hydra-python-core

Release 0.0.1

Hydra

Aug 27, 2021

CONTENTS:

1 doc_writer **3**

2 doc_maker **5**

3 Usage **7**

3.1 Create a new API Documentation and a new HydraDoc object 7

3.2 Use an existing API Documentation to create a new HydraDoc object 9

3.2.1 doc_writer 10

3.2.2 doc_maker 13

4 Indices and tables **15**

Python Module Index **17**

Index **19**

The core library is used heavily in both hydrus and agent. The core library has mainly two modules. The `doc_writer` and `doc_maker` module.

CHAPTER

ONE

DOC_WRITER

The doc_writer module contains the implementation of Hydra specification classes, which are the core building pieces of any Hydra powered REST APIs. The classes can be found at http://www.hydra-cg.com/spec/latest/core/#classes. These classes are used to build the APIDOC from scratch. To see how we can generate APIDOC from scratch, please refer this. The mapping of hydra classes and implementation of doc_writer is given below:

Table 1: Mapping

Hydra Class	class name in doc_writer
hydra:SupportedProperty	HydraClassProp, HydraCollectionProp
hydra:Status	HydraStatus
hydra:Operation	HydraClassOp, HydraCollectionOp
hydra:IriTemplateMapping	IriTemplateMapping
hydra:Link	HydraLink
hydra:Error	HydraError
hydra:Collection	HydraCollection
hydra:ApiDocumentation	HydraDoc
hydra:Class	HydraClass

HydraEntryPoint class is the template for the `entrypoint`. Entrypoint is the location from where the agent discovers all the possible locations it can go to. The `hydrus` uses the `get` method of this class to generate the entrypoint of the API.

CHAPTER

TWO

DOC_MAKER

The doc_maker module is used to parse existing JSON-LD Hydra based APIDOC. It throws error, if anything is not spec compliant. It uses the JSON-LD expansion algorithm before parsing the apidoc. The benefit this expansion provides over processing raw apidoc is the flexibility of parsing the IRIs. Some authors prefer expanded IRIs, some use compacted IRIs and some use mixture of both. So, to correctly able to parse all the IRIs we expand the IRIs and thus start parsing the doc after coming to the same page. This allows us to parse even more variety of apidocs. Then the small parsing engine traverses through entire doc and extracts classes, collections, links, property and status by calling appropriate methods of this doc_maker module.

Also another use case of doc_maker is to update the IRIs once the doc is generated by doc_writer. For example, if the server url changes, or api name changes, the changes can be reflected in the entire APIDOC by passing the doc to the `create_doc` method of the doc_maker with the updated parameters.

The doc_maker has following methods:

1. create_doc - To parse the doc or to update the doc according to new params.
2. create_collection - To parse the collection in HydraCollection of doc_writer
3. create_class - To parse classes in HydraClass of doc_writer
4. create_operation - To parse supported_operations in HydraClassOp
5. create_status - To parse the status in HydraStatus
6. create_property - To parse the supported properties in HydraClassProp
7. create_link - To parse properties referencing other class or collection in HydraLink
8. check_namespace - To check if the url provided is in the correct namespace.

CHAPTER

THREE

USAGE

3.1 Create a new API Documentation and a new HydraDoc object

The doc_writer can be used to create an API Doc itself as defined below: The first step is to create a new HydraDoc object

```
from hydra_python_core.doc_writer import HydraDoc
# Creating the HydraDoc object, this is the primary class for the Doc
API_NAME = "api"                # Name of the API, will serve as EntryPoint
BASE_URL = "http://hydrus.com/"    # The base url at which the API is hosted
# NOTE: The API will be accessible at BASE_URL + ENTRY_POINT
# (http://hydrus.com/api)
# Create ApiDoc Object
api_doc = HydraDoc(API_NAME,
                   "Title for the API Documentation",
                   "Description for the API Documentation",
                   API_NAME,
                   BASE_URL,
                   "vocab") # vocab is the name of vocabulary
```

