
    },
    "Authorization": {
        "name": "Authorization",
        "in": "header",
        "required": true,
        "type": "string",
        "description": "The parameter is used to pass an access token of the authenticated
user together with its type."
    },
    "Digest": {
        "name": "Digest",
        "in": "header",
        "type": "string",
        "description": "The Digest Header contains a Hash of the message body."
    },
    "Signature": {
        "name": "Signature",
        "in": "header",
        "type": "string",
        "description": "The Signature Header contains information about specific requirements
of PSD2 for electronic signatures."
    },
    "size": {
        "name": "size",
        "in": "query",
        "type": "integer",
        "description": "It defines the required number of records on the page. If the parameter
is not specified, the API returns the entire set."
    },
    "page": {
        "name": "page",
        "in": "query",
        "type": "integer",
        "description": "It defines the required page number. Pages are numbered from 0. If the
parameter is not specified, the API returns the first (zero) page.",
        "default": 0
    },
    "sort": {
        "name": "sort",
        "in": "query",
        "type": "array",
        "items": {
            "type": "string"
        }
    },

```

```

      "description": "It defines a list of fields for sorting. The individual fields in the
list are separated by a comma "
    },
    "order": {
      "name": "order",
      "in": "query",
      "type": "array",
      "items": {
        "type": "string",
        "enum": [
          "asc",
          "desc"
        ]
      }
    },
    "description": "It defines the sorting direction for each field listed in the sort
parameter. The individual sorting directions are separated by commas precisely as the field
names in the sort parameter and are listed in the same order as the fields listed in the sort
parameter."
  }
},
"definitions": {
  "PageMetadata": {
    "type": "object",
    "properties": {
      "pageNumber": {
        "type": "integer",
        "description": "Number of current page"
      },
      "pageCount": {
        "type": "integer",
        "description": "Total number of query pages"
      },
      "nextPage": {
        "type": "integer",
        "description": "The number of the next page. If the current page is at the same
time the last page of the list, then this parameter is not listed or is given with a null
value"
      },
      "pageSize": {
        "type": "integer",
        "description": "Number of entries per page. This parameter can match the required
size value from the query, except for the cases when it is the last page, or when the requested
page range exceeded the maximum limit defined for the particular API resource"
      },
      "totalCount": {
        "type": "integer",
        "description": "Optional data about the total number of entries for all pages. If
this value cannot be precisely determined, it is not stated"
      }
    }
  },
  "IBAN": {
    "type": "string",
    "description": "Account number in the IBAN format which is defined by the international
standard ISO 13616.",
    "pattern": "[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}"
  },
  "ISOCountryCode": {
    "type": "string",
    "description": "ISO 3166 alphanumeric 3 country code",
    "pattern": "[A-Z]{3}|[0-9]{3}"
  },
  "ISOCurrencyCode": {
    "type": "string",
    "description": "ISO 4217 currency code",
    "pattern": "[A-Z]{3}"
  },
  "PostalAddress": {
    "type": "object",
    "description": "A postal address",
    "properties": {
      "streetName": {
        "type": "string",
        "maxLength": 70,
        "description": "Street name"
      }
    }
  },

```

```

"buildingNumber": {
  "type": "string",
  "maxLength": 16,
  "description": "Land registry no."
},
"postCode": {
  "type": "string",
  "maxLength": 16,
  "description": "Postcode"
},
"townName": {
  "type": "string",
  "maxLength": 35,
  "description": "Town name"
},
"country": {
  "$ref": "#/definitions/ISOCountryCode"
},
"addressLine": {
  "type": "string",
  "maxLength": 70,
  "description": "Information that localizes and identifies the specific address as
defined by postal services, presented in the free text format"
}
},
"Errors": {
  "type": "object",
  "properties": {
    "errors": {
      "type": "array",
      "description": "The list of errors thrown.",
      "items": {
        "$ref": "#/definitions/Error"
      }
    }
  }
},
"required": [
  "errors"
],
"Error": {
  "type": "object",
  "properties": {
    "error": {
      "type": "string",
      "description": "Contains the specific error code."
    },
    "scope": {
      "type": "string",
      "description": "It determines the JSON path of the request element that caused the
error status."
    },
    "message": {
      "type": "string",
      "description": "Optional text description. It is not intended for interpretation
to the end user but, for example, to enrich the error log. "
    },
    "parameters": {
      "description": "The field of additional elements specific for the given error code.
These parameters are always specified in the description of a specific error code.",
      "type": "object"
    }
  }
},
"StandingOrderList": {
  "description": "Set of clients standing orders\n",
  "type": "array",
  "items": {
    "$ref": "#/definitions/StandingOrders"
  }
},
"StandingOrders": {
  "description": "Standing order\n",
  "type": "object",
  "required": [

```

