

Projet : Myclass.com

API

Date : 16/12/21

Cherif BALDE

Objectif :

Ce document liste les api utilisées dans le projet ainsi que la spécification de chacune d'entre elle.

Module : Gestion des utilisateurs :

1. Création d'un compte utilisateur :

Description :

Cette api permet de créer le compte utilisateur avec des données minimales. Elle envoie l'email d'activation a l'utilisateur.

url :

http://89.156.84.196:5000/myclass/api/add_user_account/

Parametres :

Cette api prend comme paramètres les informations suivantes :

- Name
- Surname
- Email
- Pwd

Output :

Cette api renvoie les informations suivantes :

- Status : True ou False comme valeurs
- Message : Le texte renvoyé après les contrôles de cohérence.

Exemple avec utilitaire Postman :

The screenshot shows a REST client interface with the following details:

- Request URL:** `http://89.156.84.196:5000/myclass/api/add_user_account/` (highlighted with a red box and '1')
- Method:** POST
- Request Body:** A table with the following data (highlighted with a red box and '2'):

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	balde	
<input checked="" type="checkbox"/> surname	cherif	
<input checked="" type="checkbox"/> email	cherif.balde@yahoo.fr	
<input checked="" type="checkbox"/> pwd	cherif	
- Response:** A JSON object (highlighted with a red box and '3'):


```

1 {
2   "message": "User account created ==> OKK",
3   "status": "True"
4 }
```
- Status:** 200 OK, 395 ms, 564 B

Email d'activation :

The email content is as follows:

- From:** MySy : test email activation compte
- Sender:** billardman01@hotmail.com (Se désinscrire)
- To:** cherif.balde@yahoo.fr
- Subject:** Bonjour
- Content:**

Pour valider votre compte, merci de cliquer ici

http://89.156.84.196:5000/myclass/api/valide_user_account/61bde264f0fb14053333c0e7

Cordialement
L'équipe informatique de MySy

2. Récupération des informations d'un compte utilisateur :

Cette api permet de faire un « get » sur une compte utilisateur et de récupérer toutes les informations associées.

La clé de recherche est l'adresse email.

url :

http://89.156.84.196:5000/myclass/api/get_user_account/

Parametres :

Cette api prend comme paramètres les informations suivantes :

- email

Output :

Cette api renvoie les informations suivantes :

- Status : True ou False comme valeurs
- Message : Le texte renvoyé après les contrôles de cohérence.

Exemple avec Postman :

The screenshot shows a Postman interface for a GET request. The URL is `http://89.156.84.196:5000/myclass/api/get_user_account/`. The request body is set to `form-data` and contains a single parameter: `email` with the value `billard@billard.com`. The response is a JSON object with a `message` array and a `status` field. The response status is `200 OK` with a response time of `21 ms` and a body size of `1011 B`.

```
1
2
3
4
5
6
{"message": [
  {"_id": "61ba609321e6c8c32d1b22f2", "name": "BALDE", "surname": "Cherif", "email": "billard@billard.com", "active": "1", "locked": "0", "date_update": "2021-12-15 23:04:47.463328", "adr_city": "Roissy en brie", "adr_county": "France", "adr_street": "3, place de la republique", "adr_zip": "75009", "facebook": "www.facebook.com?sds", "mob_phone": "0503 03 02 05", "twitter": "www.twitter.com?sds"}
],
"status": "True"}
```

3. Mise à jour d'un compte utilisateur :

Cette api met à jour le compte utilisateur. Pour commencer, les champs « updatable » sont :

- adr_street,
- adr_city,
- adr_zip,
- adr_county,
- mob_phone,

- facebook,
- twitter,
- email

Cette liste sera rallongée plus tard.

Url :

http://89.156.84.196:5000//myclass/api/update_user_account/

Parametres :

Cette api prend comme paramètres les informations suivantes :

- email => Clé de recherche pour la mise à jour => Obligatoire
- adr_street,
- adr_city,
- adr_zip,
- adr_county, → attention, c'est une coquille a corriger plus tard. On parle countRy
- mob_phone,
- facebook,
- twitter,

il n'est pas nécessaire d'ajouter tous les champ. Juste ceux qu'il faut modifier.

