
jsonldschema Documentation

Release 0.1

Dominique Batista

Nov 02, 2018

Contents

1	CEDAR utilities	1
2	UTILITIES FOR SEMANTIC COMPARISON	3
3	UTILITIES FOR PREPARING FUNCTIONS INPUT	5
4	UTILITIES TO VALIDATE SCHEMA AND INSTANCES	7
5	Indices and tables	9
	Python Module Index	11

CHAPTER 1

CEDAR utilities

The CEDAR client will provide all the links the the CEDAR API fonctionnalities such as get, post and updates on templates, template elements, folders, instances ect ...

The schema2cedar classes will help you transform your JSON schemas draft 04 into compatible CEDAR schemas

CHAPTER 2

UTILITIES FOR SEMANTIC COMPARISON

CHAPTER 3

UTILITIES FOR PREPARING FUNCTIONS INPUT

UTILITIES TO VALIDATE SCHEMA AND INSTANCES

`jsonschema_validator.validate_instance` (*schemapath, schemafile, instancepath, instancefile, error_printing, store*)

Validate a JSON instance againsts a JSON schema.

`jsonschema_validator.validate_schema` (*path, schema_file_name*)

Validate a JSON schema given the folder/path and file name of the schema file.

`jsonschema_validator.validate_schema_file` (*schema_file*)

Validate a JSON schema given the schema file.

class `miflowcyt_validate.FlowRepoClient` (*mapping, base_schema, client_id*)

A class that provides functionality to download experiments from the FlowRepository (<https://flowrepository.org/>), transform the XML into JSON and validate the instances against their schema. The transformation from XML to JSON relies on the JSONBender library (<https://github.com/Onyo/jsonbender>).

static `get_mapping` (*mapping_file_name*)

Build the mapping dictionary based on the given mapping file :param *mapping_file_name*: the name of the mapping file :return mapping: the mapping of the fields

`get_user_content_id` (*client_identifier*)

Return all IDs found in the user content XML :param *client_identifier*: the user content ID :return: a list of all IDs there were identified in the variable returned by the API

static `grab_experiment_from_api` (*client_identifier, item_identifier*)

Retrieve the experimental metadata and return it as a python object :param *client_identifier*: the client identifier (apiKey) :param *item_identifier*: the item identifier that should be retrieved :return: the python object obtained from the XML

static `grab_user_content` (*client_identifier*)

Grab all content for a given user ID as an XML and outputs it as a JSON. This method will grab all public experiments plus those only accessible to the user ID.

Args: *client_identifier*: the user ID

Returns response the dictionary containing the XML

make_validation (*number_of_items*)

Method to run the mapping for the given number of items

Args number_of_items (int): the number of items to process

Returns errors (dict): a dictionary containing the list of errors for all processed items

static validate_instance_from_file (*instance, item_id, schema_name*)

Method to output the extracted JSON into a file and validate it against the given schema :param instance: the instance to output into a file :param item_id: the instance ID needed to create the file name :param schema_name: the schema to check against :return errors: a list of fields that have an error for this instance

class miflowcyt_validate.**FlowRepoClient** (*mapping, base_schema, client_id*)

A class that provides functionality to download experiments from the FlowRepository (<https://flowrepository.org/>), transform the XML into JSON and validate the instances against their schema. The transformation from XML to JSON relies on the JSONBender library (<https://github.com/Onyo/jsonbender>).

static get_mapping (*mapping_file_name*)

Build the mapping dictionary based on the given mapping file :param mapping_file_name: the name of the mapping file :return mapping: the mapping of the fields

get_user_content_id (*client_identifier*)

Return all IDs found in the user content XML :param client_identifier: the user content ID :return: a list of all IDs there were identified in the variable returned by the API

static grab_experiment_from_api (*client_identifier, item_identifier*)

Retrieve the experimental metadata and return it as a python object :param client_identifier: the client identifier (apiKey) :param item_identifier: the item identifier that should be retrieved :return: the python object obtained from the XML

static grab_user_content (*client_identifier*)

Grab all content for a given user ID as an XML and outputs it as a JSON. This method will grab all public experiments plus those only accessible to the user ID.

Args: client_identifier: the user ID

Returns response the dictionary containing the XML

make_validation (*number_of_items*)

Method to run the mapping for the given number of items

Args number_of_items (int): the number of items to process

Returns errors (dict): a dictionary containing the list of errors for all processed items

static validate_instance_from_file (*instance, item_id, schema_name*)

Method to output the extracted JSON into a file and validate it against the given schema :param instance: the instance to output into a file :param item_id: the instance ID needed to create the file name :param schema_name: the schema to check against :return errors: a list of fields that have an error for this instance

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

j

`jsonschema_validator`, 7

m

`miflowcyt_validate`, 7

s

`schema2context`, 5

F

FlowRepoClient (class in miflowcyt_validate), 7, 8

G

get_mapping() (miflowcyt_validate.FlowRepoClient static method), 7, 8

get_user_content_id() (miflowcyt_validate.FlowRepoClient method), 7, 8

grab_experiment_from_api() (miflowcyt_validate.FlowRepoClient static method), 7, 8

grab_user_content() (miflowcyt_validate.FlowRepoClient static method), 7, 8

J

jsonschema_validator (module), 7

M

make_validation() (miflowcyt_validate.FlowRepoClient method), 7, 8

miflowcyt_validate (module), 7

S

schema2context (module), 5

V

validate_instance() (in module jsonschema_validator), 7

validate_instance_from_file() (miflowcyt_validate.FlowRepoClient static method), 8

validate_schema() (in module jsonschema_validator), 7

validate_schema_file() (in module jsonschema_validator), 7