



Nuit du HUB Challenge 1^{ère} édition

31 Octobre 2017

Améliorer l’affichage du planning des salles à EPITECH

I – Le sujet

Vous vous êtes toujours demandés à quoi servent les écrans devant la salle Sci-Fi et devant l'accueil ? Nous aussi.

C'est pourquoi nous vous proposons de développer une application graphique destinée à ces écrans. Nous allons aujourd'hui nous concentrer sur le bâtiment 2 (HUB, salle rétro, et bureau de coaching).

Les fonctionnalités demandées sont :

- Afficher le plan du bâtiment en y superposant des informations sur les activités en cours dans les salles.
- Afficher la date et l'heure
- Afficher les activités à venir

Toute fonctionnalité supplémentaire sera bienvenue (météo, fil Twitter / Facebook de l'école, animation à chaque passage à l'heure suivante, ...). Nous faisons appel à votre imagination.

II – Comment faire ?

Les explications sont destinées au langage Python. Cependant, vous êtes libres de développer dans le langage que vous voulez (même en JavaScript sur une page web).

Des exemples sont disponibles dans le dossier exemples du repo github dans lequel se trouve ce pdf.

a) Installer les modules nécessaires

```
$> wget https://bootstrap.pypa.io/get-pip.py
```

```
$> sudo python3 get-pip.py
```

```
$> sudo pip3 install pygame
```

```
$> sudo pip3 install requests
```

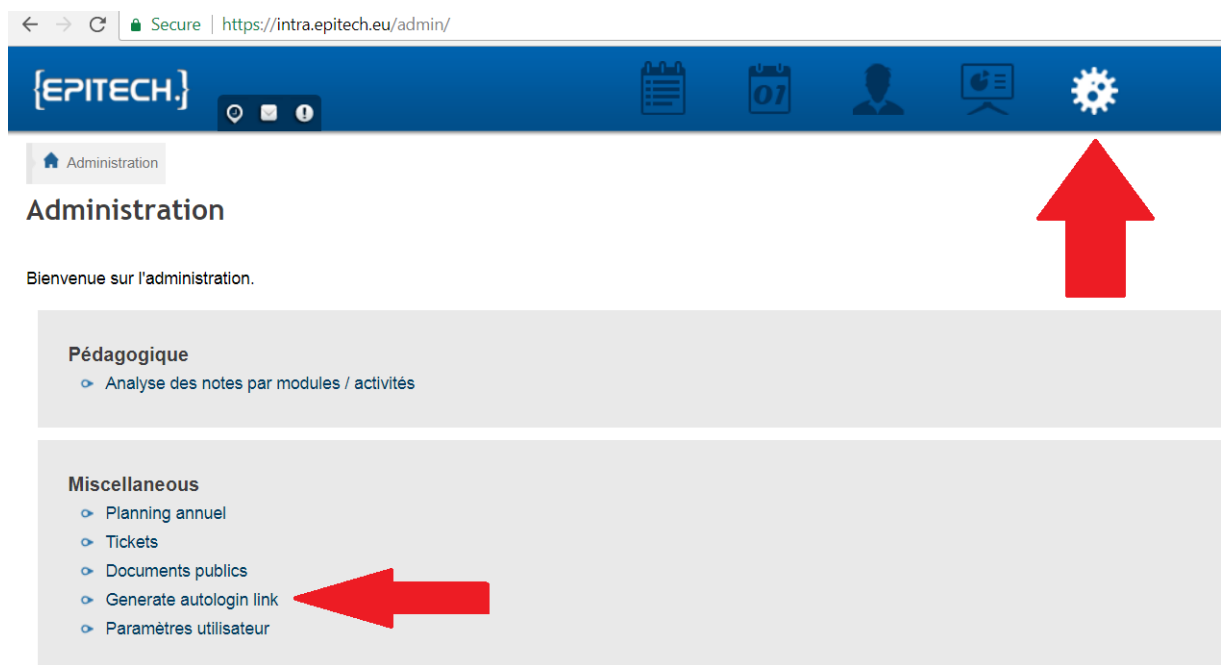
```
$> sudo pip3 install json
```

b) Récupérer les informations de l'intra

Les informations de l'intra sont disponibles via une API : une liste de requêtes qui renvoie des informations. Par exemple, une requête sur "<https://intra.epitech.eu/user?format=json>" renvoie les informations sur l'utilisateur connecté au format JSON.

Le module python « requests » permet de faire des requêtes http (voir l'exemple `epitechIntraApiTest.py`).

Pour pouvoir s'authentifier à l'intranet, il faut faire une requête sur son lien d'autologin. Vous pouvez le trouver dans l'onglet administration de votre intranet.



De nombreuses requêtes sont possibles. Pour les découvrir, l'onglet réseau de la console développeur de votre navigateur (F12) vous sera utile.

Le module python « json » permet de convertir le texte reçu en dictionnaire (tableau dont les index sont des chaînes de caractères).

c) Ouvrir une fenêtre, afficher des images et du texte

Pour l'aspect graphique, nous vous proposons d'utiliser le module pygame (voir pygameExemple.py).

Quelques fonctions utiles :

Initialiser pygame :

```
pygame.init()
pygame.font.init()
```

Ouvrir une fenêtre et afficher une image :

```
win = pygame.display.set_mode((x, y), pygame.RESIZABLE)
img = pygame.image.load(« image.png »).convert()
win.blit(img, (x_img, y_img))
pygame.display.flip() #rafraîchit la fenêtre
```

Afficher du texte :

```
font = pygame.font.SysFont('Arial', 30)
text = myfont.render('Salut ', False, (0, 0, 0))
win.blit(text, (x_text, y_text))
```

III – Modalités de rendu

Nom du repo git (blih) : hub-challenge-oct17

Droit de lecture à belugu_m

Fin du code à 6h42 (mercredi matin)

Pour les projets en Python, ceux-ci devront pouvoir être lancés comme ceci : ./hub_challenge.py <Lien autologin de l'intra>

Pour les projets dans un autre langage, vous devrez fournir un README.txt expliquant comment lancer votre projet.

Un classement des meilleurs projets sera publié avant jeudi soir. Le critère principal est le nombre et la pertinence des fonctionnalités, le design de l'ensemble ainsi que la propreté du code départageront les projets similaires. N'oubliez pas que le gagnant Tek1 recevra en cadeau un Raspberry Pi.