

Procédure de mise en place de l'environnement de développement du plugin taiga-cas-auth

I Installez un conteneur lxd :

Pour pouvoir faire tourner le plugin et vérifier la procédure d'installation c'est intéressant d'avoir un environnement vierge avec Taiga installé [en mode dev](#) et /ou [en mode prod](#) pour cela, [installez un conteneur lxd](#) ubuntu 16,04 comme conseillé pour l'installation de Taiga, suite à cela installez la version de Taiga que avec lequel vous voudrait tester le module.

II Installez Taiga et le plugin

Créez un dossier nommé Taiga à l'emplacement puis [installer Taiga en mode développement](#) à l'intérieur de ce dossier. Il devrait maintenant y avoir deux répertoires : taiga-back et taiga-front . Suite à cela installez le plugin taiga-cas-auth dans ce dossier :

Taiga Back

Clone the repo and

```
cd taiga-contrib-cas-auth/back
workon taiga (if different name of taiga-virtualenv)
pip install -e .
pip install python-cas
```

Modify taiga-back/settings/local.py and include the lines:

```
INSTALLED_APPS += ["taiga_contrib_cas_auth"]

CAS_URL = "your CAS url"
# for settings
# if your CAS attributes are generic ('id', "full_name", "email", "bio") you don't need to define
them below
# though CAS_FIELD should at least be equal to {}
CAS_FIELD = {
    "id": "corresponding CAS attribute",
    "full_name": "corresponding CAS attribute",
    "email": "corresponding CAS attribute",
    "bio": "corresponding CAS attribute"
# when true allows Taiga to register new user from CAS
CAS_CREATE = True
# when true allows Taiga to synchronize data of Taiga account with CAS's account data
CAS_SYNC = True
# when true allows Taiga to bind existent taiga account to CAS account
CAS_BIND = True
```

Taiga Front

After clone the repo link dist in taiga-front plugins directory:

```
cd taiga-front-dist/dist
mkdir -p plugins
cd plugins
ln -s ../../taiga-contrib-cas-auth/front/dist cas-auth
```

Include in your dist/conf.json casUrl and in the 'contribPlugins' list the value "/plugins/cas-auth/cas-auth.json":

```
...
"casUrl": "YOUR-CAS-URL",
"contribPlugins": [
  (...)
  "/plugins/cas-auth/cas-auth.json"
]
...
```

for i18n:

```
cd taiga-front/v-...../locales
ln -s ../../plugins/cas-auth/locales/ cas-auth
```

III Installez Pycharm et créer le projet

Vous pouvez télécharger Pycharm [ici](#) . Installez ensuite sur votre poste.

Lancez le puis choisissez créer un nouveau projet, sélectionner pure python ou django en sélectionnant le dossier taiga créé précédemment .

Il faut ensuite changer le interpréteur python proposé par celui de l'environnement virtuel de Taiga. Pour cela cliquez sur l'engrenage à côté de 'Interpreter' et choisissez "Add local Python Interpreter". Ensuite sélectionner le fichier binaire python de l'environnement virtuel de Taiga.

Ensuite cliquer sur créer, il devrait vous demander si vous voulez créer un projet depuis des sources existantes, répondez par l'affirmative.

IV Créer Run and Debug Configurations

Cliquer sur le menu déroulant à gauche de le triangle vert (bouton run) puis sur 'edit configurations' puis créer une nouvelle configuration python en cliquant sur la croix verte et en choisissant python. Pour le script choisissez taiga-back/manage.py et ajouter en paramètre runserver. Ajouter comme variables d'environnement :no_proxy, http_proxy et https_proxy ainsi que PYTHONBUFFERED à 1 et REQUEST_CA_BUNDLE si besoin.

Assurez vous ensuite que 'Add content roots to PYTHONPATH' et 'Add source roots to PYTHONPATH' soient cochés.

Puis dans 'before launch' vous pouvez ajouter une gulp task en sélectionnant le gulpfile du plugin et la tâche build pour régénérer le répertoire dist du plugin avant de lancer le back.

De plus créer une configuration pour chaque fichier de test (test_services et tests_connector) et ajouter les dans 'before launch'. Ainsi vous pourrez facilement constater une régression des fonctionnalités si il y en a une.