SLIPO API Python client

Release 0.1.6

Yannis Kouvaras

Dec 16, 2019

CONTENTS

This document provides information on how to use the *SLIPO API* implemented by the SLIPO project.

The *SLIPO* API can be used for:

- Access the remote user file system to browse, upload and download files
- Query resource catalog and download RDF datasets. All data are encoded in *N-Triples* format
- Manage existing POI data integration processes
- Execute a single SLIPO Toolkit component operation. Currently, SLIPO supports *Transform*, *Interlink*, *Fuse* and *Enrichment* operations

CHAPTER

ONE

WELCOME TO SLIPO API PYTHON CLIENT'S DOCUMENTATION!

This document provides information on how to use the *SLIPO API* implemented by the SLIPO project.

The *SLIPO* API can be used for:

- Access the remote user file system to browse, upload and download files
- Query resource catalog and download RDF datasets. All data are encoded in *N-Triples* format
- Manage existing POI data integration processes
- Execute a single SLIPO Toolkit component operation. Currently, SLIPO supports *Transform*, *Interlink*, *Fuse* and *Enrichment* operations

CHAPTER

TWO

SLIPO API CLIENT

SLIPO API entry point.

This class provides access to all SLIPO API functionality for accessing the user file system, querying the catalog, querying existing workflows and executing SLIPO Toolkit operations.

class slipo.client.**Client**(*api_key*, *base_url=None*, *requires_ssl=True*)

Class implementing all SLIPO API functionality

Parameters

- **api_key** (*str*) – SLIPO API key. An application key can be generated using the SLIPO Workbench application.
- **base_url** (*str, optional*) – Base URL for SLIPO API endpoints. The default value is `https://app.dev.slipo.eu/`.
- **requires_ssl** (*bool, optional*) – If *False*, unsecured connections are allowed (default *True*).

Returns A *Client* object.

validate() → None

Validate current application key

Raises **SlipoException** – If a network or server error has occurred.

file_browse()

Browse all files and folders on the remote file system.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

file_download(*source: str*, *target: str*, *overwrite: bool = False*) → None

Download a file from the remote file system.

Parameters

- **source** (*str*) – Relative file path on the remote file system.
- **target** (*str*) – The path where to save the file.
- **overwrite** (*bool, optional*) – Set true if the operation should overwrite any existing file.

Raises **SlipoException** – If a network, server error or I/O error has occurred.

file_upload(*source: str*, *target: str*, *overwrite: bool = False*) → None

Upload a file to the remote file system.

Note: File size constraints are enforced on the uploaded file. The default installation allows files up to 20 Mb.

Moreover, space quotas are applied on the server. The default user space is 5GB.

Directory nesting constraints are applied for the `target` value. The default installation allows nesting of directories up to 5 levels.

Parameters

- **source** (*str*) – The path of the file to upload.
- **target** (*str*) – Relative path on the remote file system where to save the file. If the directory does not exist, it will be created.
- **overwrite** (*bool, optional*) – Set true if the operation should overwrite any existing file.

Raises **SlipoException** – If a network, server error or I/O error has occurred.

catalog_query(*term: str = None*, *pageIndex: int = 0*, *pageSize: int = 10*)
Query resource catalog for RDF datasets.

Parameters

- **term** (*str, optional*) – A term for filtering resources. If specified, only the resources whose name contains the term are returned.
- **pageIndex** (*str, optional*) – Page index for data pagination.
- **pageSize** (*str, optional*) – Page size for data pagination.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

catalog_download(*resource_id: int*, *resource_version: int*, *target: str*)
Download resource to the local file system

Parameters

- **resource_id** (*int*) – The resource id.
- **resource_version** (*int*) – The resource revision.
- **target** (*str*) – The path where to save the file.

Raises **SlipoException** – If a network, server error or I/O error has occurred.

process_query(*term: str = None*, *pageIndex: int = 0*, *pageSize: int = 10*)
Query workflow instances.

Parameters

- **term** (*str, optional*) – A term for filtering workflows. If specified, only the workflows whose name contains the term are returned.
- **pageIndex** (*str, optional*) – Page index for data pagination.
- **pageSize** (*str, optional*) – Page size for data pagination.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

process_save(*process_id: int*)
Creates a new version for the specified workflow. The most recent version of the workflow is copied.