The API Documentation has been created, but it is not yet complete. Classes, properties, and operations must be added to the Doc.

```
from hydra_python_core.doc_writer import HydraClass
# Creating classes for the API
class_title = "dummyClass"                      # Title of the Class
class_description = "A dummyClass for demo"     # Description of the class
class_ = HydraClass(class_title, class_description, endpoint=False)
```

Classes need to have properties that allow them to store information related to the class. Similar to attributes in a Python class, these are stored as supportedProperty of the HydraClass. Properties are defined as HydraClassProp objects:

```
from hydra_python_core.doc_writer import HydraClassProp
# Create new properties for the class
# The URI of the class of the property
prop1_uri = "http://props.hydrus.com/prop1"
prop1_title = "Prop1"                   # Title of the property
dummyProp1 = HydraClassProp(prop1_uri, prop1_title,
                            required=False, read=False, write=True)
```

Besides these properties, classes also need to have operations that can modify the data stored within their instances. These operation are defined as HydraClassOp and are stored in supportedOperation of the HydraClass

```
from hydra_python_core.doc_writer import HydraClassOp, HydraStatus

# Create operations for the class
op_name = "UpdateClass"  # The name of the operation
op_method = "POST"  # The method of the Operation [GET, POST, PUT, DELETE]
# URI of the object that is expected for the operation
op_expects = class_.id_
op_returns = None   # URI of the object that is returned by the operation
op_returns_header = ["Content-Type", "Content-Length"]
op_expects_header = []
# List of statusCode for the operation
op_status = [HydraStatus(code=200, desc="dummyClass updated.")]

op1 = HydraClassOp(op_name,
                   op_method,
                   op_expects,
                   op_returns,
                   op_expects_header,
                   op_returns_header,
                   op_status)
```

Once the classes and properties have been defined, add them to the class.

```
class_.add_supported_prop(dummyProp1)
class_.add_supported_op(op1)
```

After defining a class along with its properties and operations, add this class to the APIDocumentation.

```
api_doc.add_supported_class(class_)
```

We can also define a collection in the same we defined class. A collection is the set of somehow related resource

```
from hydra_python_core.doc_writer import HydraCollection
collection2_title = "dummyClass collection"
collection2_name = "dummyclasses"
collection2_description = "This collection comprises of instances of dummyClass"
# A manages block is a way to declare additional, implicit statements about members␣
↪of a collection.
# Here we are defining that all the members of this collection is of type class_.
collection2_managed_by = {
    "property": "rdf:type",
    "object": class_.id_,
}
collection_2 = HydraCollection(collection_name=collection2_name,
                               collection_description=collection2_description,␣
↪manages=collection2_managed_by, get=True,
                               post=True, collection_path="DcTest")
```

Now we add this collection to the HydraDoc object.

```
# add the collection to the HydraDoc.
api_doc.add_supported_collection(collection_2)
```

Other than this, an API Documentation also needs to have the Resource and the Collection classes, so that the server can identify the class members. This can be done automatically using the add_baseResource and add_baseCollection methods.

```
# Other operations
api_doc.add_baseResource()  # Creates the base Resource Class and adds it to the API␣
↪Documentation
api_doc.add_baseCollection()    # Creates the base Collection Class and adds it to␣
↪the API Documentation
```

Finally, create the EntryPoint object for the API Documentation. All Collections are automatically assigned endpoints in the EntryPoint object. Classes that had their endpoint variables set to True are also assigned endpoints in the EntryPoint object. This object is created automatically by the HydraDoc object and can be created using the gen_EntryPoint method.

```
api_doc.gen_EntryPoint()    # Generates the EntryPoint object for the Doc using the␣
↪Classes and Collections
```

The final API Documentation can be viewed by calling the generate method which returns a Python dictionary containing the entire API Documentation. The generate method can be called for every class defined in the doc_writer module to generate its own Python dictionary.

```
doc = api_doc.generate()
```

The complete script for this API Documentation can be found in `samples/doc_writer_sample.py`, and the generated ApiDocumentation can be found in `samples/doc_writer_sample_output.py`.

