

# Web Socket : INTRO



# HTTP

Généralement pour communiquer entre un client et un serveur on utilise le protocole HTTP. Avec ce protocole, le serveur n'envoie pas d'informations sans avoir reçu une requête du client. Ainsi, le client et le serveur communiquent l'un après l'autre ce qui rend son utilisation limitée lorsqu'on a besoin d'une communication en temps réel. Et là on ne peut penser qu'aux **Web sockets**

# Web Socket

Web socket est une technologie qui permet d'ouvrir de façon permanente un canal de communication bidirectionnel entre un client et un serveur. Cette technologie repose sur le protocole TCP qui s'occupe du transport des données.

# Fonctionnement

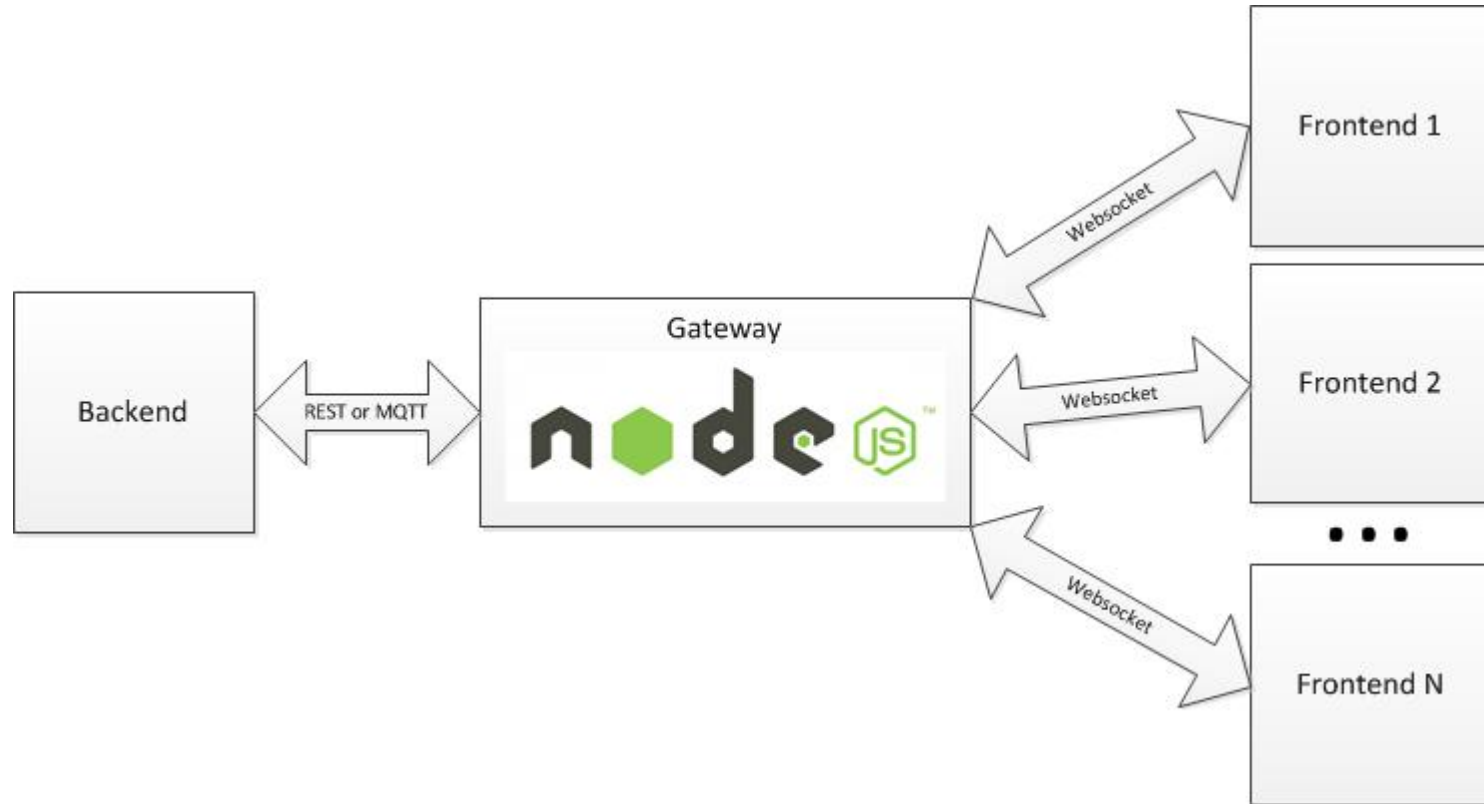
Le client commence par envoyer une requête classique, comme dans le cadre du protocole HTTP, mais dans ce cas, le processus de connexion s'effectue via une connexion TCP permanente.

