

2016

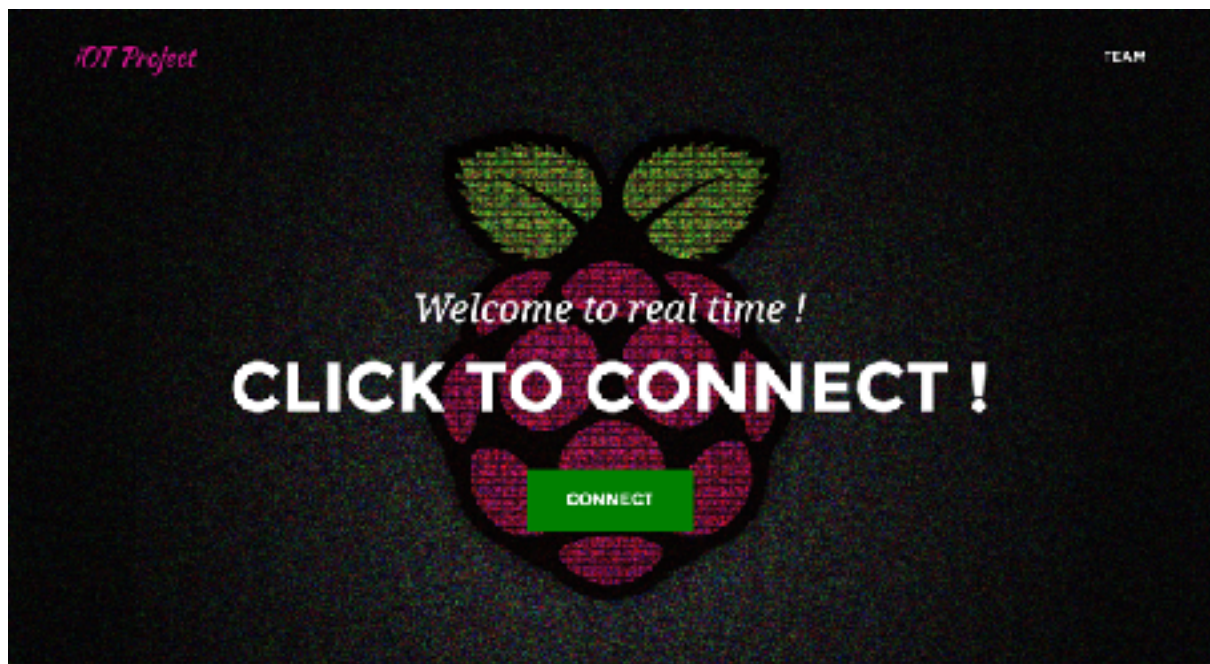
Projet IOT

INGESUP MASTERE 1

KARINE OULDBRAHAM - JANANY PERINPARAJAH - BEN NIKIEMA

Dans ce projet IOT nous avons dû de récupérer les données en temps réel du CPU, de la mémoire et des processus. Pour ce faire, nous avons choisi deux langages différents pour le client et le serveur Linux se trouvant sur une raspberry. L'interface client est en **Node.JS** et il possède un serveur propre à lui. (Node.JS est un projet open-source se basant sur le moteur « V8 » de Chrome. Le Framework que l'on a utilisé est Express.

Appuyer sur le bouton **CONNECT** afin que ce serveur client demande les renseignements sur l'adresse IP et le port afin d'envoyer une connexion socket au serveur Raspberry qui lui-même attend une connexion socket. Le serveur Raspberry est sur un système Linux et le code qui permet de compiler est en **C**.