3.2 Use an existing API Documentation to create a new HydraDoc object

If there is an existing apidoc in JSON-LD, to work with the tools of Hydra Ecosystem it needs to be converted in HydraDoc object. Any existing doc can be converted in HydraDoc by simply calling the `create_doc` method of `doc_maker` module.

Example:

```
# Sample to convert the API Doc into doc_writer classes

from hydra_python_core.doc_maker import create_doc

# Note: It would be better to use json.loads from the python json library to create
↪'doc'
    doc = {
      "@context": "http://www.w3.org/ns/hydra/context.jsonld",
      "@id": "http://api.example.com/doc/",
      "@type": "ApiDocumentation",
      "title": "The name of the API",
      "description": "A short description of the API",
      "entrypoint": "URL of the API's main entry point",
      "supportedClass": [
        # ... Classes known to be supported by the Web API ...
      ],
      "possibleStatus": [
        # ... Statuses that should be expected and handled properly ...
      ]
    }

    APIDoc = create_doc(doc)
```

3.2.1 doc_writer

API Doc templates generator.

class hydra_python_core.doc_writer.**Context**(*address: str, class_: Optional[hydra_python_core.doc_writer.HydraClass] = None, collection: Optional[hydra_python_core.doc_writer.HydraCollection] = None, entrypoint: Optional[hydra_python_core.doc_writer.HydraEntryPoint] = None*)

Class for JSON-LD context.

add(*key: str, value: Union[Dict[str, str], str]*) → None
Add entry to context.

createContext(*object_: Dict[str, Any]*) → None
Create the context for the given object.

generate() → Dict[str, Any]
Get as a python dict.

class hydra_python_core.doc_writer.**EntryPointClass**(*class_: hydra_python_core.doc_writer.HydraClass*)

Class for a Operation Entry to the EntryPoint object.

generate() → Dict[str, Any]
Get as Python Dict.

class hydra_python_core.doc_writer.**EntryPointCollection**(*collection: hydra_python_core.doc_writer.HydraCollection*)

Class for a Collection Entry to the EntryPoint object.

generate() → Dict[str, Any]
Get as a python dict.

class hydra_python_core.doc_writer.**EntryPointOp**(*id_: str, method: str, desc: str, expects: Optional[str], returns: Optional[str], expects_header: List[str] = [], returns_header: List[str] = [], possible_status: List[Union[HydraStatus, HydraError]] = [], type_: Optional[str] = None, label: str = ''*)

supportedOperation for EntryPoint.

generate() → Dict[str, Any]
Get as Python Dict.

get_type(*method: str*) → str
Return @type for op based on method type.

class hydra_python_core.doc_writer.**HydraClass**(*title: str, desc: str, path: str = None, endpoint: bool = True, sub_classof: None = None, _id: str = None*)

Template for a new class.

add_supported_op(*op: Union[EntryPointOp, HydraClassOp]*) → None
Add a new supportedOperation.

Raises: TypeError: If *op* is not an instance of *HydraClassOp* or *EntryPointOp*

add_supported_prop(*prop: Union[HydraClassProp, EntryPointClass, EntryPointCollection]*) → None

Add a new supportedProperty.

Raises:

TypeError: If *prop* is not an instance of *HydraClassProp* or *EntryPointClass* or *EntryPointCollection*

generate() → Dict[str, Any]

Get the Hydra class as a python dict.

class hydra_python_core.doc_writer.**HydraClassOp**(*title: str, method: str, expects: Optional[str], returns: Optional[str], expects_header: List[str] = [], returns_header: List[str] = [], possible_status: List[Union[HydraStatus, HydraError]] = []*)

Template for a new supportedOperation.

generate() → Dict[str, Any]

Get the Hydra op as a python dict.

get_type(*method: str*) → str

Return @type for op based on method type.

