

# Verification Services

# Tarkistuspalvelu

API Documentation

## Contents

|  |    |
|--|----|
| 1. Introduction .....  | 3  |
| 1.1. Address verification.....   | 3  |
| 1.2. Verification of customer data .....   | 3  |
| 1.3. Verification of payer information .....                                     | 3  |
| 1.4. Customer register checks .....  | 3  |
| 1.5. General information .....   | 3  |
| 2. Document Change History .....   | 4  |
| 3. Common Data Structures .....  | 5  |
| 3.1. Address data structure.....   | 5  |
| 3.1.1. Address Verification Service specific address information .....           | 6  |
| 3.1.2. Payer Information Verification Service specific address information ..... | 6  |
| 3.1.3. Verification of customer data Service specific address information.....   | 6  |
| 3.2. Party data structure.....   | 6  |
| 4. Common HTTP Response Codes .....  | 8  |
| 5. Address Verification Service.....   | 9  |
| 5.1. EXACT_MATCH .....   | 9  |
| 5.2. MATCH .....   | 10 |
| 5.3. MULTI_MATCH.....  | 10 |
| 5.4. NO_MATCH.....   | 11 |
| 6. Verification of Customer Data Service.....                                    | 12 |
| 6.1. Search by name and address .....  | 12 |
| 7. Verification of Payer Information Service .....                               | 14 |
| 7.1. EXACT_MATCH .....   | 15 |
| 7.2. MATCH .....   | 15 |
| 7.3. SSN_MISMATCH.....   | 16 |
| 7.4. UNKNOWN_NAME.....   | 17 |
| 7.5. UNKNOWN_ADDRESS.....  | 17 |
| 7.6. NO_MATCH .....  | 18 |
| 7.7. Some HTTP errors due to missing or malformed information .....              | 18 |
| 8. Customer Register Check Service .....   | 20 |
| 8.1. REST API.....   | 20 |
| 8.1.1. Check by name and address.....  | 20 |
| 8.2. SOAP API .....  | 21 |
| 8.2.1. Check by name and address.....  | 22 |
| 8.3. Return codes.....   | 22 |
| 9. Notes .....   | 24 |

20.3.2019

## 1. Introduction

With the help of Posti's APIs, you can easily integrate updated address and consumer customer information to your systems and services. The data source for all API services is Posti's address information system, which covers all mail addresses and mail recipients in Finland.

Verification Services are online based services for Posti's business customers to be used in e-commerce processes.

### 1.1. Address verification

With the service, you can control whether an address exists, how it is spelled and correct errors in the address. The service also gives you the personal identifier for Posti's address, i.e. the family address ID, which enables you to check the different language versions of the same address and synonymous addresses.

### 1.2. Verification of customer data

A consumer customer's information can be controlled and corrected in the service using the person's name and address. The service gives information on whether the person is underage, deceased and whether they live at the address in question according to the mail recipient register. If a customer's address or name is outdated, the service will provide the valid name and address information. The service will also give you the person's personal identifier (OTS-ID) and the address identifier (address family ID), which enable the identification of people who live in the same address and prevent the creation of duplicates.

### 1.3. Verification of payer information

With the service, it is possible to control whether a person's name, address and personal identity number match. The service can be used, for instance, to prevent fraud in a situation where someone tries to buy something using someone else's personal identity number. The service will return the death information for the identified person, if desired.

### 1.4. Customer register checks

The service works together with Posti's continuous update service. The service controls whether a person is already in your customer register. If the person can be found in your Posti update service customer register, the service will return the customer number or numbers. The service can be used, for instance, to prevent the creation of duplicates with several different customer numbers in the customer register.

### 1.5. General information

API can be accessed from URL: <https://www.posti.fi/tarkistuspalvelu/api/v1/rest>

Services are offered as a REST service with documented responses using JSON notation. Some services also have SOAP based web service.

Authentication is done using HTTP basic authentication and the data communication is secured using SSL.