```

    "standingOrderIdentification",
    "amount",
    "standingOrder",
    "debtorAccount"
  ],
  "properties": {
    "standingOrderIdentification": {
      "$ref": "#/definitions/StandingOrderIdentification"
    },
    "amount": {
      "$ref": "#/definitions/Amount"
    },
    "standingOrder": {
      "$ref": "#/definitions/StandingOrderSummary"
    },
    "debtorAccount": {
      "$ref": "#/definitions/DebtorAccount"
    }
  }
},
"StandingOrderIdentification": {
  "description": "Standing order identification\n",
  "type": "object",
  "properties": {
    "instructionIdentification": {
      "$ref": "#/definitions/InstructionIdentification"
    },
    "transactionIdentification": {
      "$ref": "#/definitions/TransactionIdentification"
    },
    "endToEndIdentification": {
      "$ref": "#/definitions/EndToEndIdentification"
    }
  }
},
"Amount": {
  "description": "Payment amount\n",
  "type": "object",
  "properties": {
    "instructedAmount": {
      "description": "Amount and currency in the instruction\n",
      "type": "object",
      "properties": {
        "value": {
          "type": "number",
          "description": "Value of the amount"
        },
        "currency": {
          "$ref": "#/definitions/ISOCurrencyCode"
        }
      }
    }
  }
},
"StandingOrderSummary": {
  "description": "Structure describing the summary of the standing orders.\n",
  "type": "object",
  "required": [
    "execution"
  ],
  "properties": {
    "alias": {
      "description": "Description resp. the user-defined standing order payment name.\n",
      "type": "string",
      "maxLength": 250
    },
    "execution": {
      "$ref": "#/definitions/Execution"
    }
  }
},
"StandingOrder": {
  "description": "Structure describing the parameters of the standing orders.\n",
  "type": "object",
  "required": [
    "execution"
  ]
}

```

```

    ],
    "properties": {
      "alias": {
        "description": "Description resp. the user-defined standing order payment name.\n",
        "type": "string",
        "maxLength": 250
      },
      "execution": {
        "$ref": "#/definitions/Execution"
      },
      "validity": {
        "$ref": "#/definitions/Validity"
      },
      "exceptions": {
        "$ref": "#/definitions/Exceptions"
      }
    }
  },
  "StandingOrderDetails": {
    "description": "Structure describing the parameters of the standing orders.\n",
    "type": "object",
    "required": [
      "standingOrderIdentification",
      "amount",
      "standingOrder",
      "debtorAccount",
      "creditorAccount"
    ],
    "properties": {
      "standingOrderIdentification": {
        "$ref": "#/definitions/StandingOrderIdentification"
      },
      "paymentTypeInformation": {
        "$ref": "#/definitions/PaymentType"
      },
      "amount": {
        "$ref": "#/definitions/Amount"
      },
      "requestedExecutionDate": {
        "$ref": "#/definitions/RequestedExecutionDate"
      },
      "standingOrder": {
        "$ref": "#/definitions/StandingOrder"
      },
      "exchangeRateInformation": {
        "$ref": "#/definitions/ExchangeRateInformation"
      },
      "debtor": {
        "$ref": "#/definitions/Debtor"
      },
      "debtorAccount": {
        "$ref": "#/definitions/DebtorAccount"
      },
      "creditor": {
        "$ref": "#/definitions/Creditor"
      },
      "creditorAccount": {
        "$ref": "#/definitions/CreditorAccount"
      },
      "remittanceInformation": {
        "$ref": "#/definitions/RemittanceInformation"
      },
      "standingOrderStatus": {
        "$ref": "#/definitions/StandingOrderStatus"
      },
      "purpose": {
        "$ref": "#/definitions/PaymentPurpose"
      }
    }
  },
  "StandingOrderStatus": {
    "description": "Structure describing the status parameters of the standing orders.\n",
    "type": "object",
    "properties": {
      "paymentStatus": {
        "type": "string",

```



