



Бесплатная электронная книга

УЧУСЬ

WebSockets

Free unaffiliated eBook created from
Stack Overflow contributors.

#websocket

.....	1
1: WebSockets	2
.....	2
Examples.....	2
.....	2
-.....	2
2: WebSockets AJAX	4
.....	4
Examples.....	4
.....	4
.....	6

Около

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

глава 1: Начало работы с WebSockets

замечания

В этом разделе представлен обзор того, что такое websocket, и почему разработчик может захотеть его использовать.

Следует также упомянуть любые крупные темы в websocket и ссылки на связанные темы. Поскольку документация для websocket является новой, возможно, вам придется создавать исходные версии этих связанных тем.

Examples

Установка или настройка

Подробные инструкции по настройке или установке веб-сокета.

Пример веб-сокета

Это пример клиента веб-сокета в javascript.

Это:

1. Подключается к демо-серверу.
2. Отправляет сообщение.
3. Получает сообщение (сообщения).
4. Отключается после интервала.

```
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).");
}
```

Если вы запустите этот код в консоли, вы получите результат:

```
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.
```

Прочитайте Начало работы с WebSockets онлайн:

<https://riptutorial.com/ru/websocket/topic/7958/начало-работы-с-websockets>

глава 2: Различия между WebSockets и AJAX

Вступление

Ajax использует протокол HTTP и может отправлять запросы с использованием методов POST / GET от клиента к серверу.

WebSocket сам по себе является протоколом для взаимодействия между клиентом и сервером, отличным от HTTP.

В Ajax при отправке запроса сервер отправляет ответ для этого запроса, а соединение заканчивается.

Используя WebSockets при установлении соединения с сервером, вы можете связываться между клиентом и сервером столько, сколько хотите, и оно поддерживает соединение.

Examples

пример получения уведомления

Ajax:

1. Новый запрос :

- Клиент отправляет запрос: Сервер, есть ли у вас новое уведомление?
- Сервер отправляет ответ: Клиент, НЕТ
- концы соединения

2. Новый запрос :

- Клиент отправляет запрос: Сервер, есть ли у вас новое уведомление?
- Сервер отправляет ответ: Клиент, НЕТ
- концы соединения

3. Новый запрос :

- Клиент отправляет запрос: Сервер, есть ли у вас новое уведомление?
- Сервер отправляет ответ: Клиент, НЕТ
- концы соединения

4. Новый запрос :

- Клиент отправляет запрос: Сервер, есть ли у вас новое уведомление?
- Сервер отправляет ответ: Клиент, да вот вы
- концы соединения

вы видите, что есть количество бесполезных запросов!

веб-сокеты

- Соединение установлено :
- клиент говорит: Сервер, у вас есть новое уведомление?

через xx время прошло

- Сервер говорит: Да, у меня есть

через xx время прошло

- Сервер говорит: Да, у меня есть
- Клиент говорит: Сервер, отправьте новое письмо, если у меня есть

через xx время прошло

- Сервер говорит: теперь у вас есть новый адрес электронной почты

теперь ясно, что с помощью веб-сокета у нас не будет бесполезных запросов

Прочитайте [Различия между WebSockets и AJAX онлайн:](#)

<https://riptutorial.com/ru/websocket/topic/8139/различия-между-websockets-и-ajax>

кредиты

S. No	Главы	Contributors
1	Начало работы с WebSockets	Brock Adams , Community , Farhad
2	Различия между WebSockets и AJAX	Farhad , jalanb