

## WOLFF API Plugins

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	api_hook.APIHook Class Reference . . . . .	5
3.2	api_map.APIMap Class Reference . . . . .	5
3.2.1	Detailed Description . . . . .	6
3.2.2	Constructor & Destructor Documentation . . . . .	6
3.2.2.1	__init__() . . . . .	6
3.2.3	Member Function Documentation . . . . .	6
3.2.3.1	add_special_params() . . . . .	6
3.2.3.2	get_complete_url() . . . . .	7
3.2.3.3	get_replacement() . . . . .	7
3.2.3.4	perform_uri_replacement() . . . . .	7
3.2.3.5	uri_is_substitutable() . . . . .	8
3.3	api_hook.APIMethod Class Reference . . . . .	8
3.3.1	Detailed Description . . . . .	9
3.3.2	Member Function Documentation . . . . .	9
3.3.2.1	get_http_method() . . . . .	9
3.3.2.2	get_uri() . . . . .	9
3.4	api_method.APIMethod Class Reference . . . . .	10

3.5	<a href="#">encoder.Applications Class Reference</a>	10
3.6	<a href="#">decoder.Applications Class Reference</a>	11
3.7	<a href="#">response_handler.CheckListingStockResponseHandler Class Reference</a>	12
3.8	<a href="#">client.Client Class Reference</a>	12
3.8.1	Detailed Description	12
3.8.2	Constructor & Destructor Documentation	12
3.8.2.1	<code>__init__()</code>	13
3.8.3	Member Function Documentation	13
3.8.3.1	<code>get_connection()</code>	13
3.8.3.2	<code>has_connection()</code>	13
3.8.3.3	<code>instantiate_resources()</code>	14
3.8.3.4	<code>set_connection()</code>	14
3.9	<a href="#">client_manager.ClientManager Class Reference</a>	14
3.9.1	Detailed Description	15
3.9.2	Constructor & Destructor Documentation	15
3.9.2.1	<code>__init__()</code>	15
3.9.3	Member Function Documentation	15
3.9.3.1	<code>check_existing_client()</code>	15
3.9.3.2	<code>get_client_by_service_identifier()</code>	16
3.9.3.3	<code>register_client()</code>	16
3.9.3.4	<code>register_existing_client()</code>	16
3.10	<a href="#">response_handler.CreateListingResponseHandler Class Reference</a>	17
3.11	<a href="#">decoder.DecoderFactory Class Reference</a>	17
3.11.1	Detailed Description	17
3.11.2	Member Function Documentation	17
3.11.2.1	<code>get_service()</code>	17
3.12	<a href="#">endpoint_handler.EndpointManager Class Reference</a>	18
3.13	<a href="#">decoder.EtsyDecoder Class Reference</a>	18
3.13.1	Detailed Description	19
3.13.2	Constructor & Destructor Documentation	19

3.13.2.1	<code>__init__()</code>	19
3.13.3	Member Function Documentation	19
3.13.3.1	<code>decode_update_listing_message()</code>	19
3.13.3.2	<code>get_service_decoder()</code>	19
3.14	<code>encoder.EtsyEncoder</code> Class Reference	20
3.14.1	Detailed Description	20
3.14.2	Constructor & Destructor Documentation	20
3.14.2.1	<code>__init__()</code>	20
3.14.3	Member Function Documentation	20
3.14.3.1	<code>encode()</code>	21
3.15	<code>endpoint_handler.EtsyEndpointManager</code> Class Reference	21
3.16	<code>message.EtsyMessage</code> Class Reference	22
3.17	<code>resource.FileResource</code> Class Reference	23
3.17.1	Detailed Description	24
3.17.2	Constructor & Destructor Documentation	24
3.17.2.1	<code>__init__()</code>	24
3.17.3	Member Function Documentation	24
3.17.3.1	<code>contains()</code>	25
3.17.3.2	<code>get_data()</code>	25
3.17.3.3	<code>get_file_name()</code>	25
3.17.3.4	<code>get_full_file_path()</code>	25
3.17.3.5	<code>get_full_path()</code>	26
3.17.3.6	<code>read()</code>	26
3.18	<code>message.Message</code> Class Reference	26
3.18.1	Constructor & Destructor Documentation	27
3.18.1.1	<code>__init__()</code>	27
3.18.2	Member Function Documentation	27
3.18.2.1	<code>get_data()</code>	27
3.19	<code>server.MQTTServer</code> Class Reference	27
3.19.1	Detailed Description	28

3.19.2	Member Function Documentation	28
3.19.2.1	start()	29
3.20	server_connection.MQTTServerConnection Class Reference	29
3.20.1	Detailed Description	29
3.20.2	Member Function Documentation	29
3.20.2.1	check_for_messages()	30
3.20.2.2	get_client()	30
3.20.2.3	send()	30
3.21	oauth_credentials.OAuth1Credentials Class Reference	30
3.21.1	Member Function Documentation	31
3.21.1.1	as_dict()	31
3.21.1.2	get_from_file()	31
3.21.1.3	set_client_info()	31
3.21.1.4	set_resource_owner_info()	32
3.22	resource.OAuth1Resource Class Reference	32
3.22.1	Constructor & Destructor Documentation	33
3.22.1.1	__init__()	33
3.22.2	Member Function Documentation	33
3.22.2.1	read()	33
3.22.2.2	write()	34
3.23	resource.ResourceFactory Class Reference	34
3.24	response_handler.ResponseHandler Class Reference	34
3.25	decoder.EtsyDecoder.Services Class Reference	35
3.25.1	Detailed Description	35
3.25.2	Member Function Documentation	36
3.25.2.1	has_value()	36
3.26	encoder.EtsyEncoder.Services Class Reference	36
3.26.1	Detailed Description	37
3.27	resource.ShippingTemplateIdResource Class Reference	37
3.27.1	Detailed Description	38

3.28 DBConnection.SQLite3DBConnection Class Reference . . . . .	38
3.29 server_connection.TCPServerConnection Class Reference . . . . .	38
3.29.1 Detailed Description . . . . .	39
3.29.2 Member Function Documentation . . . . .	39
3.29.2.1 get_ip() . . . . .	39
3.29.2.2 get_port() . . . . .	39
3.29.2.3 send() . . . . .	39
3.30 server.WOLFFNodeProxy Class Reference . . . . .	40
3.30.1 Detailed Description . . . . .	41
3.30.2 Member Function Documentation . . . . .	41
3.30.2.1 do_request() . . . . .	41
3.30.2.2 start() . . . . .	41
3.31 server.WOLFFServer Class Reference . . . . .	42
3.31.1 Detailed Description . . . . .	42
3.31.2 Member Function Documentation . . . . .	42
3.31.2.1 annotate_data() . . . . .	43
3.31.2.2 decode_data() . . . . .	43
3.31.2.3 do_request() . . . . .	44
3.31.2.4 get_request_handler() . . . . .	44
3.31.2.5 start() . . . . .	44
<b>Index</b>	<b>45</b>





# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

api_hook.APIHook . . . . .	5
api_map.APIMap . . . . .	5
api_hook.APIMethod . . . . .	8
api_method.APIMethod . . . . .	10
response_handler.CheckListingStockResponseHandler . . . . .	12
client.Client . . . . .	12
client_manager.ClientManager . . . . .	14
response_handler.CreateListingResponseHandler . . . . .	17
decoder.DecoderFactory . . . . .	17
endpoint_handler.EndpointManager . . . . .	18
endpoint_handler.EtsyEndpointManager . . . . .	21
decoder.EtsyDecoder . . . . .	18
encoder.EtsyEncoder . . . . .	20
resource.FileResource . . . . .	23
resource.OAuth1Resource . . . . .	32
resource.ShippingTemplateIdResource . . . . .	37
message.Message . . . . .	26
message.EtsyMessage . . . . .	22
server_connection.MQTTServerConnection . . . . .	29
oauth_credentials.OAuth1Credentials . . . . .	30
resource.ResourceFactory . . . . .	34
response_handler.ResponseHandler . . . . .	34
DBConnection.SQLite3DBConnection . . . . .	38
server_connection.TCPServerConnection . . . . .	38
server.WOLFFServer . . . . .	42
server.MQTTServer . . . . .	27
server.WOLFFNodeProxy . . . . .	40
Enum	
decoder.Applications . . . . .	11
decoder.EtsyDecoder.Services . . . . .	35
encoder.Applications . . . . .	10
encoder.EtsyEncoder.Services . . . . .	36



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">api_hook.APIHook</a>	5
<a href="#">api_map.APIMap</a>	5
<a href="#">api_hook.APIMethod</a>	8
<a href="#">api_method.APIMethod</a>	10
<a href="#">encoder.Applications</a>	10
<a href="#">decoder.Applications</a>	11
<a href="#">response_handler.CheckListingStockResponseHandler</a>	12
<a href="#">client.Client</a>	12
<a href="#">client_manager.ClientManager</a>	14
<a href="#">response_handler.CreateListingResponseHandler</a>	17
<a href="#">decoder.DecoderFactory</a>	17
<a href="#">endpoint_handler.EndpointManager</a>	18
<a href="#">decoder.EtsyDecoder</a>	18
<a href="#">encoder.EtsyEncoder</a>	20
<a href="#">endpoint_handler.EtsyEndpointManager</a>	21
<a href="#">message.EtsyMessage</a>	22
<a href="#">resource.FileResource</a>	23
<a href="#">message.Message</a>	26
<a href="#">server.MQTTServer</a>	27
<a href="#">server_connection.MQTTServerConnection</a>	29
<a href="#">oauth_credentials.OAuth1Credentials</a>	30
<a href="#">resource.OAuth1Resource</a>	32
<a href="#">resource.ResourceFactory</a>	34
<a href="#">response_handler.ResponseHandler</a>	34
<a href="#">decoder.EtsyDecoder.Services</a>	35
<a href="#">encoder.EtsyEncoder.Services</a>	36
<a href="#">resource.ShippingTemplateIdResource</a>	37
<a href="#">DBConnection.SQLite3DBConnection</a>	38
<a href="#">server_connection.TCPServerConnection</a>	38
<a href="#">server.WOLFFNodeProxy</a>	40
<a href="#">server.WOLFFServer</a>	42



## Chapter 3

# Class Documentation

### 3.1 `api_hook.APIHook` Class Reference

#### Public Member Functions

- `def __init__ (self, service="", methods=None)`
- `def get_methods (self)`
- `def add_method (self, method)`
- `def add_methods (self, methods)`
- `def get_service (self)`
- `def set_service (self, service)`

#### Public Attributes

- `service`
- `methods`

The documentation for this class was generated from the following file:

- `wolff_api_plugins/client/api_hook.py`

### 3.2 `api_map.APIMap` Class Reference

#### Public Member Functions

- `def __init__ (self)`
- `def get_base_url (self, service)`
- `def get_uri (self, service, method)`
- `def uri_is_substitutable (self, uri)`
- `def perform_uri_replacement (self, uri, uri_re, value)`
- `def get_replacement (self, service, method, data_dict, database)`
- `def get_complete_url (self, service, method, replace=None)`
- `def get_http_method (self, service, method)`
- `def get_auth_type (self, service)`
- `def get_service_identifier (self, service, method)`
- `def get_identifier_value (self, service, method, data)`
- `def add_special_params (self, service, method, data)`

## Public Attributes

- **api\_map**

### 3.2.1 Detailed Description

An APIMap contains the information necessary to retrieve the base url, methods and their arguments, and authentication types for services.

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 \_\_init\_\_()

```
def api_map.APIMap.__init__ (
    self )
```

Create the api map.  
For each service, we list its base url,  
its methods, and the authentication type.

For each method, the uri, http method, and service identifier are listed. The service\_identifier is an identifier that is unique to a user that wants to perform a method. From this service identifier, a user can be identified.

### 3.2.3 Member Function Documentation

#### 3.2.3.1 add\_special\_params()

```
def api_map.APIMap.add_special_params (
    self,
    service,
    method,
    data )
```

Add any necessary "special" parameters for the service and method to the data dictionary. A special parameter is one that is not specified by the user, but needs to be included to form a valid request. An example of this is the 'taxonomy\_id' of the create\_listing request.

@param service the string name of the service that has the special parameter  
@param method the method within the service to find the special parameter  
@param data The data dict to which the special parameter will be added

### 3.2.3.2 get\_complete\_url()

```
def api_map.APIMap.get_complete_url (
    self,
    service,
    method,
    replace = None )
```

Get the complete url for a request.

@param service The service the request is being made to

@param method the method of the request that is being made to the service.

@param replace If included, the value for this parameter will be substituted into the URL of the request

@returns the complete URL to which an API request may be made.

### 3.2.3.3 get\_replacement()

```
def api_map.APIMap.get_replacement (
    self,
    service,
    method,
    data_dict,
    database )
```

Get the replacement value for a URI value, service, method, and data dict containing the request. A Database may be queried to retrieve the value.

### 3.2.3.4 perform\_uri\_replacement()

```
def api_map.APIMap.perform_uri_replacement (
    self,
    uri,
    uri_re,
    value )
```

Perform URI replacement, given the URI the replacement should go into, the Regular expression that will allow the replacement, and the value that should go into the new URI.

### 3.2.3.5 uri\_is\_substitutable()

```
def api_map.APIMap.uri_is_substitutable (
    self,
    uri )
```

Some APIs have parameters that go in the URL of the request itself. For example, the URI for Etsy's `update_listing` has the form: `'/listings/:listing_id'` where `' :listing_id'` is a variable that is supposed to be replaced with the actual ID of the listing. This method returns true if the specified uri is one that takes a substitutable parameter.

The documentation for this class was generated from the following file:

- `wolff_api_plugins/server/api_map.py`

## 3.3 api\_hook.APIMethod Class Reference

### Public Member Functions

- `def __init__ (self, method="", args=None, http_method="", name="", uri="")`
- `def get_method (self)`
- `def set_method (self, new_method)`
- `def set_parameter_re (self, new_re)`
- `def get_http_method (self)`
- `def get_uri_replacement (self, match)`
- `def get_uri (self, substitute=True)`
- `def args_as_dict (self)`
- `def get_args (self)`
- `def set_http_method (self, method)`
- `def set_uri (self, uri)`
- `def add_arg (self, arg)`
- `def set_args (self, args)`
- `def set_name (self, name)`
- `def set_args (self, p_dict)`
- `def clear_args (self)`
- `def get_name (self)`

### Public Attributes

- `uri`
- `args`
- `http_method`
- `name`
- `method`
- `substitutable_args`
- `uri_re`



### 3.3.1 Detailed Description

Represents the data needed to call an API method. Has a uri for locating the resource necessary and a set of arguments that are included in the call of an API Method.

### 3.3.2 Member Function Documentation

#### 3.3.2.1 get\_http\_method()

```
def api_hook.APIMethod.get_http_method (
    self )
```

Get the HTTP method this APIMethod uses (GET, POST, etc.)