# Avec quel langage ?

Voici quelques exemples de bibliothèques:

- Node.js : Socket.IO / WebSocket-Node
- Java : Jetty
- Ruby : EventMachine
- Python : pyWebSocket / Tornado
- C++ : libWebSockets
- .NET : SuperWebSocket

# Exemple avec nodejs



# Exemple avec nodejs

## Installation

- Node.js: <https://nodejs.org/en>
- Express: `npm install express -save`
- Scket: `npm install socket.io --save`

# Exemple avec nodejs (index.html)

```
<!Doctype html>
<html>
  <head>
    <title>Socket.IO Chat</title>
  </head>
  <body>
    <div>
      <div class="zone_saisie">
        <input id="m" /> <button onclick="send()">Envoyer</button>
      </div>
      <ul id="messages"></ul>
    </div>
    <!-- http://localhost:3000/socket.io/socket.io.js -->
    <script src='/socket.io/socket.io.js'></script>
    <script>
      var socket = io();
      // Fonction permettant d'envoyer un msg au serveur associé à un clique
      var send = function () {
        var text = document.getElementById('m').value;
        // L'action d'envoi du msg au serveur
        socket.emit('chat message', text);
      }
      var receive = function(msg) {
        var li = document.createElement('li');
        li.innerText = msg;
        document.getElementById('messages').appendChild(li);
      }
      // Lorsqu'un "chat msg" aies lieu, la fonction receive se déclanche
      socket.on('chat message', receive);
    </script>

  </body>
</html>
```



# Exemple avec nodejs (server.js)

```
var app = require('express')();
var http = require('http').Server(app);
var io = require('socket.io')(http);

app.get("/", function(req, res){
  res.sendFile(__dirname + '/index.html');
})

// Lors d'une connection d'un client au serveur (écouteur)
io.on('connection', function(socket){
  // Lors d'un msg envoyé d'un client (écouteur)
  socket.on('chat message', function (msg){
    console.log('message reçu : ' + msg);
    // Envoi des msg à tous les clients
    io.emit('chat message', msg);
  })
})

http.listen(3000, function(){
  console.log("Server running on 3000")
})
```




# Exemple avec nodejs (lancement du serveur)

PROBLEMS

OUTPUT

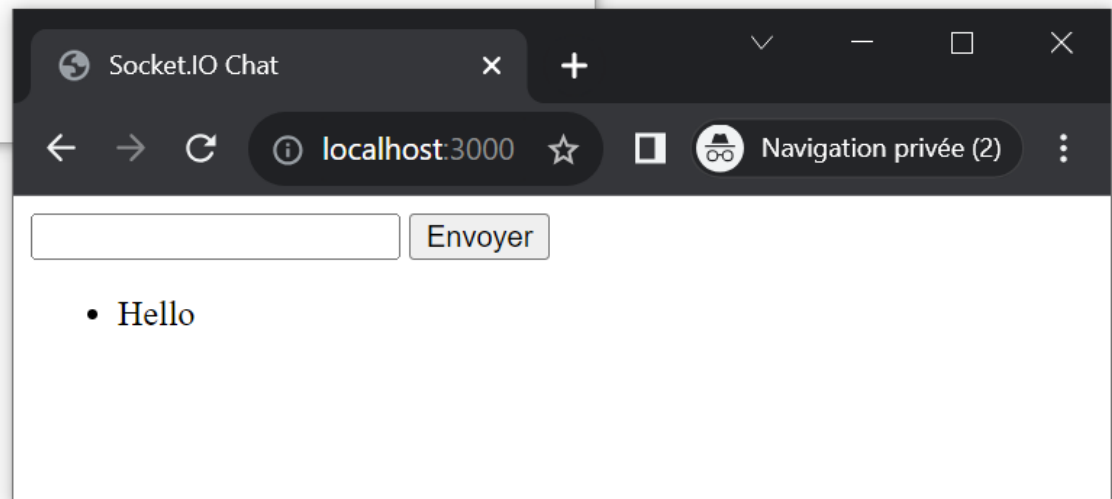
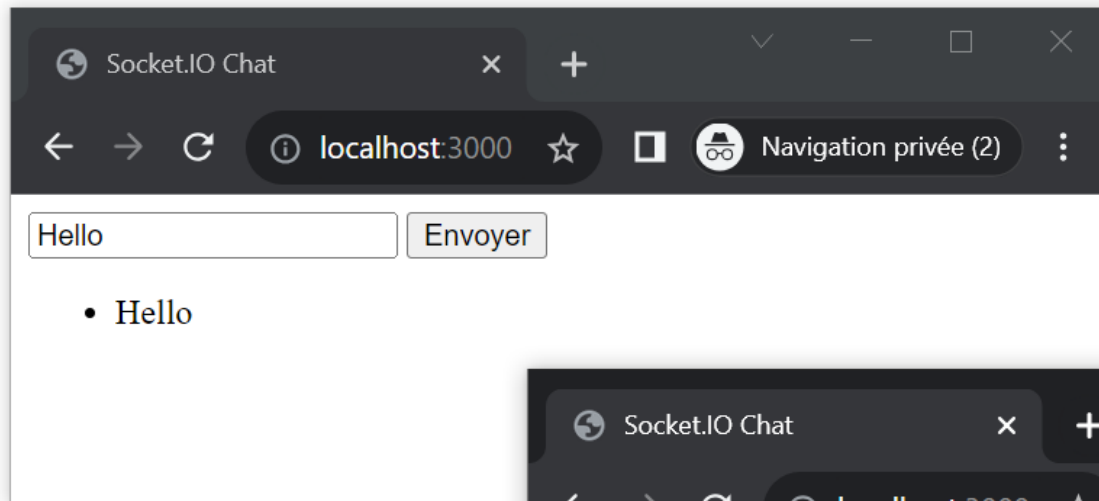
DEBUG CONSOLE

TERMINAL

 powershell + v   ... ^ X

```
PS C:\users\user\my-project> node server.js  
Server running on 3000
```

# Exemple avec nodejs (Lancer de connexion)



# Conclusion

Le protocole WebSocket est une technologie qui s'inscrit dans le cadre du développement du HTML5. C'est pour rendre le Web plus rapide, et pour pallier aux besoins auxquels le protocole HTTP ne pouvait pas répondre ! La communication en temps réel !