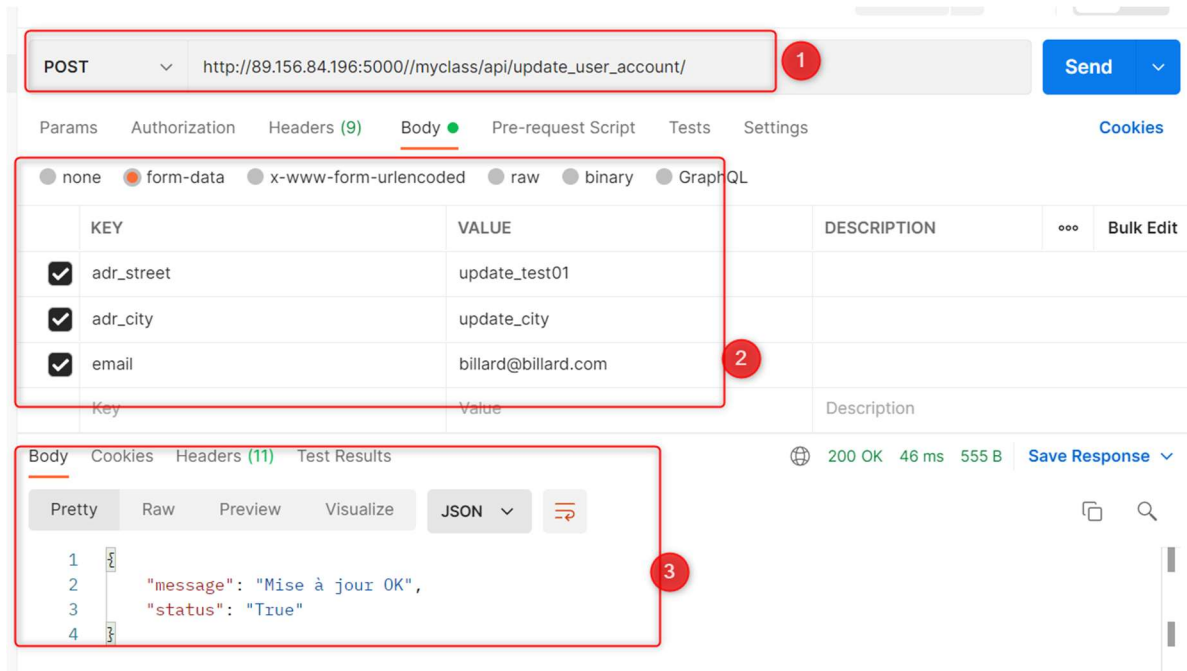
La clé de recherche pour la mise à jour est l'adresse email.

Output :

Cette api renvoie les informations suivantes :

- Status : True ou False comme valeurs
- Message : Le texte renvoyé après les contrôles de cohérence.

Exemple avec utilitaire Postman :



4 – Ajout des objectifs a un utilisateur

Les champs :

- Email_user
- Description
- Date_souhaite
- Domaine
- Niveau
- Date_modif

Url :

http://192.168.0.21:5000/myclass/api/add_update_user_objectif/

Paramètres :

- Cette API pres en en paramètre les informations suivantes :
- email_user → Clé
- name => Obligatoire
- description
- date_souhaite => Obligatoire
- domaine
- niveau

Output :

Cette api renvoie les informations suivantes :

- Status : True ou False comme valeurs
- Message : Le texte renvoyé après les contrôles de cohérence.

