

# Test technique Scibids

## Objectifs

On souhaite récupérer des documents versionnés dans une db à partir d'une api, puis les afficher sous une certaine forme dans une page web.

La db est en sqlite (document.db) et son schema se trouve dans le dossier sqlite\_schema de ce projet.

## Détails techniques

Un début d'api réalisé en python avec [flask-restfull](#) contient actuellement 1 route retournant tout les documents. Vous devrez lui rajouter 2 endpoints, qui retourneront les données au format json :

- 1 endpoint permettant de récupérer 1 document à partir de son id
- 1 endpoint permettant de récupérer les tags associés à un document

La page web devra appeler l'api, et à partir des infos retournées, comparer la version 1 et la version 2 du document dont le nom est "Scibids". (id 1 et 2 de la table document)

Nous avons plusieurs types de données:

- les Boolean, on souhaite afficher Yes/No
- des String ou des entiers, on souhaite afficher sans transformation
- une liste de String (les tags), on souhaite afficher la liste séparée par une virgule

Nous souhaitons mettre en valeur les modifications de valeur. Quand une valeur est nouvelle, nous voulons l'afficher en **gras**. Quand une valeur a été supprimée, nous voulons barrer la valeur. Dans le cas d'une liste, cette détection doit se faire niveau élément.

Pour cette partie, nous attendons :

- un code réalisé en javascript.
- le nom suivant pour le prototype de la fonction: function formatObject(old, new)

Vous êtes libre d'utiliser les librairies et framework de votre choix.

N'hésitez pas à refactorer le code existant.

## Exemple :

Pour la liste de fichiers ci-dessous

```
var old = {
  "name": "Scibids",
  "number_of_employees": 14,
  "is_serious": true,
  "tags": ["fun", "worker"]
};

var new = {
  "name": "Scibids",
  "number_of_employees": 15,
  "is_serious": true,
  "tags": ["fun", "huhu"]
};
```

Résultat attendu:

Name: Scibids

Number of employees: **15**, ~~14~~

Is serious: Yes

Tags: fun, ~~worker~~, **huhu**

## Installation de l'api

```
# crée l'env python
python3 -m venv env

# active l'env
source env/bin/activate

# installe les requirements
pip3 install -r requirements.txt
```

### **lancement de l'api**

```
python3 api.py
```

Pour accéder à la route "documents", aller sur "<http://127.0.0.1:7003/documents>"