



## Documentation : Test automatisé sur API via postman et newman

### 1. Présentation des tests :

Dans le cadre de la refonte de l'outil *Logiweb*, la mise en place de tests est nécessaire afin de vérifier le bon fonctionnement des API développées.

Afin de vérifier leurs bons fonctionnements, les tests écrits peuvent être exécutés via 2 outils : *postman* et *newman*.

Pour ce projet de refonte, plusieurs ressources existent :

***Addresses | beneficiaries | customers | listings | orders | account/password | processes | products | tokens | services | users | view/beneficiaries.***

Les tests ont été réalisés pour les API suivantes :

- **[POST] Store a newly created resource in storage** : Permet de créer une ressource spécifique qui sera ensuite ajoutée à la base de données.
- **[GET] Display a listing of the resource** : Affiche un listing complet de l'ensemble éléments d'une ressource contenu dans la base de données (par exemple pour la ressource 'user', l'api va retourner l'ensemble des utilisateurs présents dans la base de données).
- **[GET] Display the specified resource** : Retourne un élément précis d'une ressource.
- **[PUT] Update the specified resource in storage** : Met à jour un élément précis de la ressource
- **[PATCH] Update the specified resource in storage** : Met à jour un élément précis de la ressource
- **[DELETE] Remove the specified resource from storage** : Supprime un élément précis de la ressource.

Ces API sont présentes dans toutes les ressources et sont adaptées en fonction de la ressource en question.

Il y a également des tests sur des API de connexion, déconnexion et mise à jour de mot de passe :

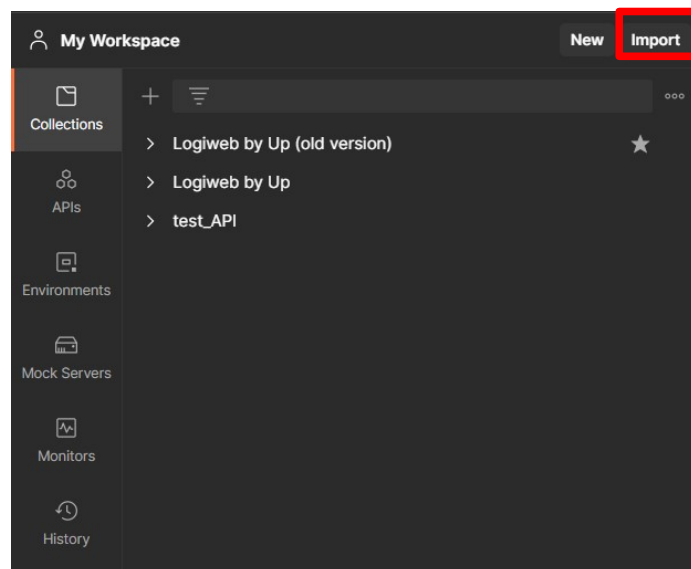
- **[POST] Sign in** : Permet de se connecter, et retourne un 'token d'identification' qui sera utilisé par la suite par l'ensemble des autres API.
- **[POST] Sign out** : Permet de se déconnecter
- **[POST] Request password change** : Envoie un lien afin de modifier son mot de passe.
- **[POST] Update password** : Modifie le mot de passe.

Pour chacune des API, les tests vérifient :