20.3.2019

## 2. Document Change History

This document describes current Verification Services API and the content is subject to change.

| Date       | Version | Description   |
|------------|---------|---|
| 2014-09-17 | 1.3.1   | Added information on municipality data and service specific fields to sections covering Address data structure. |
| 2017-01-23 | 1.3.3   | Added API for Customer register check   |
| 2017-03-07 | 1.3.4   | Added API for Verification of customer data   |
| 2018-01-08 | 1.3.5   | Added MULTI_MATCH response to Address Verification service  |
| 2019-03-20 | 1.4     | Added API for Verification of payer information   |

### 3. Common Data Structures

#### 3.1. Address data structure

All addresses returned by the services share a common data structure part:

| Field                   | Type    | Description   |
|-------------------------|---------|---|
| addressFamilyId         | Integer | A common id for the same addresses that have a different appearance as a result of different languages and so-called corner apartments.   |
| streetAddress           | String  | Street with premises or PO Box. For example Arkadiankatu 60 C 52  |
| postCode                | String  | For example 00100   |
| isExpired               | Boolean | Has this address expired due to for example municipal merge. A new address will be automatically provided for expired addresses. The address that matched the provided address is only returned in case a more detailed explanation for a possible end user is useful (e.g. a user is informed that the inputted address has been expired and this new address should be used instead). |
| municipalityNumber      | Integer | Municipality number of the street address. This is only present for the domestic street addresses.  |
| municipalityNameFinnish | String  | Finnish name of the municipality. Municipality number of the street address. This is only present for the domestic street addresses.  |
| municipalityNameSwedish | String  | Swedish name of the municipality. Municipality number of the street address. This is only present for the domestic street addresses.  |

The example below illustrates how a Finnish address Postintaival 7 A, 00230 HELSINKI is described in the address data structure.

Example 1: An address represented in the address data structure (JSON notation)

```
"Address":
{
  "addressFamilyId":12345687,
  "streetAddress":"Postintaival 7 A",
  "postCode":"00230",
  "postOffice":"Helsinki",
  "isExpired":false,
  "municipalityNumber": "091",
  "municipalityNameFinnish": "Helsinki",
  "municipalityNameSwedish": "Helsingfors"
}
```

In some cases the PO Box addresses and/or the poste restante addresses must be recognized by the application using these services. The PO Box addresses has the abbreviation 'PL' (Finnish) or 'PB' (Swedish) in the beginning of the streetAddress element.

In cases of poste restante addresses the streetAddress element includes the string 'poste restante'. Please use the case insensitive comparison.

20.3.2019

In addition to the shared part, the different services can return additional address info which is described in the next sections.

### 3.1.1. Address Verification Service specific address information

None

### 3.1.2. Payer Information Verification Service specific address information

| Field        | Type   | Description   |
|--------------|--------|---|
| status       | String | Status of the address: CURRENT, PAST, FUTURE.   |
| addressClass | String | Class of the address. Values: DOMESTIC_STREET_ADDRESS, DOMESTIC_POBOX_ADDRESS, DOMESTIC_POSTCODE_ADDRESS, FOREIGN_ADDRESS |

### 3.1.3. Verification of customer data Service specific address information

| Field        | Type   | Description   |
|--------------|--------|---|
| status       | String | Status of the address: CURRENT, PAST, FUTURE.   |
| addressClass | String | Class of the address. Values: DOMESTIC_STREET_ADDRESS, DOMESTIC_POBOX_ADDRESS, DOMESTIC_POSTCODE_ADDRESS, FOREIGN_ADDRESS |

## 3.2. Party data structure

Party holds information of a single person.

| Field           | Type  | Optional | Description   |
|-----------------|---|----------|---|
| partyId         | Integer   | No       | Unique ID for the party.  |
| firstName       | String  | No       | Person's first name. If the person has multiple first names, all first names are concatenated and separated by space characters (e.g. "Anna Maria").  |
| lastName        | String  | No       | Person's last name.   |
| yearsSinceBirth | Two-element integer array:<br>[0,14]<br>[15,17]<br>[18,300]<br>null | No       | Person's age information. First element is the lower limit of how many years have passed since the person's birth. Second element is the upper limit.<br><br>E.g. yearsSinceBirth=[0,14] implies that person is between 0 (inclusive) and 14 (inclusive) years old. |
| status          | DEAD,<br>ALIVE,<br>null   | Yes      | Shows if the person is dead or alive. This information is optional and supplied only when explicitly requested in the incoming request.   |
| partyNames      | Array of PartyName objects  | yes      | Current and former names of the person.   |

### PartyName data structure

| Field     | Type   | Optional | Description  |
|-----------|--------|----------|--|
| firstName | String | No       | Person's first name. If the person has multiple first names, all first names are concatenated and separated by space characters (e.g. "Anna Maria"). |
| lastName  | String | No       | Person's last name.  |