#### 3.3.2.2 get\_uri()

```
def api_hook.APIMethod.get_uri (
    self,
    substitute = True )
```

Get the method's URI. If there are substitutable parameters (parameters that appear in the URI in the form: `/:param_name/` will be substituted if 'substitute' is set to true. So if `user_id` is an argument to a method whose URI looks like: `/:user_id/`, then the value of `user_id` will take place of the item between the slashes.

Args:

- `substitute`: Include if URI substitution as explained above should be done. This only happens if the value in `self.args_as_dict()` is set. If false, the unchanged value will be used.

Raises:

- `ValueError` if `substitute` is true but there is no valid substitution to make in the args.

Returns:

- The string URI of this method

The documentation for this class was generated from the following file:

- `wolff_api_plugins/client/api_hook.py`

### 3.4 `api_method.APIMethod` Class Reference

#### Public Member Functions

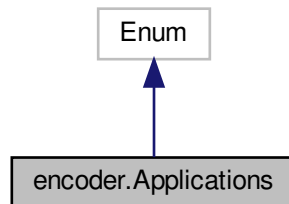
- `def __init__ (self, uri="", http_method="", name="", args=None)`
- `def get_name (self)`
- `def get_uri (self)`
- `def get_http_hethod (self)`

The documentation for this class was generated from the following file:

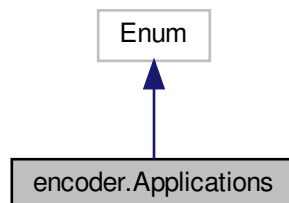
- `wolff_api_plugins/server/api_method.py`

### 3.5 `encoder.Applications` Class Reference

Inheritance diagram for `encoder.Applications`:



Collaboration diagram for `encoder.Applications`:



### Static Public Attributes

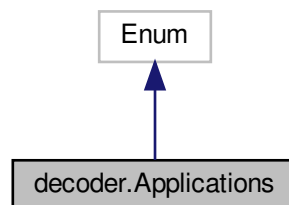
- int **ETSY** = 1

The documentation for this class was generated from the following file:

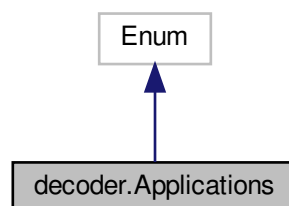
- wolff\_api\_plugins/client/encoder.py

## 3.6 decoder.Applications Class Reference

Inheritance diagram for decoder.Applications:



Collaboration diagram for decoder.Applications:



### Static Public Attributes

- int **ETSY** = 1

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/server/decoder.py

### 3.7 `response_handler.CheckListingStockResponseHandler` Class Reference

#### Public Member Functions

- `def __init__ (self, db_connection)`
- `def handle_response (self, resp_message, client_id)`

The documentation for this class was generated from the following file:

- `wolff_api_plugins/server/response_handler.py`

### 3.8 `client.Client` Class Reference

#### Public Member Functions

- `def __init__ (self, connection=None, endpoint=None, credentials=None, client_id=1, message_type=Message)`
- `def set_connection (self, server_connection)`
- `def get_id (self)`
- `def set_id (self, id)`
- `def get_connection (self)`
- `def has_connection (self)`
- `def set_endpoint (self, endpoint)`
- `def get_endpoint (self)`
- `def set_credentials (self, credential)`
- `def get_credentials (self)`
- `def specialize (self)`
- `def get_message_type (self)`
- `def __init__ (self, cli_id, resources, base_path='.')`
- `def instantiate_resources (self, resources, base)`
- `def get_id (self)`
- `def get_resource (self, service, resource)`

#### 3.8.1 Detailed Description

A client sends messages to a server, and receives them. Clients are specialized by specifying connection, endpoint, and credentials. Connection allows clients to send messages over different backbones, endpoint defines what type of client this is (Etsy, Weebly, etc.), and credentials (Typicall Oauth) identify a user using a service.

A client has a certain number of resources that are each stored in a file at a path.

#### 3.8.2 Constructor & Destructor Documentation

### 3.8.2.1 `__init__()`

```
def client.Client.__init__ (
    self,
    cli_id,
    resources,
    base_path = '.' )
```

Initialize a client with an id, certain resources, and a base path from which the client references its resources.

@param cli\_id Id of the client to create. This ID should be unique to a certain directory in which clients are located.  
@param resources The resources a client is created with.  
@param base\_path The base path for the client's resources.

## 3.8.3 Member Function Documentation

### 3.8.3.1 `get_connection()`

```
def client.Client.get_connection (
    self )
```

Get the connection stored by this client.

@returns The client's connection, or None if the client does not currently have a connection.

### 3.8.3.2 `has_connection()`

```
def client.Client.has_connection (
    self )
```

Determine whether the client has a connection.

@returns True if the client has a connection, False otherwise.

### 3.8.3.3 instantiate\_resources()

```
def client.Client.instantiate_resources (
    self,
    resources,
    base )
```

Instantiate resources for a client.

@param resources A list of resources to instantiate

@param base the base path for the resources

### 3.8.3.4 set\_connection()

```
def client.Client.set_connection (
    self,
    server_connection )
```

Set this client's server connection. The client will use this connection for sending any messages.

@param server\_connection: A server connection. This connection will be used to send any messages that the client sends.

The documentation for this class was generated from the following file:

- `wolff_api_plugins/client/client.py`

## 3.9 client\_manager.ClientManager Class Reference

### Public Member Functions

- `def __init__ (self, client_dir, db_connection)`
- `def get_client_by_id (self, client_id)`
- `def check_existing_client (self, client_id)`
- `def get_client_by_service_identifier (self, service, identifier, identifier_value)`
- `def get_clients (self)`
- `def register_existing_client (self, new_client)`
- `def register_client (self, resources)`

### Public Attributes

- `conn`

### 3.9.1 Detailed Description

A ClientManager handles clients and the resources that they use. Each managed client has a number of resources that allow the client to have persistent data.

### 3.9.2 Constructor & Destructor Documentation

#### 3.9.2.1 `__init__()`

```
def client_manager.ClientManager.__init__ (
    self,
    client_dir,
    db_connection )
```

Create a ClientManager, whose clients are in client\_dir. Within client\_dir, each client with id x is represented in client\_dir/client\_x, where x is the client's id.

Additionally stores a connection to a WOLFF database containing information regarding records that have been created

### 3.9.3 Member Function Documentation

#### 3.9.3.1 `check_existing_client()`

```
def client_manager.ClientManager.check_existing_client (
    self,
    client_id )
```

Check whether a client already exists. A client exists if 'client\_x' exists in client\_dir, where x is the id of this client.

Throws:

ValueError if the client with client\_id already exists.

### 3.9.3.2 get\_client\_by\_service\_identifier()

```
def client_manager.ClientManager.get_client_by_service_identifier (
    self,
    service,
    identifier,
    identifier_value )
```

Retrieve a client by a service identifier.  
A service identifier is used to uniquely identify a client that uses a service.

@param service the service (etsy, for example) to identify a client in.  
@param identifier the identifier to try and identify a client by.  
For example, an etsy client can be identified by a 'shipping\_template\_id'.  
@param identifier\_value the value of the identifier. A client who has this identifier (if any) will be returned.

### 3.9.3.3 register\_client()

```
def client_manager.ClientManager.register_client (
    self,
    resources )
```

Register a client with certain resources.  
Creates the necessary file structure for each resource.