Parameters **process_id** (*int*) – The process id.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

process_start(*process_id: int*, *process_version: int*) → None
Start or resume execution of a workflow instance.

Parameters

- **process_id** (*int*) – The process id.
- **process_version** (*int*) – The process revision.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

process_stop(*process_id: int*, *process_version: int*) → None
Stop a running workflow execution instance.

Parameters

- **process_id** (*int*) – The process id.
- **process_version** (*int*) – The process revision.

Raises **SlipoException** – If a network or server error has occurred.

process_status(*process_id: int*, *process_version: int*)
Check the status of a workflow execution instance.

Parameters

- **process_id** (*int*) – The process id.
- **process_version** (*int*) – The process revision.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

process_file_download(*process_id: int*, *process_version: int*, *file_id: int*, *target: str*)
Download an input or output file for a specific workflow execution instance.

During the execution of a workflow the following file types may be created:

- `CONFIGURATION`: Tool configuration
- `INPUT`: Input file
- `OUTPUT`: Output file
- `SAMPLE`: Sample data collected during step execution
- `KPI`: Tool specific or aggregated KPI data
- `QA`: Tool specific QA data
- `LOG`: Logs recorded during step execution

Parameters

- **process_id** (*int*) – The process id.

- **process_version** (*int*) – The process revision.
- **file_id** (*int*) – The file id.
- **target** (*str*) – The path where to save the file.

Raises **SlipoException** – If a network, server error or I/O error has occurred.

profiles()
Browse all SLIPO Toolkit components profiles.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

transform_csv(*path: str*, ***kwargs*)
Transforms a CSV file to a RDF dataset.

Parameters

- **path** (*str*) – The relative path for a file on the remote user file system.
- ****kwargs** – Keyword arguments to control the transform operation. Options are:
  - **attrCategory** (str, optional): Field name containing literals regarding classification into categories (e.g., type of points, road classes etc.) for each feature.
  - **attrGeometry** (str, optional): Parameter that specifies the name of the geometry column in the input dataset.
  - **attrKey** (str, optional): Field name containing unique identifier for each entity (e.g., each record in the shapefile).
  - **attrName** (str, optional): Field name containing name literals (i.e., strings).
  - **attrX** (str, optional): Specify attribute holding X-coordinates of point locations. If inputFormat is not *CSV*, the parameter is ignored.
  - **attrY** (str, optional): Specify attribute holding Y-coordinates of point locations. If inputFormat is not *CSV*, the parameter is ignored.
  - **classificationSpec** (str, optional): The relative path to a YML/CSV file describing a classification scheme.
  - **defaultLang** (str, optional): Default lang for the labels created in the output RDF (default: *en*).
  - delimiter (str, optional): Specify the character delimiting attribute values.
  - **encoding** (str, optional): The encoding (character set) for strings in the input data (default: *UTF-8*)
  - **featureSource** (str, optional): Specifies the data source provider of the input features.
  - **mappingSpec** (str, optional): The relative path to a YML file containing mappings from input schema to RDF according to a custom ontology.
  - **profile** (str, optional): The name of the profile to use. Profile names can be retrieved using *profiles()* method. If profile is not set, the *mappingSpec* parameter must be set.
  - **quote** (str, optional): Specify quote character for string values.
  - **sourceCRS** (str, optional): Specify the EPSG code for the source CRS (default: *EPSG:4326*).

- **targetCRS** (str, optional): Specify the EPSG code for the target CRS (default: *EPSG:4326*).

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

transform_shapefile(*path: str*, ***kwargs*)

Transforms a SHAPEFILE file to a RDF dataset.