Exemple d'utilisation avec postman

The screenshot shows a Postman interface for a POST request. The URL is `http://127.0.0.1:5000/myclass/api/add_update_user_objectif/`. The body is set to form-data with the following fields:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	Maitrise Python	
<input checked="" type="checkbox"/> email_user	pyhton@billard.com	
<input checked="" type="checkbox"/> description	Cette formation permet de maitriser python	
<input checked="" type="checkbox"/> date_souhaite	10/10/2022	
<input checked="" type="checkbox"/> domaine	programmation	
Key	Value	Description

The response body is shown in JSON format:

```
1  "message": "Objectif créé ==> OKK",  
2  "status": "True"
```

Resultat dans mongodb :

```
_id: ObjectId("61c0eea8084eda3fb1d19766")  
email_user: "pyhton@billard.com"  
name: "Maitrise Python"  
description: "Cette formation permet de maitriser python"  
date_souhaite: "10/10/2022"  
domaine: "programmation"  
niveau: "programmation"  
active: "1"  
date_update: "2021-12-20 21:59:20.302223"
```

5 – récupération des objectifs d'un utilisateur (get_objectif) :

Cette API retourne les objectifs d'un utilisateur.

L'info obligatoire est l'adresse email. Si un nom d'objectif est fourni, le système renverra uniquement cet objectif. Si non tous les objectifs de cet utilisateur sont renvoyés.

Url :

http://192.168.0.21:5000//myclass/api/get_user_objectif

Paramètres :

Cette api prend les paramètres ci-dessous :

- email_user => Clé/ obligatoire
- name => nom de la formation

Exemple avec postman :

The image displays two screenshots of the Postman interface, illustrating an API request and its response.

First Screenshot (Top):

- Request:** Method: GET, URL: `http://127.0.0.1:5000//myclass/api/get_user_objectif`. The request body is set to `form-data`. Parameters are defined as follows:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> email_user	pyhton@billard.com	
<input checked="" type="checkbox"/> name	formation_name	
- Response:** Status: 200 OK, Time: 71 ms, Size: 857 B. The response body is JSON:

```
1 {
2   "message": [
3     {
4       "\_id": "\_61bfbb638cf8858dacf97b", "\_email_user": "\_pyhton@billard.com", "\_name": "\_formation_name", "\_description": "\_form_descrito2", "\_date_souhaitel": "\_date_sou2",
5       "\_domainel": "\_domainel", "\_niveau": "\_domainel", "\_active": "\_1", "\_date_update": "\_2021-12-20 21:43:57.669423"}
6     ]
7   },
8   "status": "True"
```

Second Screenshot (Bottom):

- Request:** Method: GET, URL: `http://127.0.0.1:5000//myclass/api/get_user_objectif`. The request body is set to `form-data`. Only the `email_user` parameter is present:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> email_user	pyhton@billard.com	
- Response:** Status: 200 OK, Time: 52 ms, Size: 118 KB. The response body is JSON:

```
1 {
2   "message": [
3     {
4       "\_id": "\_61bfbb638cf8858dacf97b", "\_email_user": "\_pyhton@billard.com", "\_name": "\_formation_name", "\_description": "\_form_descrito2", "\_date_souhaitel": "\_date_sou2",
5       "\_domainel": "\_domainel", "\_niveau": "\_domainel", "\_active": "\_1", "\_date_update": "\_2021-12-20 21:43:57.669423"},
6     {
7       "\_id": "\_61c0eaa8084eda3fb1d19766", "\_email_user": "\_pyhton@billard.com", "\_name": "\_Maitrise Python", "\_description": "\_Cette formation permet de maitriser python",
8       "\_date_souhaitel": "\_18/10/2022", "\_domainel": "\_programmation", "\_niveau": "\_programmation", "\_active": "\_1", "\_date_update": "\_2021-12-20 21:59:20.382223"}
9     ]
10  },
11  "status": "True"
```


6 – Ajout des profils temporaire (en cours)

Un profil temporaire est défini par :

- nom du profil
- niveau etude
- domaine d'étude
- diplôme
- pays
- expérience la plus élevée :
 - pays
 - poste
 - année expérience

7 – Login et gestion des token :

Ici sera traité l'authentification d'un utilisateur avec l'utilisation d'un token et de son temps d'activité.

Process :

Lorsqu'un utilisateur se connecte, après vérification de son login et pass, un token lui est attribué.

Ce token sera valide une période donnée (exemple : 30 min).