```

        "description": "the status of the payment"
    },
    "cancelationReason": {
        "type": "string",
        "description": "the reason why the payment was cancelled"
    },
    "editable": {
        "type": "boolean",
        "description": "marks if the payment is editable or not."
    }
}
},
"InstructionIdentification": {
    "description": "Payment identification entered by a third party, or a constant symbol
may be entered.\n",
    "maxLength": 35,
    "type": "string"
},
"TransactionIdentification": {
    "description": "Transaction identification\n",
    "maxLength": 200,
    "type": "string"
},
"EndToEndIdentification": {
    "description": "End to end identification. Used for SEPA payments.\n",
    "maxLength": 35,
    "type": "string"
},
"PaymentPurpose": {
    "description": "Description of the payment purpose\n",
    "type": "object",
    "properties": {
        "code": {
            "description": "Purpose of the payment",
            "type": "string",
            "maxLength": 70
        },
        "proprietary": {
            "description": "Free form of payment purpose",
            "type": "string",
            "maxLength": 35
        }
    }
}
},
"Execution": {
    "description": "Features of execution a standing order for payment.\n",
    "type": "object",
    "required": [
        "mode",
        "modeDue",
        "interval",
        "intervalDue"
    ],
    "properties": {
        "mode": {
            "description": "The execution mode defines when or how standing order will be
cancelled, processed the last time. Possible values: UNTIL_DATE (standing order is valid
until specific date - field lastExecutionDate), UNTIL_CANCELLATION (standing order is valid
forever and must be cancelled by client), AFTER_MAX_ITERATION_EXCEEDED (certain count of
executions is specified - field maxIterations) or MAX_AMOUNT_EXCEEDED (maximum amount which
can be transferred for this order is specified, if next iteration would exceed this amount
it is not executed - field maxAmount).\n",
            "maxLength": 35,
            "type": "string",
            "enum": [
                "UNTIL_DATE",
                "UNTIL_CANCELLATION",
                "FTER_MAX_ITERATION_EXCEEDED",
                "MAX_AMOUNT_EXCEEDED"
            ]
        },
        "modeDue": {
            "description": "The execution due mode defines how the date when order should be
executed is specified. Possible values: DUE_DAY_OF_MONTH (specific number of day in the month
is defined) or DUE_OR_BEFORE_DAY_OF_MONTH (is defined as a specific day of the month or the
previous day if the day falls on a non-banking day) or DUE_OR_NEXT DAY_OF_MONTH (is defined

```

```

as a specific day of the month or the next day if the day falls on a non-banking day) or
DUE_LAST_DAY_OF_MONTH (order is executed on last day of particular month).\n",
  "maxLength": 35,
  "type": "string",
  "enum": [
    "DUE_DAY_OF_MONTH",
    "DUE_OR_BEFORE_DAY_OF_MONTH",
    "DUE_OR_NEXT_DAY_OF_MONTH",
    "DUE_LAST_DAY_OF_MONTH"
  ]
},
"interval": {
  "description": "Execution interval defines how often order is executed. Possible
values: DAILY, WEEKLY, MONTHLY, BI_MONTHLY, QUARTERLY, HALFYEARLY, YEARLY, IRREGULAR.\n",
  "type": "string",
  "maxLength": 10,
  "enum": [
    "DAILY",
    "WEEKLY",
    "MONTHLY",
    "BI_MONTHLY",
    "QUARTERLY",
    "HALFYEARLY",
    "YEARLY",
    "IRREGULAR"
  ]
},
"intervalDue": {
  "description": "Value represents order number of the day within particular period
when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY
interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12
for YEARLY.\n",
  "type": "number"
}
},
"DebtorAccount": {
  "type": "object",
  "description": "Clear identification of the payer account in which a debit item will
be executed as the transaction result.",
  "required": [
    "identification"
  ],
  "properties": {
    "identification": {
      "type": "object",
      "properties": {
        "iban": {
          "$ref": "#/definitions/IBAN"
        },
        "other": {
          "type": "object",
          "description": "Unique account identification assigned by the party
administering the account.",
          "properties": {
            "identification": {
              "type": "string",
              "maxLength": 34,
              "description": "An account number used for identification of the client
account."
            }
          }
        }
      }
    }
  }
},
"currency": {
  "$ref": "#/definitions/ISOCurrencyCode"
}
},
"CreditorAccount": {
  "type": "object",
  "description": "Clear identification of the payee account in which a credit item will
be executed as the transaction result.",
  "required": [
    "identification"
  ]
}
}

```