Parameters

- **path** (*str*) – The relative path for a file on the remote user file system.
- ****kwargs** – Keyword arguments to control the transform operation. Options are:
  - **attrCategory** (str, optional): Field name containing literals regarding classification into categories (e.g., type of points, road classes etc.) for each feature.
  - **attrGeometry** (str, optional): Parameter that specifies the name of the geometry column in the input dataset.
  - **attrKey** (str, optional): Field name containing unique identifier for each entity (e.g., each record in the shapefile).
  - **attrName** (str, optional): Field name containing name literals (i.e., strings).
  - **classificationSpec** (str, optional): The relative path to a YML/CSV file describing a classification scheme.
  - **defaultLang** (str, optional): Default lang for the labels created in the output RDF (default: *en*).
  - **encoding** (str, optional): The encoding (character set) for strings in the input data (default: *UTF-8*)
  - **featureSource** (str, optional): Specifies the data source provider of the input features.
  - **mappingSpec** (str, optional): The relative path to a YML file containing mappings from input schema to RDF according to a custom ontology.
  - **profile** (str, optional): The name of the profile to use. Profile names can be retrieved using *profiles()* method. If profile is not set, the *mappingSpec* parameter must be set.
  - **sourceCRS** (str, optional): Specify the EPSG code for the source CRS (default: *EPSG:4326*).
  - **targetCRS** (str, optional): Specify the EPSG code for the target CRS (default: *EPSG:4326*).

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

interlink(*profile: str, left: Union[str, Tuple[int, int], Tuple[int, int, int]], right: Union[str, Tuple[int, int], Tuple[int, int, int]]*)

Generates links for two RDF datasets.

Arguments *left*, *right* and *links* may be either:

- A `str` that represents a relative path to the remote user file system
- A `tuple` of two integer values that represents the id and revision of a catalog resource.
- A `tuple` of three integer values that represents the process id, process revision and output file id for a specific workflow or SLIPO API operation execution.

Parameters

- **profile** (*str*) – The name of the profile to use. Profile names can be retrieved using *profiles()* method.
- **left** (*Union[str, Tuple[int, int], Tuple[int, int, int]*) – The *left* RDF dataset.
- **right** (*Union[str, Tuple[int, int], Tuple[int, int, int]*) – The *right* RDF dataset.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

fuse(*profile: str, left: Union[str, Tuple[int, int], Tuple[int, int, int]], right: Union[str, Tuple[int, int], Tuple[int, int, int]], links: Union[str, Tuple[int, int], Tuple[int, int, int]]*)
Fuses two RDF datasets using Linked Data and returns a new RDF dataset.

Parameters

- **profile** (*str*) – The name of the profile to use. Profile names can be retrieved using *profiles()* method.
- **left** (*Union[str, Tuple[int, int], Tuple[int, int, int]*) – The *left* RDF dataset.
- **right** (*Union[str, Tuple[int, int], Tuple[int, int, int]*) – The *right* RDF dataset.
- **links** (*Union[str, Tuple[int, int], Tuple[int, int, int]*) – The links for the *left* and *right* datasets.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

enrich(*profile: str, source: Union[str, Tuple[int, int], Tuple[int, int, int]]*)
Enriches a RDF dataset.

Parameters

- **profile** (*str*) – The name of the profile to use. Profile names can be retrieved using *profiles()* method.
- **source** (*Union[str, Tuple[int, int], Tuple[int, int, int]*) – The RDF dataset to enrich.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

export_csv(*profile: str, source: Union[str, Tuple[int, int], Tuple[int, int, int]], **kwargs*) → dict
Exports a RDF dataset to a CSV file.

Parameters

- **profile** (*str*) – The name of the profile to use. Profile names can be retrieved using *profiles()* method.
- **source** (*Union[str, Tuple[int, int], Tuple[int, int, int]*) – The RDF dataset to export.
- ****kwargs** – Keyword arguments to control the transform operation. Options are:

- **defaultLang** (str, optional): The default language for labels created in output RDF. The default is "en".
- delimiter (str, optional):A field delimiter for records (default: *;*).
- **encoding** (str, optional): The encoding (character set) for strings in the input data (default: *UTF-8*)
- **quote** (str, optional): Specify quote character for string values (default *"*).
- **sourceCRS** (str, optional): Specify the EPSG code for the source CRS (default: *EPSG:4326*).
- **sparqlFile** (str, optional): The relative path to a file containing a user-specified SELECT query (in SPARQL) that will retrieve results from the input RDF triples. This query should conform with the underlying ontology of the input RDF triples.
- **targetCRS** (str, optional): Specify the EPSG code for the target CRS (default: *EPSG:4326*).