class hydra_python_core.doc_writer.**HydraClassProp**(*prop: Union[str, HydraLink], title: str, read: bool, write: bool, required: bool, desc: str = '', **kwargs*)

Template for a new property.

generate() → Dict[str, Any]

Get the Hydra prop as a python dict.

class hydra_python_core.doc_writer.**HydraCollection**(*collection_name: str = None, collection_path: str = None, collection_description: str = None, manages: Union[Dict[str, Any], List] = None, get: bool = True, post: bool = True, put: bool = True, delete: bool = True*)

Class for Hydra Collection.

generate() → Dict[str, Any]

Get as a python dict.

class hydra_python_core.doc_writer.**HydraCollectionOp**(*id_: str, type_: str, method: str, desc: str, expects: Optional[str], returns: Optional[str], expects_header: List[str] = [], returns_header: List[str] = [], possible_status: List[Union[HydraStatus, HydraError]] = []*)

Operation class for Collection operations.

generate() → Dict[str, Any]

Get as a Python dict.

class hydra_python_core.doc_writer.**HydraDoc**(*API: str, title: str, desc: str, entrypoint: str, base_url: str, doc_name: str*)

Class for an API Doc.

add_baseCollection() → None
Add Collection class to the API Doc.

add_baseResource() → None
Add Resource class to the API Doc.

add_possible_status(*status: Union[HydraStatus, HydraError]*) → None
Add a new possibleStatus.

Raises: TypeError: If *status* is not an instance of *HydraStatus*.

add_supported_class(*class_:* hydra_python_core.doc_writer.HydraClass) → None
Add a new supportedClass.

Raises: TypeError: If *class_* is not an instance of *HydraClass*

add_supported_collection(*collection_:* hydra_python_core.doc_writer.HydraCollection) → None
Add a supported Collection

Raises: TypeError: If *collection_* is not an instance of *HydraCollection*

add_to_context(*key: str, value: Union[Dict[str, str], str]*) → None
Add entries to the vocabs context.

gen_EntryPoint() → None
Generate the EntryPoint for the Hydra Doc.

generate() → Dict[str, Any]
Get the Hydra API Doc as a python dict.

class hydra_python_core.doc_writer.**HydraEntryPoint**(*base_url: str, entrypoint: str*)

Template for a new entrypoint.

add_Class(*class_:* hydra_python_core.doc_writer.HydraClass) → None
Add supportedProperty to the EntryPoint.

Raises: TypeError: If *class_* is not an instance of *HydraClass*.

add_Collection(*collection:* hydra_python_core.doc_writer.HydraCollection) → None
Add supportedProperty to the EntryPoint.

Raises: TypeError: If *collection* is not an instance of *HydraCollection*.

generate() → Dict[str, Any]
Get as a Python dict.

get() → Dict[str, str]
Create the EntryPoint object to be returnd for the get function.

class hydra_python_core.doc_writer.**HydraError**(*code: int, id_: str = None, title: str = '', desc: str = ''*)

Class for Hydra Error to represent error details.

generate() → Dict[str, Any]
Get error response body

class hydra_python_core.doc_writer.**HydraIriTemplate**(*template: str, iri_mapping: List[*hydra_python_core.doc_writer.IriTemplateMapping*] = [], basic_representation: bool = True*)

Class for hydra IriTemplates

generate() → Dict[str, Any]

Get IriTemplate as a python dict

class hydra_python_core.doc_writer.**HydraLink**(*id_: str, title: str = '', desc: str = '', domain: str = '', range_: str = ''*)

Template for a link property.

add_supported_op(*op: Union[EntryPointOp, HydraClassOp]*) → None

Add a new supportedOperation.

