

# Technical Specification

## Person API



## Table of Contents

|  |           |
|--|-----------|
| <b>1 Introduction</b>                    | <b>3</b>  |
| 1.1 Prerequisites                        | 3         |
| <b>2 Domain model</b>                    | <b>4</b>  |
| <b>3 Entities</b>                        | <b>5</b>  |
| 3.1 Properties                           | 5         |
| 3.2 Person                               | 5         |
| 3.2.1 Person attributes                  | 5         |
| 3.2.2 Person extensions                  | 6         |
| 3.2.3 Person extra attributes            | 7         |
| 3.3 Bus card information                 | 8         |
| 3.3.1 BusCardInfo                        | 8         |
| 3.3.2 Student                            | 8         |
| <b>4 Services</b>                        | <b>9</b>  |
| 4.1 Export of person                     | 9         |
| 4.2 Export of persons using mail address | 9         |
| 4.3 Export of multiple person            | 9         |
| 4.4 Update person contact information    | 10        |
| 4.5 Create extra attributes              | 10        |
| 4.6 Get extra attributes                 | 11        |
| 4.7 Update extra attributes              | 11        |
| 4.8 Delete extra attributes              | 11        |
| 4.9 Get bus card information             | 12        |
| 4.10 Update bus card information         | 12        |
| 4.11 Delta export of person information  | 13        |
| 4.11.1 Use cases                         | 14        |
| <b>5 Appendix</b>                        | <b>15</b> |
| 5.1 References                           | 15        |
| 5.2 Vocabulary                           | 15        |

## 1 Introduction

The Person API provides services that make it possible to get, create and update information of persons – and some person related information – in the Tieto Education system. The services in the Person API are accessed via URIs.

This document describes following:

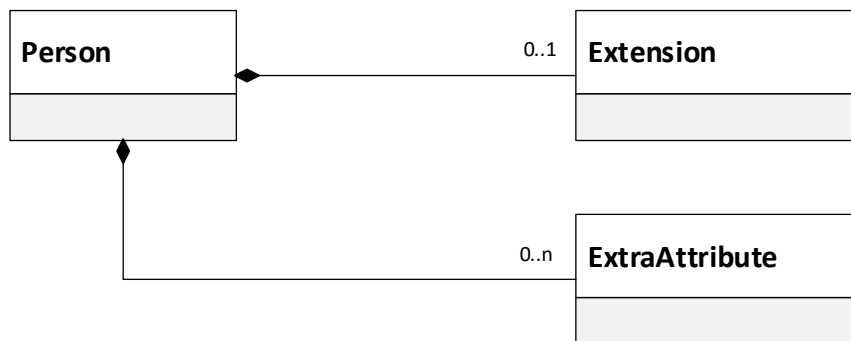
- Which person information that can be fetched and updated.
- How the entities are structured.

### 1.1 Prerequisites

Before you can get access to the Person API you must request a license key from Tieto. You should have basic knowledge about the W3C XML before you start using the API.

## 2 Domain model

The diagram below describes how the information of person and extra attributes are related.



The diagram below describes how the information of bus card information are related.



## 3 Entities

### 3.1 Properties

The delivery information of the Person packaging that is used to manage the exchange of data between source and target system.

| Name       | Description  |
|------------|--|
| datasource | This is the source identifier of data source   |
| comments   | Input values for the service.  |
| type       | Enumerated as PersonInfo, PersonSearch, DeltaPersonInfo, ExtraAttributes and BusCardInfo     |
| datetime   | The date and time the XML information were generated. The format is as [YYYY-MM-DDTHH:MM:SS] |
| timeframe  | Timeframe for the person data  |

### 3.2 Person

The person object contains information about person in Tieto Education Student management system. All persons are identified with a personal identity and Globally Unique Identifier (GUID). The personal identity could be temporary or generated from Skatteverket, for more information see [www.skatteverket](http://www.skatteverket)

