



PSA VOP SERVICE API DESCRIPTION SINGLE DATA REQUEST

FOR BANK EMPLOYEES, FOR INTERNAL USE ONLY
VERSION 1.0, 28.02.2025

PREAMBLE

Dear PSA VoP Service-participant,

If you decided to use the PSA VoP Service with Option B, this document describes the PSA VoP Service API for Single Data Request.

The following specification includes the technology used and the prerequisites needed to send VoP requests and receive according responses.

The current version of this document is intended as a guide and overview.

We look forward to continuing our successful and constructive cooperation!



PSA Payment Services Austria GmbH

Handelskai 92
Gate 2, 6th floor
1200 Vienna
office@psa.at
www.psa.at

TABLE OF CONTENTS

CHANGE HISTORY	4
1. GENERAL DESCRIPTION	5
2. API ENDPOINT DOMAINS DNS NAMES	5
3. API ENDPOINTS.....	5
3.1 Single Payee DATA	5
3.1.1 Request Header (Example)	5
3.1.2 Request Body (Example).....	6
3.1.3 Full Request – (Example)	6
3.1.4 Server Response (Example)	6
3.1.5 Response Codes.....	9
3.2 Schemes.....	10

CHANGE HISTORY

Version	Author	Date	Description of change
1.0	PSA	28.02.2025	Creation of the document
1.1.	PSA	04.08.2025	Typo fix

1. GENERAL DESCRIPTION

This request is called by PSA VoP service and response has to be provided by the financial institution being member that hasn't provided data to PSA, but using the matching facilities.

2. API ENDPOINT DOMAINS DNS NAMES

Calling the APIs differs in the various environments:

DOMAIN TEST	api-test.vop.psa.at
DOMAIN PRODUCTION	api.vop.psa.at

All API URLs are built by combining the domain with the path of the required environment.

3. API ENDPOINT

3.1 SINGLE PAYEE DATA

URL:	https://[domain]/{BIC}/vop/v1/payee-data
Method:	POST

3.1.1 REQUEST HEADER (EXAMPLE)

Name	M/O ¹	Description
X-Request-ID <code>string(\$uuid)</code>	M	Unique ID for the request - is defined by the initiator: e.g.: 2313ab62-02e8-4f97-9c3f-6eeb6abd2a76 for linking the response to the request
X-Request- Timestamp <code>string(\$date-time)</code>	M	Timestamp of the requests: 2024-12-12T12:12:12Z 2024-12-12T12:12:12.02Z 2024-12-12T12:12:10+02
X-Original-Re- quest-Id <code>string(\$uuid)</code>	M	Unique ID for the request - is defined by the initiator: e.g.: 2313ab62-02e8-4f97-9c3f-6eeb6abd2a76 for linking the response to the request
X-Original-Re- quest- Timestamp <code>string(\$date-time)</code>	M	Timestamp of the requests: 2024-12-12T12:12:12Z 2024-12-12T12:12:12.02Z 2024-12-12T12:12:10+02
X-Original-Re- questers-BIC * <code>string</code>	O	ABCD BEBB FIM

¹ Mandatory / Optional

3.1.2 REQUEST BODY (EXAMPLE)

```
{
  "partyAccount": {
    "iban": "AB121234567812345678901234"
  }
}
```

3.1.3 FULL REQUEST – (EXAMPLE)

```
curl -X 'POST' \
  'https://api.vop.psa.at/%7BBIC%7D/vop/v1/payee-data' \
  -H 'accept: application/json' \
  -H 'X-Request-ID: 9f45a3e5-aa49-43d6-9f1c-36ae3b5a295e' \
  -H 'X-Request-Timestamp: 2024-12-12T12:12:12.02Z' \
  -H 'X-Original-Request-Id: 9f45a3e5-aa49-43d6-9f1c-36ae3b5a295e' \
  -H 'X-Original-Request-Timestamp: 2024-12-12T12:12:12.002Z' \
  -H 'X-Original-Requesters-BIC: ABCDBEBBFIM' \
  -H 'Content-Type: application/json' \
  -d '{
    "partyAccount": {
      "iban": "AB121234567812345678901234"
    }
  }'
```

3.1.4 SERVER RESPONSE (EXAMPLE)

Code	Details
200	Data assigned by the Responder to be used for VoP processing by the RVM (Routing and/or Verification Mechanism) <pre>{ "partyAccount": { "iban": "AB121234567812345678901234" }, "accountSince": "2024-12-12Z", "associatedNamesAndIds": { "person": [{ "additionalInformation": { "salutation": "string", "academicTitles": [{ "title": "string", "position": "before" }] }, "otherTitles": [</pre>

```

        "string"
      ],
      "knownNames": [
        "string"
      ]
    },
    "firstNames": [
      "string"
    ],
    "lastNames": [
      "string"
    ],
    "name": "string",
    "aliases": [
      "string"
    ],
    "address": {
      "department": "string",
      "subDepartment": "string",
      "streetName": "string",
      "buildingNumber": "string",
      "buildingName": "string",
      "floor": "string",
      "postBox": "string",
      "room": "string",
      "postCode": "string",
      "townName": "string",
      "townLocationName": "string",
      "districtName": "string",
      "countrySubDevision": "string",
      "country": "AT"
    },
    "birthName": "string",
    "officialGender": "Other",
    "dateAndPlaceOfBirth": {
      "birthDay": "2024-12-12Z",
      "provinceOfBirth": "string",
      "cityOfBirth": "string",
      "countryOfBirth": "AT"
    },
    "identification": [
      {
        "identification": "string",
        "schemeName": {
          "code": "CCPT"
        },
        "issuer": "string"
      }
    ]
  },
  "organisation": [
    {
      "type": "string",
      "identification": {
        "organisationId": {
          "lei": "549300DTUYXVMJXZNY75"
        }
      },
      "names": [
        "string"
      ],
      "aliases": [
        "string"
      ]
    }
  ]
}

```

```
    ],  
    "address": {  
      "department": "string",  
      "subDepartment": "string",  
      "streetName": "string",  
      "buildingNumber": "string",  
      "buildingName": "string",  
      "floor": "string",  
      "postBox": "string",  
      "room": "string",  
      "postCode": "string",  
      "townName": "string",  
      "townLocationName": "string",  
      "districtName": "string",  
      "countrySubDevision": "string",  
      "country": "AT"  
    }  
  }  
]  
}
```

