

**Obsolete**

**People Profile Import specification document**

**Contact**: Pascal Grossniklaus, Founder  
 +41 58 329 31 02 | mobile: +41 79 474 97 66

[pascal@aheadintranet.com](mailto:pascal@aheadintranet.com)

**Version**: 0.9.5

Table of Contents

[Version History 3](#_Toc144280313)

[Introduction 4](#_Toc144280314)

[Accepted values for field types 5](#_Toc144280315)

[Authentication & Authorization 6](#_Toc144280316)

[Endpoint 7](#_Toc144280317)

[Request & Response format 8](#_Toc144280318)

[Person update JSON document 8](#_Toc144280319)

[Multipart-Request 8](#_Toc144280320)

[JSON request 9](#_Toc144280321)

[Response 9](#_Toc144280322)

[Examples 11](#_Toc144280323)

[People profile definition (PPD) 11](#_Toc144280324)

[Update of simple fields 11](#_Toc144280325)

[Update of a file 11](#_Toc144280326)

# Version History

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 0.9 | 21.07.2023 | First version |
| 0.9.1 | 10.08.2023 | * “userPrincipalName” may be used as correlation id. * Return code of 413 when a request exceeds the maximum number of updated people |
| 0.9.2 | 17.08.2023 | * Clarification of the HTTP Verb to use on the API * Ability to send a JSON Array for profile updates without file uploads |
| 0.9.3 | 30.08.2023 | * Specifying the valid input for certain field types |
| 0.9.4 | 14.09.2023 | * Simplify file upload API usage |
| 0.9.5 | 18.09.2023 | * Mentioning the total size limit of a request of 104MB |
| --- | 22.11.2023 | * This document has been superseded by the documentation at <https://aheadintranet.github.io/ahead.people-profile-import/> |

# Introduction

This document describes ahead’s *people profile import* (**PPI**) API, which allows to merge external values into ahead user profiles.

By default, ahead provides access to all common fields stored in the *Azure Active Directory* (**AAD**, or via its new product name, MS Entra). By providing additional values through the API described in this document, customers will be able to reference values unknown to AAD such that they become part of ahead intranet’s user interface.

A diagram of a group of people

Description automatically generated

The relationships between the API, the people profile definition (PPD), the ahead people store (APS) and its result in the user interface

The *people profile definition* (**PPD**[[1]](#footnote-2)) is a JSON document that will be prepared in collaboration with ahead’s customer success managers and that describes all the known fields of “people” in ahead’s user interface and how those fields are used throughout the UI.

The *ahead people store* (**APS**) is our database that is regularly synchronized with the customer’s AAD and provides the basis to merge any additional fields into the profile of a person.

The intended audience of this document are developers and administrators tasked with synchronizing values into ahead’s people store.

## Accepted values for field types

Most field types are string-based and hence accept any string. Some of those types will be used in a different way in ahead’s UI.

* **Email**: will be rendered as link with “mailto:” prefix
* **Phone**: will be rendered as link with “tel:” prefix
* **Link**: will be rendered as link
* **TranslationKey**: a string that will be used for a lookup in the translation files. You can consult with ahead’s Customer Success Managers (CSMs) how to import your own translations into ahead for specific keys

The schema’s *isList*-property for field definitions is currently constrained to the field types **String,** **TranslationKey** and **File**.

* When using a field of type **Date**, the date input must be formatted like the date part of the ISO8601 formatting, namely yyyy-MM-dd, e.g. 2023-12-24 for Christmas Eve in 2023.

# Authentication & Authorization

Authentication & Authorization is done by generating an API-Key through ahead’s administration settings. There it is possible to:

* obtain one or several API-Keys and give them a descriptive name to be able to identify import protocols (see chapter on “Responses”)
* delete an API-Key in case that it is not in use anymore or has otherwise leaked into a system with unrestricted visibility.

Please take care of your API-Keys. They allow you to import data into the profiles of your users that use ahead. If you suspect that the key has been compromised, use the ahead administration interface to revoke the API-Key in question.

When calling the API, you need to provide the API-Key as copied out from ahead as value to the custom “*x-functions-key*” HTTP-Header.

