

# Guide de l'API QCM (FastAPI)

## API

Avant d'utiliser cet API, les utilisateurs doivent vérifier leurs identifiants

L'utilisateur doit pouvoir choisir un type de test (`use`) ainsi qu'une ou plusieurs catégories (`subject`). De plus, l'application peut produire des QCMs de 5, 10 ou 20 questions. Les questions doivent être retournées dans un ordre aléatoire : ainsi, une requête avec les mêmes paramètres pourra retourner des questions différentes.

API permet à un utilisateur admin de créer une nouvelle question.

L'API est implémenté un point de terminaison pour vérifier que l'API est bien fonctionnelle

**Installation** : `requirements.txt`

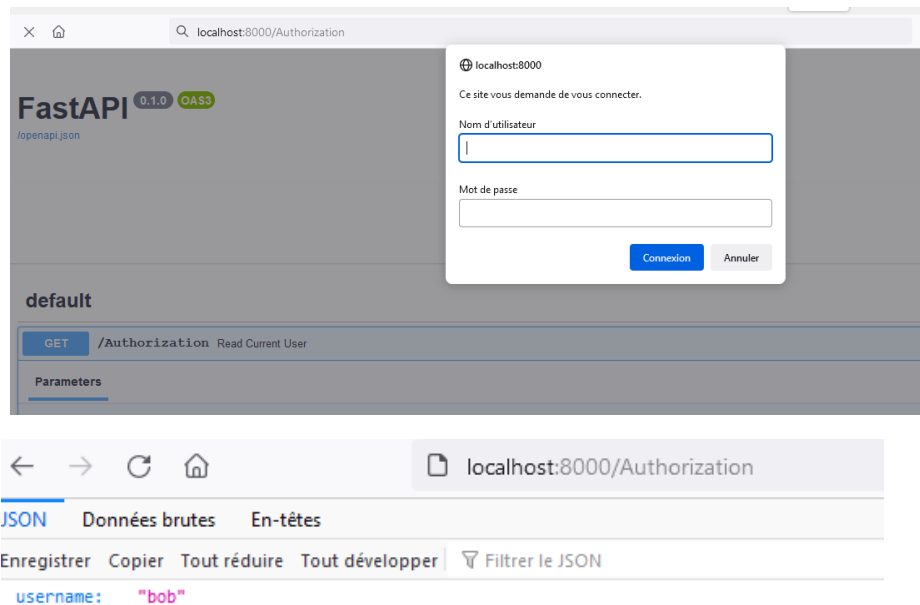
`fastapi`

`uvicorn`

`pandas`

`secrets`

## Authorization



Vérifier :

```
ubuntu@ip-172-31-6-211:~$ curl http://localhost:8000/headers -u "bob:builder"
{"username":"bob"}ubuntu@ip-172-31-6-211:~$
```

## QCM

Les 3 variables sont requises :

1. Variable 'use' : type string
2. Variable 'subject' : type string
3. nb\_QCM : type integer (nombre de QCM à afficher)

Les types de 'use' et 'subject' suivants sont acceptés :

subject	use
Automation	Test de validation
BDD	Test de positionnement
Classification	Test de validation
Data Science	Total Bootcamp
Docker	Test de positionnement
Machine Learning	Total Bootcamp
Streaming de données	Test de positionnement
	Test de validation
Systèmes distribués	Test de positionnement
	Test de validation
Sytèmes distribués	Test de validation

Vérifier la fonction de l'API :

FastAPI - Swagger UI

localhost:8000/docs#/default/get\_QCM\_QCM\_get

**GET** /QCM Get Qcm

**Parameters**

Name	Description
<b>use</b> * required string (query)	Test de validation
<b>subject</b> * required string (query)	Automation
<b>nb_QCM</b> * required integer (query)	5

**Execute**

Exécuter 2 fois pour voir si les questions affichées sont différentes et dans un ordre aléatoire

