|  |  |
| --- | --- |
| C:\Users\HP\Downloads\logo-esprit (1).png | Année Universitaire : 2022-2023  **Javascript côté serveur** |
| **Atelier :**  **Socket.io** | |

# Objectifs

* Installer socket.io
* Utilisation de socket.io
* Créer un système de notification
* Créer une application Chat

# Installation de socket.io

1. Créer un nouveau projet dans un dossier « atelier6 ».

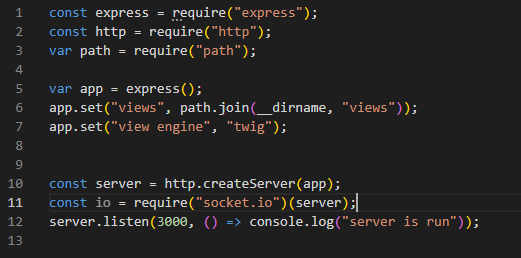
**npm init**

1. Installer les packages socket.io et express.

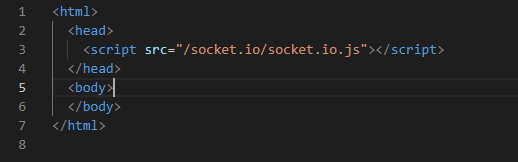
**npm install express socket.io --save**

# Utilisation de socket.io

1. Dans le fichier (app.js), ajouter le code serveur suivant.



1. Dans le fichier (chat.twig), ajouter le code client suivant.



1. Lancer un évènement par défaut lorsque l’utilisateur se connecte sur un socket.

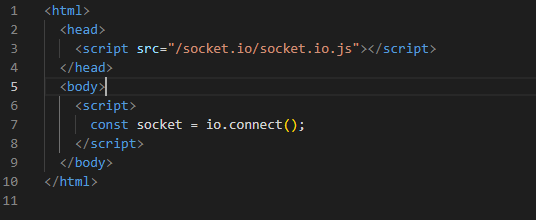
* Ajouter la ligne suivante dans le fichier (app.js) :

**io.on('connection', function(socket) {**

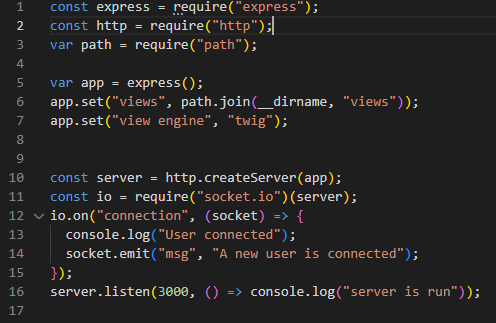
**console.log ('User Connected..');**

**});**

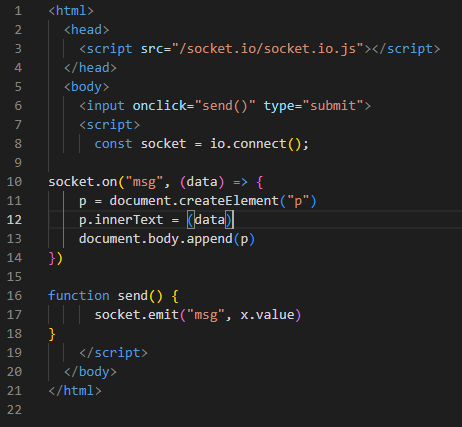
* Ajouter cette ligne dans un script dans la page chat.twig :



1. Créer un évènement qui envoie un message à un client dès qu'il se connecte



1. Dans la page twig, il faut écouter l’évènement « msg » et récupérer son contenu.



1. **Travail à faire :**

* Créer une Application de chat :
  + Afficher une notification lorsqu’un nouvel utilisateur se connecte.
  + Afficher une notification lorsqu’un utilisateur se déconnecte.
  + Afficher une notification lorsqu’un utilisateur est en train d’écrire un message à l’aide de l'événement "**onkeyup**" de JavaScript.
  + Chaque utilisateur doit avoir un pseudo.
  + Ajouter les messages dans votre base de données MongoDB dans l’entité « Chat » avec la date d’envoi du message.
  + Ajouter la fonctionnalité “notification” pour chaque fonction de CRUD.

Le lien ci-dessous explique comment faire un chat simple:

<http://socket.io/get-started/chat/>