Returns A `dict` representing the parsed JSON response.

Raises **`SlipoException`** – If a network or server error has occurred.

`export_shapefile`(*source: Union[str, Tuple[int, int], Tuple[int, int, int]], profile: str, **kwargs*) → dict

Exports a RDF dataset to a SHAPEFILE file.

Parameters

- **`profile`** (*`str`*) – The name of the profile to use. Profile names can be retrieved using *`profiles()`* method.
- **`source`** (*`Union[str, Tuple[int, int], Tuple[int, int, int]`*) – The RDF dataset to export.
- **`**kwargs`** – Keyword arguments to control the transform operation. Options are:
  - **defaultLang** (str, optional): The default language for labels created in output RDF. The default is "en".
  - delimiter (str, optional):A field delimiter for records (default: *;*).
  - **encoding** (str, optional): The encoding (character set) for strings in the input data (default: *UTF-8*)
  - **quote** (str, optional): Specify quote character for string values (default *"*).
  - **sourceCRS** (str, optional): Specify the EPSG code for the source CRS (default: *EPSG:4326*).
  - **sparqlFile** (str, optional): The relative path to a file containing a user-specified SELECT query (in SPARQL) that will retrieve results from the input RDF triples. This query should conform with the underlying ontology of the input RDF triples.
  - **targetCRS** (str, optional): Specify the EPSG code for the target CRS (default: *EPSG:4326*).

Returns A `dict` representing the parsed JSON response.

Raises **`SlipoException`** – If a network or server error has occurred.

CHAPTER

THREE

FILE SYSTEM

Client for accessing the user remote file system

class slipo.filesystem.**FileSystemClient**(*base_url*, *api_key*)

FileSystemClient provides methods for browsing, uploading and downloading files from the user remote file system.

Details about the API responses are available at the SLIPO site.

Parameters

- **base_url** (*str*) – Base URL for SLIPO API endpoints. The default value is `https://app.dev.slipo.eu/`.
- **api_key** (*str*) – SLIPO API key. An application key can be generated using the SLIPO Workbench application.

Returns A *FileSystemClient* object.

browse() → dict

Browse all files and folders on the remote file system.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

download(*source: str*, *target: str*, *overwrite: bool = False*) → None

Download a file from the remote file system.

Parameters

- **source** (*str*) – Relative file path on the remote file system.
- **target** (*str*) – The path where to save the file.
- **overwrite** (*bool, optional*) – Set true if the operation should overwrite any existing file.

Raises **SlipoException** – If a network, server error or I/O error has occurred.

upload(*source*, *target*, *overwrite=False*) → dict

Upload a file to the remote file system.

Note: File size constraints are enforced on the uploaded file. The default installation allows files up to 20 Mb.

Moreover, space quotas are applied on the server. The default user space is 5GB.

Directory nesting constraints are applied for the `target` value. The default installation allows nesting of directories up to 5 levels.

Parameters

- **source** (*str*) – The path of the file to upload.
- **target** (*str*) – Relative path on the remote file system where to save the file. If the directory does not exist, it will be created.
- **overwrite** (*bool, optional*) – Set true if the operation should overwrite any existing file.

Raises **SlipoException** – If a network, server error or I/O error has occurred.

CHAPTER

FOUR

RESOURCE CATALOG

Client for accessing the resource catalog

class slipo.catalog.**CatalogClient**(*base_url: str*, *api_key: str*)

CatalogClient provides methods for querying the resource catalog and downloading RDF datasets. All datasets are encoded in *N-Triples* format.

Details about the API responses are available at the SLIPO site.

Parameters

- **base_url** (*str*) – Base URL for SLIPO API endpoints. The default value is `https://app.dev.slipo.eu/`.
- **api_key** (*str*) – SLIPO API key. An application key can be generated using the SLIPO Workbench application.

Returns A *CatalogClient* object.

query(*term: str = None*, *pageIndex: int = 0*, *pageSize: int = 10*) → dict

Query resource catalog for RDF datasets.