20.3.2019

|           |                   |     |   |
|-----------|-------------------|-----|---|
| validFrom | String (yyyymmdd) | No  | Date when the name became valid   |
| validTo   | String (yyyymmdd) | Yes | Date when the name became expired.  |
| Status    | CURRENT,<br>PAST  | No  | Status of the name; if the name is the person's current name (CURRENT) or former name (PAST). |

The example below illustrates how Anna Maria Johansson (maiden name Nieminen) is described in the party data structure. She is alive and over 18 years.

```
{
  "firstName": "Anna Maria",
  "lastName": "Johansson",
  "partyId": "15693256",
  "status": "ALIVE",
  "yearsSinceBirth": [18, 300],
  "partyNames": [
    {
      "firstName": "Anna Maria",
      "lastName": "Johansson",
      "validFrom": "20060302",
      "status": "CURRENT"
    },
    {
      "firstName": "Anna Maria",
      "lastName": "Nieminen",
      "validFrom": "19990329",
      "validTo": "20060302",
      "status": "PAST"
    }
  ]
}
```

20.3.2019

#### 4. Common HTTP Response Codes

| Status Code                        | REST services  | SOAP services  |
|------------------------------------|--|--|
| HTTP 200 - OK                      | No errors  |  |
| HTTP 400 – Bad Request             | The request format is corrupted, required parameters are missing or request data is malformed.                                   |  |
| HTTP 401 – Authentication Required | The request did not contain valid HTTP Basic Auth header, i.e. the header was missing, or the username or password were invalid. |  |
| HTTP 403 – Not Authorized          | User authentication succeeded (username and password matched) but the user does not have the role to access the service.         |  |
| HTTP 500 – Internal Server Error   | Internal error at Posti's service.   | Internal error at Posti's service OR the response contains a SOAP fault. A SOAP fault can result in e.g. requesting data that the user is not permitted to access. |
| HTTP 503 – Service Unavailable     | An invocation threshold has been exceeded for the service.   |  |



## 5. Address Verification Service

This service tries to identify and correct the given address. The service returns one or more addresses with the correct spelling when the identification is successful. If identification fails only the information of no match is returned.

### Input

| Field         | Type   | Optional | Description           |
|---------------|--------|----------|-----------------------|
| streetAddress | String | No       | e.g. Postintaival 7 A |
| postcode      | String | No       | e.g. 00230            |
| postOffice    | String | No       | e.g. Helsinki         |

### Output

| Field            | Type  | Description  |
|------------------|---|--|
| matchType        | EXACT_MATCH<br>MATCH<br>MULTI_MATCH<br>NO_MATCH | Describes the match type. See details below.   |
| validAddresses   | List<Address>                                   | If the match type was EXACT_MATCH or MATCH this field will contain the matching address.<br><br>Note: This field's type is defined as a list in case we are able to provide a list of addresses for a multimatch response in the future. |
| expiredAddresses | List<Address>                                   | This field contains the address if it has been expired   |

### 5.1. EXACT\_MATCH

The exact match is returned when the given address is found, and it is written exactly in the same way as in the Contact information Management system. The exact match is given even if the address has been expired or the street name is abbreviated correctly.

#### Request

```
GET
/tarkistuspalvelu/api/v1/rest/addresscorrection/correct?postcode=00100&streetaddress=Arka
diankatu%2060%20C%2052&postoffice=Helsinki HTTP/1.1
Host: posti.fi
Connection: keep-alive
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

#### Response

```
HTTP/1.1 200 OK
Pragma: No-cache
Cache-Control: no-cache
Content-Type: application/json
Transfer-Encoding: chunked
```

```
{ "matchType": "EXACT_MATCH",
  "validAddresses": [
    { "addressFamilyId": "121374369",
      "streetAddress": "Arkadiankatu 60 C 52",
```

20.3.2019

```
"postCode": "00100",
"postOffice": "HELSINKI",
"isExpired": false}],
"expiredAddresses": null}
```

## 5.2. MATCH

The match is returned when the given address is found but it's misspelled. The match is given even if the address has been expired. The response includes the correctly spelled version of the address.

### Request