#### 3.2.1 Person attributes

| Name                         | Description  | Read-only |
|------------------------------|--|-----------|
| sourcedid.id                 | The persistent unique identity of the person as a GUID                   | Yes       |
| userid.value                 | Alternative unique identifiers of the person                             | Yes       |
| userid.useridtype            | Enumerated as PID, GUID  | Yes       |
| name.fn                      | Formatted name ([middle name] [lastname], [firstname])                   | Yes       |
| name.n.given                 | The given name and not necessarily the first name                        | Yes       |
| name.n.partname.partnametype | The type of partname. "Middle" is used for middle name.                  | Yes       |
| name.n.partname.Value        | Middle name  | Yes       |
| name.n.family                | This is the family name  | Yes       |
| demographics.bday            | The birth date of the Person. The format is [YYYY-MM-DD]                 | Yes       |
| demographics.gender          | The gender of the Person. This is enumerated as Unknown, Female and Male | Yes       |
| emailhome                    | Email home address   | No        |
| emailworkschool              | Work email for staff, school email for students                          | No        |
| emailworkschool2             | Work email for staff, school email for students                          | No        |

| Name         | Description   | Read-only |
|--------------|---|-----------|
| tel          | Phone numbers   | No        |
| tel.teltype  | Identifies the type of telephone number. Enumerated as Voice, Mobile and Work, where Voice is home phone.   | No        |
| adr.pcode    | The post identity number  | Yes       |
| adr.street   | The actual street address   | Yes       |
| adr.locality | The locality component of the address e.g., the City  | Yes       |
| adr.extadd   | C/O address   | Yes       |
| mergeduser   | The persistent unique identity of the person, is always a GUID. Only present when recstatus="4"   | Yes       |
| recstatus    | An attribute present in delta indicating what type of change has happened to the person element. If no recstatus is present, it can be assumed that the person was created or updated.<br>1 The person is created<br>2 The person is updated<br>3 The person is removed<br>4 The person has been merged with another person | Yes       |

### 3.2.2 Person extensions

Common extended properties for all persons

| Name              | Description  | Read-only |
|-------------------|--|-----------|
| privacy           | true, if the person has privacy restrictions, false if no restrictions.<br>"level" indicates type of privacy<br>1 Address information is protected.<br>2 National registration is protected<br>3 1 and 2 | Yes       |
| geographickeycode | Geographic key code  | Yes       |
| municipalitycode  | Municipality code, formatted [NNNN], for more information see <a href="http://www.skl.se">www.skl.se</a>   | Yes       |
| municipalityname  | Municipality name  | Yes       |
| nativelanguage    | Native language, formatted as longer a form of language name   | Yes       |
| timestamp         | Date and time when person is created, updated or deleted.  | Yes       |
| status            | Status<br>Normal The person is alive and lives in Sweden<br>Deceased The person has deceased<br>Emigrated The person has emigrated   | Yes       |
| residentstatus    | AsylumSeeker Asylum seeker<br>EuEssSwitzerlandCitizen Citizen within EU, ESS, Schweiz<br>NordicCitizen Nordic citizen<br>ExchangeStudent Exchange student<br>SwedishForeign Swedes that live abroad      | Yes       |

| Name                                | Description  | Read-only |
|-------------------------------------|--|-----------|
|                                     | OtherForeignCitizen      Other foreign citizen<br>DiplomatChildrenWithin      Diplomatic children within EU, ESS,<br>EuEssSwitzerland      Schweiz<br>DiplomatChildrenOutside      Diplomatic children outside EU, ESS,<br>EuEssSwitzerland      Schweiz<br>PermitApplicationSeeker      Permit application seeker<br>Unknown      Unknown Resident status |           |
| contactpersons                      | List of contact persons  | Yes       |
| contactpersons.contactpersonType    | Type of contact person<br>Guardian<br>OtherResponsible<br>Contact  | Yes       |
| contactpersons.contactperson.usedid | The persistent unique identity of the person, can be PID or GUID   | Yes       |
| children                            | List of children   | Yes       |
| children.child.type                 | Type of child type<br>Guardian<br>OtherResponsible<br>Contact  | Yes       |
| children.child.userid               | The persistent unique identity of the person, can be PID or GUID   | Yes       |
| extrainformation                    | List of extra information  | Yes       |
| extrainformation.title              | Title of the extra information   | Yes       |
| extrainformation.start              | Start date for the extra information   | Yes       |
| extrainformation.end                | End date for the extra information   | Yes       |

