Activité : les bases des websockets avec Flask-socketio

Une application de base

app.py

```
from flask import Flask, render_template
from flask_socketio import SocketIO

app = Flask(__name__) # creation de l'application Web utilisant Flask
app.config['SECRET_KEY'] = 'secret!' #activation du chiffrement
socketio = SocketIO(app) # SockeyIO est associe a app

@app.route('/') # vue/page d'accueil du site Web
def index():
    return render_template('index.html')

@socketio.on('messagepourserveur')
def messagerecu(message, methods=['GET', 'POST']): # la variable message contient l'information
recue du client
    print('information reçue: '+ message)
    socketio.emit( 'messagepourclient', 'Bonjour cher client')
    print('information envoyee')

if __name__ == '__main__':
    socketio.run(app, debug=True) # valeur par défaut pour l'hôte (127.0.0.1) et le port (5000)
                                  # activation de debug pour visualiser les erreurs
```

Recevoir des messages dans l'application

```
@socketio.on('my event')
def handle_my_custom_event(json):
    print('received json: ' + str(json))
```

- Le message reçu peut être au format **string**, **bytes**, **int** ou **JSON**.

Envoyer des messages au client Web

```
from flask_socketio import send, emit

@socketio.on('message')
def handle_my_custom_event(message):
    emit('my response', message)
```

- **emit()** permet de donner un nom à la réponse ;
- sinon je peux utiliser **send(message)**.

Activité Les langages pour créer des sites Web ...

- Les langages pour créer des sites Web

From:
/ - **Les cours du BTS SIO**

Permanent link:
/doku.php/isn/websocketflask_02?rev=1584024630

Last update: **2020/03/12 15:50**