```

    ],
    "properties": {
      "identification": {
        "type": "object",
        "properties": {
          "iban": {
            "$ref": "#/definitions/IBAN"
          },
          "other": {
            "type": "object",
            "description": "Unique account identification assigned by the party
administering the account.",
            "properties": {
              "identification": {
                "type": "string",
                "maxLength": 34,
                "description": "An account number used for identification of the client
account."
              }
            }
          }
        }
      },
      "currency": {
        "$ref": "#/definitions/ISOCurrencyCode"
      }
    }
  },
  "Validity": {
    "description": "Elements restricting the validity of the standing order.\n",
    "type": "object",
    "properties": {
      "lastExecutionDate": {
        "type": "string",
        "format": "date"
      },
      "maxIterations": {
        "type": "number"
      },
      "maxAmount": {
        "type": "object",
        "properties": {
          "value": {
            "type": "number",
            "description": "Value of the amount"
          },
          "currency": {
            "$ref": "#/definitions/ISOCurrencyCode"
          }
        }
      }
    }
  },
  "Exceptions": {
    "description": "Elements defining exceptions to executing a standing order.\n",
    "type": "object",
    "properties": {
      "stoppages": {
        "type": "array",
        "items": {
          "type": "string",
          "maxLength": 20,
          "enum": [
            "JANUARY",
            "FEBRUARY",
            "MARCH",
            "APRIL",
            "MAY",
            "JUNE",
            "JULY",
            "AUGUST",
            "SEPTEMBER",
            "OCTOBER",
            "NOVEMBER",
            "DECEMBER"
          ]
        }
      }
    }
  }
]

```

```

    },
    "breaks": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/Break"
      }
    }
  },
  "Break": {
    "description": "Break periods.\n",
    "type": "object",
    "required": [
      "validFromDate",
      "validToDate"
    ],
    "properties": {
      "validFromDate": {
        "type": "string",
        "format": "date"
      },
      "validToDate": {
        "type": "string",
        "format": "date"
      }
    }
  },
  "PaymentType": {
    "type": "object",
    "description": "A set of elements used to further determine the transaction type.",
    "properties": {
      "instructionPriority": {
        "type": "string",
        "description": "If the element is not filled in, the payer's provider assigns a NORM value and makes a payment by default. If a HIGH value is indicated, the payer's provider will make the payment as priority"
      },
      "serviceLevel": {
        "type": "object",
        "required": [
          "code"
        ],
        "description": "Service placement(within type of payment)",
        "properties": {
          "code": {
            "type": "string"
          }
        }
      }
    }
  },
  "RequestedExecutionDate": {
    "type": "string",
    "format": "date-time",
    "description": "The date on which the initiating party requests payment processing by a settlement agent. Date (or date and time) in the ISO 8601 format"
  },
  "ExchangeRateInformation": {
    "type": "object",
    "description": "The element provides details about the exchange rate and Contract",
    "properties": {
      "exchangeRate": {
        "type": "number",
        "format": "float",
        "description": "Detail of Agreed Exchange rate."
      },
      "rateType": {
        "type": "string",
        "enum": [
          "AGREED",
          "SALE",
          "SPOT"
        ]
      }
    }
  },
  "contractIdentification": {

```

