

MusicGenerationWebService

Release 0.0.1

n4mespace

May 25, 2020

1	Subpackages	1
2	Module contents	3
3	Module contents	5
4	Subpackages	7
5	Submodules	9
6	server.algorythm.music_generation module	11
7	server.algorythm.routes module	13
8	Module contents	15
9	Submodules	17
10	server.errors.handlers module	19
11	Module contents	21
12	Submodules	23
13	server.main.routes module	25
14	Module contents	27
	Python Module Index	29
	Index	31

Subpackages

Module contents

`server.create_app (config_class: config.Config) → quart.app.Quart`

Create Quart app given Config class

Use:

Compress for files sizes reduction JWTManager for tokens CORS for savity

Parameters {Config} -- class with configuration attrs (config_class) –

Returns Quart – created app instance

Module contents

Subpackages

Submodules

server.algorythm.music_generation module

`server.algorythm.music_generation.generate` (*genre: str, checkpoint_dir: str, sess: tensorflow.python.client.session.Session, params: dict*) → str

Generate music given model session and params

Parameters

- {str} -- **genre name** (*genre*) –
- {str} -- **checkpoint path** (*checkpoint_dir*) –
- {tf.Session} -- **model session** (*sess*) –
- {dict} -- **additional params** (*params*) –

Returns str – generated music

server.algorythm.routes module

Module contents

Submodules

server.errors.handlers module

```
server.errors.handlers.bad_request ( message )  
server.errors.handlers.error_response ( status_code, message=None )  
async server.errors.handlers.internal_error ( error )  
async server.errors.handlers.not_found_error ( error )  
server.errors.handlers.wants_json_response ( )
```

Module contents

Submodules

server.main.routes module

```
class server.main.routes.HealthCheck
    Bases: quart_openapi.Resource
    Server success startup checker
    Extends:
        Resource
    async get ( ) → quart.wrappers.response.Response
        GET checker
        Must return json with success status
    methods: Optional[List[str]] = ['GET', 'POST']
    async post ( ) → quart.wrappers.response.Response
        POST checker
        Must get data and return json with success status

class server.main.routes.Service
    Bases: quart_openapi.Resource
    Main page
    Serve client side
    Extends:
        Resource
    genres = ['classical', 'jazz', 'rock']
    async get ( ) → quart.wrappers.response.Response
        Main page
        Renders main page template
    methods: Optional[List[str]] = ['GET']
```

Module contents

- *Index*
- *Module Index*
- *Search Page*

S

server, [3](#)
 server.algorythm.music_generation,
 [11](#)
 server.errors, [21](#)
 server.errors.handlers, [19](#)
 server.main, [27](#)
 server.main.routes, [25](#)

B

`bad_request()` (in module `server.errors.handlers`),
19

C

`create_app()` (in module `server`), 3

E

`error_response()` (in module `server.errors.handlers`), 19

G

`generate()` (in module `server.algorythm.music_generation`), 11

`genres` (`server.main.routes.Service` attribute), 25

`get()` (`server.main.routes.HealthCheck` method),
25

`get()` (`server.main.routes.Service` method), 25

H

`HealthCheck` (class in `server.main.routes`), 25

I

`internal_error()` (in module `server.errors.handlers`), 19

M

`methods` (`server.main.routes.HealthCheck`
attribute), 25

`methods` (`server.main.routes.Service` attribute), 25

module

`server`, 3

`server.algorythm.music_generation`, 11

`server.errors`, 21

`server.errors.handlers`, 19

`server.main`, 27

`server.main.routes`, 25

N

`not_found_error()` (in module `server.errors.handlers`), 19

P

`post()` (`server.main.routes.HealthCheck` method),
25

S

`server`

module, 3

`server.algorythm.music_generation`

module, 11

`server.errors`

module, 21

`server.errors.handlers`

module, 19

`server.main`

module, 27

`server.main.routes`

module, 25

`Service` (class in `server.main.routes`), 25

W

`wants_json_response()` (in module `server.errors.handlers`), 19

