

Alan BELLEC

Awen BOUILLIS

Néo LEBORGNE

DEF'IUT

Documentation générale d'installation et d'administration

Les commandes à exécuter seront marquées en **vertes**.

I - Installation locale

A - Installation des packages nécessaires au lancement de l'application

```
sudo apt install git npm -y
```

B - Clonage du répertoire

A partir de GitHub :

```
git clone https://github.com/Ne0re0/Def-IUT.git
```

```
cd Def-IUT
```

Si vous avez une archive .tar.gz :

```
tar -xzf <nom archive>.tar.gz
```

Si vous avez une archive .zip :

```
unzip <nom archive>.zip
```

C - Installation des librairies npm

```
./install-libraries
```

D - Réinitialisation de la base de données

```
./reset-database
```

/!\ N'exécutez cette commande que si vous souhaitez vraiment vider entièrement la base de donnée

E - Configurer le serveur SMTP

Pour que l'envoi de courriels puisse être effectif, il convient de compléter cette configuration en suivant les étapes ci-dessous :

```
mv conf/mail.yml.template conf/mail.yml
```

Ouvrez conf/mail.yml avec l'éditeur de votre choix et compléter les informations suivantes :

- smtpServer
- login
- password
- smtpPort

Sauvegardez et quittez.

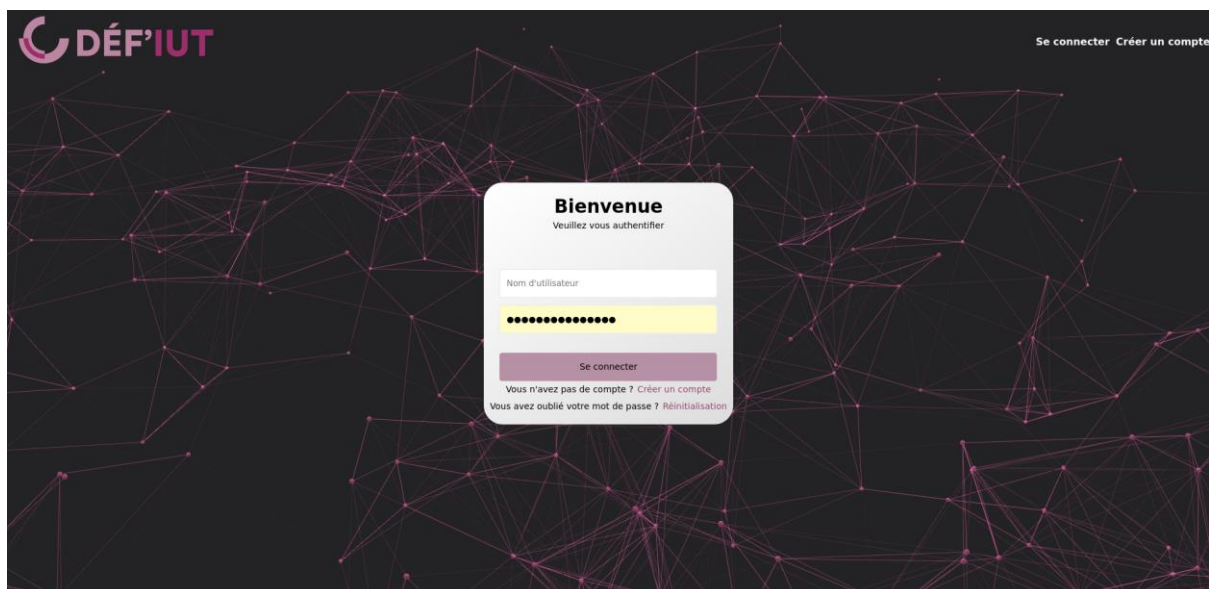
F - Lancement de l'application

```
./start
```

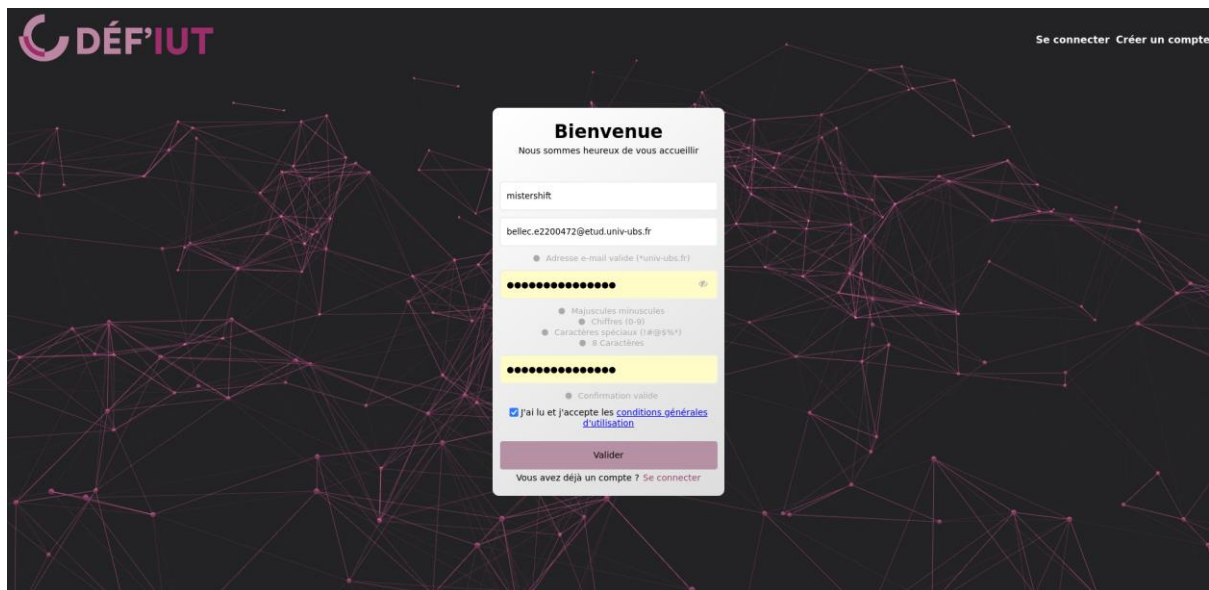
L'application est lancée sur le port 3000 : <http://localhost:3000>