Raises: TypeError: If *op* is not an instance of *HydraClassOp* or *EntryPointOp*

generate() → Dict[str, Any]

Get the Hydra link as a python dict.

class hydra_python_core.doc_writer.**HydraStatus**(*code: int, id_: str = None, title: str = '', desc: str = ''*)

Class for possibleStatus in Hydra Doc.

generate(*status_type: str = 'Status'*) → Dict[str, Any]

Get as Python dict.

class hydra_python_core.doc_writer.**IriTemplateMapping**(*variable: str, prop: str, required: bool = False*)

Class for hydra IriTemplateMapping

generate() → Dict[str, Any]

Get IriTemplateMapping as a python dict

3.2.2 doc_maker

Contsructor to take a Python dict containing an API Documentation and create a HydraDoc object for it

hydra_python_core.doc_maker.**check_namespace**(*id_: str = None*) → str

A helper method to check if the classes and properties are in the same namespace and if not bring them into the right namespace :param **id_** The id to check :return: correct url

hydra_python_core.doc_maker.**create_class**(*expanded_class: Dict[str, Any], endpoint: bool*) → *hydra_python_core.doc_writer.HydraClass*

Creates HydraClass from the expanded API document;

Parameters

- **apidoc** – object of HydraDoc type
- **expanded_class** – the expanded class
- **endpoint** – boolean True if class is an endpoint, False if class is not endpoint

Returns HydraClass object that can be added to api doc

hydra_python_core.doc_maker.**create_collection**(*endpoint_collection: Dict[str, Any]*) → *hydra_python_core.doc_writer.HydraCollection*

Creates the instance of HydraCollection from expanded APIDOC

Parameters **endpoint_collection** – creates HydraCollection from expanded API doc

Returns instance of HydraCollection

`hydra_python_core.doc_maker.create_doc`(*doc: Dict[str, Any], HYDRUS_SERVER_URL: str = None, API_NAME: str = None*) → *hydra_python_core.doc_writer.HydraDoc*

Create the HydraDoc object from the API Documentation.

Parameters

- **doc** – dictionary of hydra api doc
- **HYDRUS_SERVER_URL** – url of the hydrus server
- **API_NAME** – name of the api

Returns instance of HydraDoc which server and agent can understand

Raises **SyntaxError** – If the *doc* doesn't have an entry for *@id* , *@context*, *@type* key.

`hydra_python_core.doc_maker.create_link`(*supported_property: Dict[str, Any]*) → *hydra_python_core.doc_writer.HydraLink*

Creates the instances of HydraLink

Parameters **supported_property** – expanded Property

Returns instance of HydraLink

`hydra_python_core.doc_maker.create_operation`(*supported_operation: Dict[str, Any]*) → *hydra_python_core.doc_writer.HydraClassOp*

Creates the instance of HydraClassOp

Parameters **supported_operation** – The expanded supported operation from the API DOC

Returns HydraClassOp

`hydra_python_core.doc_maker.create_property`(*supported_property: Dict[str, Any]*) → Union[*hydra_python_core.doc_writer.HydraLink*, *hydra_python_core.doc_writer.HydraClassProp*]

Creates the HydraClassProp from the expanded supported property

Parameters **supported_property** – supported property dict from the expanded api doc

Returns HydraClassProp

`hydra_python_core.doc_maker.create_status`(*possible_status: List[Any]*) → List[*hydra_python_core.doc_writer.HydraStatus*]

Creates instance of HydraStatus from expanded API doc

Parameters **possible_status** – possible status from the expanded API doc

Returns List of instances of HydraStatus

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

h

hydra_python_core.doc_maker, 13
hydra_python_core.doc_writer, 10

INDEX

A

add() (*hydra_python_core.doc_writer.Context method*), 10
add_baseCollection() (*hydra_python_core.doc_writer.HydraDoc method*), 12
add_baseResource() (*hydra_python_core.doc_writer.HydraDoc method*), 12
add_Class() (*hydra_python_core.doc_writer.HydraEntryPoint method*), 12
add_Collection() (*hydra_python_core.doc_writer.HydraEntryPoint method*), 12
add_possible_status() (*hydra_python_core.doc_writer.HydraDoc method*), 12
add_supported_class() (*hydra_python_core.doc_writer.HydraDoc method*), 12
add_supported_collection() (*hydra_python_core.doc_writer.HydraDoc method*), 12
add_supported_op() (*hydra_python_core.doc_writer.HydraClass method*), 10
add_supported_op() (*hydra_python_core.doc_writer.HydraLink method*), 13
add_supported_prop() (*hydra_python_core.doc_writer.HydraClass method*), 10
add_to_context() (*hydra_python_core.doc_writer.HydraDoc method*), 12