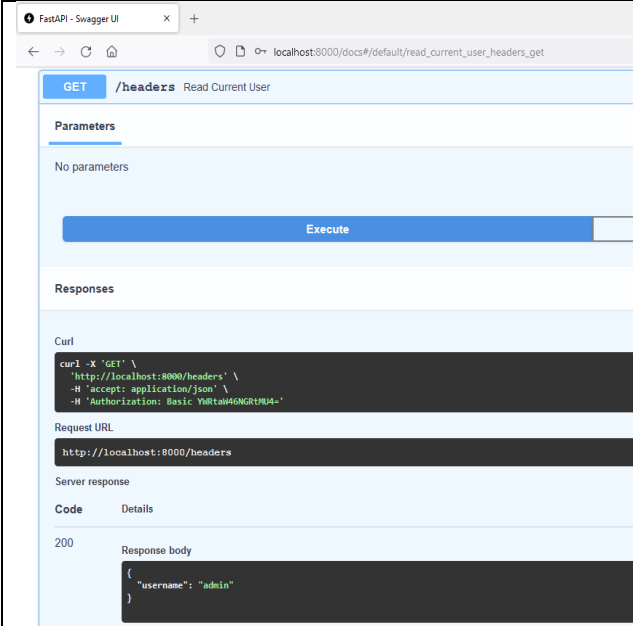
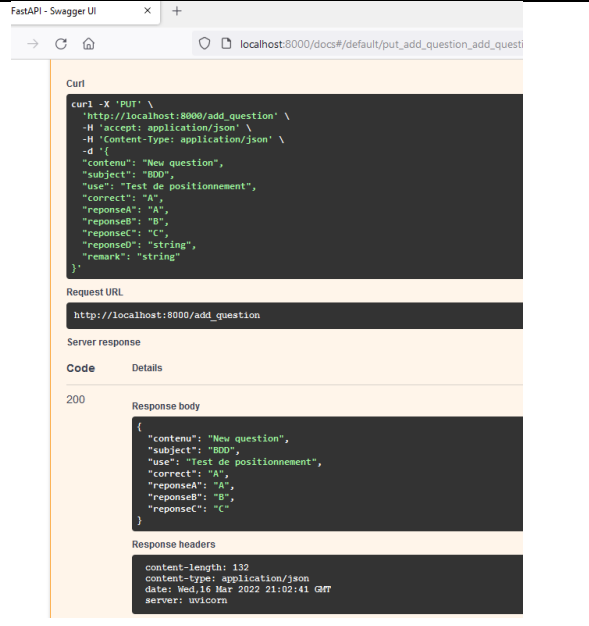
Code	Details
200	Response body <pre>[   "Dans Airflow, le DAG répertorie",   "Pour faciliter le passage de fichiers entre la machine hôte et un container Docker, on peut utiliser les",   "Pourquoi utiliser des APIs?",   "Pour faciliter la communication entre la machine hôte et un container Docker, on peut utiliser les",   "Si on fait un parallèle avec la programmation orientée objet, les images de containers Docker sont l'équivalent des" ]</pre>

Code	Details
200	Response body <pre>[   "Pourquoi utiliser des APIs?",   "Pourquoi utiliser Docker?",   "Pour faciliter la communication entre la machine hôte et un container Docker, on peut utiliser les",   "Dans Airflow, le DAG répertorie",   "Pour faciliter le passage de fichiers entre la machine hôte et un container Docker, on peut utiliser les" ]</pre>

## Admin & nouvelle question

Après la vérification que l'utilisateur est l'admin, API permet l'admin d'ajouter une nouvelle question.

 <p>FastAPI - Swagger UI</p> <p>GET /headers Read Current User</p> <p>Parameters</p> <p>No parameters</p> <p>Execute</p> <p>Responses</p> <p>Curl</p> <pre>curl -X 'GET' \   'http://localhost:8000/headers' \   -H 'accept: application/json' \   -H 'Authorization: Basic YWRtaW46MTIzNDU2Nzg='</pre> <p>Request URL</p> <p>http://localhost:8000/headers</p> <p>Server response</p> <table><tr><th>Code</th><th>Details</th></tr><tr><td>200</td><td>Response body<pre>{   "username": "admin" }</pre></td></tr></table>	Code	Details	200	Response body <pre>{   "username": "admin" }</pre>	 <p>FastAPI - Swagger UI</p> <p>PUT /add_question Add New Question</p> <p>Curl</p> <pre>curl -X 'PUT' \   'http://localhost:8000/add_question' \   -H 'accept: application/json' \   -H 'Content-Type: application/json' \   -d '{     "contenu": "New question",     "subject": "BD",     "use": "Test de positionnement",     "correct": "A",     "responseA": "A",     "responseB": "B",     "responseC": "C",     "responseD": "string",     "remark": "string"   }'</pre> <p>Request URL</p> <p>http://localhost:8000/add_question</p> <p>Server response</p> <table><tr><th>Code</th><th>Details</th></tr><tr><td>200</td><td>Response body<pre>{   "contenu": "New question",   "subject": "BD",   "use": "Test de positionnement",   "correct": "A",   "responseA": "A",   "responseB": "B",   "responseC": "C" }</pre></td></tr></table> <p>Response headers</p> <pre>content-length: 132 content-type: application/json date: Wed, 16 Mar 2022 21:02:41 GMT server: uvicorn</pre>	Code	Details	200	Response body <pre>{   "contenu": "New question",   "subject": "BD",   "use": "Test de positionnement",   "correct": "A",   "responseA": "A",   "responseB": "B",   "responseC": "C" }</pre>
Code	Details								
200	Response body <pre>{   "username": "admin" }</pre>								
Code	Details								
200	Response body <pre>{   "contenu": "New question",   "subject": "BD",   "use": "Test de positionnement",   "correct": "A",   "responseA": "A",   "responseB": "B",   "responseC": "C" }</pre>								

## Vérifier les points de terminaisons : Authorization & QCM

```
ubuntu@ip-172-31-6-211:~$ python3 main.py
making 10 requests on the '/Authorization' endpoint ...
took 0.028337860107421876 seconds
making 10 requests on the '/QCM' endpoint
took 0.03213164806365967 seconds
```