Parameters

- **term** (*str, optional*) – A term for filtering resources. If specified, only the resources whose name contains the term are returned.
- **pageIndex** (*str, optional*) – Page index for data pagination.
- **pageSize** (*str, optional*) – Page size for data pagination.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

download(*resource_id: int*, *resource_version: int*, *target: str*) → None

Download a resource to the local file system

Parameters

- **resource_id** (*int*) – The resource id.
- **resource_version** (*int*) – The resource revision.
- **target** (*str*) – The path where to save the file.

Raises **SlipoException** – If a network, server error or I/O error has occurred.

CHAPTER

FIVE

POI DATA INTEGRATION

Client for managing existing POI data integration workflows

class slipo.process.**ProcessClient**(*base_url*, *api_key*)

ProcessClient provides methods for managing existing POI data integration workflows.

Details about the API responses are available at the SLIPO site.

Parameters

- **base_url** (*str*) – Base URL for SLIPO API endpoints. The default value is `https://app.dev.slipo.eu/`.
- **api_key** (*str*) – SLIPO API key. An application key can be generated using the SLIPO Workbench application.

Returns A *ProcessClient* object.

query(*term: str = None*, *pageIndex: int = 0*, *pageSize: int = 10*) → dict

Query workflow instances.

Parameters

- **term** (*str, optional*) – A term for filtering workflows. If specified, only the workflows whose name contains the term are returned.
- **pageIndex** (*str, optional*) – Page index for data pagination.
- **pageSize** (*str, optional*) – Page size for data pagination.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

save(*process_id: int*) → None

Creates a new version for the specified workflow. The most recent version of the workflow is copied.

Parameters **process_id** (*int*) – The process id.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

start(*process_id: int*, *process_version: int*) → dict

Start or resume the execution of a workflow instance.

Parameters

- **process_id** (*int*) – The process id.
- **process_version** (*int*) – The process revision.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

stop(*process_id: int*, *process_version: int*) → None
Stop a running workflow execution instance.

Parameters

- **process_id** (*int*) – The process id.
- **process_version** (*int*) – The process revision.

Raises **SlipoException** – If a network or server error has occurred.

status(*process_id: int*, *process_version: int*) → dict
Check the status of a workflow execution instance.

Parameters

- **process_id** (*int*) – The process id.
- **process_version** (*int*) – The process revision.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

download(*process_id: int*, *process_version: int*, *file_id: int*, *target: str*) → None
Download an input or output file for a specific workflow execution instance.

During the execution of a workflow the following file types may be created:

- `CONFIGURATION`: Tool configuration
- `INPUT`: Input file
- `OUTPUT`: Output file
- `SAMPLE`: Sample data collected during step execution
- `KPI`: Tool specific or aggregated KPI data
- `QA`: Tool specific QA data
- `LOG`: Logs recorded during step execution

Parameters

- **process_id** (*int*) – The process id.
- **process_version** (*int*) – The process revision.
- **file_id** (*int*) – The file id.
- **target** (*str*) – The path where to save the file.

Raises **SlipoException** – If a network, server error or I/O error has occurred.

CHAPTER

SIX

SLIPO TOOLKIT OPERATIONS

Client for executing SLIPO Toolkit component operations

class slipo.operation.**EnumDataFormat**
Supported data formats for transform operations

class slipo.operation.**EnumInputType**
Supported input types for SLIPO Toolkit components operations

class slipo.operation.**OperationClient**(*base_url*, *api_key*)
OperationClient provides methods for executing SLIPO Toolkit components operations.

Details about the API responses are available at the SLIPO site.

Parameters

- **base_url** (*str*) – Base URL for SLIPO API endpoints. The default value is `https://app.dev.slipo.eu/`.
- **api_key** (*str*) – SLIPO API key. An application key can be generated using the SLIPO Workbench application.

Returns A *OperationClient* object.

profiles() → dict
Browse all SLIPO Toolkit components profiles.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

transform_csv(*path: str*, ***kwargs*) → dict
Transforms a CSV file to a RDF dataset.

