

TP6

DJANGO (2 séances)

Il s'agit d'améliorer une interface web pour un code de simulation basé sur le modèle FitzHugh-Nagumo.

Pour information, ces deux équations forment le modèle de FitzHugh-Nagumo :

$$\frac{dV}{dt} = (\alpha - V)(V - 1)V - w \quad (1)$$

$$\frac{dw}{dt} = c(\beta V - \gamma w - \delta) \quad (2)$$

Elles simulent la valeur du potentiel électrique d'un neurone (V) en fonction du temps et des paramètres : α , δ , ϵ . w est une variable d'inactivation sans correspondance physique, nécessaire au calcul.

Actuellement cette interface permet de créer et de relancer des simulations qui sont accessibles via un système d'authentification, les utilisateurs et les simulations sont stockés dans une base de données.

Travail préparatoire :

Étape 1 :

- Récupérer depuis Moodle et dézipper les fichiers :
 - djangoTP.zip .
 - pyfhn.zip qui est une bibliothèque nécessaire.
- Sous PyCharm, ouvrir le projet djangoTP et ajouter les bibliothèques pandas, spacy et pyfhn avec python packages.
- Lancer le serveur Web : « python manage.py runserver ».
- Se rendre sur <http://127.0.0.1:8000/>. Si l'interface s'ouvre bien, vous pouvez passer à l'étape 2.

Étape 2 :

Pour pouvoir utiliser l'interface, il faut créer un superutilisateur qui sera sauvegardé dans la table « auth_user » de la base de données.

Pour cela, tapez la commande :

```
python manage.py createsuperuser --username=joe --email=joe@example.com
```

Dès lors, vous pouvez vous authentifier sur l'interface et faire des simulations.

1 – Utilisateurs :

1. Mettre en place le formulaire de création d'un utilisateur, il faudra vérifier que le compte n'existe pas.
2. Permettre à l'utilisateur de modifier son mot de passe.
3. Permettre à l'utilisateur de supprimer son compte.

Inspirez-vous de la doc : <https://docs.djangoproject.com/fr/3.2/topics/auth/default/>

2 – Simulations :

1. Permettre à un utilisateur de partager une simulation avec un autre utilisateur.
2. Ajouter un système de favoris sur les simulations. Les simulations favoris se placeront en haut de la liste.

Pour les deux tâches, il faut modifier « models.py » pour l'ajout et modification de classes. La synchronisation de la base de données avec les modèles s'effectue par l'intermédiaire des commandes :

- python manage.py makemigrations
- python manage.py migrate