### 3.9.3.4 register\_existing\_client()

```
def client_manager.ClientManager.register_existing_client (
    self,
    new_client )
```

Register a client object that has already been initialized.  
@param new\_client a Client object that will be initialized.  
@pre new\_client.get\_id() cannot be in self.client\_dir

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/server/client\_manager.py



## 3.10 response\_handler.CreateListingResponseHandler Class Reference

### Public Member Functions

- `def __init__ (self, db_connection)`
- `def handle_response (self, resp_message, client_id)`

The documentation for this class was generated from the following file:

- `wolff_api_plugins/server/response_handler.py`

## 3.11 decoder.DecoderFactory Class Reference

### Public Member Functions

- `def get_decoder (self, message)`
- `def get_service (self, message)`

### 3.11.1 Detailed Description

Returns a decoder based upon the service specified in the encoded method

@param message the message for which a decoder is reduced  
@returns A decoder that can decode the message

### 3.11.2 Member Function Documentation

#### 3.11.2.1 get\_service()

```
def decoder.DecoderFactory.get_service (  
    self,  
    message )
```

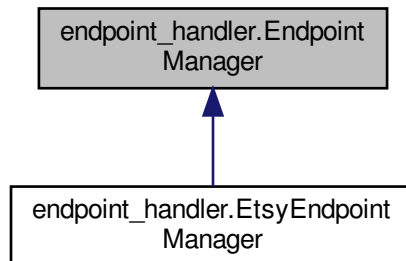
Get a service-specific decoder.

The documentation for this class was generated from the following file:

- `wolff_api_plugins/server/decoder.py`

### 3.12 endpoint\_handler.EndpointManager Class Reference

Inheritance diagram for endpoint\_handler.EndpointManager:



#### Public Member Functions

- def `__init__` (self)

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/server/endpoint\_handler.py

### 3.13 decoder.EtsyDecoder Class Reference

#### Classes

- class [Services](#)

#### Public Member Functions

- def `__init__` (self)
- def `is_etsy_message` (self, message)
- def `get_service_decoder` (self, message)
- def `decode` (self, message)
- def `decode_update_listing_message` (self, message)

#### Public Attributes

- `title_map`
- `description_map`
- `who_made_map`
- `when_made_map`

### 3.13.1 Detailed Description

A class that is capable of decoding messages for the Etsy service.

### 3.13.2 Constructor & Destructor Documentation

#### 3.13.2.1 `__init__()`

```
def decoder.EtsyDecoder.__init__ (
    self )
```

A map associating bytes with titles.

### 3.13.3 Member Function Documentation

#### 3.13.3.1 `decode_update_listing_message()`

```
def decoder.EtsyDecoder.decode_update_listing_message (
    self,
    message )
```

#### 3.13.3.2 `get_service_decoder()`

```
def decoder.EtsyDecoder.get_service_decoder (
    self,
    message )
```

Get a method that will decode messages for a certain service.

The documentation for this class was generated from the following file:

- `wolff_api_plugins/server/decoder.py`

## 3.14 encoder.EtsyEncoder Class Reference

### Classes

- class [Services](#)

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)
- def [encode](#) (self, etsy\_data)

### Public Attributes

- **title\_map**
- **description\_map**
- **who\_made\_map**
- **when\_made\_map**

### 3.14.1 Detailed Description

An EtsyDecode handles the details required for the encoding and decoding of Etsy message, including the maps necessary for mapping strings -> byte values in the encoded message

### 3.14.2 Constructor & Destructor Documentation

#### 3.14.2.1 [\\_\\_init\\_\\_\(\)](#)

```
def encoder.EtsyEncoder.__init__ (
    self )
```

The title map, specifying how titles should be mapped to byte values.  
These can be expanded until there are 0xFF values mapped.

### 3.14.3 Member Function Documentation

## 3.14.3.1 encode()

```
def encoder.EtsyEncoder.encode (
    self,
    etsy_data )
```

Encode a dictionary containing etsy data.

@param etsy\_data A dictionary containing the data necessary to perform a request to an Etsy method. The dictionary should be of the form:

```
{ 'service': 'etsy',
  'method': {'params': {}, 'name': ''
            },
  'client_id': ''
}
```

Where 'name' is the name of the method, such as 'create\_listing', and 'params' contains a dictionary of parameters and their values for the method.

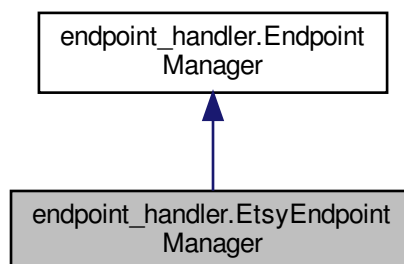
@returns A method that is capable of encoding either etsy\_data, or a dictionary of the same type

The documentation for this class was generated from the following file:

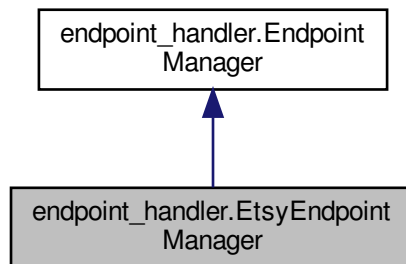
- wolff\_api\_plugins/client/encoder.py

## 3.15 endpoint\_handler.EtsyEndpointManager Class Reference

Inheritance diagram for endpoint\_handler.EtsyEndpointManager:



Collaboration diagram for `endpoint_handler.EtsyEndpointManager`:



### Public Member Functions

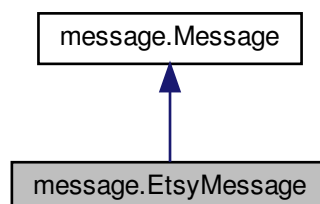
- `def __init__(self)`
- `def get_base_url(self)`
- `def get_method(self, method_name)`

The documentation for this class was generated from the following file:

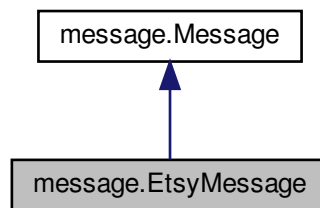
- `wolff_api_plugins/server/endpoint_handler.py`

## 3.16 message.EtsyMessage Class Reference

Inheritance diagram for `message.EtsyMessage`:



Collaboration diagram for message.EtsyMessage:



### Public Member Functions

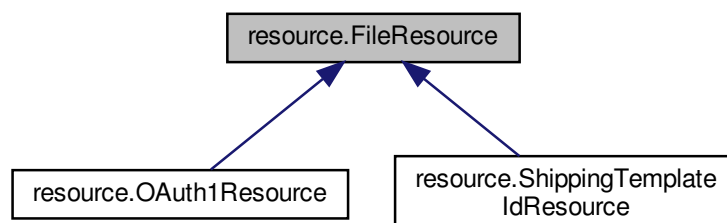
- `def __init__(self, data)`
- `def __str__(self)`
- `def encode(self)`

The documentation for this class was generated from the following file:

- `wolff_api_plugins/client/message.py`

## 3.17 resource.FileResource Class Reference

Inheritance diagram for resource.FileResource:



## Public Member Functions

- `def __init__ (self, resource_id, base_path, resource_name="resource.txt", values=None, root_path='.')`
- `def contains (self, val)`
- `def get_data (self)`
- `def read (self)`
- `def write (self, str_data)`
- `def get_file_name (self)`
- `def get_full_path (self, include_trailing=True)`
- `def get_full_file_path (self)`
- `def __eq__ (self, other)`
- `def __ne__ (self, other)`

### 3.17.1 Detailed Description

A FileResource is a resource that is located in a file.  
This class provides methods for creation, editing, etc. of FileResources.