# Endpoint

The API URL of the import is:

**https://connect.aheadintranet.com/api/PeopleImport**

and accepts **PUT**s With the following headers:

|  |  |
| --- | --- |
| Key | Value |
| x-functions-key | The API-Key necessary to authenticate the request |
| Content-Type | **multipart/form-data**[[2]](#footnote-3) or **application/json**  (see chapters Multipart-Request and JSON request) |

The following content types are supported for the parts of a multipart request:

|  |  |
| --- | --- |
| Content type | Description |
| application/json | A JSON document |
| application/pdf, | A \*.pdf document |
| *image/jpeg, png, webp[[3]](#footnote-4)* | *A jpg, png or webp image* |

The specifics of the request will be described in the following chapters.

# Request & Response format

## Person update JSON document

A person update JSON document may appear as

* part with content-type: application/json in a multipart
* as element of a JSON array within the body of a request whose content-type is application/json.

Each JSON document **must** contain **one of**

* the “**employeeId**” field. This field needs to correlate with a person’s “employeeId” value as stored in the AAD.
* the “**userPrincipalName**” (UPN). This field needs to correlate with a person’s user principal name in the AAD / MS Entra.

The document should furthermore contain values for all or a subset of the fields that are marked with source = API in the people profile definition.

Provided that ahead is successful in correlating the person via the one of the correlation fields, the provided values will be merged with the existing values. The following rules apply:

* Any fields not referenced will be left untouched.
* Provided fields will overwrite any previously existing values.
* If the API call sends “null” for a field, ahead will take this value and use it to overwrite / reset that specific value.

## ­Multipart-Request

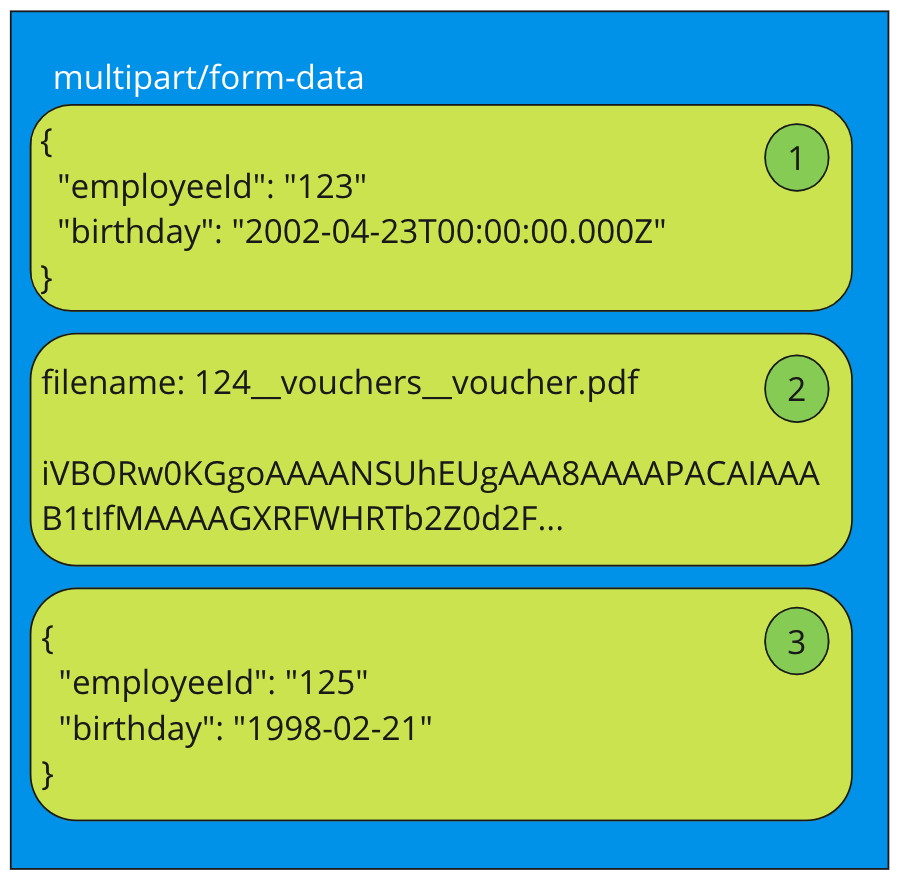
A request may contain up to **50** parts as described in “Person update JSON document”. Each part corresponds to the update of one person.

