ESCO REST API Docs

Doc Writer EC

Version 1.0.2, 09/07/2021

Table of Contents

Introduction

The `ESCO REST API` exposes `ESCO` as a simple, `RESTful API` .

Overview

HTTP verbs

The ESCO API tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP verbs.

Verb	Usage
`GET`	Used to retrieve an ESCO resource

HTTP status codes

The ESCO API tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP status codes.

Status code	Usage
`200 OK`	The request completed successfully
`400 Bad Request`	The request was malformed (e.g. illegal value for a given path variable or query parameter). The response body will include an error providing further information
`404 Not Found`	The requested resource did not exist
`500 Internal Server Error`	An exception was raised

Headers

Every response has the following header(s):

Name	Description
`Content-Type`	The media type of the returned content ('application/json;charset=UTF-8' or 'application/hal+json;charset=UTF-8')
`Content-Length`	The size of the response body

Errors

Whenever an error response (status code >= 400) is returned, the body will contain a `JSON object` that describes the problem. The error object has the following structure:

Path	**Type**	**Description**
`logref`	`String`	Log ref
`status`	`Number`	The HTTP status code applicable to this problem.
`message`	`String`	Summary of the problem.
`_links`	`Object`	Links section

For example, a request that attempts to retrieve a given ESCO resource but applies a non-existent class path in the request will produce a `400 Bad Request` response:

Request

```
  GET /resource/taxonom?uri=http://data.europa.eu/esco/concept-scheme/occupations HTTP/1.1
Accept: application/json
```

Response

```
  HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
Content-Length: 146

{
  "logref" : "BadInputException",
  "status" : 400,
  "message" : "Illegal/Unknown type parameter values: ('taxonom')",
  "_links" : null
}
```

ESCO resources

The ESCO API describes and exposes resources that belong to a certain class. The main classes described in the ESCO API are listed in Main classes.

In ESCO each resource can be identified and retrieved by means of its URI, the unique and persistent identifier of the resource.

JSON HAL representation

The ESCO API is hypermedia driven. `Resources` include links to other `resources` in their responses. Responses are in Hypertext Application Language (HAL) (http://stateless.co/hal_specification.html) format. `Links` can be found beneath the `_links` key. Users of the API should not create URIs themselves, instead they should use the above-described links to navigate.

The payload of a response in the ESCO API is a `JSON` object according the `HAL` model representing the requested `Resource` .

General format of the JSON representation of a Resource

```
  {
  "someResourcePropertyA":null,
  "someResourcePropertyB":null,
  "title" : "",
  "_links" : {
     "self": {
        "uri" : "",
        "href" : "",
        "title" : "",
     },
     "someRelationA":[{
        "uri" : "",
        "href" : "",
        "title" : ""
     },{}],
    "someRelationB" :[]
  },
  "_embedded" : {
     "someRelationC" : [{
        "someResourcePropertyX":null,
        "title" : "",
        "_links" :{
            "self": {},
            "someRelation":[]
        },
        "_embedded" :{}
      },{}],
     "someRelationD" :[]
  }
}
```

Properties on the `JSON` -object representing a `Resource` :

Property	**Type**	**Description**
Resource properties (Direct properties on the JSON object)	Any (i.e. `String` , `Number` , `Object` , `Array` ,...)	The resource section contains general metadata fields describing the returned `Resource`

Property	Type	Description
`title`	`String`	The default label of the resource. This property provides a human-readable identifier in the negotiated language.
`_links`	`Object`	The links section. The value of this property is a map containing links to other related resources indexed by relation name. A link to a `Resource` is represented by a `Link-object`. Each `Resource`-object contains a 'self' link, a relation whose target is the described resource itself.
`_embedded`	`Object`	The embedded resource section. The value of this property is a map containing embedded resources (i.e. other resources contained within it) indexed by relation name.

Properties on a `Link`-object representing a link to a related `Resource`:

Property	Type	Description
`uri`	`String`	The uri of the related resource. It is the unique and persistent identifier of the related resource.`
`href`	`String`	The `hyperlink` to navigate to the related resource.

Property	**Type**	**Description**
`title`	`String`	The default label of the related resource. This property provides a human-readable identifier in the negotiated language.
`code`	`String`	An optional code of the related resource. See codelists.

All `Link`-objects and (embedded) `Resource`-objects are labelled with a `title`-property:

- a human-readable identifier in the negotiated language.
- If the default label is not available in the negotiated language, the label in the fallback language 'english' is shown.

The `title`-property should not be seen as metadata. It is purely descriptive and provides a default label with a human-readable identifier.

Use Cases ('How-to' guides)

How to use it with Docker

After running the following command the ESCO API will be accessible on http://localhost/esco/api/ (http://localhost/esco/api/).

Note: All occurences of `${PATH_TO_ESCO_LOCAL_API}` should be replaced by the real path of the downloaded and uncompressed ESCO API.