Response Headers

```
content-type: application/json; charset=utf-8  
server: Kestrel  
transfer-encoding: chunked  
x-request-id: 2313ab62-02e8-4f97-9c3f-6eeb6abd2a76  
x-response-timestamp: 2024-12-12T12:12:13.751Z'
```


3.1.5 RESPONSE CODES

Code	Details
400	Bad Request - malformed payload.
Example Value scheme	<pre>{ "type": "https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/400", "code": "FORMAT_ERROR", "title": "Malformed JSON", "status": 400, "detail": "Request Body could not be parsed or otherwise recognised" }</pre>
403	Forbidden - authorisation issue, certificate not acceptable, e.g. certificate's NAN do not match BIC in request, etc.
Example Value scheme	<pre>{ "type": "https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/403", "code": "CLIENT_INCONSISTENT", "title": "NAN and BIC don't match", "status": 403, "detail": "BIC derived from NAN of certificate doesn't match BIC in request" }</pre>
404	Not Found - resource not existing, e.g. resource according path in request not existing.
Example Value scheme	<pre>{ "type": "https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/404", "code": "NOT_FOUND", "title": "Cannot find the requested resource", "status": 404, "detail": "Resource is missing or has never existed" }</pre>
409	Conflict - header carry duplicate entry, i.e. request id already used.
Example Value scheme	<pre>{ "type": "https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/409", "code": "DUPLICATE", "title": "RequestId already in use", "status": 409, "detail": "The request identification is already in use and therefore cannot be accepted" }</pre>
500	Internal Server Error - technical problem occurred in the application code or application server preventing request execution. Timeout
Example Value scheme	<pre>{ "type": "https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/500", "code": "SERVER_ERROR", "title": "Internal error occurred while processing request", "status": 500, "detail": "Server reported an error during computing. Technical personal has been informed about this issue" }</pre>



Details

BICType **string**

pattern: `^[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3})?$`

example: `ABCDDEBBFIM`

Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.

IBANType **string**

pattern: `^[A-Z]{2}[0-9]{2}[A-Z0-9]{1,30}$`

example: `AB121234567812345678901234`

InternationalBankAccountNumber according ISO 13616, as assigned by the responding PSP.

LEIType **string**

pattern: `^[A-Z0-9]{18}[0-9]{2}$`

example: `549300DTUYXVMJXZNY75`

Code LEI (Legal Entity Identifier) according ISO 17442.

UUIDType **string**(\$uuid)

pattern: `^[a-f0-9]{8}-[a-f0-9]{4}-4[a-f0-9]{3}-[89ab][a-f0-9]{3}-[a-f0-9]{12}$`

example: `9f45a3e5-aa49-43d6-9f1c-36ae3b5a295e`

A UUID according RFC 4122 Version 4.

academicTitleType{

description: Acquired academic title and title position relative to name

title* max35Textstring

minLength: 1

maxLength: 35

position*

{

}

}

accountType{

description: InternationalBankAccountNumber according ISO 13616, as assigned by the responding PSP.

iban*

IBANType **string**

pattern: `^[A-Z]{2}[0-9]{2}[A-Z0-9]{1,30}$`

example: `AB121234567812345678901234`

InternationalBankAccountNumber according ISO 13616, as assigned by the responding PSP.

```
}
agentType{
  description:      Financial Institution Identifier within clearing and/or settlement systems.
```

```
  financialInsti-  financialInstitutionId{
    tutionId*      description:      Financial Institution Identifier within clearing and/or settlement systems.

                  bicfi*          BICType string
                                   pattern: ^[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3})?$
                                   example: ABCDBEBBFIM
                                   Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.
  }
}
```

```
addressType{
  description:      Address information

  department        max70Text string
                   minLength: 1
                   maxLength: 70
  subDepartment      max70Text string
                   minLength: 1
                   maxLength: 70
  streetName         max70Text string
                   minLength: 1
                   maxLength: 70
  buildingNumber     max16Text string
                   minLength: 1
                   maxLength: 16
  buildingName       max35Text string
                   minLength: 1
                   maxLength: 35
  floor              max70Text string
                   minLength: 1
                   maxLength: 70
  postBox            max16Text string
                   minLength: 1
                   maxLength: 16
  room               max70Text string
                   minLength: 1
                   maxLength: 70
  postCode           max16Text string
                   minLength: 1
                   maxLength: 16
}
```

```

townName*      max35Text string
                minLength: 1
                maxLength: 35
townLocationName max35Text string
                minLength: 1
                maxLength: 35
districtName    max35Text string
                minLength: 1
                maxLength: 35
countrySubDivision max35Text string
                minLength: 1
                maxLength: 35
country*        countryCode string
                pattern: ^[A-Z]{2}$
                example: AT
                Two alpha country code according ISO 3166

```

```

}

```

```

birthType{
  description:    Information on person's birth

  birthDay*      dateType string($date-time)
                pattern: ^\d{4}(-\d{2}){2}(Z|([-+]\d{2}(:\d{2})?))$
                example: 2024-12-12Z
                Date in ISO 8601 format, either UTC (YYYY-MM-DDZ) or local time with UTC offset (YYYY-MM-DD+/-hh:mm).

  provinceOfBirth max35Text string
                minLength: 1
                maxLength: 35
  cityOfBirth*    max35Text string
                minLength: 1
                maxLength: 35
  countryOfBirth* countryCode string
                pattern: ^[A-Z]{2}$
                example: AT
                Two alpha country code according ISO 3166

```

```

}

```

```

countryCodestring
pattern: ^[A-Z]{2}$
example: AT

```

```

Two alpha country code according ISO 3166

```

```
dateType string($date-time)
pattern: ^\d{4}(-\d{2}){2}T\d{2}(:\d{2}){2}(\.\d{0,2}[1-9])?([Z]([-+]\d{2}(:\d{2})?)?)$
example: 2024-12-12T12:12:12Z ; 2024-12-12T12:12:12.02Z ; 2024-12-12T12:12:10+02
```