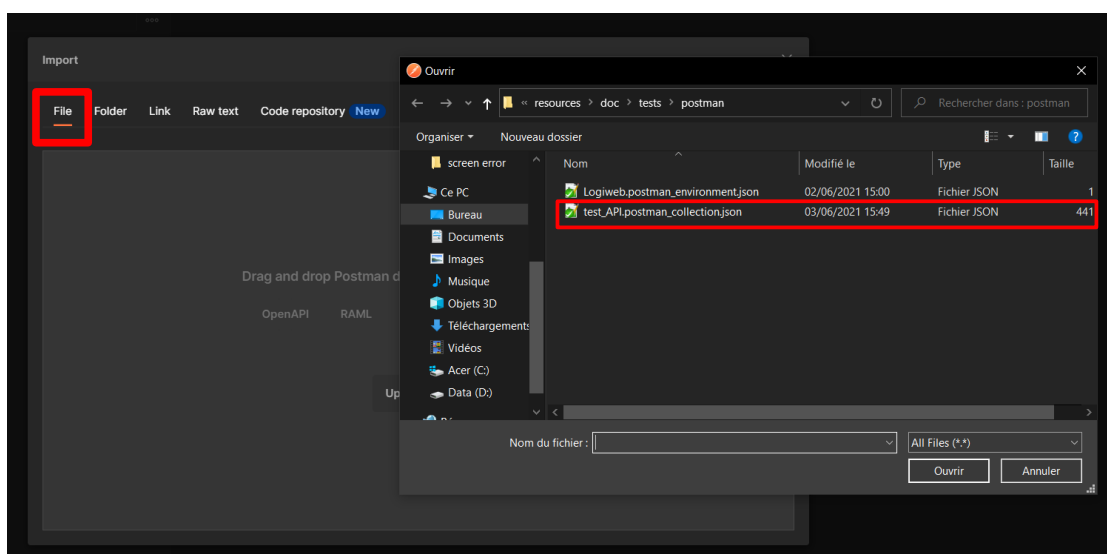
- Le code statut 200 → réussite de la requête
- Temps de réponse < 500 ms
- Le retour de la requête soit bien un JSON
- Success égale à 'true'
- Vérification au niveau du contenu du JSON qui est retourné (vérification de certains champs (le champ doit être non 'null' et non 'undefined').

## 2. Exécution via postman :

Pour exécuter le scénario de test via 'postman', il faut importer le fichier 'test\_API.postman\_collection.json' de la branche 'integration' qui se trouve dans le dossier 'ressources/doc/tests/postman'.



*Figure 1 – Import via postman*



*Figure 2 – Import fichier 'test\_API.postman\_collection.json'*

On obtient par la suite dans ‘My Workspace’ une collection qui contient l’ensemble des scénarios qui ont été développés. En cliquant sur la collection, un onglet s’ouvre, et il est possible de lancer le scénario de test en cliquant sur ‘run’.

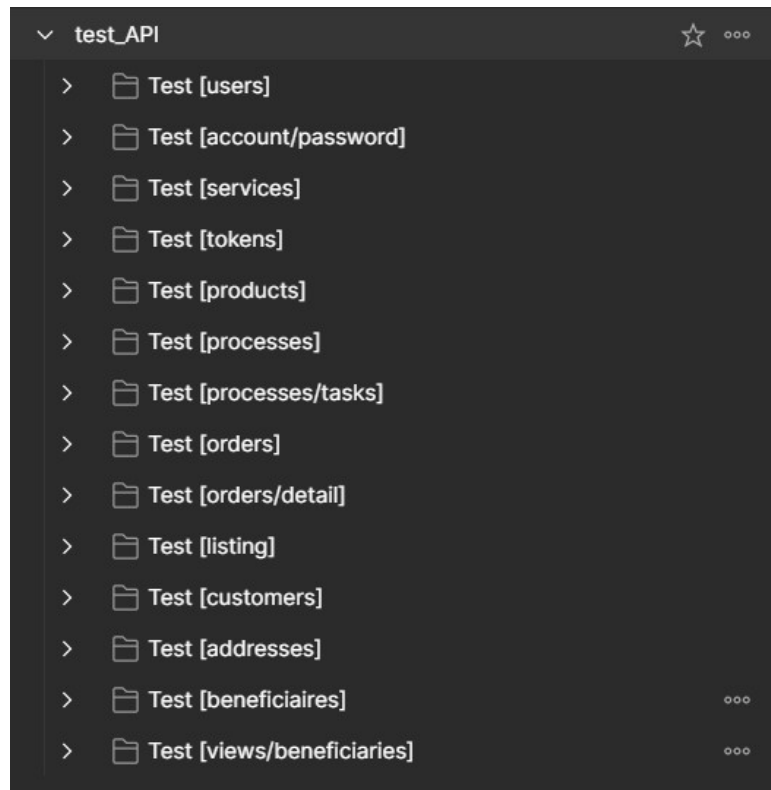


Figure 3 : Collection ‘test API’