### 3.2.3 Person extra attributes

It is possible to attach extra information to a person. This information will not be visible anywhere in the Tieto Education interface. The API ensures that the data is stored and can be retrieved on demand. When a person is deleted in the UI the extra attributes attached to that person are also deleted. These are modelled on a combination of sources to provide enough freedom for a range of integration scenarios. See Section 5.1 for more information.

| Name        | Description  | Read-only                                  |
|-------------|--|--|
| attributeld | GUID that uniquely identifies this set of attributes   | Yes  |
| service     | A descriptive identifier what these attributes are for. Examples can be Azure Ad, Google.<br>Three service names are reserved by Tieto Education for possible future use, these are <b>eduroam</b> , <b>swamid</b> , <b>skolfederation</b> | Required during create, cannot be changed. |

| Name                       | Description  | Read-only |
|----------------------------|--|-----------|
| eduPersonPersistentId      | Max 256 characters. Free to use for any purpose. Modelled on section 8.3.7 of SAML 2.0 core                            | No        |
| eduPersonPrincipalName     | Max 256 characters. Free to use for any purpose. Recommendation is to map this to UserPrincipalName in AD or Azure AD. | No        |
| eduPersonUniqueId          | Max 321 characters. Free to use for any purpose.   | No        |
| eduPersonTargetedId        | Max 2500 characters. Free to use for any purpose   | No        |
| eduPersonAffiliation       | A list of strings, each string can be max 256 characters. Free to use for any purpose                                  | No        |
| eduPersonScopedAffiliation | A list of strings, each string can be max 256 characters. Free to use for any purpose                                  | No        |
| eduPersonEntitlement       | A list of strings, each string can be max 1024 characters. Free to use for any purpose                                 | No        |

### 3.3 Bus card information

The bus card information contains information about a bus card, and its user.

#### 3.3.1 BusCardInfo

| Name                 | Description   | Read-only |
|----------------------|---|-----------|
| student              | User of the bus card  | Yes       |
| eligible-for-buscard | Tells if the student eligible for a bus card or not. Not currently used   | Yes       |
| dateframe.start      | First day when the bus card is valid. Iso-8601 formatted date   | No        |
| dateframe.end        | Last day when the bus card is valid. ISO-8601 formatted date. Defaults to 9999-12-31 which means no end date set. | No        |
| distance             | Distance in km between where the student lives and the school.  | No        |
| buscardtype          | Type of bus card; possible values are depending on the settings in Tieto Education                                | No        |
| buscardnumber        | Number of the bus card assigned to this student   | No        |

#### 3.3.2 Student

| Name             | Description                       | Read-only |
|------------------|-----------------------------------|-----------|
| student.id       | Unique identifier for the student | Yes       |
| student.personid | Person Id for the student         | Yes       |
| student.name     | Students full name                | Yes       |



## 4 Services

All services are based on HTTP REST technology. “Get services” are using the method HTTP GET, “Update services” and “Create services” are using the method HTTP POST and “Delete services” are using the method HTTP DELETE.

### 4.1 Export of person

This service returns person information, in the <person-root>-<person>element.

Parameters:

| Parameter       | Mandatory | Description  |
|-----------------|-----------|--|
| PersonId        | Yes       | Id can be passed as GUID or PID for the person.  |
| ExtraAttributes | No        | Boolean indicating if any existing extra attributes associated to the person should be included in the response. If this parameter is not present, the value <b>false</b> is assumed |

Services:

| Service       | Description                |
|---------------|----------------------------|
| GetPersonInfo | Returns person information |

### 4.2 Export of persons using mail address

This service returns person information, in the <person-root>-<person>element.

