
consumer Documentation

Release 0.0.3

Zeeshan Khan

Jan 17, 2019

Contents:

1	AbstractBaseWebService	1
2	AbstractWebAuth	3
3	AbstractBaseJsonWebService	5
4	AbstractBaseXmlWebService	7
5	Indices and tables	9
	Python Module Index	11

CHAPTER 1

AbstractBaseWebService

```
class consumer.AbstractBaseWebService (env=None)
```

Parameters **env** – env name to use.

get (***params*)
return a string or None representing response from calling the service URI

headers ()
optional headers that should be passed with the HTTP request

hosts
Concrete Service should define a “hosts” property that should be a ServiceHostCollection with UAT and PROD values

isAuthenticationRequired ()
Concrete Service should define a “hosts” property that should be a ServiceHostCollection with UAT and PROD values

mandatory_param_rules
Service can optionally define a property “mandatory_params” which is a tuple of parameters combination, representing set of required query parameter for the service.

Parameters passed to get method will be checked to have all of these params else the web service call will not continue default [] empty list no parameters are mandatory

Returns: a tuple of pairs of parameters

```
>>>“((‘param1’,), (‘param2’,),)“ param1 or param2 should be present
```

```
((‘param1’, ‘param2’), (‘param3’,),) either param1 and param2 paramters should  
be present or just Name parameter should be present in the query parameters
```

protocol
http or https

service_path
Concrete Service must have a property “service_path” that should be a string representing a abso-

lute path(except the protocol and hostname/port) Example: for a service <http://supercomputer:8080/servicepath/some-end-point> service_path will be 'servicepath/some-end-point'

url

Construct the complete service URI using proper protocol host, port and service_path

valid_params

Service must have a property "valid_params" which is a list of valid query parameter for the service, any other parameter passed other than what is defined here will be dropped with a warn log

verify()

optional verify parameter that should be passed with the HTTP request for verification

AbstractWebAuth

```
class consumer.AbstractWebAuth
```

```
    auth_headers ()
```

```
        return a dict containing headers that should ne included in a Authenticated HTTP request
```

```
    certificates ()
```

```
        return a string representing the certificate content
```

AbstractBaseJsonWebService

class consumer.**AbstractBaseJsonWebService** (*env=None*)

Use this abstract class when the service is providing JSON response and you want the result as a python dictionary

as_dict ()

return a dictionary or None representing response from calling the service URI

get (***params*)

delegate the call to super *get* method, also try parsing the response as json and store it in self.json

AbstractBaseXmlWebService

class `consumer.AbstractBaseXmlWebService` (*env=None*)

Use this abstract class when the service is providing XML response and you want the result as a python dictionary

as_dict (***params*)

return a dictionary or None representing response from calling the service URI

get (***params*)

delegate the call to super *get* method, also try parsing the response as XML and store it in `self.xml` as a `defusedxml.ElementTree`

CHAPTER 5

Indices and tables

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

C

consumer, ??

A

AbstractBaseJsonWebService (class in consumer), 5
AbstractBaseWebService (class in consumer), 1
AbstractBaseXmlWebService (class in consumer), 7
AbstractWebAuth (class in consumer), 3
as_dict() (consumer.AbstractBaseJsonWebService method), 5
as_dict() (consumer.AbstractBaseXmlWebService method), 7
auth_headers() (consumer.AbstractWebAuth method), 3

C

certificates() (consumer.AbstractWebAuth method), 3
consumer (module), 1

G

get() (consumer.AbstractBaseJsonWebService method), 5
get() (consumer.AbstractBaseWebService method), 1
get() (consumer.AbstractBaseXmlWebService method), 7

H

headers() (consumer.AbstractBaseWebService method), 1
hosts (consumer.AbstractBaseWebService attribute), 1

I

isAuthunticationRequired() (consumer.AbstractBaseWebService method), 1

M

mandatory_param_rules (consumer.AbstractBaseWebService attribute), 1

P

protocol (consumer.AbstractBaseWebService attribute), 1

S

service_path (consumer.AbstractBaseWebService attribute), 1

U

url (consumer.AbstractBaseWebService attribute), 2

V

valid_params (consumer.AbstractBaseWebService attribute), 2
verify() (consumer.AbstractBaseWebService method), 2