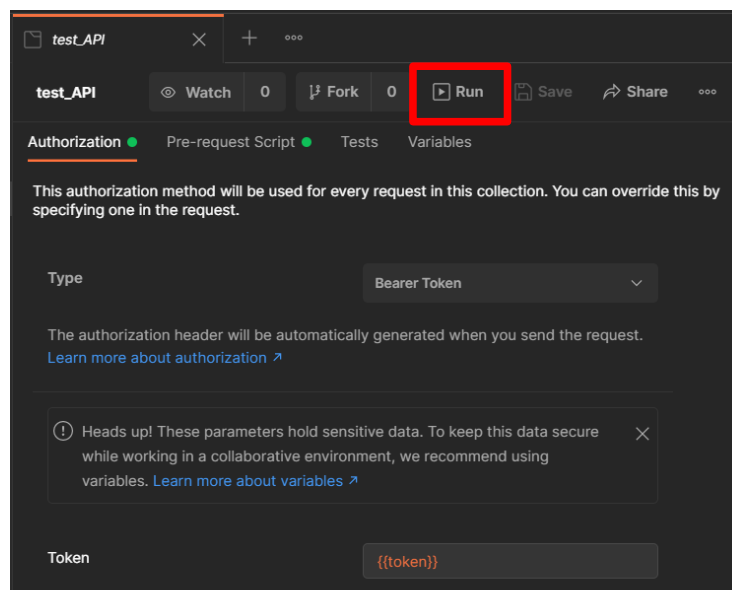
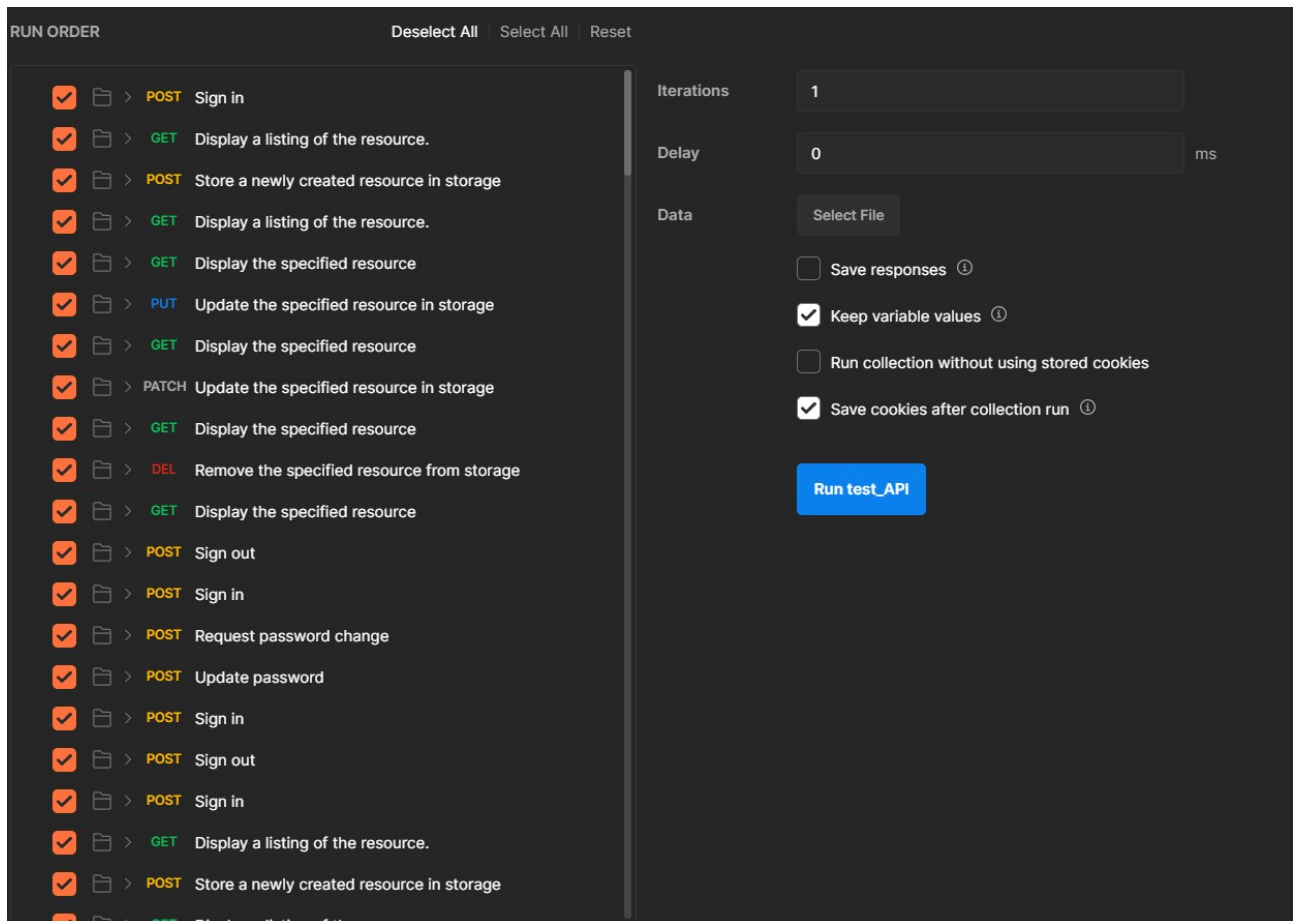