Timestamp in ISO 8601 format, either UTC (YYYY-MM-DDThh:mm:ss.sssZ) or local time with UTC offset (YYYY-MM-DDThh:mm:ss.sss+/-hh:mm), milliseconds precision (observe truncation rules!).

```
dateType string($date-time)
pattern: ^\d{4}(-\d{2}){2}[Z]([-+]\d{2}(:\d{2})?)?)$
example: 2024-12-12Z
```

Date in ISO 8601 format, either UTC (YYYY-MM-DDZ) or local time with UTC offset (YYYY-MM-DD+/-hh:mm).

```
genderType string
example: Other
```

Gender of person from documents used in KYC process

```
Enum:
[ Male, Female, Other ]
```

```
genericOrganisationId{
  description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier issuer
                information
```

oneOf ->	organisationCodedId{ description: Unique and unambiguous identification of an organisation according provided coded scheme within issuers space.
identification*	max256Text string minLength: 1 maxLength: 256
schemeNameCode*	organisationIdCode string example: BOID Specifies the external organisation identification scheme name code (maximum 4 characters) as defined in an external list (iso20022.org).
	Enum: [BANK, CBID, CHID, CINC, COID, CUST, DUNS, EMPL, GS1G, SREN, SRET, TXID, BDID, BOID]
issuer	max35Text string minLength: 1 maxLength: 35
	}
	organisationProprietaryId{

description: Unique and unambiguous identification of an organisation according provided proprietary scheme eventually within issuers space.

identification* max256Text string
minLength: 1
maxLength: 256

schemeNameProprietary* max35Text string
minLength: 1
maxLength: 35

issuer max35Text string
minLength: 1
maxLength: 35

}

}

genericPersonId{

description: A person identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier issuer information

identification* max35Text string
minLength: 1
maxLength: 35

schemeName

{

oneOf ->

schemeCode{

description: Coded information about the kind and/or nature of a code

code*

schemeType string

example: CCPT

Codes for the kind and/or nature of a code

Enum:

[ARNU, CCPT, CUST, DRIC, EMPL, NIDN, POID, SOSE, TELE, TXID]

}

proprietaryName{

description: Proprietary information about the kind and/or nature of a code

proprietary*

max35Text string

minLength: 1

maxLength: 35

}

}

issuer

max35Text string

minLength: 1

maxLength: 35

}

```
idAsLEI{
  description: Legal Entity Identifier according ISO 17442.

  lei      LEIType string
           pattern: ^[A-Z0-9]{18}[0-9]{2}$
           example: 549300DTUYXVMJXZNY75
           Code LEI (Legal Entity Identifier) according ISO 17442.

}
```

```
idAsBIC{
  description: Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.

  anyBIC    BICType string
           pattern: ^[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3})?$
           example: ABCDEBBFIM
           Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.

}
```

```
idAsOther{
  description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier issuer information.
               Although the structure is an array, only one identifier is allowed

  others    [
             maxItems: 1
             genericOrganisationId{
               description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier
                           issuer information

               oneOf -> organisationCodedId{
                 description: Unique and unambiguous identification of an organisation according provided coded scheme within issuers space.

                 identification* max256Text string
                               minLength: 1
                               maxLength: 256

                 schemeNameCode* organisationIdCode string
                               example: BOID
                               Specifies the external organisation identification scheme name code (maximum 4 characters) as defined in an external list
                               (iso20022.org).

                 Enum:
                 [ BANK, CBID, CHID, CINC, COID, CUST, DUNS, EMPL, GS1G, SREN, SRET, TXID, BDID, BOID ]
               }
             ]
}
```



```

        issuer          max35Text string
                        minLength: 1
                        maxLength: 35
    }
    organisationProprietaryId{
        description:      Unique and unambiguous identification of an organisation according provided proprietary scheme eventually within
                           issuers space.

        identification*    max256Text string
                           minLength: 1
                           maxLength: 256
        schemeNameProprietary* max35Text string
                           minLength: 1
                           maxLength: 35
        issuer            max35Text string
                           minLength: 1
                           maxLength: 35
    }
}
}
}

```

```

idType{
    description:      Unique identification of an organisation, as assigned by an institution, using an identification scheme.

    organisationId*   organisationIdType{
        description:    Unique identification of an organisation, as assigned by an institution, using an identification scheme.

        oneOf ->
            idAsLEI{
                description: Legal Entity Identifier according ISO 17442.

                lei          LEIType string
                            pattern: ^[A-Z0-9]{18}[0-9]{2}$
                            example: 549300DTUYXVMJXZNY75
                            Code LEI (Legal Entity Identifier) according ISO 17442.
            }
            idAsBIC{
                description: Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.

                anyBIC       BICType string
                            pattern: ^[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3})?$
                            example: ABCDEBBFIM
                            Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.
            }
            idAsOther{

```

```

description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier issuer information. Although the structure is an array, only one identifier is allowed

others [
  maxItems: 1
  genericOrganisationId{
    description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier issuer information

    oneOf -> organisationCodedId{
      description: Unique and unambiguous identification of an organisation according provided coded scheme within issuers space.

      identification* max256Text string
        minLength: 1
        maxLength: 256
      schemeNameCode* organisationIdCode string
        example: BOID
        Specifies the external organisation identification scheme name code (maximum 4 characters) as defined in an external list (iso20022.org).

        Enum:
        [ BANK, CBID, CHID, CINC, COID, CUST, DUNS, EMPL, GS1G, SREN, SRET, TXID, BDID, BOID ]
      issuer max35Text string
        minLength: 1
        maxLength: 35
    }
    organisationProprietaryId{
      description: Unique and unambiguous identification of an organisation according provided proprietary scheme eventually within issuers space.

      identification* max256Text string
        minLength: 1
        maxLength: 256
      schemeNameProprietary* max35Text string
        minLength: 1
        maxLength: 35
      issuer max35Text string
        minLength: 1
        maxLength: 35
    }
  }
]
}
}

```

```
max140Text string
minLength: 1
maxLength: 140
```

```
max16Text string
minLength: 1
maxLength: 16
```

```
max256Text string
minLength: 1
maxLength: 256
```

```
max35Text string
minLength: 1
maxLength: 35
```

```
max500Text string
minLength: 1
maxLength: 500
```

```
max70Text string
minLength: 1
maxLength: 70
```

```
messageCode string
example: TIMESTAMP_INVALID
```

Specifies the codes Error messages.

```
Enum:
[ FORMAT_ERROR, CLIENT_INVALID, CLIENT_INCONSISTENT, TIMESTAMP_INVALID, DUPLICATE, NOT_FOUND, UNKNOWN_ERROR ]
```

```
organisationCodedId {
  description: Unique and unambiguous identification of an organisation according provided coded scheme within issuers space.

  identification* max256Text string
    minLength: 1
    maxLength: 256
  schemeNameCode* organisationIdCode string
    example: BOID
    Specifies the external organisation identification scheme name code (maximum 4 characters) as defined in an external list (iso20022.org).

  Enum:
  [ BANK, CBID, CHID, CINC, COID, CUST, DUNS, EMPL, GSIG, SREN, SRET, TXID, BDID, BOID ]
  issuer max35Text string
    minLength: 1
    maxLength: 35
}
```

organisationIdCode **string**
example: BOID

Specifies the external organisation identification scheme name code (maximum 4 characters) as defined in an external list (iso20022.org).