G – Tester le fonctionnement

Connectez-vous à l'application via l'url <http://localhost:3000>.



Créez un compte.



Vérifiez le compte via le mail envoyé.

Si **aucun serveur SMTP** n'a été configuré préalablement, vous allez devoir mettre à jour l'état de vérification du compte manuellement :

```
cd src/express_webapp
```

```
sqlite3 database.db
```

```
UPDATE Users SET accountVerified=1 WHERE mail='<votre mail>';
```

Pour accéder à une documentation plus complète sur l'utilisation de l'application, veuillez utiliser [ce document](#).

II - Installation avec Docker

A - Installation des packages nécessaires au bon fonctionnement de la conteneurisation

```
sudo apt install docker-compose git -y
```

B - Clonage du répertoire

```
git clone https://github.com/Ne0re0/Def-IUT.git
```

```
cd Def-IUT
```

C - Configurer le serveur SMTP

Pour que l'envoi de courriels puisse être effectif, il convient de compléter cette configuration en suivant les étapes ci-dessous :

```
mv conf/mail.yml.template conf/mail.yml
```

Ouvrez conf/mail.yml avec l'éditeur de votre choix et compléter les informations suivantes :

- smtpServer
- login
- password
- smtpPort

Sauvegardez et quittez.

D - Lancement de la conteneurisation

Si vous n'avez pas installé docker auparavant, veuillez effectuer les commandes suivantes :

```
sudo apt-get install -y docker.io
```

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

```
Sudo apt-get install docker-compose
```

Afin de lancer la conteneurisation, effectuez la commande suivante :

```
sudo docker-compose up --build -d
```

*La première installation peut durer un certain moment en fonction de votre débit.

*L'application est lancée sur le port 3000 : <http://localhost:3000>

III - Accès distant

Pour accéder à l'intérieur du Docker, il suffit d'exécuter la commande suivante :

```
sudo docker exec -it defiut /bin/bash
```

IV - Démarrage et arrêt

Depuis le répertoire Def-IUT/

Démarrage :

```
sudo docker-compose up
```

Arrêt : `sudo docker-compose down`

V – Ajout, modification et publication des challenges

A – Publication des challenges

Comme demandée dans le cahier des charges, la gestion des challenges se fait depuis un accès terminal ainsi qu'à un fichier de configuration `conf/challenges.yml`.

Modifiez le fichier `/app/conf/challenges.yml` selon les règles de sa documentation (à l'intérieur du fichier)

Mise à jour des challenges (depuis la racine):

`./update-challenges`

B – Lancement des challenges

Afin de pouvoir lancer les challenges, il est nécessaire de construire les dockers et de les lancer. Pour ce faire faites :

`sudo apt-get install -y docker.io`

`sudo systemctl start docker`

`sudo systemctl enable docker`

`sudo apt-get install docker-compose`

`cd challenge/<categorie_du_challenge>/<nom_du_challenge>`

`sudo docker-compose up --build -d`

Le challenge en question est maintenant disponible sur l'application.

VI - Présentation de l'API

`/` liste les challenges disponibles

`/challenge/ID` affiche le challenge dont l'identifiant correspond à ID

`/user/ID` affiche le profil public de l'utilisateur dont l'identifiant correspond à l'ID

`/myprofile` affiche le profil privé de l'utilisateur et lui permet de modifier et supprimer son compte

`/scoreboard` affiche un tableau des scores

/recover permet à l'utilisateur de s'envoyer un mail de récupération de mot de passe avec un jeton de réinitialisation

/recover/JETON permet à l'utilisateur dont le jeton correspond à changer son mot de passe

/verify/JETON permet à l'utilisateur de vérifier son adresse e-mail, le jeton est récupéré via un mail

Autres ressources utiles

[Présentation de la base de données](#) : /src/database/

[Présentation de la gestion des challenges et des mails](#) : /conf/

[Présentation de la journalisation](#) : /log/

[Documentation utilisateur](#) : /documents/Doc_utilisateur.pdf

[Politique de confidentialité](#) : /documents/confidentialité.pdf

[Conditions générales d'utilisation](#) : /documents/utilisation.pdf