### 3.17.2 Constructor & Destructor Documentation

#### 3.17.2.1 \_\_init\_\_()

```
def resource.FileResource.__init__ (
    self,
    resource_id,
    base_path,
    resource_name = "resource.txt",
    values = None,
    root_path = '.' )
```

Create a new FileResource, with resource\_id and a base path.

@param resource\_id The id of this resource. It should be unique.  
@param resource\_name: The name of the value of the resource. This will be the name of the text file.

@param base\_path Base of the path to write files to.  
Files will be written to resource\_id/base\_path

### 3.17.3 Member Function Documentation



### 3.17.3.1 contains()

```
def resource.FileResource.contains (
    self,
    val )
```

Determine if this resource contains a value.  
This method is generally not useful for the base class.

### 3.17.3.2 get\_data()

```
def resource.FileResource.get_data (
    self )
```

Return the data stored in this  
resource, reading it from the file if  
necessary

### 3.17.3.3 get\_file\_name()

```
def resource.FileResource.get_file_name (
    self )
```

Get the filename for the resource,  
without any of the leading path information.

Returns:  
the name of the file this resource is associated with

### 3.17.3.4 get\_full\_file\_path()

```
def resource.FileResource.get_full_file_path (
    self )
```

Get the entire path of the resource.

### 3.17.3.5 get\_full\_path()

```
def resource.FileResource.get_full_path (
    self,
    include_trailing = True )
```

Get the full path of the location of the resource.  
@note This method does not return the complete path of the resource,  
only the path of its location.

### 3.17.3.6 read()

```
def resource.FileResource.read (
    self )
```

Read the data for a resource from its file.  
The complete filepath is constructed from  
id, base\_path, and resource\_name.

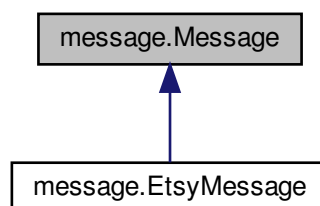
@returns the lines from the resource file in a list.

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/server/resource.py

## 3.18 message.Message Class Reference

Inheritance diagram for message.Message:



### Public Member Functions

- def `__init__` (self, data)
- def `get_data` (self)
- def `__str__` (self)
- def `encode` (self)

### 3.18.1 Constructor & Destructor Documentation

#### 3.18.1.1 \_\_init\_\_()

```
def message.Message.__init__ (
    self,
    data )
```

Initialize a message with data.

Args:

data: the data this message contains.  
Should be a dictionary that is  
transformable by 'json.dumps'

### 3.18.2 Member Function Documentation

#### 3.18.2.1 get\_data()

```
def message.Message.get_data (
    self )
```

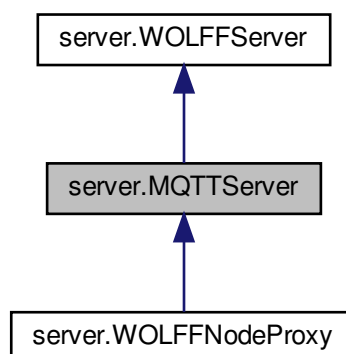
Get the data that is

The documentation for this class was generated from the following file:

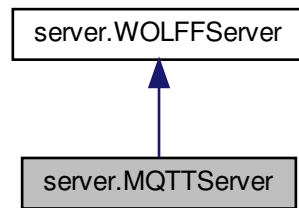
- wolff\_api\_plugins/client/message.py

## 3.19 server.MQTTServer Class Reference

Inheritance diagram for server.MQTTServer:



Collaboration diagram for server.MQTTServer:



### Public Member Functions

- `def __init__ (self, db_connection, ip, port, update_port, channels=None)`
- `def start (self, timeout=60)`
- `def handle_update_requests (self)`
- `def get_client (self)`
- `def add_channel (self, channel)`
- `def add_channels (self, channels)`

### Public Attributes

- `on_connect`
- `on_message`

#### 3.19.1 Detailed Description

An MQTTServer that performs requests on behalf of the user, similar to the base WOLFFServer. However, this server interacts with the client through an MQTT broker.

Incoming messages are expected to have the form `posts/client_x`, where `x` uniquely identifies a client.

The server expects any requests coming from a client to be in the `'posts/#'` topic. The response will be sent to the client via the `'respones/client_x'` topic.

#### 3.19.2 Member Function Documentation

### 3.19.2.1 start()

```
def server.MQTTServer.start (
    self,
    timeout = 60 )
```

Start the server, running continuously.  
Server will handle requests on behalf of the user via the MQTT broker.

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/server/server.py

## 3.20 server\_connection.MQTTServerConnection Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, ip="127.0.0.1", port=0, timeout=60)
- def **get\_channels** (self)
- def **get\_ip** (self)
- def **get\_port** (self)
- def **get\_timeout** (self)
- def **get\_received\_messages** (self)
- def **check\_for\_messages** (self)
- def **get\_client** (self, connect=False)
- def **subscribe\_to** (self, topic)
- def **send** (self, message)

### Public Attributes

- **on\_message**
- **on\_connect**
- **on\_disconnect**

### 3.20.1 Detailed Description

A delay-tolerant connection to an MQTT broker.  
Clients can send a message to the broker and check  
to see if any responses have been sent.

### 3.20.2 Member Function Documentation

### 3.20.2.1 check\_for\_messages()

```
def server_connection.MQTTServerConnection.check_for_messages (
    self )
```

Check if our message queue has had any items placed into it.  
Uses this class' mutex to avoid race conditions with the message-watch thread.

@note Calling this method clears the cache of any existing messages  
@returns A list of string messages that have been received by the client

### 3.20.2.2 get\_client()

```
def server_connection.MQTTServerConnection.get_client (
    self,
    connect = False )
```

Return this class' client, optionally connect  
to the MQTT broker  
If connect is true, a thread will be started that  
watches for messages.

### 3.20.2.3 send()

```
def server_connection.MQTTServerConnection.send (
    self,
    message )
```

Sends a message to the broker through the client.  
Subscribes to the topic where the response will be sent.

@param message The message to send  
@returns status of the attempt to send the message

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/client/server\_connection.py

## 3.21 oauth\_credentials.OAuth1Credentials Class Reference

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, api\_key="", api\_secret="", access\_token="", access\_secret="")
- def [get\\_from\\_file](#) (self, fname)
- def [set\\_client\\_info](#) (self, client\_key, client\_secret)
- def [set\\_resource\\_owner\\_info](#) (self, resource\_owner\_token, resource\_owner\_secret)
- def [as\\_dict](#) (self)

## Public Attributes

- **client\_key**
- **client\_secret**
- **resource\_owner\_key**
- **resource\_owner\_secret**

### 3.21.1 Member Function Documentation

#### 3.21.1.1 as\_dict()

```
def oauth_credentials.OAuth1Credentials.as_dict (
    self )
```

Return the dictionary representation of this object.

Returns:

Dictionary representing this object, for example:

```
{ 'resource_owner_secret': 'adfsdfsdfllflsfllds' }
```

#### 3.21.1.2 get\_from\_file()

```
def oauth_credentials.OAuth1Credentials.get_from_file (
    self,
    fname )
```

Parse oauth credentials from a file. The file must be tab-delimited with two columns, one for each item used in the authentication process. The first column must have names that correspond to the items needed for authentication, e.g.:

- client\_key
- client\_secret
- resource\_owner\_token
- resource\_owner\_secret

#### 3.21.1.3 set\_client\_info()

```
def oauth_credentials.OAuth1Credentials.set_client_info (
    self,
    client_key,
    client_secret )
```

Set the client information (token and secret) for this class

Args:

client\_token: the client token to set.  
access\_secret: the client secret to set.