```
GET
/tarkistuspalvelu/api/v1/rest/addresscorrection/correct?postcode=00100&streetaddress=Fredrikink%2060%20C%2052&postoffice=HELSINK HTTP/1.1
Host: posti.fi
Connection: keep-alive
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

### Response

```
HTTP/1.1 200 OK
Pragma: No-cache
Cache-Control: no-cache
Content-Type: application/json
Transfer-Encoding: chunked

{"matchType": "MATCH",
"validAddresses":
[{"addressFamilyId": "121374369",
"streetAddress": "Fredrikinkatu 60 C 52",
"postCode": "00100",
"postOffice": "HELSINKI",
"isExpired": false}],
"expiredAddresses": null}
```

## 5.3. MULTI\_MATCH

Address query that has incomplete info about property, stair and apartment numbers and letters may result to MULTI\_MATCH because query matches more than one address. For example when querying Taivastie 13 that has properties Taivastie 13a, 13b and 13c results to MULTI\_MATCH.

### Request

```
GET /tarkistuspalvelu/api/v1/rest/addresscorrection/correct?postcode=00100&streetaddress=Taivastie%2013&postoffice=Helsinki HTTP/1.1
Host: Posti.fi
Connection: keep-alive
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

### Response

```
{
"matchType": "MULTIMATCH",
"validAddresses": null,
```

20.3.2019

```
"expiredAddresses": null,  
"addresses": null  
}
```

#### 5.4. NO\_MATCH

The no match is returned when the given address is misspelled in a way that it can't be recognized, or the address doesn't exist.

##### Request

```
GET  
/tarkistuspalvelu/api/v1/rest/addresscorrection/correct?postcode=00100&streetaddress=Fred  
rik&postoffice=HLESINK HTTP/1.1  
Host: posti.fi  
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

##### Response

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Transfer-Encoding: chunked  
{  
  "matchType": "NOMATCH",  
  "validAddresses": null,  
  "expiredAddresses": null  
}
```

20.3.2019

## 6. Verification of Customer Data Service

This service returns name and address information of a person identified by name and address. The given name or address can be current or expired. Service returns information whether match was done against past name or address, this can be used to check whether address needs to be updated

The result is EXACT\_MATCH (person name and address are written correctly), MATCH (spelling mistakes in name or address), NO\_MATCH (name and address do not match) or MULTI\_MATCH (multiple matches found, i.e. multiple persons with same name live in the given address).

Status of whether the person is deceased or alive is always included.

If the requested data is non-public (“luovutuskiellossa olevat tiedot” in Finnish), Posti is not permitted to supply the data and the service returns NO\_MATCH by default. Non-public data access is not available for customer users.

### 6.1. Search by name and address

To request customer data verification by name and address, the following URL is used

<https://www.posti.fi/tarkistuspalvelu/api/v1/rest/partyinfo/findByNameAndAddress?firstName=<firstName>&lastName=<lastName>&streetAddress=<streetAddress>&postCode=<postCode>&postOffice=<postOffice>>

#### Input

| Field         | Type   | Optional | Description  |
|---------------|--------|----------|--|
| firstName     | String | No       | One to all first names of the person. Multiple first names are separated by space character. |
| lastName      | String | No       | Last name of the person.   |
| streetAddress | String | No       | e.g. Postintaival 7 A  |
| postCode      | String | No       | e.g. 00230   |
| postOffice    | String | No       | e.g. Helsinki  |

#### Output

| Field                | Type  | Description   |
|----------------------|---|---|
| matchType            | EXACT_MATCH<br>MATCH<br>MULTI_MATCH<br>NO_MATCH | Describes the type of the result. In case of MULTI_MATCH and NO_MATCH no other information is included in the response. |
| addresses            | Address[]                                       | Address of the matched party.   |
| party                | Party   | Identified party.   |
| publicityLevel       | PUBLIC<br>NON_PUBLIC                            | Publicity level of the returned information.  |
| matchPastName        | Boolean   | True if matched against past name.  |
| matchPastAddress     | Boolean   | True if matched against past address.   |
| matchFutureAddress   | Boolean   | True if matched against future address.   |
| matchAddressFamilyId | Integer   | Common id of address that matched.  |

### Request

GET

```
/tarkistuspalvelu/api/v1/rest/partyinfo/findByNameAndAddress?firstName=Anna%20Maria&lastName=Nieminen&streetAddress=Puistokatu%201%20B%2034&postCode=40100&postOffice=JYV%C3%84SKYL%C3%84
```

20.3.2019

## Response