Parameters:

| Parameter     | Mandatory | Description                             |
|---------------|-----------|---|
| EEmailAddress | Yes       | The email address to search persons for |

Services:

| Service                        | Description                |
|--------------------------------|----------------------------|
| GetPersonInfoFromEEmailAddress | Returns person information |

### 4.3 Export of multiple person

This service returns multiple person information, in the <person-root>-<person>element. This method is called as a HTTP POST. The data returned will consist of the persons actually found.

Parameters:

| Parameter       | Mandatory | Description   |
|-----------------|-----------|---|
| ExtraAttributes | No        | Boolean indicating if any existing extra attributes associated to the persons should be included in the response. If this parameter is not present, the value <b>false</b> is assumed |

Input data:

The list of persons requested shall be appended as an xml file in the content of the request with the elements <person-root>-<personsearch>. The <properties>-<type> must have the value PersonSearch.

Services:

| Service             | Description                         |
|---------------------|-------------------------------------|
| GetPersonSearchInfo | Returns multiple person information |

## 4.4 Update person contact information

This service is used to update contact information of persons (email and telephone numbers). If an email or phone number is not supplied in the supplied data, the corresponding value in Tieto Education is not changed.

As input this service takes XML information according to the XML schema for “<person-root><person>” in the Person API xsd. The <properties>-<type> must have the value PersonInfo.

Services:

| Service                  | Description                        |
|--------------------------|------------------------------------|
| UpdateContactInformation | Update persons contact information |

## 4.5 Create extra attributes

This service is used to create a new set of attributes for a service and person. A complete person object with services is returned.

It is not allowed to create extra attributes for services that the person already has.

As input this service takes XML information according to the XML schema for <person> in the Person API xsd. The identification of the person is done by setting one of the <userid>-elements. No other elements (beside “extraattribute”) will be handled in this operation and can therefore be omitted.

Services:

| Service               | Description                                     |
|-----------------------|---|
| CreateExtraAttributes | Create attributes for one or more new services. |

#### 4.6 Get extra attributes

This service allows fetching the attributes for a single service and person.

Parameters

| Parameter   | Mandatory | Description  |
|-------------|-----------|--|
| attributeld | Yes       | GUID that uniquely identifies this service and person. |

Services:

| Service            | Description  |
|--------------------|--|
| GetExtraAttributes | Returns the attributes for a single service and person |

#### 4.7 Update extra attributes

This service allows updating attributes for one or more services and persons. Any attributes left out of the request will be deleted if they are previously stored.

As input this service takes XML information according to the XML schema for <person-root>-<extraattributes> in the Person API xsd.

Services:

| Service               | Description  |
|-----------------------|--|
| UpdateExtraAttributes | update attributes for one or more stored services. |

#### 4.8 Delete extra attributes

This service allows deleting attributes for a single service

Parameters

| Parameter   | Mandatory | Description  |
|-------------|-----------|--|
| attributeld | Yes       | GUID that uniquely identifies this service and person. |

Services:

| Service               | Description  |
|-----------------------|--|
| DeleteExtraAttributes | deletes the attributes for a single service and person |

## 4.9 Get bus card information

This service allows getting bus card information, for either a single student, or all on-going and future bus cards.

Parameters:

| Parameter  | Mandatory | Description   |
|------------|-----------|---|
| PersonId   | No        | GUID or PID that uniquely identifies the person to retrieve bus card information about. Supersedes SearchDate.                                      |
| SearchDate | No        | The search date. Current date will be used if SearchDate is not specified. The oldest search date that can be used is current date minus ten years. |
| SchoolType | Yes       | Must be "GR" for handling bus cards for students in Compulsory School, or "GY" for handling bus cards for students in Upper Secondary School        |

Services:

| Service               | Description   |
|-----------------------|---|
| GetBusCardInformation | Returns the bus card information according to the given parameters. |

## 4.10 Update bus card information

This service allows updating of bus card information for a number of students.

Each given bus card information in the input is compared to existing data in the system. For a given input that has one counterpart in the system, the system will be updated. Otherwise, a new object will be created. If more than one counterpart is found, no update is done.