C

check_namespace() (*in module hydra_python_core.doc_maker*), 13
Context (*class in hydra_python_core.doc_writer*), 10
create_class() (*in module hydra_python_core.doc_maker*), 13
create_collection() (*in module hydra_python_core.doc_maker*), 13
create_doc() (*in module hydra_python_core.doc_maker*), 14
create_link() (*in module hydra_python_core.doc_maker*), 14
create_operation() (*in module hydra_python_core.doc_maker*), 14
create_property() (*in module hydra_python_core.doc_maker*), 14
create_status() (*in module hydra_python_core.doc_maker*), 14
createContext() (*hydra_python_core.doc_writer.Context method*), 10

E

EntryPointClass (*class in hydra_python_core.doc_writer*), 10
EntryPointCollection (*class in hydra_python_core.doc_writer*), 10
EntryPointOp (*class in hydra_python_core.doc_writer*), 10

G

gen_EntryPoint() (*hydra_python_core.doc_writer.HydraDoc method*), 12
generate() (*hydra_python_core.doc_writer.Context method*), 10
generate() (*hydra_python_core.doc_writer.EntryPointClass method*), 10
generate() (*hydra_python_core.doc_writer.EntryPointCollection method*), 10
generate() (*hydra_python_core.doc_writer.EntryPointOp method*), 10
generate() (*hydra_python_core.doc_writer.HydraClass method*), 11
generate() (*hydra_python_core.doc_writer.HydraClassOp method*), 11
generate() (*hydra_python_core.doc_writer.HydraClassProp method*), 11

generate() (*hydra_python_core.doc_writer.HydraCollection method*), 11
generate() (*hydra_python_core.doc_writer.HydraCollectionOp method*), 11
generate() (*hydra_python_core.doc_writer.HydraDoc method*), 12
generate() (*hydra_python_core.doc_writer.HydraEntryPoint method*), 12
generate() (*hydra_python_core.doc_writer.HydraError method*), 12
generate() (*hydra_python_core.doc_writer.HydraIriTemplate method*), 13
generate() (*hydra_python_core.doc_writer.HydraLink method*), 13
generate() (*hydra_python_core.doc_writer.HydraStatus method*), 13
generate() (*hydra_python_core.doc_writer.IriTemplateMapping method*), 13
get() (*hydra_python_core.doc_writer.HydraEntryPoint method*), 12
get_type() (*hydra_python_core.doc_writer.EntryPointOp method*), 10
get_type() (*hydra_python_core.doc_writer.HydraClassOp method*), 11

H

hydra_python_core.doc_maker
module, 13
hydra_python_core.doc_writer
module, 10
HydraClass (*class in hydra_python_core.doc_writer*), 10
HydraClassOp (*class in hydra_python_core.doc_writer*), 11
HydraClassProp (*class in hydra_python_core.doc_writer*), 11
HydraCollection (*class in hydra_python_core.doc_writer*), 11
HydraCollectionOp (*class in hydra_python_core.doc_writer*), 11
HydraDoc (*class in hydra_python_core.doc_writer*), 11
HydraEntryPoint (*class in hydra_python_core.doc_writer*), 12
HydraError (*class in hydra_python_core.doc_writer*), 12
HydraIriTemplate (*class in hydra_python_core.doc_writer*), 12
HydraLink (*class in hydra_python_core.doc_writer*), 13
HydraStatus (*class in hydra_python_core.doc_writer*), 13

I

IriTemplateMapping (*class in hydra_python_core.doc_writer*), 13

M

module
hydra_python_core.doc_maker, 13
hydra_python_core.doc_writer, 10