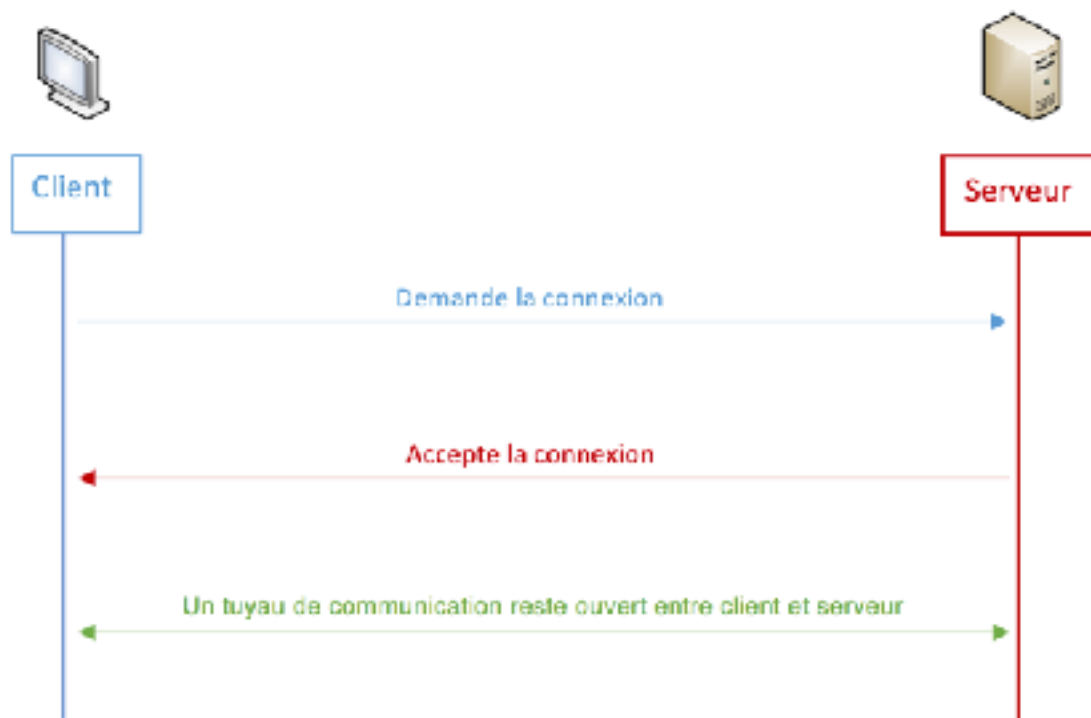


La bibliothèque utilisé est **Socket.IO** cela nous permet de faire la liaison en temps réel avec le serveur afin d'appeler ses propres fonctions, de même pour le serveur qui peut appeler des fonctions propres au côté client. Cette liaison se fait par le biais d'événements que l'on crée et que l'on appelle. Une fois la connexion faite le serveur Raspberry envoie en flux continu les infos sur *CPU*, *Memory* et *Processus*.

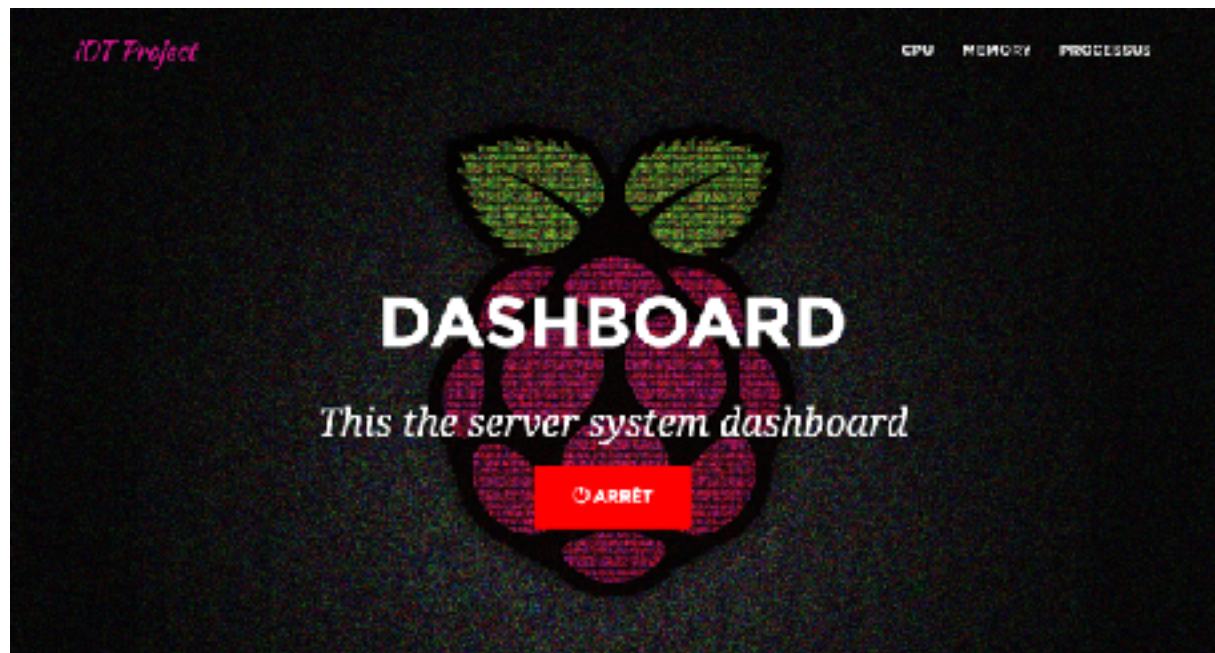


Le module **net** de Socket nous permet de connecter le serveur et le client. Ce module fournit une enveloppe de réseau asynchrone et il peut être importé en utilisant la syntaxe suivante.

`<net.Socket>` L'objet de connexion émet quand une nouvelle connexion est établie. Socket est une instance de `net.Socket`.



Si le client veut arrêter l'envoi de flux de donnée, une commande **ARRET** le permettra grâce à un bouton créer sur l'interface client.

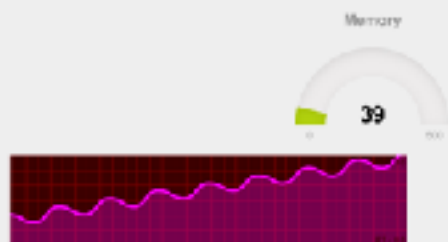


Le serveur Raspberry ne garde pas les informations en mémoire, il fait une boucle qui écrase les anciennes données. Mais ces données sont stockées sur l'interface clients étant donné que cette interface possède plus de mémoire que le raspberry.

CPU DASHBORAD



MEMORY DASHBORAD



PROCESSUS DASHBORAD