#### 3.21.1.4 set\_resource\_owner\_info()

```
def oauth_credentials.OAuth1Credentials.set_resource_owner_info (
    self,
    resource_owner_token,
    resource_owner_secret )
```

Set the resource\_owner information (token and secret) for this class

Args:

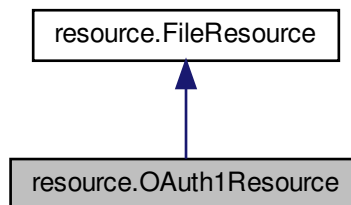
- resource\_owner\_token: the resource\_owner token to set.
- access\_secret: the resource\_owner secret to set.

The documentation for this class was generated from the following file:

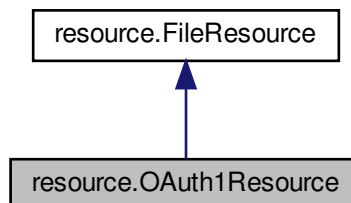
- wolff\_api\_plugins/client/oauth\_credentials.py

## 3.22 resource.OAuth1Resource Class Reference

Inheritance diagram for resource.OAuth1Resource:



Collaboration diagram for resource.OAuth1Resource:





## Public Member Functions

- def `__init__` (self, resource\_id, base\_path, root\_path='.', values=None)
- def `read` (self)
- def `write` (self, values, override=False)
- def `__hash__` (self)

### 3.22.1 Constructor & Destructor Documentation

#### 3.22.1.1 `__init__()`

```
def resource.OAuth1Resource.__init__ (
    self,
    resource_id,
    base_path,
    root_path = '.',
    values = None )
```

Create a new OAuth1 resource, with resource\_id and a base path.

@param resource\_id: The id of this resource. It should be unique.

@param base\_path Base of the path to write files to.  
Files will be written to resource\_id/base\_path

### 3.22.2 Member Function Documentation

#### 3.22.2.1 `read()`

```
def resource.OAuth1Resource.read (
    self )
```

Read the value for the resource from a file.

@returns A dictionary containing the necessary key/value pairs for OAuth1 identification.

### 3.22.2.2 write()

```
def resource.OAuth1Resource.write (
    self,
    values,
    override = False )
```

Write a dictionary of OAuth1 key/value pairs to this resource.

```
@throws OSError if override is false and the specified file already exists
@param values a dictionary of key/value pairs necessary for oauth authentication
@param override include if you want an existing file with this resources'
    path to be overwritten. If this flag is not True,
    an OSError will be raised upon existing file.
```

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/server/resource.py

## 3.23 resource.ResourceFactory Class Reference

### Public Member Functions

- def **get** (str\_type)

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/server/resource.py

## 3.24 response\_handler.ResponseHandler Class Reference

### Public Member Functions

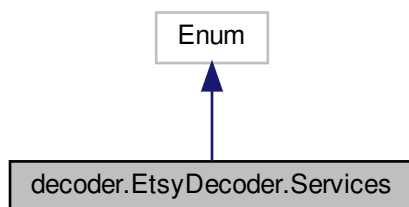
- def **\_\_init\_\_** (self, db\_conn)
- def **get\_handler** (self, data\_dict)

The documentation for this class was generated from the following file:

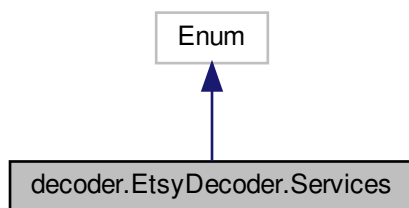
- wolff\_api\_plugins/server/response\_handler.py

## 3.25 decoder.EtsyDecoder.Services Class Reference

Inheritance diagram for decoder.EtsyDecoder.Services:



Collaboration diagram for decoder.EtsyDecoder.Services:



### Public Member Functions

- def `has_value` (cls, value)

### Static Public Attributes

- int `CREATE_LISTING` = 1
- int `CHECK_LISTING_STOCK` = 2
- int `UPDATE_LISTING` = 3

#### 3.25.1 Detailed Description

An enum listing the services that are currently available.

### 3.25.2 Member Function Documentation

#### 3.25.2.1 has\_value()

```
def decoder.EtsyDecoder.Services.has_value (
    cls,
    value )

Determine if this enum has a value.
We use this to check if a method is supported
by the decoder.

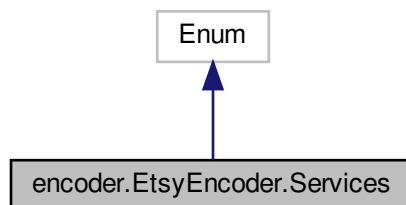
@param value The value to check for
@returns True if this class has value,
        false otherwise.
```

The documentation for this class was generated from the following file:

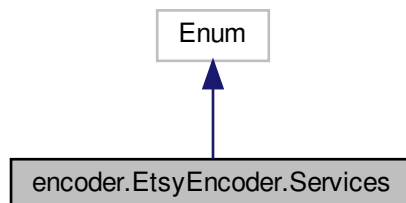
- wolff\_api\_plugins/server/decoder.py

### 3.26 encoder.EtsyEncoder.Services Class Reference

Inheritance diagram for encoder.EtsyEncoder.Services:



Collaboration diagram for encoder.EtsyEncoder.Services:



### Static Public Attributes

- int **CREATE\_LISTING** = 1

#### 3.26.1 Detailed Description

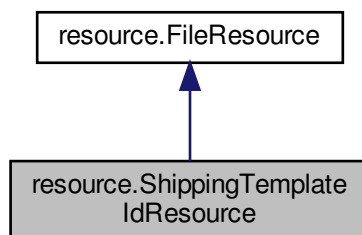
An enum detailing the services that are currently supported by the Encoder

The documentation for this class was generated from the following file:

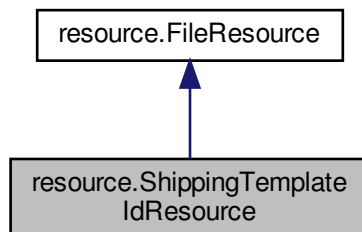
- wolff\_api\_plugins/client/encoder.py

### 3.27 resource.ShippingTemplateIdResource Class Reference

Inheritance diagram for resource.ShippingTemplateIdResource:



Collaboration diagram for resource.ShippingTemplateIdResource:



### Public Member Functions

- `def __init__ (self, resource_id, base_path, root_path='.', values=None)`
- `def write (self, values, override=False)`
- `def read (self)`
- `def __hash__ (self)`

#### 3.27.1 Detailed Description

A resource representing the ShippingTemplateId for Etsy's create listing method.

The documentation for this class was generated from the following file:

- `wolff_api_plugins/server/resource.py`

### 3.28 DBConnection.SQLite3DBConnection Class Reference

#### Public Member Functions

- `def __init__ (self, file_name)`
- `def add_listing (self, listing_id, client_id)`
- `def get_client_by_listing_id (self, listing_id)`
- `def get_listing_id (self, record_id)`
- `def get_listing_stock (self, record_id)`
- `def update_listing_stock (self, record_id, quantity)`
- `def add_listing_stock (self, record_id, quantity)`
- `def get_record_id (self, listing_id)`

#### Public Attributes

- `db_file_name`
- `conn`

The documentation for this class was generated from the following file:

- `wolff_api_plugins/server/DBConnection.py`

### 3.29 server\_connection.TCPServerConnection Class Reference

#### Public Member Functions

- `def __init__ (self, ip="127.0.0.1", port=0)`
- `def get_ip (self)`
- `def get_port (self)`
- `def send (self, message)`

### 3.29.1 Detailed Description

A connection to a server that takes place over the TCP/IP stack using TCP.