```
{
  "party": {
    "firstName": "Anna Maria",
    "lastName": "Johansson",
    "partyId": "15693255",
    "status": "ALIVE",
    "partyNames": [
      {
        "firstName": "Anna Maria",
        "lastName": "Johansson",
        "validFrom": "20060302",
        "status": "CURRENT"
      }
    ],
    "yearsSinceBirth": [18,300]
  },
  "addresses": [
    {
      "streetAddress": "Arkadiankatu 60 C 52",
      "postCode": "00100",
      "postOffice": "HELSINKI",
      "expired": false,
      "status": "CURRENT",
      "validFrom": "20050126"
    }
  ],
  "publicityLevel": "PUBLIC",
  "matchPastName": false,
  "matchPastAddress": false,
  "matchFutureAddress": false,
  "matchAddressFamilyId": "123443210"
}
```

20.3.2019

## 7. Verification of Payer Information Service

This service verifies the consistency of payers (person buying a product) information (social security number, name and address). The verification result is one of the following codes:

- EXACT\_MATCH – person's name, address and ssn match
- MATCH – spelling mistakes in name or address, ssn matches the person
- 
- non-matching cases:
  - SSN\_MISMATCH – name and address match, but ssn belongs to another person
    - The possibility of fraud should be investigated thoroughly before accepting the transaction
  - UNKNOWN\_NAME – name is not found in the person's official name history
    - The possibility of fraud should be investigated before accepting the transaction
  - UNKNOWN\_ADDRESS – address is not found in person's address history
    - This occurs e.g. when a person has newly moved and has not yet registered the new address
  - SSN\_NOT\_IN\_REGISTER – Posti's register does not contain ssn for the person found by name and address. Posti can not confirm if the ssn provided in the query is the person's real ssn.
    - This occurs e.g. when Posti has not yet been informed that an immigrant has been granted an official Finnish ssn.
  - NO\_MATCH – verification failed due to misspelled address, non-existing ssn, or Posti is not authorized to supply the information, etc.

Parameter `includeDeceasedCheck=true` can be included in the request to check if the matched person is dead or alive.

If the name, address or ssn are non-public ("henkilötiedot luovutuskiellossa" in Finnish), Posti is not authorized to supply the matching data and `NO_MATCH` is returned.

Missing required parameters (such as `lastName`) or malformed parameters (malformed ssn) result HTTP 400 – Bad request with descriptive text.

Input

| Field                             | Type    | Optional | Description   |
|-----------------------------------|---------|----------|---|
| <code>firstName</code>            | String  | No       | One to all first names of the person. Multiple first names are separated by space character.  |
| <code>lastName</code>             | String  | No       | Last name of the person.  |
| <code>streetAddress</code>        | String  | No       | e.g. Postintaival 7 A   |
| <code>ssn</code>                  | String  | No       | Syntactically valid Finnish social security number.   |
| <code>postcode</code>             | String  | No       | e.g. 00230  |
| <code>postOffice</code>           | String  | No       | e.g. Helsinki   |
| <code>includeDeceasedCheck</code> | Boolean | Yes      | If set to true and there is a match a query is made to determine whether the person is alive or deceased.<br><br>Note: Setting this value to true will have price implications.                       |
| <code>Api-Version</code>          | String  | Yes      | Defines the version of the service. The currently supported version is 2.<br><br>This parameter can be given either as a URL query parameter or as an HTTP header, query parameter having precedence. |

20.3.2019

|  |  |  |   |
|--|--|--|---|
|  |  |  | Can be either "1" or "2" (without quotes), defaults to "1". |
|--|--|--|---|

#### Output

| Field           | Type   | Description  |
|-----------------|--|--|
| matchType       | EXACT_MATCH<br>MATCH<br>SSN_MISMATCH<br>UNKNOWN_NAME<br>UNKNOWN_ADDRESS<br>SSN_NOT_IN_REGISTER<br>NO_MATCH | Describes the type of the result. See examples below.  |
| address         | Address  | Address of the matched party.  |
| party           | Party  | Identified party if there is an exact match.   |
| partyIsDeceased | Boolean  | If the includeDeceasedCheck parameter was set to true and match type is exact match then this value will describe whether the identified party has deceased. |

### 7.1. EXACT\_MATCH

#### Request