Parameters

- **path** (*str*) – The relative path to a file on the remote user file system.
- ****kwargs** – Keyword arguments to control the transform operation. Options are:
  - **attrCategory** (str, optional): Field name containing literals regarding classification into categories (e.g., type of points, road classes etc.) for each feature.
  - **attrGeometry** (str, optional): Parameter that specifies the name of the geometry column in the input dataset.
  - **attrKey** (str, optional): Field name containing unique identifier for each entity (e.g., each record in the shapefile).
  - **attrName** (str, optional): Field name containing name literals (i.e., strings).

- **attrX** (str, optional): Specify attribute holding X-coordinates of point locations. If inputFormat is not *CSV*, the parameter is ignored.
- **attrY** (str, optional): Specify attribute holding Y-coordinates of point locations. If inputFormat is not *CSV*, the parameter is ignored.
- **classificationSpec** (str, optional): The relative path to a YML/CSV file describing a classification scheme.
- **defaultLang** (str, optional): Default lang for the labels created in the output RDF (default: *en*).
- delimiter (str, optional): Specify the character delimiting attribute values.
- **encoding** (str, optional): The encoding (character set) for strings in the input data (default: *UTF-8*)
- **featureSource** (str, optional): Specifies the data source provider of the input features.
- **mappingSpec** (str, optional): The relative path to a YML file containing mappings from input schema to RDF according to a custom ontology.
- **profile** (str, optional): The name of the profile to use. Profile names can be retrieved using `profiles()` method. If profile is not set, the *mappingSpec* parameter must be set.
- **quote** (str, optional): Specify quote character for string values.
- **sourceCRS** (str, optional): Specify the EPSG code for the source CRS (default: *EPSG:4326*).
- **targetCRS** (str, optional): Specify the EPSG code for the target CRS (default: *EPSG:4326*).

Returns A `dict` representing the parsed JSON response.

Raises **`SlipoException`** – If a network or server error has occurred.

`transform_shapefile`(*path: str*, ****kwargs*) → dict

Transforms a SHAPEFILE file to a RDF dataset.

Parameters

- **`path`** (*`str`*) – The relative path for a file on the remote user file system.
- **`**kwargs`** – Keyword arguments to control the transform operation. Options are:
  - **attrCategory** (str, optional): Field name containing literals regarding classification into categories (e.g., type of points, road classes etc.) for each feature.
  - **attrGeometry** (str, optional): Parameter that specifies the name of the geometry column in the input dataset.
  - **attrKey** (str, optional): Field name containing unique identifier for each entity (e.g., each record in the shapefile).
  - **attrName** (str, optional): Field name containing name literals (i.e., strings).
  - **classificationSpec** (str, optional): The relative path to a YML/CSV file describing a classification scheme.
  - **defaultLang** (str, optional): Default lang for the labels created in the output RDF (default: *en*).
  - **encoding** (str, optional): The encoding (character set) for strings in the input data (default: *UTF-8*)

- **featureSource** (str, optional): Specifies the data source provider of the input features.
- **mappingSpec** (str, optional): The relative path to a YML file containing mappings from input schema to RDF according to a custom ontology.
- **profile** (str, optional): The name of the profile to use. Profile names can be retrieved using `profiles()` method. If profile is not set, the *mappingSpec* parameter must be set.
- **sourceCRS** (str, optional): Specify the EPSG code for the source CRS (default: *EPSG:4326*).
- **targetCRS** (str, optional): Specify the EPSG code for the target CRS (default: *EPSG:4326*).

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

interlink(*profile: str, left: Union[str, Tuple[int, int], Tuple[int, int, int]], right: Union[str, Tuple[int, int], Tuple[int, int, int]]*) → dict

Generates links for two RDF datasets.

Arguments *left*, *right* and *links* may be either:

- A `str` that represents a relative path to the remote user file system
- A `tuple` of two integer values that represents the id and revision of a catalog resource.
- A `tuple` of three integer values that represents the process id, process revision and output file id for a specific workflow or SLIPO API operation execution.

