

Gestion des challenges

Dans ce dossier `conf/` se trouvent les fichiers :

- `challenges.yml` qui contient les challenges de l'application
- `update-db.sh` qui doit être exécuté pour mettre à jour la base de données
- `mail.yml` qui doit être complété avec les informations sur le relai SMTP
- `README.md` ce fichier

Toutes la gestion des challenges passe au travers du fichier `challenges.yml`

Modification de `challenges.yml`

Chaque challenge prends la forme suivante :

```
# Le titre du challenge
- title: "Premier challenge"
  # La catégorie du challenge (Programmation, Reverse, Web,...)
  # Accepte des chaines de caractères
  category: "Reverse"
  # La difficulté du challenge
  # Accepte des entiers entre 1 et 5 inclus.
  # 1 est la difficulté la plus basse
  # 5 est la difficulté la plus haute
  difficulty: 4
  # La description du challenge
  # Accepte une chaine de caractères
  description: "The best description"
  # Le flag de validation pour le challenge
  # Accepte une chaine de caractères
  flag: "p@$w0rd"
  # Cet attribut donne des informations sur la connexion au challenge
  # Accepte une chaine de caractères
  # Cet attribut peut être commenté s'il n'est pas nécessaire
  connection: "http://example.com/"
```

Pour ajouter un challenge il suffit de l'ajouter à la fin du fichier

```
- title: "Premier challenge"
  category: "Reverse"
  difficulty: 4
  description: "The best description"
```

```
flag: "p@$w0rd"
# connection: "http://example.com/"

- title: "Deuxième challenge"
  category: "Programmation"
  difficulty: 2
  description: "Ceci est une description"
  flag: "DEFIUT{flag_de_validation}"
  connection: "ssh carl@defiut.fr -p 2222"
```

A savoir

Lorsqu'un challenge de la base de données possédant le même titre qu'un challenge du fichier est trouvé :

- Si le challenge est identique, rien ne se passe
- Si certaines valeurs sont différentes, le script demande à l'utilisateur s'il souhaite mettre à jour la base de données.

Attention, si vous changez le titre d'un challenge, il sera considéré comme un nouveau challenge à part entière

Application des modifications

Si vous utilisez une instance Docker

```
sudo docker exec -it defiut /bin/bash # Si vous utilisez une instance docker
cd /app/ && ./update-db.sh
```

Sinon

```
cd /chemin/vers/Def-IUT/
./update-db.sh
```

PS : Il est possible d'insérer un challenge, de modifier un challenge existant mais il n'est pas possible de supprimer un challenge pour le moment.

Gestion des mails

Les fichiers `conf/recover.yml` ainsi que `conf/verify.yml` sont respectivement les fichiers de configuration liés à l'envoi de mail en cas de **perte de mot de passe** et à l'envoi de mail lié à la **vérification d'adresse email**.

Ils sont tous les deux identiques et composés de choses simples, veuillez vous référer à leurs documentations respectives pour plus de détails

Dans la partie `text`, l'apparition de `#link` est obligatoire, il sera remplacé par les liens à destination de l'utilisateur automatiquement

Exemple : `verify.yml`

Notez que la quantité de documentation de l'exemple est extrêmement limitée par rapport au fichier actuel.

```
# Le domain sur lequel est hébergé l'application
domain : "localhost"
# Le port qui est ouvert pour l'application
port : 3000 # 3000 par défaut
# Le titre du mail
subject :
  "Vérification de votre e-mail Déf'IUT"
# Le contenu du mail
# "#link" représente l'emplacement du lien à destination de l'utilisateur
text : "
Madame,Monsieur,\n
\n
Nous sommes ravis de pouvoir vous compter parmi nos membres !\n
\n
Vous pouvez dès à présent vérifier votre compte Déf'IUT en cliquant sur le
lien suivant : #link \n
\n
Cybèrement vôtre,\n
Le staff Déf'IUT
"
```

Exemple : `mail.yml`

```
# Ce fichier yaml est utilisé pour configurer un serveur SMTP en tant que
relais de courrier
# Par défaut, cette application utilise un serveur SMTP GMAIL

# Le nom de domaine complet du serveur SMTP
# Accepte une chaîne de caractères
# Requis
smtpServer: "smtp.exemple.com"

# Identifiant utilisé pour s'authentifier auprès du serveur SMTP
```

```
# Souvent une adresse e-mail
# Accepte une chaîne de caractères
# Requis
login: "exemple@exemple.com"

# Mot de passe utilisé pour s'authentifier auprès du serveur SMTP (clé API)
# Accepte une chaîne de caractères
# Requis
password: 'password_here'

# Port utilisé par le serveur SMTP (souvent 587 ou 25)
# Accepte un entier
# Requis
smtpPort: 587
```