
translationConverter (tx) Documentation

Release 1.0

Door43

Jan 15, 2018

Services:

1	src.services package	3
1.1	Subpackages	3
1.1.1	src.services.notes package	3
1.2	Submodules	4
1.2.1	src.services.sample module	4
2	src.main module	5
3	src.service_manager module	7
4	Indices and tables	9
	Python Module Index	11

This site is the official documentation for translationConverter (tx) tool. This tool provides a number of RESTful services for processing and publishing translations on the [Door43 Content Service](#).

src.services package

This package contains services and supporting code that may be registered with the Flask api.

1.1 Subpackages

1.1.1 src.services.notes package

This is a sample service

Submodules

src.services.notes.notes module

This is a more complex illustration of a sample service.

`src.services.notes.notes.note_repr(key)`

Prepares the note response

Parameters `key` (*int*) – The note key

Returns The formatted note response

`src.services.notes.notes.notes_detail(key)`

Retrieve, update or delete note instances.

Parameters `key` (*int*) – The note key

Returns When PUTing a new note this will return the note object. When DELETEing a note this will return an empty body with a 204 header When GETing a note this will return the note object

`src.services.notes.notes.notes_list()`

Lists the notes

Returns A list of note objects

1.2 Submodules

1.2.1 src.services.sample module

`src.services.sample.sample_service` (*name=None*)

This is a sample service. Give it your name and prepare to be greeted!

Parameters `name` (*basestring*) – Your name

Returns A greeting or an error

This module bootstraps the API setup and registers all of the available services.

`src.main.index()`

Displays the available services.

Returns A dictionary of registered endpoints.

`src.main.page_not_found(e)`

Handles 404 errors experienced by the clients.

Parameters `e` – The exception object

Returns The exception object

`src.main.register_services(api)`

Registers the available services with the Service Manager. New services should be registered inside this function.

Parameters `api` (`ServiceManager`) – An instance of the Service Manager

src.service_manager module

class `src.service_manager.ServiceManager` (*app*)

The ServiceManager is a light wrapper around FlaskAPI that allows us to perform additional processing when routes are registered.

formatted_routes (*host_url*)

Returns a dictionary of endpoints that have been registered with the api. This is useful for displaying a summary of available endpoints.

Parameters `host_url` (*basestring*) – The host address e.g. <https://example.com>

Returns

A dict mapping endpoints to their corresponding route and summary. For example:

```
>>> {
>>>   'my_service': {
>>>     'route': 'https://example.com/myservice/<string:some_
↳arg>',
>>>     'doc': 'This is my function. It takes one argument.'
>>>   }
>>> }
```

register_route (*func, rule, **options*)

Like `Flask.route()` but with additional processing. The route is stored in a list so we can keep track of all the exposed API endpoints.

Parameters

- **func** (*callable*) – The function that will be executed for the URL rule.
- **rule** (*basestring*) – The URL rule to match in order to execute the endpoint.
- **endpoint** (*basestring*) – The name of the exposed endpoint. Flask itself assumes the name of the view function as the endpoint.

Returns The registered function.

route (*rule*, ***options*)

A decorator version of `register_route()`. This allows you to use the decorator syntax like:

```
>>> @api.route('/')
>>> def index():
```

Parameters

- **rule** (*basestring*) – The URL rule to match in order to execute the endpoint.
- **endpoint** (*basestring*) – The name of the exposed endpoint. Flask itself assumes the name of the view function as the endpoint.

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`

S

src.main, 5
src.service_manager, 7
src.services, 3
src.services.notes, 3
src.services.notes.notes, 3
src.services.sample, 4

F

formatted_routes() (src.service_manager.ServiceManager method), 7

I

index() (in module src.main), 5

N

note_repr() (in module src.services.notes.notes), 3

notes_detail() (in module src.services.notes.notes), 3

notes_list() (in module src.services.notes.notes), 3

P

page_not_found() (in module src.main), 5

R

register_route() (src.service_manager.ServiceManager method), 7

register_services() (in module src.main), 5

route() (src.service_manager.ServiceManager method), 7

S

sample_service() (in module src.services.sample), 4

ServiceManager (class in src.service_manager), 7

src.main (module), 5

src.service_manager (module), 7

src.services (module), 3

src.services.notes (module), 3

src.services.notes.notes (module), 3

src.services.sample (module), 4