Special rules apply to fields of type “Image” and “File”. In this case we expect a part with a specific file name that informs ahead about how to proceed with the uploaded file.

The filename should be constructed as:

{userIdentifier}\_\_{fieldName}\_\_{fileName}

The *userIdentifier* is either the *employeeId* or the *upn*. The fieldname refers to the field as defined by the PPD



Example of a request and how the parts are counted towards the maximum limit of 50 per request. Here, 3 employees obtain updates, 123 and 125 for their birthdate, 124 gets a new uploaded file related to the “vouchers” field that needs to be defined in the PPD.

## 

Regardless of whether you send 50 files on a request or, please note that the total size of the request should NOT exceed **104 Megabytes**. The request will not be processed if it exceeds that size.

## JSON request

A request with the content-type application/json may contain a JSON array with **up to** **50** people update documents as described in “Person update JSON document”.

This kind of API usage only supports the update of fields directly related to profiles. It will **not** allow the upload of files.

## Response

**Status Code 200**: This means that the request has been processed without issues. The body of the response will contain a GUID with which this import call can be identified in the import logs (which will be accessible through ahead’s admin interface)

**Status Code 400**: If there’s any issue with the request at a basic level (e.g. wrong content type), a status code of 400 (Bad request) will be returned. Another possible reason is when all parts of the request failed to be processed.

**Status Code 403:** A request without API-Key or an unknown API-Key will cause a return of 403. Such calls will not appear in the protocol.

**Status Code 413:** A request contains more than 50 people to update.

**Status Code 422**: If there are issues with the data itself, which may only apply to a part of the sent data, a status code of 422 (unprocessable content) will be returned. The body of the response will contain a GUID with which this import call can be identified such that it can be analyzed which specific import command went wrong.

Conditions that would lead to a 422 response would be:

* A particular JSON document could not be correlated with an existing entry (missing / wrong employeeId field)
* A JSON document contains fields that are not known as defined in the People Profile Definition
* A field is known, but contains the wrong kind of data

Imports will be accepted on an employee-by-employee basis. That means that when the API returns the 422 error code, some of the sent employees may have been successfully imported. In such cases, you will need to identify the proper logs that are correlated with the GUID provided in the body of the response through ahead’s “Integrations” section in the management pages

# Examples

## People profile definition (PPD)

Provided the people profile definition contains the following field definitions:

{

"name": "nickName",

"fieldType": "String",

"source": "Api",

"icon": "comment"

},

{

"name": "yearsOfService",

"fieldType": "Number",

"source": "Api",

"icon": "cake"

},

{

"name": "payroll",

"fieldType": "File",

"isList": true,

"source": "Api"

}

## Update of simple fields

Then an update of the simple fields could look like a multipart/form-data request with a single part of type application/json containing:

{

"employeeId": "1234",

"nickName": "Joe"

}

This would update the employee’s “nickname” field who has the employee id “1234” with Joe.

## Update of a file

An upload of a file would need the file name to be   
“1234\_\_payroll\_\_March 2023.pdf” – the user with the id 1234 will then find this file in their personal space under the category “payroll” with the name “March 2023.pdf”

At <https://github.com/aheadintranet/ahead.people-profile-import> you will find supplementary material (e.g. code samples) to help you implement calling the import API.

1. 1 Describes which fields are used where and with what functionality (e.g. editable or not). [↑](#footnote-ref-2)
2. The structure of a multipart/form-data request is formally described at [https://ww w.rfc-editor.org/rfc/rfc2388](https://www.rfc-editor.org/rfc/rfc2388). It forms the basis of e.g., file uploads from HTML form submissions. [↑](#footnote-ref-3)
3. Image upload is not implemented yet [↑](#footnote-ref-4)