### 3.29.2 Member Function Documentation

#### 3.29.2.1 get\_ip()

```
def server_connection.TCPServerConnection.get_ip (  
    self )
```

Return the (IPv4) IP this connection will bind to when a connection is initialized

#### 3.29.2.2 get\_port()

```
def server_connection.TCPServerConnection.get_port (  
    self )
```

Return the port this connection will bind to when initiating a connection.

#### 3.29.2.3 send()

```
def server_connection.TCPServerConnection.send (  
    self,  
    message )
```

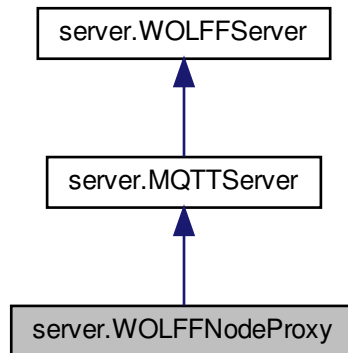
Sends an object of type `message` to the specified TCP ip address over the specified port. Waits for a specified amount of time for a response from the server, disconnects after waiting and not getting a response.

The documentation for this class was generated from the following file:

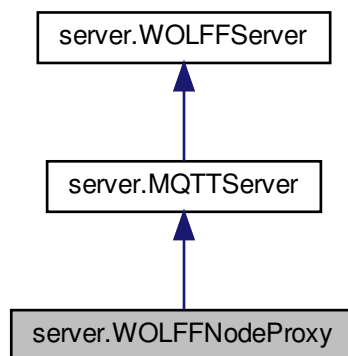
- `wolff_api_plugins/client/server_connection.py`

### 3.30 server.WOLFFNodeProxy Class Reference

Inheritance diagram for server.WOLFFNodeProxy:



Collaboration diagram for server.WOLFFNodeProxy:



#### Public Member Functions

- def **\_\_init\_\_** (self, client\_ip="127.0.0.1", client\_port=5555, broker\_ip="127.0.0.1", broker\_port=1883, channels=None)
- def **get\_client\_ip** (self)
- def **get\_client\_port** (self)
- def **decode\_data** (self, encoded\_message)
- def **do\_request** (self, data, topic)
- def **start** (self)



## Public Attributes

- **client\_ip**
- **client\_port**
- **broker\_ip**
- **broker\_port**
- **cond**
- **message\_buffer**
- [on\\_connect](#)

*Note: topic is of the form /posts/client\_x, where x is the ID for the client.*

- **on\_message**

### 3.30.1 Detailed Description

A WOLFFNodeProxy is a MQTTServer that sends requests on behalf of a user to the WOLFF gateway through MQTT via TCP/IP. Incoming messages are expected to have the form posts/client\_x, where x uniquely identifies a client.

The server expects any requests coming from a client to be in the 'posts/#' topic. The response will be sent to the client via the 'respones/client\_x' topic.

### 3.30.2 Member Function Documentation

#### 3.30.2.1 do\_request()

```
def server.WOLFFNodeProxy.do_request (
    self,
    data,
    topic )
```

Send a request ot the MQTT broker, with the specified topic.

@param data\_dict A dictionary containing the data to send.  
@param topic the string topic to publish the message to,  
              'posts/client\_1' for example.

#### 3.30.2.2 start()

```
def server.WOLFFNodeProxy.start (
    self )
```

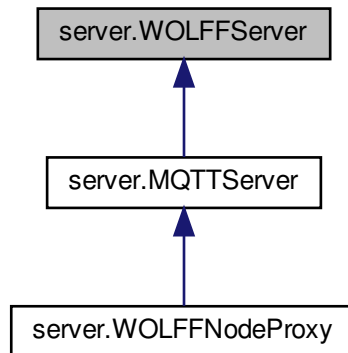
Start the server, allow it to run continuously. Requests will be forwarded to the MQTT broker on behalf of the user.

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/server/server.py

### 3.31 server.WOLFFServer Class Reference

Inheritance diagram for server.WOLFFServer:



#### Public Member Functions

- def **\_\_init\_\_** (self, db\_connection, ip="127.0.0.1", port=5555)
- def **get\_ip** (self)
- def **set\_ip** (self, ip)
- def **set\_port** (self, port)
- def **get\_port** (self)
- def **encode\_data** (self, data)
- def [decode\\_data](#) (self, data)
- def [do\\_request](#) (self, data\_dict)
- def [start](#) (self)
- def [get\\_request\\_handler](#) (self, credentials)
- def [annotate\\_data](#) (self, data\_dict, client\_manager)

#### Public Attributes

- **ip**
- **port**
- **conn**

#### 3.31.1 Detailed Description

A server for handling messages sent over the WOLFF network.

#### 3.31.2 Member Function Documentation

3.31.2.1 `annotate_data()`

```
def server.WOLFFServer.annotate_data (
    self,
    data_dict,
    client_manager )
```

Annotate a data dictionary with client and service-specific information that comes from the message.

@pre data\_dict contains the following:

```
{'api_details': ('etsy', 'create_listing'),
 'message': {'title': 'title_1', 'description': 'desc_1',
             'quantity': 1, 'price': 16.04, 'who_made': 'collective',
             'when_made': '1990s', 'is_supply': 1, 'shipping_template_id': 76575991147
            }
}
```

'api\_details' is a tuple containing the service and method being used.

Any non-dictionary values are examples, and will be dependent upon the message itself, but the keys are required. 'Message' will be dependent upon the service/method used.

@post data\_dict contains:

```
{'api_details': ('etsy', 'create_listing'),
 'message': {'title': 'title_1', 'description': 'desc_1',
             'quantity': 1, 'price': 16.04, 'who_made': 'collective',
             'when_made': '1990s', 'is_supply': 1, 'shipping_template_id': 76575991147
            },
 'method': {'http_method': 'post'},
 'url': 'https://openapi.etsy.com/v2/listings/',
 'credentials': {'client_key': '', 'client_secret': '',
                 'resource_owner_key': '',
                 'resource_owner_secret': ''
                }
 'client_id': 'client_x'
}
```

The values of method, url, and credentials are again dependent upon api\_details and message

@param client\_manager A client manager that can be used to identify clients based upon a service.

@param A dictionary containing the data from a decoded message

@returns a dictionary that is ready to be sent as an api request by `do_request`

3.31.2.2 `decode_data()`

```
def server.WOLFFServer.decode_data (
    self,
    data )
```

Decode a message, generating a dictionary of arguments and their values.

Details of the encoded message can be found here:  
[https://github.com/CANIS-NAU/WOLFF\\_Protocol/wiki](https://github.com/CANIS-NAU/WOLFF_Protocol/wiki)

### 3.31.2.3 do\_request()

```
def server.WOLFFServer.do_request (
    self,
    data_dict )
```

Perform a request on behalf of the user. Uses the appropriate request handler that allows for OAuth authentication using the users' credentials.

@param data\_dict: A dictionary containing data that allows us to determine the parameters for our request.