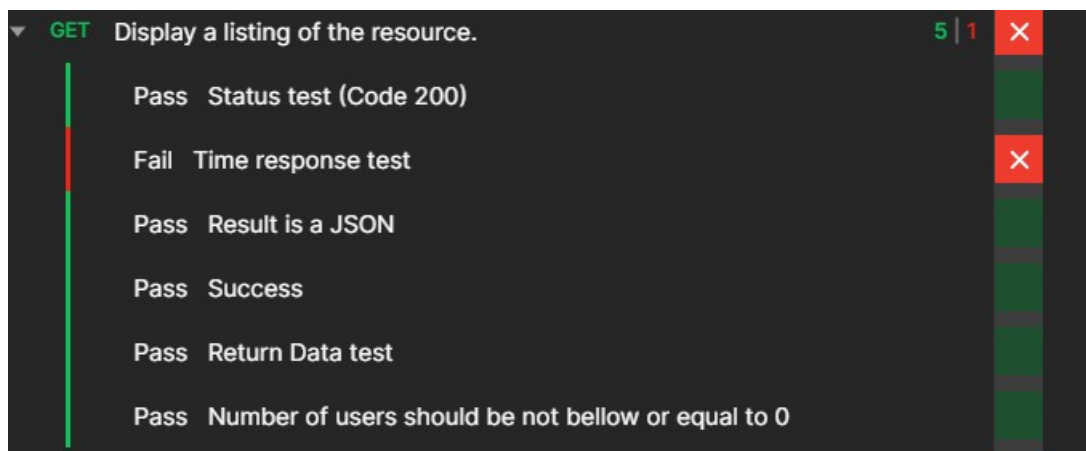