Enum:
 [BANK, CBID, CHID, CINC, COID, CUST, DUNS, EMPL, GS1G, SREN, SRET, TXID, BDID, BOID]

```

organisationIdType{
  description: Unique identification of an organisation, as assigned by an institution, using an identification scheme.
  tion:

  oneOf - idAsLEI{
    >      description: Legal Entity Identifier according ISO 17442.

    lei      LEIType string
            pattern: ^[A-Z0-9]{18}[0-9]{2}$
            example: 549300DTUXXVMJXZNY75
            Code LEI (Legal Entity Identifier) according ISO 17442.

  }
  idAsBIC{
    description: Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.

    anyBIC      BICType string
            pattern: ^[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3})?$
            example: ABCDBBBFIM
            Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.

  }
  idAsOther{
    description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier issuer information. Although the structure is an array, only one identifier is allowed

    others      [
            maxItems: 1
            genericOrganisationId{
              description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier issuer information

              oneOf ->      organisationCodedId{
                description: Unique and unambiguous identification of an organisation according provided coded scheme within issuers space.
              }
            }
          ]
  }
}

```

```

        identification* max256Text string
                        minLength: 1
                        maxLength: 256
        schemeNameCode* organisationIdCode string
                        example: BOID
                        Specifies the external organisation identification scheme name code (maximum 4 characters) as
                        defined in an external list (iso20022.org).

        issuer Enum:
        [ BANK, CBID, CHID, CINC, COID, CUST, DUNS, EMPL, GSIG, SREN, SRET, TXID, BDID, BOID ]
        max35Text string
        minLength: 1
        maxLength: 35
    }
    organisationProprietaryId{
        description: Unique and unambiguous identification of an organisation according provided proprietary
        scheme eventually within issuers space.

        identification* max256Text string
                        minLength: 1
                        maxLength: 256
        schemeNameProprietary* max35Text string
                        minLength: 1
                        maxLength: 35
        issuer max35Text string
                minLength: 1
                maxLength: 35
    }
}]]
}

```

```

organisationProprietaryId{
    description: Unique and unambiguous identification of an organisation according provided proprietary scheme eventually within issuers space.

    identification* max256Text string
                    minLength: 1
                    maxLength: 256
    schemeNameProprietary* max35Text string
                            minLength: 1
                            maxLength: 35
    issuer max35Text string
            minLength: 1
            maxLength: 35
}

```

```

organisationType{
  description: Structure to hold information about an organisation

  type          max35Text string
                minLength: 1
                maxLength: 35

  identification idType{
    description: Unique identification of an organisation, as assigned by an institution, using an identification scheme.

    organisationId* organisationIdType{
      description: Unique identification of an organisation, as assigned by an institution, using an identification scheme.

      oneOf -> idAsLEI{
        description: Legal Entity Identifier according ISO 17442.

        lei      LEIType string
                  pattern: ^[A-Z0-9]{18}[0-9]{2}$
                  example: 549300DTUYXVMJXZNY75
                  Code LEI (Legal Entity Identifier) according ISO 17442.

      }
      idAsBIC{
        description: Business Identifier Code according ISO 9362. May identify financial institutions and other organisation

        anyBIC      BICType string
                     pattern: ^[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3})?$
                     example: ABCDBEBBFIM
                     Business Identifier Code according ISO 9362. May identify financial institutions and other organisation

      }
      idAsOther{
        description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNamePr
                     and optionally an identifier issuer information. Although the structure is an array, only one identifie

        others      [genericOrganisationId{
                      description: An organisation identifier and an identifier type qualified by either 'schemeName-
                               Code' or 'schemeNameProprietary' and optionally an identifier issuer information

                      oneOf -> organisationCodedId{
                        description: Unique and unambiguous identification of an organi-
                               sation according provided coded scheme within issu-
                               ers space.

                        identification* max256Text string
                                       minLength: 1
                                       maxLength: 256

```

```

}
names      [max140Text string
            minLength: 1
            maxLength: 140]
aliases    [max140Text string
            minLength: 1
            maxLength: 140]
address    addressType{
            description:    Address information

            department      max70Text string
                           minLength: 1
                           maxLength: 70
            subDepartment   max70Text string
                           minLength: 1
                           maxLength: 70
            streetName      max70Text string
                           minLength: 1
                           maxLength: 70

```

```

buildingNum-    max16Text string
ber             minLength: 1
                maxLength: 16
buildingName    max35Text string
                minLength: 1
                maxLength: 35
floor           max70Text string
                minLength: 1
                maxLength: 70
postBox         max16Text string
                minLength: 1
                maxLength: 16
room            max70Text string
                minLength: 1
                maxLength: 70
postCode        max16Text string
                minLength: 1
                maxLength: 16
townName*       max35Text string
                minLength: 1
                maxLength: 35
townLocation-   max35Text string
Name            minLength: 1
                maxLength: 35
districtName    max35Text string
                minLength: 1
                maxLength: 35
countrySubDe-   max35Text string
vision          minLength: 1
                maxLength: 35
country*        countryCode string
                pattern: ^[A-Z]{2}$
                example: AT
                Two alpha country code according ISO 3166

```

```

}

```

```

}

```

```

partyData{
  descrip-  Structure holding information about a party / parties associated with an account.
  tion:
  person    [personType{
              description: Structure to hold information about a person

              additionalInformation titlesType{
                description:  Structure for titles and other not categorised information connected to a person

                salutation    max35Text string
                             minLength: 1
                             maxLength: 35
                academicTitles [...]
                otherTitles    [...]
              }
            }

