

 eBook Gratuit

# APPRENEZ

---

# WebSockets

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#websocket

# Table des matières

<b>À propos</b> .....	<b>1</b>
<b>Chapitre 1: Démarrer avec WebSockets</b> .....	<b>2</b>
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Web-socket Exemple client.....	2
<b>Chapitre 2: Différences entre WebSockets et AJAX</b> .....	<b>4</b>
Introduction.....	4
Exemples.....	4
exemple obtenir une notification.....	4
<b>Crédits</b> .....	<b>6</b>

---

# À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [websockets](#)

It is an unofficial and free WebSockets ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official WebSockets.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapitre 1: Démarrer avec WebSockets

## Remarques

Cette section fournit une vue d'ensemble de ce qu'est le WebSocket et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets au sein de websocket, et établir un lien avec les sujets connexes. La documentation de websocket étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

## Exemples

### Installation ou configuration

Instructions détaillées sur la configuration ou l'installation du socket Web.

### Web-socket Exemple client

Voici un exemple de client de socket Web en javascript.

Il:

1. Se connecte à un serveur de démonstration en direct.
2. Envoie un message.
3. Recevoir des messages.
4. Se déconnecte après un intervalle.

```
var mySocket      = null;
var serverUrl     = 'wss://echo.websocket.org'; // wss: is ws: but using SSL.
var oWebSocket    = window.WebSocket || window.MozWebSocket;
if (oWebSocket) {
  mySocket = new oWebSocket (serverUrl);
  if (mySocket) {
    console.log (mySocket);
    mySocket.onopen      = onSocketOpen;
    mySocket.onclose     = onSocketClose;
    mySocket.onmessage   = onSocketMessage;
    mySocket.onerror     = onSocketError;

    setTimeout (closeSocket, 5000); // Be polite and free socket when done.
  }
}

function onSocketOpen (evt) {
  console.log ("Socket is now open.");
  mySocket.send ("Hello from my first live web socket!");
}

function onSocketClose (evt) {
  console.log ("Socket is now closed.");
}
```

```
}

function onSocketMessage (evt) {
    console.log ("Recieved from socket: ", evt.data);
}

function onSocketError (evt) {
    console.log ("Error with/from socket!");
    console.log (evt);
}

function closeSocket () {
    if (mySocket.readyState !== mySocket.CLOSED) {
        console.log ("Closing socket from our end (timer).");
        mySocket.close ();
    }
    else
        console.log ("Socket was already closed (timer).");
}
}
```

Si vous exécutez ce code dans une console, vous obtenez un résultat tel que:

```
WebSocket { url: "wss://echo.websocket.org/", readyState: 0, bufferedAmount: 0, onopen: null,
  onerror: null, onclose: null, extensions: "", protocol: "", onmessage: null,
  binaryType: "blob"
}
2 // <--- 2 was the socket ID in this instance.
Socket is now open.
Recieved from socket: Hello from my first live web socket!
Closing socket from our end (timer).
Socket is now closed.
```

Lire Démarrer avec WebSockets en ligne: <https://riptutorial.com/fr/websocket/topic/7958/demarrer-avec-websockets>

---

# Chapitre 2: Différences entre WebSockets et AJAX

## Introduction

Ajax utilise le protocole HTTP et peut envoyer des requêtes à l'aide des méthodes POST / GET du client vers le serveur.

WebSocket est lui-même un protocole de communication entre le client et le serveur, distinct du HTTP.

Dans Ajax, lorsque vous envoyez une demande, le serveur envoie une réponse pour cette demande et la connexion prend fin.

En utilisant WebSockets lorsque vous établissez une connexion avec le serveur, vous pouvez communiquer entre le client et le serveur autant que vous le souhaitez, tout en conservant la connexion.

## Exemples

### exemple obtenir une notification

Ajax:

1. Nouvelle requête :
  - Le client envoie la demande: Serveur, vous avez une nouvelle notification?
  - Le serveur envoie une réponse: Client, NO
  - connexion se termine
2. Nouvelle requête :
  - Le client envoie la demande: Serveur, vous avez une nouvelle notification?
  - Le serveur envoie une réponse: Client, NO
  - connexion se termine
3. Nouvelle requête :
  - Le client envoie la demande: Serveur, vous avez une nouvelle notification?
  - Le serveur envoie une réponse: Client, NO
  - connexion se termine
4. Nouvelle requête :
  - Le client envoie la demande: Serveur, vous avez une nouvelle notification?
  - Le serveur envoie une réponse: client, oui vous êtes ici
  - connexion se termine

vous voyez qu'il y a beaucoup de demandes inutiles!

socket web

- Connexion établie :
- le client dit: Serveur, vous avez une nouvelle notification?  
après xx fois passé
- Server dit: Oui, j'ai  
après xx fois passé
- Server dit: Oui, j'ai
- Le client dit: Serveur, envoyez un nouvel email si j'ai  
après xx fois passé
- Le serveur dit: maintenant vous avez un nouvel email

maintenant il est clair que, avec web-socket, nous n'aurons pas de demandes inutiles

Lire Différences entre WebSockets et AJAX en ligne:

<https://riptutorial.com/fr/websocket/topic/8139/differences-entre-websockets-et-ajax>

---

# Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec WebSockets	<a href="#">Brock Adams</a> , <a href="#">Community</a> , <a href="#">Farhad</a>
2	Différences entre WebSockets et AJAX	<a href="#">Farhad</a> , <a href="#">jalanb</a>