```
GET
/tarkistuspalvelu/api/v1/rest/partyverification/verify?firstName=Antti&lastName=Testinen&
streetAddress=Arkadiankatu%2060%20C%2052&postCode=00100&postOffice=HELSINKI&ssn=180878-
213B&includeDeceasedCheck=true HTTP/1.1
Host: Posti.fi
Connection: keep-alive
Api-Version: 2
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

#### Response

```
{
  "matchType": "EXACT_MATCH",
  "address": {
    "streetAddress": "Arkadiankatu 60 C 52",
    "postCode": "00100",
    "postOffice": "HELSINKI",
    "expired": false,
    "status": "CURRENT"
  },
  "party": {
    "firstName": "ANTTI",
    "lastName": "TESTINEN",
    "partyId": "14481691",
    "status": "ALIVE",
    "yearsSinceBirth": [ 18, 300 ]
  },
  "partyIsDeceased": false
}
```

### 7.2. MATCH

Post office name is misspelled in the request (HLESINKI, should be HELSINKI). This is why matchType is MATCH (not EXACT\_MATCH).

20.3.2019

## Request

```
GET
/tarkistuspalvelu/api/v1/rest/partyverification/verify?firstName=Antti&lastName=Testinen&streetAddress=Arkadiankatu%2060%20C%2052&postCode=00100&postOffice=HELSINKI&ssn=180878-213B HTTP/1.1
Host: Posti.fi
Connection: keep-alive
Api-Version: 2
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

## Response

```
{
  "matchType": "MATCH",
  "address": {
    "streetAddress": "Arkadiankatu 60 C 52",
    "postCode": "00100",
    "postOffice": "HELSINKI",
    "expired": false,
    "status": "CURRENT"
  },
  "party": {
    "firstName": "ANTTI",
    "lastName": "TESTINEN",
    "partyId": "14481691",
    "yearsSinceBirth": [ 18, 300 ]
  }
}
```

### 7.3. SSN\_MISMATCH

Name and address match other ssn than expected. The possibility of fraud should be investigated thoroughly before accepting the transaction.

The following persons are found in Posti's records:

- Tapani Testinen, ssn 200254-883X, address Postintaival 7 A, 00230 HELSINKI
- Maija Mäkinotkelma, ssn 131052-308T, address Fredrikinkatu 60 C 52 00100 HELSINKI

Querying "Tapani Testinen, ssn 131052-308T, address Postintaival 7 A, 00230 HELSINKI" results SSN\_MISMATCH.

Tapani Testinen could have payment defaults (poor payment record) which he tries to conceal by using Maija Mäkinotkelma's ssn.

## Request

```
GET
/tarkistuspalvelu/api/v1/rest/partyverification/verify?firstName=Tapani&lastName=Testinen&streetAddress=Postintaival%207%20A&postCode=00230&postOffice=HELSINKI&ssn=131052-308T HTTP/1.1
```



20.3.2019

```
Host: Posti.fi
Connection: keep-alive
Api-Version: 2
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

### Response

```
{
  "matchType": "SSN_MISMATCH"
}
```

## 7.4. UNKNOWN\_NAME

Name does not match ssn.

The following person is found in Posti's records:

- Tapani Testinen, ssn 200254-883X, address Postintaival 7 A, 00230 HELSINKI

Querying “Tapsa Testinen, ssn 200254-883X, address Postintaival 7 A, 00230 HELSINKI” results UNKNOWN\_NAME.

### Request

```
GET
/tarkistuspalvelu/api/v1/rest/partyverification/verify?firstName=Tapsa&lastName=Testinen&streetAddress=Postintaival%207%20A&postCode=00230&postOffice=HELSINKI&ssn=200254-883X
HTTP/1.1
Host: Posti.fi
Connection: keep-alive
Api-Version: 2
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

### Response

```
{
  "matchType": "UNKNOWN_NAME"
}
```

## 7.5. UNKNOWN\_ADDRESS

Address is not found in person's address history

The following person is found in Posti's records:

- Tapani Testinen, ssn 200254-883X, address Postintaival 7 A, 00230 HELSINKI

Querying “Tapani Testinen, ssn 200254-883X, address Fredrikinkatu 60 C 52 00100 HELSINKI” results UNKNOWN\_ADDRESS.

It could be that Mr. Testinen has just moved in the Fredrikinkatu address and haven't yet registered the new address.

### Request

```
GET
/tarkistuspalvelu/api/v1/rest/partyverification/verify?firstName=Tapani&lastName=Testinen&streetAddress=Fredrikinkatu%2060%20C%2052&postCode=00100&postOffice=HELSINKI&ssn=200254-883X&includeDeceasedCheck=false
HTTP/1.1
```

20.3.2019

```
Host: Posti.fi
Connection: keep-alive
Api-Version: 2
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

### Response

```
{
  "matchType": "UNKNOWN_ADDRESS"
}
```

## 7.6. NO\_MATCH

Address or ssn could not be recognized.

