

swagger-bootstrap-ui RESTful Api Docs

Description:

Coachingmate2020 API

HOST: <https://coachingmate-backend2020.herokuapp.com>

Contact:coachingmate team 2020-unimelb

Version:1.0.0

api url: <https://github.com/chenyuguo/coachingmate/tree/master/docs/api.pdf>

activity-data-retrieve-controller

retrieve Data By Username

note:query all activity data of specific user

url: `/activity/getall`

method: `GET`

consumes: ``

produces: `["*"/*"]`

Request Params:

name	description	in	require	data type	schema
username	username	query	true	string	

Response Example:

```
[
  {
    "avg_cadence": {},
    "avg_heart_rate": {},
    "avg_power": {},
    "avg_speed": {},
    "avg_stroke_count": {},
    "avg_stroke_distance": {},
    "intensity_factor": {},
```

```

    "max_cadence": {},
    "max_heart_rate": {},
    "max_power": {},
    "max_speed": {},
    "normalized_power": {},
    "num_active_length": {},
    "num_laps": {},
    "pool_length": {},
    "serial_number": "",
    "session_id": "",
    "sport_type": "",
    "start_position_lat": {},
    "start_position_long": {},
    "start_time": "",
    "statistic_id": "",
    "sub_sport": {},
    "threshold_power": {},
    "tick": {},
    "total_ascent": {},
    "total_calories": {},
    "total_cycles": {},
    "total_descent": {},
    "total_distance": {},
    "total_elapsed_time": {},
    "total_fat_calories": {},
    "total_work": {},
    "training_stress_score": {},
    "user_id": ""
  }
]

```

Response Params:

name	description	type	schema
avg_cadence		Number	Number
avg_heart_rate		Number	Number
avg_power		Number	Number
avg_speed		Number	Number
avg_stroke_count		Number	Number
avg_stroke_distance		Number	Number
intensity_factor		Number	Number

max_cadence		Number	Number
max_heart_rate		Number	Number
max_power		Number	Number
max_speed		Number	Number
normalized_power		Number	Number
num_active_length		Number	Number
num_laps		Number	Number
pool_length		Number	Number
serial_number		string	
session_id		string	
sport_type		string	
start_position_lat		Number	Number
start_position_long		Number	Number
start_time		string	
statistic_id		string	
sub_sport		Number	Number
threshold_power		Number	Number
tick		Number	Number
total_ascent		Number	Number
total_calories		Number	Number
total_cycles		Number	Number
total_descent		Number	Number
total_distance		Number	Number
total_elapsed_time		Number	Number
total_fat_calories		Number	Number
total_work		Number	Number

training_stress_score		Number	Number
user_id		string	

Response Status:

code	description	schema
200	OK	Statistic
401	Unauthorized	
403	Forbidden	
404	Not Found	

garmin-push-controller

push data url

note:configure this url to enpoint configuration, and the garmin endpoint will transfer the data to this server

url: `/push`

method: `POST`

consumes: `["multipart/form-data"]`

produces: `["*"/*"]`

Request Params:

name	description	in	require	data type	schema
file	file	formData	false	file	
uploadMetaData	uploadMetaData	body	false	string	

Response Status:

code	description	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

login

login interface

note:user login

url: /login

method: POST

consumes: ["application/json"]

produces: ["*"/*"]

Request Params:

name	description	in	require	data type	schema
password	password	body	true	string	
username	username	body	true	string	

Response Example:

```
{
  "password": "",
  "userAccessSecret": "",
  "userAccessToken": "",
  "userId": 0,
  "username": ""
}
```

Response Params:

name	description	type	schema
password		string	
userAccessSecret		string	
userAccessToken		string	
userId		integer(int64)	integer(int64)
username		string	

Response Status:

code	description	schema
200	OK	UserPartner
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

o-auth-controller

callback url

note: this is used to receive the request from garmin connect

url: `/auth/accessToken`

method: `GET`

consumes: ``

produces: `[" */ * "]`

Request Params:

name	description	in	require	data type	schema
oauth_token	oauth_token	query	true	string	
oauth_verifier	oauth_verifier	query	true	string	

Response Status:

code	description	schema
200	OK	
401	Unauthorized	
403	Forbidden	
404	Not Found	

oauth RequestToken

note:Acquire Unauthorized Request Token and Token Secret

url: /auth/requestToken

method: POST

consumes: ["application/json"]

produces: ["*"/*"]

Request Params:

name	description	in	require	data type	schema
username	username	body	true	string	

Response Status:

code	description	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

generate Access Token

note:when user first invoke this url, it will authorisethe user, when this user has been authorised, it just need to get token from mongodb

url: `/auth/userAccessToken`

method: `POST`

consumes: `["application/json"]`

produces: `["*"/*"]`

Request Params:

name	description	in	require	data type	schema
username	username	query	true	string	

Response Status:

code	description	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

registry-controller

register interface

note:user registration

url: `/register`

method: `GET`

consumes: ``

produces: ["*"/*"]

Request Params:

name	description	in	require	data type	schema
password	password	query	true	string	
username	username	query	true	string	

Response Example:

```
{
  "password": "",
  "userAccessSecret": "",
  "userAccessToken": "",
  "userId": 0,
  "username": ""
}
```

Response Params:

name	description	type	schema
password		string	
userAccessSecret		string	
userAccessToken		string	
userId		integer(int64)	integer(int64)
username		string	

Response Status:

code	description	schema
200	OK	UserPartner
401	Unauthorized	
403	Forbidden	
404	Not Found	