```

        "type": "string",
        "maxLength": 35,
        "description": "Detail of Contract agreed with client for the execution of payments
at the Agreed Exchange rate"
    }
},
"Debtor": {
    "type": "object",
    "description": "A party that owes money to the payee.",
    "properties": {
        "name": {
            "type": "string",
            "maxLength": 70,
            "description": "Name under which the party is known."
        },
        "postalAddress": {
            "$ref": "#/definitions/PostalAddress"
        }
    }
},
"Creditor": {
    "type": "object",
    "description": "Payee of the transaction.",
    "properties": {
        "name": {
            "type": "string",
            "maxLength": 105,
            "description": "Name under which the party is known."
        },
        "postalAddress": {
            "$ref": "#/definitions/PostalAddress"
        }
    }
},
"RemittanceInformation": {
    "type": "object",
    "description": "Information that allows match (i.e. pairing, reconciliation) of
payments with which the payment should be settled.",
    "properties": {
        "unstructured": {
            "type": "string",
            "maxLength": 104,
            "description": "Unstructured message for payee."
        },
        "structured": {
            "type": "string",
            "maxLength": 140,
            "description": "Structured message for payee."
        }
    }
},
"SignInfo": {
    "type": "object",
    "required": [
        "state",
        "signId"
    ],
    "properties": {
        "state": {
            "$ref": "#/definitions/State"
        },
        "signId": {
            "type": "string"
        }
    }
},
"description": "Status information and id of unauthorized transactions"
},
"State": {
    "type": "string",
    "enum": [
        "OPEN",
        "REJECTED",
        "CLOSED",
        "NONE"
    ],
},

```

```

    "description": "Transaction authorisation status"
  },
  "StandingOrderInitiationRequest": {
    "description": "Structure describing the parameters of the standing orders.\n",
    "type": "object",
    "required": [
      "standingOrderIdentification",
      "amount",
      "requestedExecutionDate",
      "standingOrder",
      "debtorAccount",
      "creditorAccount"
    ],
    "properties": {
      "standingOrderIdentification": {
        "$ref": "#/definitions/StandingOrderIdentificationRequest"
      },
      "paymentTypeInformation": {
        "$ref": "#/definitions/PaymentType"
      },
      "amount": {
        "$ref": "#/definitions/Amount"
      },
      "requestedExecutionDate": {
        "$ref": "#/definitions/RequestedExecutionDate"
      },
      "standingOrder": {
        "$ref": "#/definitions/StandingOrder"
      },
      "debtor": {
        "$ref": "#/definitions/Debtor"
      },
      "debtorAccount": {
        "$ref": "#/definitions/DebtorAccount"
      },
      "creditor": {
        "$ref": "#/definitions/Creditor"
      },
      "creditorAccount": {
        "$ref": "#/definitions/CreditorAccount"
      },
      "remittanceInformation": {
        "$ref": "#/definitions/RemittanceInformation"
      },
      "purpose": {
        "$ref": "#/definitions/PaymentPurpose"
      }
    }
  },
  "StandingOrderInitiationResponse": {
    "description": "Structure describing the parameters of the standing orders.\n",
    "type": "object",
    "required": [
      "standingOrderIdentification",
      "paymentTypeInformation",
      "signInfo"
    ],
    "properties": {
      "standingOrderIdentification": {
        "$ref": "#/definitions/StandingOrderIdentification"
      },
      "paymentTypeInformation": {
        "$ref": "#/definitions/PaymentType"
      },
      "signInfo": {
        "$ref": "#/definitions/SignInfo"
      }
    }
  },
  "StandingOrderIdentificationRequest": {
    "description": "Standing order identification request\n",
    "type": "object",
    "required": [
      "instructionIdentification"
    ],
    "properties": {

```