Docker command

DOCKER

```
docker run \
  --cpus 2 \
  -it \
  --rm \
  -p 80:8080 \
  -v ${PATH_TO_ESCO_LOCAL_API}/webapps/fuseki.war:/usr/local/tomcat/webapps/fuseki.war \
  -v ${PATH_TO_ESCO_LOCAL_API}/webapps/esco-solr.war:/usr/local/tomcat/webapps/esco-solr.wa
  -v ${PATH_TO_ESCO_LOCAL_API}/webapps/esco#api.war:/usr/local/tomcat/webapps/esco#api.war
  -v ${PATH_TO_ESCO_LOCAL_API}/work:/usr/local/tomcat/work \
  -v ${PATH_TO_ESCO_LOCAL_API}/bin/setenv.sh:/usr/local/tomcat/bin/setenv.sh \
  tomcat:9.0-jdk8
```

Supported Model Specifications

Main classes

The main classes of resources described by the ESCO API:

Class name	Class path	Description	Example
Taxonomy	taxonomy	The class of all concept scheme resources. A concept scheme represents a 'compiled vocabulary' such as a classification, taxonomy, name authority list, code list or collection. ESCO organizes different 'kind' of concept resources in different concept schemes. Concepts come in 'compiled vocabularies'. Concepts might also be grouped or organised into collections.	see used concept schemes in ESCO

Class name	Class path	Description	Example
Concept	concept	The class of all generic concept resources. The generic class used to represent resources in 'compiled vocabularies' such as classifications, taxonomies, name authority lists, code lists or collections.	ISCO Groups and skill groups
Occupation	occupation	The class of all occupation member resources. An occupation resource is a more specific concept resource.	All ESCO occupations
Skill	skill	The class of all skill member resources. A skill resource is a more specific concept resource.	All ESCO skills

Known concept schemes

Main concept schemes

The main concept schemes used to organize the concept resources described by the ESCO API:

Concept Scheme URI	Name	Description

Concept Scheme URI	Name	Description
http://data.europa.eu/esco/concept-scheme/occupations	ESCO Occupations	Represents the concept hierarchy of the full occupation pillar in ESCO. The top level concepts in the hierarchy are the ISCO groups.
http://data.europa.eu/esco/concept-scheme/skills	ESCO skills	Represents the concept hierarchy of the full skill pillar in ESCO. The transversal skills are organized into skill transversal groups and language groups.
http://data.europa.eu/esco/concept-scheme/isco	ISCO 2008	Represents the ISCO group hierarchy.

Supporting concept schemes

The concept schemes used to organize small supporting controlled vocabularies in the ESCO API:

Concept Scheme URI	Name	Description
http://data.europa.eu/esco/concept-scheme/skill-competence-type	ESCO Skill Pillar concept (sub-) Types	Controlled vocabulary of the different 'skill types': `skill/competence and knowledge`

Concept Scheme URI	Name	Description
http://data.europa.eu/esco/concept-scheme/skill-reuse-level	ESCO Skill Pillar concept reusability levels	Controlled vocabulary of the different skill reuse levels
http://data.europa.eu/esco/concept-scheme/regulated-professions	Regulated Professions	Controlled vocabulary of regulated profession notes

Supporting collections

Concept Scheme URI	Name	Description
http://data.europa.eu/esco/concept-scheme/skill-transversal-groups	ESCO transversal skill groups	Represents the transversal skill(group) hierarchy excluding the languages
http://data.europa.eu/esco/concept-scheme/skill-ict-groups	Information and communications technology	Represents the transversal ICT skill(group) hierarchy excluding the languages. This collection is a subset of the ESCO transversal skill groups
http://data.europa.eu/esco/concept-scheme/skill-language-groups	Language	Represents the transversal language skill hierarchy

Supporting code lists

Concepts might have one or more codes or notations. A notation is a string of characters used to uniquely identify a concept within the scope of a given code list. A notation is different from a lexical label in that a notation is not normally recognizable as a word or sequence of words in any natural language.

Available code lists in ESCO:

Code list URI	Name	Description
`http://data.europa.eu/esco/Notation/ISCO08`	`ISCO 08 code`	`All ISCO 08 groups have a unique ISCO 08 code`

Contributing

The ESCO API is intended to make it easy for you to explore, browse and search for ESCO resources in an automated way. However, we can’t achieve that goal without your contributions.

Questions

If you have any questions about the ESCO API, please contact the ESCO Secretariat.
email EMPL-ESCO-SECRETARIAT@ec.europa.eu

Bugs

If you believe you have found a bug, please contact the ESCO Secretariat.
email EMPL-ESCO-SECRETARIAT@ec.europa.eu

Enhancements

If you’d like an enhancement to be made to the ESCO API, please contact the ESCO Secretariat.
email EMPL-ESCO-SECRETARIAT@ec.europa.eu

Privacy Statement

During the usage of the ESCO Service API some information from your input is retained. No personal data is stored, only the following information:

(1) Type and parameters of the call

(2) Time of the call

(3) Location of the request.

The purpose of this collection of data is: to track the usage of the ESCO classification; to use the information as a feedback mechanism for future releases of the ESCO classification