We need the following:

- credentials: oauth1 (oauth2 not supported currently) credentials containing the necessary args to perform authenticated requests.
- http\_method: the http method to perform
- url: the base url + the uri to submit the request to.
- params: The parameters (and their arguments) to include in the request

@returns the result object returned by OAuthRequestsLib

### 3.31.2.4 get\_request\_handler()

```
def server.WOLFFServer.get_request_handler (
    self,
    credentials )
```

Get the request handler that allows us to perform authenticated http requests on behalf of the client.

@param credentials: The user credentials to use for authenticating the user

### 3.31.2.5 start()

```
def server.WOLFFServer.start (
    self )
```

Start the server, allow it to run continuously. Requests will be performed on behalf of users, and responses will be forwarded.

The documentation for this class was generated from the following file:

- wolff\_api\_plugins/server/server.py

# Index

- `__init__`
  - `api_map::APIMap`, 6
  - `client::Client`, 12
  - `client_manager::ClientManager`, 15
  - `decoder::EtsyDecoder`, 19
  - `encoder::EtsyEncoder`, 20
  - `message::Message`, 27
  - `resource::FileResource`, 24
  - `resource::OAuth1Resource`, 33
- `add_special_params`
  - `api_map::APIMap`, 6
- `annotate_data`
  - `server::WOLFFServer`, 42
- `api_hook.APIHook`, 5
- `api_hook.APIMethod`, 8
- `api_hook::APIMethod`
  - `get_http_method`, 9
  - `get_uri`, 9
- `api_map.APIMap`, 5
- `api_map::APIMap`
  - `__init__`, 6
  - `add_special_params`, 6
  - `get_complete_url`, 6
  - `get_replacement`, 7
  - `perform_uri_replacement`, 7
  - `uri_is_substitutable`, 7
- `api_method.APIMethod`, 10
- `as_dict`
  - `oauth_credentials::OAuth1Credentials`, 31
- `check_existing_client`
  - `client_manager::ClientManager`, 15
- `check_for_messages`
  - `server_connection::MQTTServerConnection`, 29
- `client.Client`, 12
- `client::Client`
  - `__init__`, 12
  - `get_connection`, 13
  - `has_connection`, 13
  - `instantiate_resources`, 13
  - `set_connection`, 14
- `client_manager.ClientManager`, 14
- `client_manager::ClientManager`
  - `__init__`, 15
  - `check_existing_client`, 15
  - `get_client_by_service_identifier`, 15
  - `register_client`, 16
  - `register_existing_client`, 16
- `contains`
  - `resource::FileResource`, 24
- `DBConnection.SQLite3DBConnection`, 38
- `decode_data`
  - `server::WOLFFServer`, 43
- `decode_update_listing_message`
  - `decoder::EtsyDecoder`, 19
- `decoder.Applications`, 11
- `decoder.DecoderFactory`, 17
- `decoder.EtsyDecoder`, 18
- `decoder.EtsyDecoder.Services`, 35
- `decoder::DecoderFactory`
  - `get_service`, 17
- `decoder::EtsyDecoder`
  - `__init__`, 19
  - `decode_update_listing_message`, 19
  - `get_service_decoder`, 19
- `decoder::EtsyDecoder::Services`
  - `has_value`, 36
- `do_request`
  - `server::WOLFFNodeProxy`, 41
  - `server::WOLFFServer`, 43
- `encode`
  - `encoder::EtsyEncoder`, 20
- `encoder.Applications`, 10
- `encoder.EtsyEncoder`, 20
- `encoder.EtsyEncoder.Services`, 36
- `encoder::EtsyEncoder`
  - `__init__`, 20
  - `encode`, 20
- `endpoint_handler.EndpointManager`, 18
- `endpoint_handler.EtsyEndpointManager`, 21
- `get_client`
  - `server_connection::MQTTServerConnection`, 30
- `get_client_by_service_identifier`
  - `client_manager::ClientManager`, 15
- `get_complete_url`
  - `api_map::APIMap`, 6
- `get_connection`
  - `client::Client`, 13
- `get_data`
  - `message::Message`, 27
  - `resource::FileResource`, 25
- `get_file_name`
  - `resource::FileResource`, 25
- `get_from_file`
  - `oauth_credentials::OAuth1Credentials`, 31
- `get_full_file_path`

- resource::FileResource, 25
- get\_full\_path
  - resource::FileResource, 25
- get\_http\_method
  - api\_hook::APIMethod, 9
- get\_ip
  - server\_connection::TCPServerConnection, 39
- get\_port
  - server\_connection::TCPServerConnection, 39
- get\_replacement
  - api\_map::APIMap, 7
- get\_request\_handler
  - server::WOLFFServer, 44
- get\_service
  - decoder::DecoderFactory, 17
- get\_service\_decoder
  - decoder::EtsyDecoder, 19
- get\_uri
  - api\_hook::APIMethod, 9
- has\_connection
  - client::Client, 13
- has\_value
  - decoder::EtsyDecoder::Services, 36
- instantiate\_resources
  - client::Client, 13
- message.EtsyMessage, 22
- message.Message, 26
- message::Message
  - \_\_init\_\_, 27
  - get\_data, 27
- oauth\_credentials.OAuth1Credentials, 30
- oauth\_credentials::OAuth1Credentials
  - as\_dict, 31
  - get\_from\_file, 31
  - set\_client\_info, 31
  - set\_resource\_owner\_info, 31
- perform\_uri\_replacement
  - api\_map::APIMap, 7
- read
  - resource::FileResource, 26
  - resource::OAuth1Resource, 33
- register\_client
  - client\_manager::ClientManager, 16
- register\_existing\_client
  - client\_manager::ClientManager, 16
- resource.FileResource, 23
- resource.OAuth1Resource, 32
- resource.ResourceFactory, 34
- resource.ShippingTemplateIdResource, 37
- resource::FileResource
  - \_\_init\_\_, 24
  - contains, 24
  - get\_data, 25
  - get\_file\_name, 25
  - get\_full\_file\_path, 25
  - get\_full\_path, 25
  - read, 26
- resource::OAuth1Resource
  - \_\_init\_\_, 33
  - read, 33
  - write, 33
- response\_handler.CheckListingStockResponseHandler, 12
- response\_handler.CreateListingResponseHandler, 17
- response\_handler.ResponseHandler, 34
- send
  - server\_connection::MQTTServerConnection, 30
  - server\_connection::TCPServerConnection, 39
- server.MQTTServer, 27
- server.WOLFFNodeProxy, 40
- server.WOLFFServer, 42
- server::MQTTServer
  - start, 28
- server::WOLFFNodeProxy
  - do\_request, 41
  - start, 41
- server::WOLFFServer
  - annotate\_data, 42
  - decode\_data, 43
  - do\_request, 43
  - get\_request\_handler, 44
  - start, 44
- server\_connection.MQTTServerConnection, 29
- server\_connection.TCPServerConnection, 38
- server\_connection::MQTTServerConnection
  - check\_for\_messages, 29
  - get\_client, 30
  - send, 30
- server\_connection::TCPServerConnection
  - get\_ip, 39
  - get\_port, 39
  - send, 39
- set\_client\_info
  - oauth\_credentials::OAuth1Credentials, 31
- set\_connection
  - client::Client, 14
- set\_resource\_owner\_info
  - oauth\_credentials::OAuth1Credentials, 31
- start
  - server::MQTTServer, 28
  - server::WOLFFNodeProxy, 41
  - server::WOLFFServer, 44
- uri\_is\_substitutable
  - api\_map::APIMap, 7
- write
  - resource::OAuth1Resource, 33