```

    "instructionIdentification": {
      "$ref": "#/definitions/InstructionIdentification"
    },
    "endToEndIdentification": {
      "$ref": "#/definitions/EndToEndIdentification"
    }
  }
},
"StandingOrderStatusResponse": {
  "type": "object",
  "required": [
    "instructionStatus"
  ],
  "description": "Status of entered payment",
  "properties": {
    "instructionStatus": {
      "$ref": "#/definitions/StatusCode"
    },
    "realizedPayments": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/RealizedPayment"
      }
    }
  }
},
"RealizedPayment": {
  "description": "Details of realized payment.\n",
  "type": "object",
  "required": [
    "transactionIdentification"
  ],
  "properties": {
    "transactionIdentification": {
      "type": "string",
      "maxLength": 35,
      "description": "Identifier of established transaction"
    },
    "date": {
      "type": "string",
      "format": "date",
      "description": "Due date/payment foreign currency in the format ISODate, , or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.STZD, depending on the transaction type and method how the bank presents data (and time) of due date/ payment foreign currency. Mainly for card or cash"
    }
  }
},
"StatusCode": {
  "type": "string",
  "enum": [
    "ACTC",
    "RJCT",
    "ACSP",
    "ACSC",
    "ACWC"
  ]
},
"PaymentSigningResponse": {
  "type": "object",
  "required": [
    "scenarios",
    "signInfo"
  ],
  "properties": {
    "scenarios": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "description": "A set of possible authorization scenarios"
  },
  "signInfo": {
    "$ref": "#/definitions/SignInfo"
  }
}
}

```

```

    },
    "AccountsListLinks": {
      "type": "object",
      "properties": {
        "self": {
          "type": "string",
          "description": "The link to the resource itself"
        },
        "first": {
          "type": "string",
          "description": "GET the first page of the account list"
        },
        "last": {
          "type": "string",
          "description": "GET the last page of the account list"
        },
        "next": {
          "type": "string",
          "description": "GET the next page of the account list"
        },
        "previous": {
          "type": "string",
          "description": "GET the previous page of the account list"
        }
      }
    },
    "AccountLinks": {
      "type": "object",
      "properties": {
        "balance": {
          "type": "string",
          "description": "GET the account's balance(s)"
        },
        "transactions": {
          "type": "string",
          "description": "GET the list of transactions on the account"
        },
        "balanceCheck": {
          "type": "string",
          "description": "POST an inquiry about availability of funds. Requires the
authorization of user/client as part of the API call"
        },
        "thirdPartyBalanceCheck": {
          "type": "string",
          "description": "POST an inquiry about availability of funds. Does not require the
authorization of user/client as part of the API call"
        },
        "payment": {
          "type": "string",
          "description": "POST a new payment. Requires the authorization of user/client as
part of the API call"
        }
      }
    },
    "BalancesListLinks": {
      "type": "object",
      "properties": {
        "self": {
          "type": "string",
          "description": "The link to the resource itself"
        },
        "transactions": {
          "type": "string",
          "description": "GET the list of transactions on the account"
        },
        "accounts": {
          "type": "string",
          "description": "GET the list of accounts"
        },
        "payment": {
          "type": "string",
          "description": "POST a new payment. Requires the authorization of user/client as
part of the API call"
        }
      }
    }
  },

```



```

"TransactionsListLinks": {
  "type": "object",
  "properties": {
    "self": {
      "type": "string",
      "description": "The link to the resource itself"
    },
    "first": {
      "type": "string",
      "description": "GET the first page of the account list"
    },
    "last": {
      "type": "string",
      "description": "GET the last page of the account list"
    },
    "next": {
      "type": "string",
      "description": "GET the next page of the account list"
    },
    "previous": {
      "type": "string",
      "description": "GET the previous page of the account list"
    },
    "balance": {
      "type": "string",
      "description": "GET the account's balance(s)"
    },
    "balanceCheck": {
      "type": "string",
      "description": "POST an inquiry about availability of funds. Requires the
authorization of user/client as part of the API call"
    },
    "thirdPartyBalanceCheck": {
      "type": "string",
      "description": "POST an inquiry about availability of funds. Does not require the
authorization of user/client as part of the API call"
    },
    "payment": {
      "type": "string",
      "description": "POST a new payment. Requires the authorization of user/client as
part of the API call"
    },
    "accounts": {
      "type": "string",
      "description": "GET the list of accounts"
    }
  }
}
}
}

```