```



```

        knownNames    [...]
    }
firstNames    [(max140Text string
               minLength: 1
               maxLength: 140)]
lastNames    [(max140Text string
               minLength: 1
               maxLength: 140)]
name         max140Text string
               minLength: 1
               maxLength: 140
aliases     [(max140Text string
               minLength: 1
               maxLength: 140)]
address      addressType{
               description:    Address information

               department      max70Text string
                               minLength: 1
                               maxLength: 70
               subDepartment   max70Text string
                               minLength: 1
                               maxLength: 70
               streetName      max70Text string
                               minLength: 1
                               maxLength: 70
               buildingNumber   max16Text string
                               minLength: 1
                               maxLength: 16
               buildingName     max35Text string
                               minLength: 1
                               maxLength: 35
               floor            max70Text string
                               minLength: 1
                               maxLength: 70
               postBox          max16Text string
                               minLength: 1
                               maxLength: 16
               room             max70Text string
                               minLength: 1
                               maxLength: 70
               postCode         max16Text string
                               minLength: 1
                               maxLength: 16
               townName*       max35Text string
                               minLength: 1
                               maxLength: 35
               townLocationName max35Text string
                               minLength: 1
                               maxLength: 35
               districtName     max35Text string
                               minLength: 1
                               maxLength: 35

```

```

        countrySubDev-    max35Text string
        sion              minLength: 1
                        maxLength: 35
        country*          countryCode string
                        pattern: ^[A-Z]{2}$
                        example: AT
                        Two alpha country code according ISO 3166
    }
    birthName            max140Text string
                        minLength: 1
                        maxLength: 140
    officialGender       genderType string
                        example: Other
                        Gender of person from documents used in KYC process

    Enum:
    [ Male, Female, Other ]
    dateAnd-             birthType{
    PlaceOfBirth          description:      Information on person's birth

                        birthDay*          dateType string($date-time)
                        pattern: ^\d{4}(-\d{2}){2}{Z|([-+]\d{2}(:\d{2})?)}$
                        example: 2024-12-12Z
                        Date in ISO 8601 format, either UTC (YYYY-MM-DDZ) or local time with UTC offset (YYYY-MM-DD+/-hh:mm).

                        provinceOfBirth    max35Text string
                        minLength: 1
                        maxLength: 35
                        cityOfBirth*        max35Text string
                        minLength: 1
                        maxLength: 35
                        countryOfBirth*     countryCode string
                        pattern: ^[A-Z]{2}$
                        example: AT
                        Two alpha country code according ISO 3166
    }
    identifica-         [genericPersonId{
    tion                 description:      A person identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier
                        information

                        identification*     max35Text string
                        minLength: 1
                        maxLength: 35
                        schemeName          {
                                oneOf ->    schemeCode{
                                        description: Coded information about the kind and/or nature of a code

```

```

        code*      schemeType string
                    example: CCPT
                    Codes for the kind and/or nature of a code

                    Enum:
                    [ ARNU, CCPT, CUST, DRLC, EMPL, NIDN, POID, SOSE, TELE, TXID ]

    }
    proprietaryName{
        description: Proprietary information about the kind and/or nature of a code

        proprietary* max35Text string
                     minLength: 1
                     maxLength: 35

    }

    issuer          max35Text string
                     minLength: 1
                     maxLength: 35

    ]]

organisation [organisationType{
    description: Structure to hold information about an organisation

    type        max35Text string
                 minLength: 1
                 maxLength: 35

    identification idType{
        description: Unique identification of an organisation, as assigned by an institution, using an identification scheme.

        organisationId* organisationIdType{
            description: Unique identification of an organisation, as assigned by an institution, using an identification scheme.

            oneOf -> idAsLEI{
                descrip- Legal Entity Identifier according ISO 17442.
                tion:
                lei      LEIType string
                         pattern: ^[A-Z0-9]{18}[0-9]{2}$
                         example: 549300DTUYXVMJXZNY75
                         Code LEI (Legal Entity Identifier) according ISO 17442.

            }
            idAsBIC{
                descrip- Business Identifier Code according ISO 9362. May identify financial institutions and
                tion:      other organisations.
            }
        }
    }
}

```

```

anyBIC      BICTypeBICTypestring
            pattern: ^[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3})?
            example: ABCDBBBBIFIM
            Business Identifier Code according ISO 9362. May identify financial institutions and
            other organisations.

}
idAsOther{
  descrip-  An organisation identifier and an identifier type qualified by either 'schemeNameCode' or
tion:       'schemeNameProprietary' and optionally an identifier issuer information. Although the
            structure is an array, only one identifier is allowed

  others   [genericOrganisationId{
            descrip-  An organisation identifier and an identifier type quali-
tion:       fied by either 'schemeNameCode' or 'schemeName-
            Proprietary' and optionally an identifier issuer infor-
            mation

            oneOf ->  organisationCodedId{
                        description: Unique and unambiguous identi-
                                fication of an organisation
                                according provided coded
                                scheme within issuers space.

                        identifica-  max256Text string
tion*       minLength: 1
                        maxLength: 256
                        schemeName-  organisationIdCode string
Code*       example: BOID
                        Specifies the external organi-
                        sation identification scheme
                        name code (maximum 4 charac-
                        ters) as defined in an exter-
                        nal list (iso20022.org).

                        Enum:
                        [ BANK, CBID, CHID, CINC,
                        COID, CUST, DUNS, EMPL, GS1G,
                        SREN, SRET, TXID, BDID, BOID ]

                        issuer      max35Textm string
                                minLength: 1
                                maxLength: 35
                    }
            organisationProprietaryId{
                        description: Unique and unambiguous identi-
                                fication of an organisation
                                according provided proprietary
                                scheme eventually within issu-
                                ers space.