Figure 4 : Exécuter collection



*Figure 5 : Interface d'exécution des tests*

Après avoir lancé les tests sur la collection, on obtient un résultat pour chaque requête, et qui nous indique les tests qui ont été exécutés sans problème, et ceux qui ont échoué. De plus, pour conserver ces résultats, il est possible de les exporter sous la forme d'un fichier json.



*Figure 6 : Exemple de résultat (via postman)*

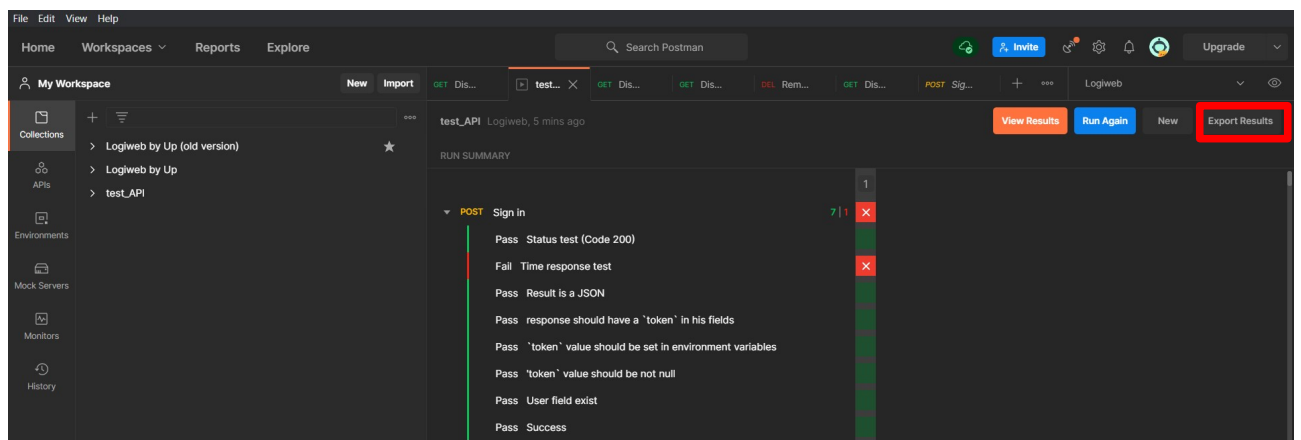


Figure 7 : Exportation des résultats

### 3. Exécution via en ligne de commande (newman) :

Pour exécuter la collection via newman, il faut déjà installer l'outil grâce à la commande "`npm install -g newman`" (via `cmd`). Si vous ne disposez pas de `npm`, il faut l'installer, sinon il existe une autre alternative qui est présentée sur le site de la documentation de `newman`.

Ensuite, il faut se rendre dans le dossier du projet, et plus précisément là où se trouve la collection ainsi que l'environnement de postman. ('logiweb\_v2\resources\doc\tests\postman').

```
03/06/2021 16:40 <DIR> .
03/06/2021 16:40 <DIR> ..
02/06/2021 15:00      325 Logiweb.postman_environment.json
03/06/2021 16:59      450 593 test_API.postman_collection.json
                2 fichier(s)      450 918 octets
                2 Rép(s)  42 239 479 808 octets libres

C:\Users\user\OneDrive\Bureau\up\logiweb_v2\resources\doc\tests\postman>
```

Figure 8 : Emplacement des fichiers

Il faut ensuite exécuter la commande : `newman run test_API.postman_collection.json -e Logiweb.postman_environment.json`. Après avoir exécuté la commande, on obtient les résultats aux tests pour chacune des requêtes. Il y a également un tableau qui récapitule le nombre de requêtes qui ont été exécutées, ainsi que les 'failure'. En dessous de ce tableau, il y a le détail des 'failure'.

```
Display a listing of the resource.
GET http://localhost:8000/orange/api/v1/users [200 OK, 1.69KB, 255ms]
✓ Status test (Code 200)
✓ Time response test
✓ Result is a JSON
✓ Success
✓ Return Date test
✓ Number of users should be not bellow or equal to 0

{
  'Total = 13'
  'Success = true'
}

Display the specified resource
GET http://localhost:8000/orange/api/v1/users/13 [200 OK, 382B, 255ms]
✓ Status test (Code 200)
✓ Time response test
✓ Result is a JSON
✓ Data field exist
✓ Name not null?
✓ Name not equal to undefined ?
✓ Email not null ?
✓ Email not equal to undefined ?
✓ Success

{
  'Id = 13'
  'Name = Juan T. Shamblin'
  'Email = juantshamblin240@up.coop'
  'Success = true'
}

Update the specified resource in storage
PUT http://localhost:8000/orange/api/v1/users/13 [200 OK, 379B, 607ms]
✓ Status test (Code 200)
4. Time response test
✓ Result is a JSON
✓ Data field exist
✓ Name not null?
✓ Name not equal to undefined ?
✓ Email not null ?
✓ Email not equal to undefined ?
✓ Success

{
  'Id = 13'
  'Name = Juan T. Shamblin'
  'Email = jtshamblin481@up.coop'
  'Success = true'
}
```

Figure 9 : Exemple de résultats des tests (via newman)

	executed	failed
iterations	1	0
requests	155	0
test-scripts	303	12
prerequest-scripts	183	0
assertions	1494	265
total run duration: 2m 26.5s		
total data received: 83.94KB (approx)		
average response time: 439ms [min: 163ms, max: 3.4s, s.d.: 371ms]		