The following person is found in Posti's records:

- Tapani Testinen, ssn 200254-883X, address Postintaival 7 A, 00230 HELSINKI

Misspelling address as "Po7Aintaiva 99999 HELSINKI" results NO\_MATCH.

### Request

```
GET
/tarkistuspalvelu/api/v1/rest/partyverification/verify?firstName=Tapani&lastName=Testinen
&streetAddress=Po7Aintaiva&postCode=99999&postOffice=HELSINKI&ssn=200254-883X HTTP/1.1
Host: Posti.fi
Connection: keep-alive
Api-Version: 2
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

### Response

```
{
  "matchType": "NO_MATCH"
}
```

## 7.7. Some HTTP errors due to missing or malformed information

HTTP 400 – Bad Request (missing lastName) - The request does not contain parameter "lastName".

### Request

```
GET
/tarkistuspalvelu/api/v1/rest/partyverification/verify?firstName=Anna&streetAddress=Arkadi
ankatu%2060%20C%2052&postCode=00100&postOffice=HELSINKI&ssn=180878-213B HTTP/1.1
Host: Posti.fi
Connection: keep-alive
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

### Response

```
HTTP/1.1 400 Bad Request
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 27
```

20.3.2019

```
Date: Mon, 02 Sep 2013 12:53:11 GMT  
Connection: close
```

```
Errors: lastName is missing, malformed ssn "180878-213B"
```

HTTP 400 – Bad Request (malformed ssn) - The supplied ssn value “180878-21” is malformed.

### Request

```
GET  
/tarkistuspalvelu/api/v1/rest/partyverification/verify?firstName=Anna&lastName=Johansson&streetAddress=Arkadiankatu%2060%20C%2052&postCode=00100&postOffice=HLESINKI&includeYearsSinceBirth=false&ssn=180878-21 HTTP/1.1
```

```
Host: Posti.fi
```

```
Connection: keep-alive
```

```
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

### Response

```
HTTP/1.1 400 Bad Request  
Server: Apache-Coyote/1.1  
Content-Type: application/json  
Content-Length: 33  
Date: Mon, 02 Sep 2013 12:45:33 GMT  
Connection: close
```