```

```

identifica- max256Text string
tion*      minLength: 1
           maxLength: 256
schemeName- max35Text string
Proprietary* minLength: 1
           maxLength: 35
issuer      max35Text string
           minLength: 1
           maxLength: 35

```

```

    }
  }
}

```

```

}
names      [max140Text string
           minLength: 1
           maxLength: 140]
aliases    [max140Text string
           minLength: 1
           maxLength: 140]
address     addressType{
  description: Address information

  department max70Text string
             minLength: 1
             maxLength: 70
  subDepartment max70Text string
               minLength: 1
               maxLength: 70
  streetName max70Text string
             minLength: 1
             maxLength: 70
  buildingNumber max16Text string
                 minLength: 1
                 maxLength: 16
  buildingName max35Text string
               minLength: 1
               maxLength: 35
  floor max70Text string
        minLength: 1
        maxLength: 70
  postBox max16Text string
          minLength: 1
          maxLength: 16
  room max70Text string
        minLength: 1
        maxLength: 70
  postCode max16Text string
           minLength: 1
           maxLength: 16
  townName* max35Text string
            minLength: 1
            maxLength: 35
}

```

```

    townLocation-    max35Text string
    Name             minLength: 1
                    maxLength: 35

    districtName     max35Text string
                    minLength: 1
                    maxLength: 35

    countrySubDe-    max35Text string
    vision           minLength: 1
                    maxLength: 35

    country*         countryCode string
                    pattern: ^[A-Z]{2}$
                    example: AT
                    Two alpha country code according ISO 3166
}

}
}

```

```

personType{
  description: Structure to hold information about a person

  additionalInformation titlesType{
    description: Structure for titles and other not categorised information connected to a person

    salutation max35Text string
                  minLength: 1
                  maxLength: 35

    academicTitles [academicTitleType{
                      description: Acquired academic title and title position relative to name

                      title* max35Text string
                              minLength: 1
                              maxLength: 35

                      position* {

                      }

                    }]

    otherTitles [max35Text string
                  minLength: 1
                  maxLength: 35]

    knownNames [max35Text string
                 minLength: 1
                 maxLength: 35]

  }

  firstNames [max140Text string
               minLength: 1
               maxLength: 140]
}

```

```

lastNames  [max140Text string
            minLength: 1
            maxLength: 140]
name       max140Text string
            minLength: 1
            maxLength: 140
aliases    [max140Text string
            minLength: 1
            maxLength: 140]
address    addressType{
            description: Address information

            department max70Text string
                        minLength: 1
                        maxLength: 70
            subDepartment max70Text string
                        minLength: 1
                        maxLength: 70
            streetName max70Text string
                        minLength: 1
                        maxLength: 70
            buildingNumber max16Text string
                        minLength: 1
                        maxLength: 16
            buildingName max35Text string
                        minLength: 1
                        maxLength: 35
            floor max70Text string
                        minLength: 1
                        maxLength: 70
            postBox max16Text string
                        minLength: 1
                        maxLength: 16
            room max70Text string
                        minLength: 1
                        maxLength: 70
            postCode max16Text string
                        minLength: 1
                        maxLength: 16
            townName* max35Text string
                        minLength: 1
                        maxLength: 35
            townLocationName max35Text string
                        minLength: 1
                        maxLength: 35
            districtName max35Text string
                        minLength: 1
                        maxLength: 35
            countrySubDev- max35Text string
            sion           minLength: 1
                        maxLength: 35
            country*      countryCode string
                        pattern: ^[A-Z]{2}$
                        example: AT

```

Two alpha country code according ISO 3166

```

    }
    birthName      max140Text string
                  minLength: 1
                  maxLength: 140
    officialGender genderType string
                  example: Other
                  Gender of person from documents used in KYC process

    Enum:
    [ Male, Female, Other ]
    dateAndPlaceOfBirth birthType{
      description:      Information on person's birth

      birthDay*         dateTyped string($date-time)
                        pattern: ^\d{4}(-\d{2}){2}(Z|([-+]\d{2}(:\d{2})?))$
                        example: 2024-12-12Z
                        Date in ISO 8601 format, either UTC (YYYY-MM-DDZ) or local time with UTC offset (YYYY-MM-DD+/-hh:mm).

      provinceOfBirth   max35Text string
                        minLength: 1
                        maxLength: 35
      cityOfBirth*      max35Text string
                        minLength: 1
                        maxLength: 35
      countryOfBirth*   countryCode string
                        pattern: ^[A-Z]{2}$
                        example: AT
                        Two alpha country code according ISO 3166

    }
    identification {genericPersonId{
      description:      A person identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier issuer info

      identification*   max35Text string
                        minLength: 1
                        maxLength: 35

      schemeName        {
        oneOf -> schemeCode{
          description: Coded information about the kind and/or nature of a code

          code*         schemeTypes string
                        example: CCPT
                        Codes for the kind and/or nature of a code

          Enum:
          [ ARNU, CCPT, CUST, DRLC, EMPL, NIDN, POID, SOSE, TELE, TXID ]
        }
      }
    }
  }

```



```

        proprietaryName{
            description: Proprietary information about the kind and/or nature of a code

            proprietary* max35Text string
                minLength: 1
                maxLength: 35
        }
    }
    issuer max35Text string
        minLength: 1
        maxLength: 35
}]]
}

```

```

verificationOfId{
    description: Defined codes for IBAN - Id verification.

    partyIdMatch* partyIdMatchType string
        example: MTCH
        Specifies the codes -in sequence- for Match, NoMatch and MatchNotApplicable.

    Enum:
    [ MTCH, NMTC, NOAP ]
}

```

```

verificationOfName{
    description: Defined codes for IBAN - Name verification (not including close match).

    partyNameMatch* nameMatchType string
        example: MTCH
        Specifies the codes -in sequence- for Match, NoMatch and MatchNotApplicable.

    Enum:
    [ MTCH, NMTC, NOAP ]
}

```

```

verificationOfNameClose{
    description: Defined code for IBAN - Name verification on close match together with proposed name.

    partyNameMatch* nameMatchTypeClose string
        example: CMTc
        Specifies the code CloseMatch.

    Enum:
    [ CMTc ]
}

```

```

    matchedName*      max140Text string
                      minLength: 1
                      maxLength: 140
  }

