

# Desenvolvimento Web II

## Aula 09 - Server Sent Events - SSE

Prof. Fabricio Bizotto

Instituto Federal Catarinense  
*[fabricio.bizotto@ifc.edu.br](mailto:fabricio.bizotto@ifc.edu.br)*

Ciência da Computação  
30 de janeiro de 2024

## 1 Server Sent Events - SSE

## 2 Experimentos

## Server Sent Events - SSE

Uma requisição, uma resposta muito muito longa

# Server Sent Events - SSE

## Definição

- Funcionalidade disponível apenas no protocolo HTTP/1.1.
- SSE é uma API para criar *streams* de eventos a partir do servidor para o cliente.
- O cliente quer receber notificações do servidor em tempo real .
- SSE é uma alternativa para *WebSockets* para comunicação entre cliente e servidor.
- SSE é mais simples de implementar do que *WebSockets*.
- Continua sendo uma requisição HTTP, mas a resposta não é fechada.

# Server Sent Events - SSE

## Exemplo de uso

- O cliente faz uma requisição HTTP para o servidor.
- O servidor responde com um cabeçalho `Content-Type : text/event-stream`.
- O servidor envia dados para o cliente **em tempo real**.
- O cliente recebe os dados e pode processá-los.
- O servidor pode enviar dados para o cliente **a qualquer momento**.

# Server Sent Events - SSE

Ilustração

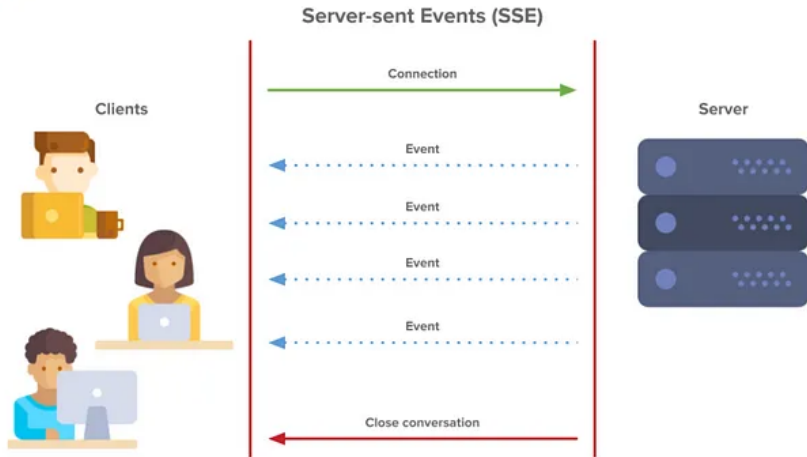


Figura: Ilustração de um evento SSE

### ■ Prós

- Simplicidade para real-time.
- **Padrão W3C**: é compatível com a maioria dos navegadores.
- **Conexão persistente**: o servidor pode enviar dados sem que o cliente faça uma nova requisição.

### ■ Contras

- **Não é bidirecional**. O cliente não pode enviar dados para o servidor.
- Cliente precisa estar **conectado** para receber os eventos.
- **Limite de conexões simultâneas** (6 conexões por domínio), ou seja, o cliente pode abrir no máximo 6 abas para o mesmo domínio.

Server Sent Events - SSE

## Exemplo prático



# Server Sent Events - SSE

Quando usar?

- **Notificações em tempo real:** por exemplo, notificações de novas mensagens em um chat.
- **Atualizações em tempo real:** por exemplo, atualizações de preços de ações.
- **Atualizações de dados:** por exemplo, atualizações de dados de sensores.
- **Atualizações de progresso:** por exemplo, progresso de um upload de arquivo.
- **Atualizações de eventos:** por exemplo, atualizações de eventos em um jogo.

# Server Sent Events - SSE

## Exemplo prático

exemplos > sse > JS index.js > ...

```
1  const app = require('express')();
2
3  app.get('/', (req, res) => res.send("Hello World!"));
4  app.get('/stream', (req, res) => {
5    res.setHeader('Content-Type', 'text/event-stream');
6    res.setHeader('Cache-Control', 'no-cache');
7    send(res);
8  });
9
10 const port = process.env.PORT || 8000;
11
12 let i = 0;
13 const send = (res) => {
14   res.write("data: "+`Hello World from Server Side -> [${i++}]\n\n`);
15   setTimeout(() => send(res), 1000);
16 }
17
18 app.listen(port, () => console.log(`Server running on port ${port}`));
19
20 // Para rodar o exemplo, execute o comando:
21 // node index.js
22 // Acesse o endereço http://localhost:8000/stream
23 // O servidor irá enviar mensagens a cada segundo.
24 // Tente abrir o endereço em mais de uma aba do navegador.
25 // ou
26 // curl -N http://localhost:8000/stream
```

Figura: Exemplo prático de uso de SSE

# Server Sent Events - SSE

## Exemplo prático

- Vamos ver um exemplo prático de uso de SSE.
- O exemplo será feito em Node.js para simular uma lista de chamadas dos alunos.
- O exemplo será disponibilizado em <https://github.com/fabricioifc/sse-votes>.

Escolha um dos exercícios a seguir para implementar um exemplo prático de uso de SSE:

- 1 **Exercício 1:** Contador de votos em tempo real.
- 2 **Exercício 2:** Chat em tempo real.
- 3 **Exercício 3:** Sistema de notificações em tempo real.
- 4 **Exercício 4:** Sistema de atualização de dados em tempo real.
- 5 **Exercício 5:** Contador de visitas em tempo real.
- 6 **Exercício 6:** Sistema leilão em tempo real.
- 7 **Exercício 7:** Sistema de lista de compras colaborativa em tempo real.
- 8 **Exercício 8:** Sistema de rastreamento de entregas em tempo real.
- 9 **Exercício 9:** Outros sistemas em tempo real.