```
Errors: malformed ssn "180878-21"
```

20.3.2019

## 8. Customer Register Check Service

This service answers to the question whether a person identified by name and address is among your specified customer Maintenance Register. The service lists all identifiers by which the matched person occurs in your register.

The service provides two check types; they determine how the person to be checked for has to be matched against the register information. For PERSON\_MATCH, it suffices that the register contains the person's name, whereas for PERSON\_AND\_ADDRESS\_MATCH it is necessary that the register contains both the person's name and address as a combination. The first check type is more relaxed since it provides a match even though the person in your register has a different address to the one provided. It should be noted that this check type requires the address so that the person can be reliably identified against the reference register.

This service is available both as REST API and as SOAP API.

Your Posti customer account information, e.g. SAP order customer and billing address, are set when user account for authentication is created by Posti. Default values can be changed by Posti customer service.

If you have multiple SAP order customer accounts and they have different customer registries in Maintenance Service, the values can be changed per request basis by using optional http headers. Thus same user account (basic auth) can be used to query multiple Posti customer accounts. Customer registry specified must be owned by given SAP order customer. SAP order and billing customer must belong to authenticated user.

Headers for setting customer account information (optional, otherwise your default values for user are used):  
X-Posti-OrderCustomer: <SAP order customer number>  
X-Posti-BillingAddress: <SAP billing address number>

### 8.1. REST API

Update register check service can be accessed at URL:

<https://www.posti.fi/tarkistuspalvelu/api/v1/rest/updateregister/>

Currently, there is only one resource "check", which returns matched persons in your register.

The response is in JSON format.

#### 8.1.1. Check by name and address

To check update register, the following URL is used

[https://www.posti.fi/tarkistuspalvelu/api/v1/rest/updateregister/check?checkType=person\\_match&updateRegisterId=<id>&firstname=<firstname>&lastName=<lastname>&streetaddress=<street address>&postcode=<postcode>&postOffice=<postoffice>](https://www.posti.fi/tarkistuspalvelu/api/v1/rest/updateregister/check?checkType=person_match&updateRegisterId=<id>&firstname=<firstname>&lastName=<lastname>&streetaddress=<street address>&postcode=<postcode>&postOffice=<postoffice>)

Input parameters

| Field            | Type   | Optional | Description   |
|------------------|--------|----------|---|
| checkType        | Enum   | No       | PERSON_MATCH or PERSON_AND_ADDRESS_MATCH  |
| updateRegisterId | String | No       | Unique Id of your customer register in Posti maintenance service. Given by Posti. |

20.3.2019

|               |        |    |  |
|---------------|--------|----|--|
| firstname     | String | No | Person's first name. If the person has multiple first names, all first names are concatenated and separated by space characters (e.g. "Anna Maria"). |
| lastName      | String | No | Person's last name.  |
| streetaddress | String | No | Street with premises or PO Box. For example Kauppakatu 1 C 2   |
| postcode      | String | No | For example 00100  |
| postOffice    | String | No | For example Helsinki   |

#### Output

| Field           | Type     | Description  |
|-----------------|----------|--|
| checkType       | Enum     | PERSON_MATCH or PERSON_AND_ADDRESS_MATCH                                 |
| matchingPersons | String[] | List of matching person identifiers in the registers. Empty if no match. |

#### Request

```
GET
/tarkistuspalvelu/api/v1/rest/updateregister/check?checkType=person_match&updateRegisterId=123&firstname=Antti&lastName=Testinen&streetaddress=Katutie%201&postcode=00100&postOffice=Helsinki HTTP/1.1
Host: Posti.fi
Connection: keep-alive
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
X-Posti-OrderCustomer: 765432
X-Posti-BillingAddress: 9876543
```

#### Response

```
{
  "checkType": "PERSON_MATCH",
  "matchingPersons": ["123", "321"]}
}
```

## 8.2. SOAP API

The WSDL for Maintenance Register Check Service SOAP API is available at

<https://www.posti.fi/tarkistuspalvelu/api/v1/soap/updateregister?wsdl>

Note that wsdl is autogenerated and soap:address is missing some parts of the URL, so please use SOAP API address above.

The interface provides one operation

- check

These operations correspond with the rest interface and use the same parameters.

20.3.2019

### 8.2.1. Check by name and address

#### Request example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fron="http://frontend.tapa.itella.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <fron:check>
      <check>
        <checkType>PERSON_AND_ADDRESS_MATCH</checkType>
        <updateRegisterId>123</updateRegisterId>
        <firstName>Antti</firstName>
        <lastName>Testinen</lastName>
        <streetAddress>Katutie 1</streetAddress>
        <postCode>00100</postCode>
        <postOffice>Helsinki</postOffice>
      </check>
    </fron:check>
  </soapenv:Body>
</soapenv:Envelope>
```

#### Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:checkResponse xmlns:ns2="http://frontend.tapa.itella.com/">
      <response>
        <checkType>PERSON_AND_ADDRESS_MATCH</checkType>
        <matchingPersons>
          <personIdentifier>123</personIdentifier>
          <personIdentifier>321</personIdentifier>
        </matchingPersons>
      </response>
    </ns2:checkResponse>
  </soap:Body>
</soap:Envelope>
```

### 8.3. Return codes

#### HTTP return codes

| Return code | Description  |
|-------------|--|
| 200         | Success, also when customer is not found (matchingPersons list is empty) |
| 400         | Invalid request, missing parameter                                       |
| 401         | Authentication headers for http basic auth are missing                   |

20.3.2019

|     |   |
|-----|---|
| 403 | Access forbidden, invalid credentials, updateRegisterId is invalid, SAP order or billing customer does not belong to authenticated user, given updateRegisterId does not belong to given SAP order customer |
| 404 | Page not found, invalid url   |
| 500 | Internal server error from service side   |

20.3.2019

## **9. Notes**

- This service is only meant to be used for Finnish addresses and people