# Activité : gérer les données du formulaire de la page HTML

# Présentation

La page index.html contient un formulaire avec :

- comme action à exécuter la route demandée sur le serveur : /ajout
- la méthode d'envoi des données : POST
- un champ de saisie nommé : tache
- un bouton de soumission du formulaire

## Ajout d'un nouvelle route

• ajout d'une nouvelle route dans le fichier de l'application **app.py** :

```
@app.route('/ajout', methods=['POST'])
def ajout():
   vartache = request.form['tache']
   # traitement à faire pour mémoriser la nouvelle tâche ...
   return render_template('index.html', message = "Tâche ajoutée.")
```

#### Explications

- le décorateur @app.route définit une nouvelle vue ajout() pour la route '/ajout' en prenant en compte la méthode d'envoi POST des données du formulaire ;
- request.form['tache'] contient l'information saisie dans le champ du formulaire de la page HTML qui a pour nom tache. Il s'agit de la balise <input type="text" name="tache">
- l'instruction return permet de renvoyer comme réponse au navigateur la même page index.html mais en fournissant une variable pour personnaliser cette page.

L'objet **Request** représente la **requête HTTP envoyée** par le client et reçue par le serveur : le **chemin** de la page demandée au serveur, le **type** de la requête (**GET ou POST**), des informations sur le client, ainsi que les **données** transmises par le formulaire.

#### **ATTENTION** :

Mon application **ne doit pas avoir** deux routes identiques avec les mêmes méthodes HTTP, ou deux vues portant le même nom !

### Le fichier HTML modifié

L'affichage du contenu la variable message dans le template index.html se fait en utilsant %%et%% : <file html index.html><!DOCTYPE html> <html lang="fr"</li>

```
>
< 
head> <meta charset="UTF-8"

>
<title>Application Todo</title>
</head>
```

<

body> <h1>Todo liste</h1>

Ajouter une nouvelle tâche :

```
<form action="/ajout" method="POST">
<input type="text" name="tache">
<input type="submit"></div>
</div>
</div>
<div>{{message}}</div>
<div>
<input type="checkbox">Tâche 1
<input type="checkbox">Tâche 2
<input type="checkbox">Tâche 3
```

</body> </html> </file>

Le résultat après avoir saisi une nouvelle tâche :

### Traitement des cases à cocher

Retour à l'activité : Utiliser le framework Flask pour créer un site Web dynamique

• Activité : Utiliser le framework Flask pour créer un site Web dynamique

From: / - Les cours du BTS SIO

Permanent link: /doku.php/isn/flask\_todo02

Last update: 2019/12/10 17:46