The criteria for finding a counterpart is that:

- Same student
- Overlapping periods

A special case for upper secondary school is that if more counterparts are found but only one on the school where the student is placed, that bus card will be overwritten.

Valid bus card types are:

- for compulsory school

- the code name for active codes in the code register for code type "SKJUTSTYP"
- for upper secondary school
  - the code text for active codes for the school where the student's class placement is, and with code type "RESE\_FARDMEDEL"

As input this service takes XML information according to the XML schema for <person-root> with <buscardinfo> elements in the Person API xsd.

Parameters:

| Parameter  | Mandatory | Description  |
|------------|-----------|--|
| SchoolType | Yes       | Must be "GR" for handling bus cards for students in Compulsory School, or "GY" for handling bus cards for students in Upper Secondary School |

Services:

| Service                  | Description  |
|--------------------------|--|
| UpdateBusCardInformation | Update bus card information according to the input data. |

#### 4.11 Delta export of person information

This service gives the person changes done within a specified timeframe. The output is XML information with changed person entities.

These services return specific person delta information, in the <person-root><person>-element.

Parameters:

| Parameter | Mandatory | Description     |
|-----------|-----------|-----------------|
| StartDate | Yes       | Start date&time |
| EndDate   | Yes       | End date&time   |

Parameter value limitations:

- EndDate – StartDate >= 5 minutes
- EndDate – StartDate <= 24 hours
- Current time – EndDate >= 5 minutes
- Current time – StartDate <= 30 days

Services:

| Service            | Description                 |
|--------------------|-----------------------------|
| GetPersonInfoDelta | Returns person information. |

#### 4.11.1 Use cases

Some use cases for delta export of person:

| Person change   | The XML information that is output from the service contains   |
|---|--|
| Create a new person in student registry                 | Newly created person is exported if person creation date is between start and end date of the delta request.   |
| Update any existing person in student registry          | Updated person is exported if person creation date is between start and end date of the delta request.   |
| Merging of two person posts in Tieto Education into one | Following person posts are exported if person merging date is between start and end date of the delta request: <ul style="list-style-type: none"><li>- the remaining person post will have recstatus=4, and the element mergeduser present</li><li>- the deleted person post will have recstatus = 3</li></ul> |

## 5 Appendix

### 5.1 References

| Name   | Address   |
|--|---|
| IMS Enterprise Information Model                       | <a href="http://www.imsglobal.org/enterprise/entv1p1/imsent_infov1p1.html">http://www.imsglobal.org/enterprise/entv1p1/imsent_infov1p1.html</a> |
| IMS Enterprise XML Binding Specification               | <a href="http://www.imsglobal.org/enterprise/entv1p1/imsent_bindv1p1.html">http://www.imsglobal.org/enterprise/entv1p1/imsent_bindv1p1.html</a> |
| IMS Enterprise Best Practices and Implementation Guide | <a href="http://www.imsglobal.org/enterprise/entv1p1/imsent_bestv1p1.html">http://www.imsglobal.org/enterprise/entv1p1/imsent_bestv1p1.html</a> |
| Person API XML schemas and examples                    | <a href="https://github.com/ExchangeServices">https://github.com/ExchangeServices</a>   |
| SAML 2.0   | <a href="http://saml.xml.org/saml-specifications">http://saml.xml.org/saml-specifications</a>   |
| SWAMID   | <a href="https://www.sunet.se/swamid/">https://www.sunet.se/swamid/</a>   |
| Skolfederation   | <a href="https://www.skolfederation.se/">https://www.skolfederation.se/</a>   |
| REFEDS EduPerson                                       | <a href="https://wiki.refeds.org/pages/viewpage.action?pageId=44957737">https://wiki.refeds.org/pages/viewpage.action?pageId=44957737</a>       |

### 5.2 Vocabulary

| Word | Description                                 |
|------|---|
| PID  | Personal identity, formatted [YYYYMMDDNNNN] |
| GUID | Global unique identifier                    |