# failure	detail
001. AssertionError	Time response test expected 827 to be below 500 at assertion:1 in test-script inside "Test [users] / Sign in"
002. AssertionError	Time response test expected 782 to be below 500 at assertion:1 in test-script inside "Test [users] / Display a listing of the resource."
003. AssertionError	Time response test expected 1256 to be below 500 at assertion:1 in test-script inside "Test [users] / Store a newly created resource in storage"
004. AssertionError	Time response test expected 607 to be below 500 at assertion:1 in test-script inside "Test [users] / Update the specified resource in storage"
005. AssertionError	Time response test expected 557 to be below 500 at assertion:1 in test-script inside "Test [users] / Update the specified resource in storage"
006. AssertionError	Time response test expected 517 to be below 500 at assertion:1 in test-script inside "Test [users] / Remove the specified resource from storage"
007. AssertionError	Time response test expected 2073 to be below 500

Figure 10 : Détails sur les ‘failure’

Il est également possible d'avoir un report en html des résultats en tapant la commande :  
`newman run test_API.postman_collection.json -e Logiweb.postman_environment.json -r htmlextra`

Mais il faut au préalable installer le reporter avec la commande :

```
- npm install -g newman-reporter-htmlextra
```

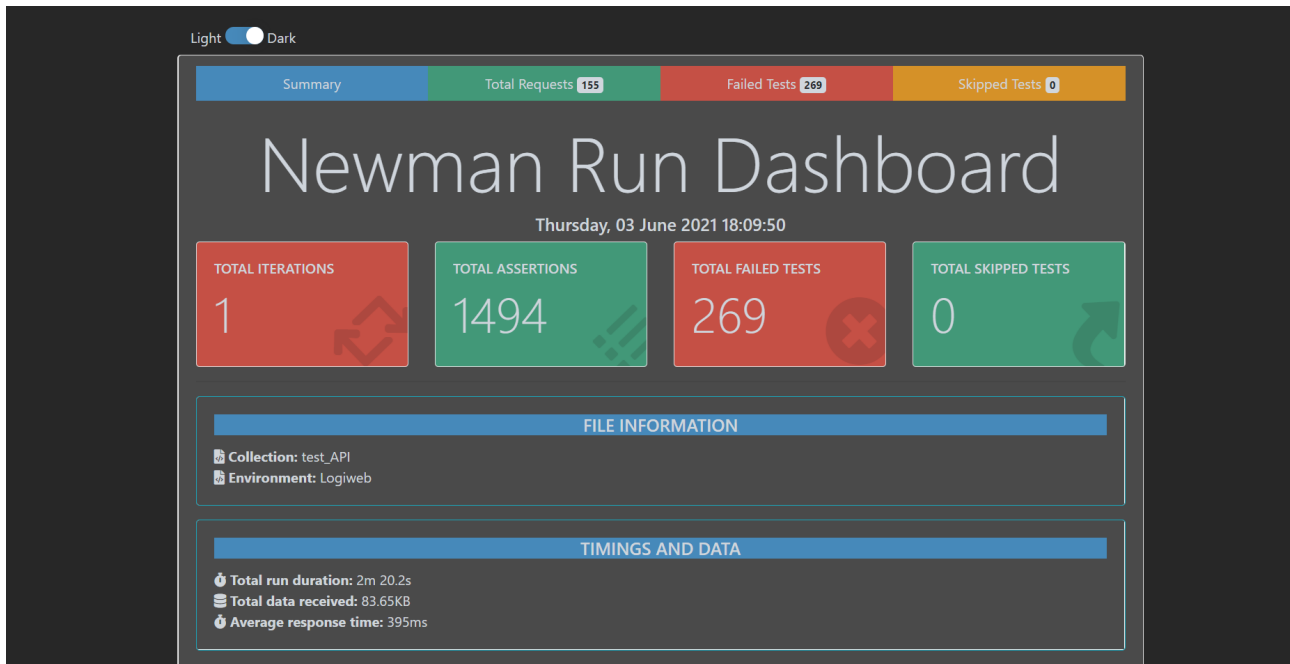


Figure 11 : Dashboard newman

Pour plus d'informations :

Documentation postman : <https://learning.postman.com/docs/getting-started/introduction/>

Documentation newman : <https://learning.postman.com/docs/running-collections/using-newman-cli/command-line-integration-with-newman/>

Documentation pour le reporter htmlextra (newman) :

- <https://github.com/DannyDainton/newman-reporter-htmlextra>
- <https://www.npmjs.com/package/newman-reporter-htmlextra>