Parameters

- **profile** (*str*) – The name of the profile to use. Profile names can be retrieved using `profiles()` method.
- **left** (*Union[str, Tuple[int, int], Tuple[int, int, int]]*) – The *left* RDF dataset.
- **right** (*Union[str, Tuple[int, int], Tuple[int, int, int]]*) – The *right* RDF dataset.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

fuse(*profile: str, left: Union[str, Tuple[int, int], Tuple[int, int, int]], right: Union[str, Tuple[int, int], Tuple[int, int, int]], links: Union[str, Tuple[int, int], Tuple[int, int, int]]*) → dict

Fuses two RDF datasets using Linked Data and returns a new RDF dataset.

Parameters

- **profile** (*str*) – The name of the profile to use. Profile names can be retrieved using `profiles()` method.
- **left** (*Union[str, Tuple[int, int], Tuple[int, int, int]]*) – The *left* RDF dataset.
- **right** (*Union[str, Tuple[int, int], Tuple[int, int, int]]*) – The *right* RDF dataset.
- **links** (*Union[str, Tuple[int, int], Tuple[int, int, int]]*) – The links for the *left* and *right* datasets.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

enrich(*profile: str, source: Union[str, Tuple[int, int], Tuple[int, int, int]]*) → dict
Enriches a RDF dataset.

Parameters

- **profile** (*str*) – The name of the profile to use. Profile names can be retrieved using *profiles()* method.
- **source** (*Union[str, Tuple[int, int], Tuple[int, int, int]]*) – The RDF dataset to enrich.

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

export_csv(*profile: str, source: Union[str, Tuple[int, int], Tuple[int, int, int]], **kwargs*) → dict
Exports a RDF dataset to a CSV file.

Parameters

- **profile** (*str*) – The name of the profile to use. Profile names can be retrieved using *profiles()* method.
- **source** (*Union[str, Tuple[int, int], Tuple[int, int, int]]*) – The RDF dataset to export.
- ****kwargs** – Keyword arguments to control the transform operation. Options are:
  - **defaultLang** (str, optional): The default language for labels created in output RDF. The default is "en".
  - delimiter (str, optional):A field delimiter for records (default: *;*).
  - **encoding** (str, optional): The encoding (character set) for strings in the input data (default: *UTF-8*)
  - **quote** (str, optional): Specify quote character for string values (default *"*).
  - **sourceCRS** (str, optional): Specify the EPSG code for the source CRS (default: *EPSG:4326*).
  - **sparqlFile** (str, optional): The relative path to a file containing a user-specified SELECT query (in SPARQL) that will retrieve results from the input RDF triples. This query should conform with the underlying ontology of the input RDF triples.
  - **targetCRS** (str, optional): Specify the EPSG code for the target CRS (default: *EPSG:4326*).

Returns A `dict` representing the parsed JSON response.

Raises **SlipoException** – If a network or server error has occurred.

export_shapefile(*profile: str, source: Union[str, Tuple[int, int], Tuple[int, int, int]], **kwargs*) → dict
Exports a RDF dataset to a SHAPEFILE file.

Parameters

- **profile** (*str*) – The name of the profile to use. Profile names can be retrieved using *profiles()* method.
- **source** (*Union[str, Tuple[int, int], Tuple[int, int, int]]*) – The RDF dataset to export.

- **`**kwargs`** – Keyword arguments to control the transform operation. Options are:
  - **defaultLang** (str, optional): The default language for labels created in output RDF. The default is "en".
  - delimiter (str, optional):A field delimiter for records (default: *;*).
  - **encoding** (str, optional): The encoding (character set) for strings in the input data (default: *UTF-8*)
  - **quote** (str, optional): Specify quote character for string values (default *"*).
  - **sourceCRS** (str, optional): Specify the EPSG code for the source CRS (default: *EPSG:4326*).
  - **sparqlFile** (str, optional): The relative path to a file containing a user-specified SELECT query (in SPARQL) that will retrieve results from the input RDF triples. This query should conform with the underlying ontology of the input RDF triples.
  - **targetCRS** (str, optional): Specify the EPSG code for the target CRS (default: *EPSG:4326*).

Returns A `dict` representing the parsed JSON response.

Raises **`SlipoException`** – If a network or server error has occurred.

CHAPTER

SEVEN

INDICES AND TABLES

- genindex

CHAPTER

EIGHT

VERSION

Version 0.1.6

PYTHON MODULE INDEX

S

INDEX

U

V