```

```

verificationOfPayeeError{
  description: Provides information on an error occurred.

  type*      max70Text string
              minLength: 1
              maxLength: 70
  code*      messageCode string
              example: TIMESTAMP_INVALID
              Specifies the codes Error messages.

  title      Enum:
              [ FORMAT_ERROR, CLIENT_INVALID, CLIENT_INCONSISTENT, TIMESTAMP_INVALID, DUPLICATE, NOT_FOUND, UNKNOWN_ERROR ]
              max70Text string
              minLength: 1
              maxLength: 70
  status      integerEnum:
              [ 400, 401, 403, 404, 409, 500 ]
  detail      max500Text string
              minLength: 1
              maxLength: 500
  instance    max256Text string
              minLength: 1
              maxLength: 256
}

```

```

proprietaryName{
  description: Proprietary information about the kind and/or nature of a code

  proprietary*      max35Text string
                    minLength: 1
                    maxLength: 35
}

```

```

schemeCode{
  description: Coded information about the kind and/or nature of a code

  code*      schemeType string
              example: CCPT
              Codes for the kind and/or nature of a code

  Enum:
  [ ARNU, CCPT, CUST, DRLC, EMPL, NIDN, POID, SOSE, TELE, TXID ]
}

```

schemeType **string**
example: CCPT

Codes for the kind and/or nature of a code

Enum:
[ARNU, CCPT, CUST, DRLC, EMPL, NIDN, POID, SOSE, TELE, TXID]

```
titlesType{
  description:   Structure for titles and other not categorised information connected to a person

  salutation    max35Text string
                minLength: 1
                maxLength: 35
  academicTitles [academicTitleType{
    description:   Acquired academic title and title position relative to name

    title*        max35Text string
                  minLength: 1
                  maxLength: 35
    position*     {
                  }

  }}
  otherTitles   [max35Text string
                 minLength: 1
                 maxLength: 35]
  knownNames    [max35Text string
                 minLength: 1
                 maxLength: 35]
}
```

```
verificationData{
  description:   Data returned to apply matching algorithms on.

  partyAccount* accountType{
    description: InternationalBankAccountNumber according ISO 13616, as assigned by the responding PSP.

    iban*        IBANType string
                  pattern:  ^[A-Z]{2}[0-9]{2}[A-Z0-9]{1,30}$
                  example:  AB121234567812345678901234
                  InternationalBankAccountNumber according ISO 13616, as assigned by the responding PSP.

  }
  accountSince  dateType string($date-time)
                 pattern:  ^\d{4}(-\d{2}){2}(Z|([-+]\d{2}(:\d{2})?))$
                 example:  2024-12-12Z
}
```

Date in ISO 8601 format, either UTC (YYYY-MM-DDZ) or local time with UTC offset (YYYY-MM-DD+/-hh:mm).

```

associated-NamesAndIds* partyData{
  description: Structure holding information about a party / parties associated with an account.

  person [personType{...}]
  organisation [organisationType{
    description: Structure to hold information about an organisation
    type max35Text string
      minLength: 1
      maxLength: 35
    identification idType{
      description: Unique identification of an organisation, as assigned by an institution, using an identification scheme.

      organisationIdType{
        nId* description: Unique identification of an organisation, as assigned by an institution, using an identification scheme.

        oneOf -> idAsLEI{
          description: Legal Entity Identifier according ISO 17442.

          lei LEIType string
            pattern: ^[A-Z0-9]{18}[0-9]{2}$
            example: 549300DTUYXVMJXZNY75
            Code LEI (Legal Entity Identifier) according ISO 17442.

        }
          idAsBIC{
            description: Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.

            anyBIC BICType string
              pattern: ^[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3})?$
              example: ABCDEBBBFIM
              Business Identifier Code according ISO 9362. May identify financial institutions and other organisations.

          }
          idAsOther{
            description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or 'schemeNameProprietary' and optionally an identifier issuer information. Although the structure is an array, only one identifier is allowed

            others [genericOrganisationId{
              description: An organisation identifier and an identifier type qualified by either 'schemeNameCode' or

```

'schemeNameProprietary' and optionally an identifier issuer information

```

oneOf ->
  organisationCodedId {
    description: Unique and unambiguous identification of an organisation according provided coded scheme within issuers space.

    identification* max256Text string
                  minLength: 1
                  maxLength: 256
    schemeNameCode* organisationIdCode string
                  example: BOID
                  Specifies the external organisation identification scheme name code (maximum 4 characters) as defined in an external list (iso20022.org).

    issuer max35Textmax35Textstring
          minLength: 1
          maxLength: 35
  }
  organisationProprietaryId {
    description: Unique and unambiguous identification of an organisation according provided proprietary scheme eventually within issuers space.

    identification* max256Text string
                  minLength: 1
                  maxLength: 256
    schemeNameProprietary* max35Text string
                          minLength: 1
                          maxLength: 35
    issuer max35Text string
          minLength: 1
          maxLength: 35
  }
}

```

```

names      [max140Text string
            minLength: 1
            maxLength: 140]
aliases    [max140Text string
            minLength: 1
            maxLength: 140]
address    addressType{
            description: Address information

            department    max70Text string
                           minLength: 1
                           maxLength: 70
            subDepartment max70Text string
                           minLength: 1
                           maxLength: 70
            streetName    max70Text string
                           minLength: 1
                           maxLength: 70
            buildingNumber max16Text string
                           minLength: 1
                           maxLength: 16
            buildingName   max35Text string
                           minLength: 1
                           maxLength: 35
            floor          max70Text string
                           minLength: 1
                           maxLength: 70
            postBox        max16Text string
                           minLength: 1
                           maxLength: 16
            room           max70Text string
                           minLength: 1
                           maxLength: 70
            postCode       max16Text string
                           minLength: 1
                           maxLength: 16
            townName*      max35Text string
                           minLength: 1
                           maxLength: 35
            townLocation-  max35Text string
            Name           minLength: 1
                           maxLength: 35
            districtName   max35Text string
                           minLength: 1
                           maxLength: 35
            countrySubDevi- max35Text string
            sion           minLength: 1
                           maxLength: 35
            country*       countryCode string
                           pattern: ^[A-Z]{2}$
                           example: AT
                           Two alpha country code according ISO 3166
        }
    }

```

```

    ]]
  }
}

```

```

verificationDataError{
  description: No data returnable.

  partyMatch* string
  example: NOAP
  In case no data can or shall be returned

  Enum:
  [ NOAP ]
}

