

Activité : première application websocket avec Flask-socketio

Les dossiers du projet

Je crée les dossiers de mon projet :

- la **gestion** du Websocket côté client se réalise en **javascript**, dans le fichier **websocket.js**.

Programme javascript de base du client : la page index.html

La page **index.html** contient le code HTML ainsi que le code javascript pour gérer les échanges côté client.

Dans la page Web HTML, le script **javascript** va créer un **client WebSocket** en utilisant la bibliothèque (API) javascript afin de **communiquer** avec le **serveur WebSocket** grâce au **protocole WebSocket**.

```
<script src="//cdnjs.cloudflare.com/ajax/libs/socket.io/2.2.0/socket.io.js" integrity="sha256-yr4fRk/GU1ehYJPAs8P4JLTgu0Hdsp4ZKrx8bDEDC3I=" crossorigin="anonymous"></script>
```

Création d'un objet WebSocket

L'instruction suivante permet d'ouvrir une connexion websocket vers le serveur en connaissant son adresse IP. Pour cette activité, le serveur sera situé sur le même ordinateur que le client : **l'adresse IP est donc 127.0.0.1** (localhost) :

```
var socket = io.connect('http://' + document.domain + ':' + location.port); // creation de la connexion
```

Explications :

- la **variable socket** va contenir la connexion vers le serveur,
- le **protocole** est **http** suivi de **l'adresse IP et du port d'écoute** du serveur websocket.

Recevoir des données du serveur

Quand un **message arrive** du serveur, un événement (**event**) **message** est envoyé à la fonction **onmessage()**. Pour utiliser ce message voici un exemple de code :

```
socket.on('messagepourclient', function( message ) {  
  alert(message);  
  information.innerHTML = message;  
  console.log('recu');  
});
```

Envoyer des données au serveur

Les messages sont **envoyés** avec la fonction **send()**. Cependant, les connexions étant asynchrones, l'envoi du **premier message** immédiatement après la création de la connexion **peut échouer**. Il est alors préférable ...

```
socket.on('connect', function() {  
  socket.emit('messagepourserveur', 'je souhaite me connecter !');  
  console.log('envoi');  
});
```

Le code complet de ce premier exemple dans le fichier javascript websocket.js

websocket.js

```
var information = document.getElementById('information');
```

```

    information.innerHTML = 'test2';
    var socket = io.connect('http://' + document.domain + ':' + location.port); // creation de la
    connexion
    //var socket = io();
    socket.on('connect', function() {
        socket.emit('messagepourserveur', 'je souhaite me connecter !');
        console.log('envoi');
    });

    socket.on('messagepourclient', function( message ) {
        alert(message);
        information.innerHTML = message;
        console.log('recu');
    });
    // Ajout d'un gestionnaire d'événement sur le bouton
    var envoi = document.getElementById('envoi')
    envoi.addEventListener('click',envoyerMessage);
    var numeroMessage = 1;
    // fonction qui envoie un message au serveur
    function envoyerMessage() {
        socket.emit('messagepourserveur', "J'envoie message " + numeroMessage + " au serveur.");
        console.log('envoi');
        numeroMessage += 1;
    }

    // fonction qui envoie un message au serveur
    function envoyerMessage() {
        ws.send("J'envoie message au serveur.");
    }

```

La page index.html qui utilise le fichier websocket.js

Voici le code HTML complet de ce premier exemple. Une balise `%%%%` est utilisé pour visualiser la réponse du serveur :

[index.html](#)

```

<!DOCTYPE html>
<html>
<head>
  <title>Socket</title>
</head>
<body>
  <p>Gérer un socket</p>
  <div id="information"></div>
  <input type="button" name="envoi" id="envoi" value="Envoyer message"/>

  <script src="//cdnjs.cloudflare.com/ajax/libs/socket.io/2.2.0/socket.io.js" integrity="sha256-
  yr4fRk/GU1ehYJPAs8P4JLTgu0Hdsp4ZKrx8bDEDC3I=" crossorigin="anonymous"></script>
  <script type="text/javascript" src="websocket.js" charset="utf-8"></script>
</body>
</html>

```

L'application sur le serveur

[app.py](#)

```

from flask import Flask, render_template
from flask_socketio import SocketIO

app = Flask(__name__)
app.config['SECRET_KEY'] = 'secret!'
socketio = SocketIO(app)

@app.route('/')

```

```
def index():  
    """Fournir la page index.html"""  
    return render_template('index.html')  
  
if __name__ == '__main__':  
    socketio.run(app, debug=True)
```

Les messages **textuels** échangés lors d'une connexion WebSocket sont au **format UTF-8**.

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_03](#)

Last update: **2020/03/12 16:04**