A chaque fois qu'un utilisateur lance une action, si c'est l'API vient avec un user/email et un token (ces 2 infos sont obligatoires), on va procéder à la vérif suivante :

- Est-ce que l'association : user/email ↔ token est valide ?
- Si oui, alors on vérifie le temps « de vie » du token (a chaque token, il aura les infos : date creation, date_time_dernière activité).
- Si les 2 étapes sont ok, le champs « date_time_dernière_activité » est mise à jour.

Url :

<http://192.168.0.21:5000//myclass/api/login>

Paramètres :

Cette api prend en paramètre les informations suivantes :

- Email → Obligatoire
- Pwd → Obligatoire

Output :

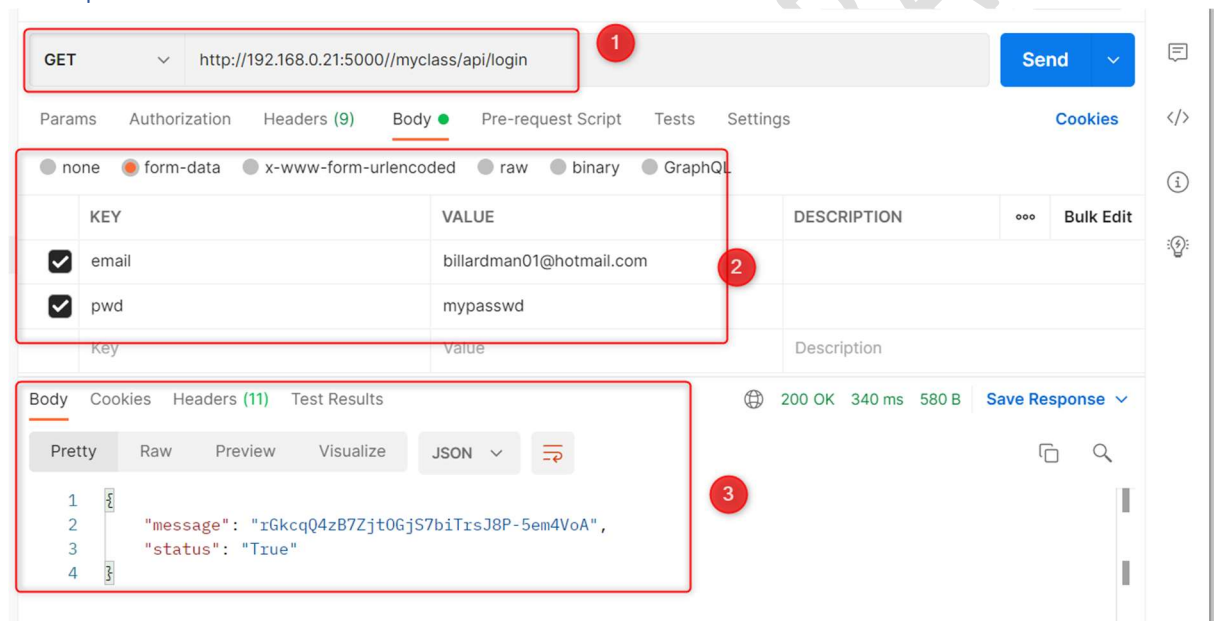
Cette API retourne le statut de la connexion et un token de connexion.

Si le login et le mot de passe sont ok, la variable « status » aura la valeur « True », si non « False ».

Si status = « True », alors la variable « message » contiendra le « token » de session.

Important : Par la suite, toute action lancée en mode connecté devra comporter le token.

Exemple avec Postman :



The screenshot shows a Postman interface for a REST client. The request is a GET method to the URL `http://192.168.0.21:5000//myclass/api/login`. The body is set to form-data with two fields: `email` with value `billardman01@hotmail.com` and `pwd` with value `mypasswd`. The response is a 200 OK status with a 340 ms response time and 580 B of data. The response body is shown in JSON format:

```
1 {
2   "message": "rGkcqQ4zB7Zjt0GjS7biTisJ8P-5em4VoA",
3   "status": "True"
4 }
```