```

```

verificationOfPayeeDataRequest{
  description: Request for the data needed for verification process.

  partyAccount* accountType{
    description: InternationalBankAccountNumber according ISO 13616, as assigned by the responding PSP.

    iban* IBANType string
    pattern: ^[A-Z]{2}[0-9]{2}[A-Z0-9]{1,30}$
    example: AB121234567812345678901234
    InternationalBankAccountNumber according ISO 13616, as assigned by the responding PSP.

  }
}

```

```

verificationOfPayeeDataResponse{
  description: Result of the verification data request.

  oneOf -> verificationData{
    description: Data returned to apply matching algorithms on.

    partyAccount* accountType{
      description: InternationalBankAccountNumber according ISO 13616, as assigned by the responding PSP.

      iban* IBANType string
      pattern: ^[A-Z]{2}[0-9]{2}[A-Z0-9]{1,30}$
      example: AB121234567812345678901234
      InternationalBankAccountNumber according ISO 13616, as assigned by the responding PSP.
    }
  }
}

```

```

    }
    ac-      dateType string($date-time)
countSince pattern: ^\d{4}(-\d{2}){2}(Z|([-+]\d{2}(:\d{2})?))$
           example: 2024-12-12Z
           Date in ISO 8601 format, either UTC (YYYY-MM-DDZ) or local time with UTC offset (YYYY-MM-DD+/-hh:mm).

associa-   partyData{
tedName-   descrip- Structure holding information about a party / parties associated with an account.
sAndIds*   tion:
           person [personType{
                                description: Structure to hold information about a person

                                additionalIn- titlesType{
formation          description: Structure for titles and other not categorised information connected to a person

                                salutation      max35Text string
                                                minLength: 1
                                                maxLength: 35
                                academicTitles [academicTitleType{
                                                description: Acquired academic title and title position relative
                                                                to name

                                                                title*      max35Text string
                                                                minLength: 1
                                                                maxLength: 35
                                                                position*
                                                                {
                                                                }

                                                                }]
                                otherTitles      [max35Text string
                                                minLength: 1
                                                maxLength: 35]
                                knownNames      [max35Text string
                                                minLength: 1
                                                maxLength: 35]

                                }
                                firstNames      [max140Text string
                                                minLength: 1
                                                maxLength: 140]
                                lastNames      [max140Text string
                                                minLength: 1
                                                maxLength: 140]
                                name            max140Text string
                                                minLength: 1
                                                maxLength: 140
                                aliases          [max140Text string
                                                minLength: 1
                                                maxLength: 140]
                                address          addressType{
                                                description: Address information

```



```

    department      max70Text string
                     minLength: 1
                     maxLength: 70
    subDepartment    max70Text string
                     minLength: 1
                     maxLength: 70
    streetName       max70Text string
                     minLength: 1
                     maxLength: 70
    buildingNumber    max16Text string
                     minLength: 1
                     maxLength: 16
    buildingName      max35Text string
                     minLength: 1
                     maxLength: 35
    floor            max70Text string
                     minLength: 1
                     maxLength: 70
    postBox          max16Text string
                     minLength: 1
                     maxLength: 16
    room             max70Text string
                     minLength: 1
                     maxLength: 70
    postCode         max16Text string
                     minLength: 1
                     maxLength: 16
    townName*        max35Text string
                     minLength: 1
                     maxLength: 35
    townLocation-    max35Text string
Name               minLength: 1
                     maxLength: 35
    districtName     max35Text string
                     minLength: 1
                     maxLength: 35
    countrySubDevi-  max35Text string
sion               minLength: 1
                     maxLength: 35
    country*         countryCode string
                     pattern: ^[A-Z]{2}$
                     example: AT
                     Two alpha country code according ISO 3166
  }
  birthName          max140Text string
                     minLength: 1
                     maxLength: 140
  officialGen-       genderType string
der                 example: Other
                     Gender of person from documents used in KYC process

  Enum:
  [ Male, Female, Other ]

```

```

dateAndPlace- birthType{
OfBirth      description: Information on person's birth

              birthDay*    dateType string($date-time)
                               pattern: ^\d{4}(-\d{2}){2}(Z|([-+]\d{2}(:\d{2})?))$
                               example: 2024-12-12Z
                               Date in ISO 8601 format, either UTC (YYYY-MM-DDZ) or local time with UTC offset
                               (YYYY-MM-DD+/-hh:mm).

              province-    max35Text string
OfBirth      minLength: 1
              maxLength: 35
              cityOfBirth* max35Text string
              minLength: 1
              maxLength: 35
              countryO-    countryCode string
fBirth*      pattern: ^[A-Z]{2}$
              example: AT
              Two alpha country code according ISO 3166

    }
}
[genericPersonId{
  description: A person identifier and an identifier type qualified by either 'schemeNameCode' or
               'schemeNameProprietary' and optionally an identifier issuer information

  identifica- max35Textm string
tion*         minLength: 1
              maxLength: 35

  schemeName {
    oneOf -> schemeCode{
      description: Coded information about the kind
                  and/or nature of a code

      code*      schemeTypes string
                  example: CCPT
                  Codes for the kind and/or nature
                  of a code

      Enum:
      Array [ 10 ]

    }
    proprietaryName{
      description: Proprietary information about the
                  kind and/or nature of a code

      proprietary* max35Text string
                  minLength: 1
                  maxLength: 35
    }
  }
}

```

```

        issuer
        }
        max35Text string
        minLength: 1
        maxLength: 35
    }
}

    organisation
    }
}

verificationDataError{
    description: No data returnable.

    partyMatch* string
    example: NOAP
    In case no data can or shall be returned

    Enum:
    [ NOAP ]
}

}

verificationOfPayeeError{
    description: Provides information on an error occurred.

    type* max70Text string
    minLength: 1
    maxLength: 70
    code* messageCode string
    example: TIMESTAMP_INVALID
    Specifies the codes Error messages.

    Enum:
    [ FORMAT_ERROR, CLIENT_INVALID, CLIENT_INCONSISTENT, TIMESTAMP_INVALID, DUPLICATE, NOT_FOUND, UNKNOWN_ERROR ]
    title max70Text string
    minLength: 1
    maxLength: 70
    status integerEnum:
    [ 400, 401, 403, 404, 409, 500 ]
    detail max500Text string
    minLength: 1
    maxLength: 500
    instance max256Text string
    minLength: 1
    maxLength: 256
}

```