

2024-06-01

WebSocket
└─ Background

└─ Background

Background

- Historically, creating web applications that need bidirectional communication between a client and a server has required an abuse of HTTP to poll the server for updates while sending upstream notifications as distinct HTTP calls.

bidirectional means the server and the client can send data to each other at any time

2024-06-01

WebSocket
└─ Background

└─ Background

Background

- Historically, creating **web applications** that need bidirectional communication between a client and a server has required an abuse of HTTP to poll the server for updates while sending upstream notifications as distinct HTTP calls.

bidirectional means the server and the client can send data to each other at any time

2024-06-01

WebSocket
└─ Background

└─ Background

Background

- Historically, creating **web applications** that need **bidirectional communication** between a client and a server has required an abuse of HTTP to poll the server for updates while sending upstream notifications as distinct HTTP calls.

bidirectional means the server and the client can send data to each other at any time

2024-06-01

WebSocket
└─ Background

└─ Background

Background

- Historically, creating **web applications** that need **bidirectional communication** between a **client** and a server has required an abuse of HTTP to poll the server for updates while sending upstream notifications as distinct HTTP calls.

bidirectional means the server and the client can send data to each other at any time

2024-06-01

WebSocket
└─ Background

└─ Background

Background

- Historically, creating **web applications** that need **bidirectional communication** between a **client** and a **server** has required an abuse of HTTP to poll the server for updates while sending upstream notifications as distinct HTTP calls.

bidirectional means the server and the client can send data to each other at any time

2024-06-01

WebSocket
└─ Background

└─ Background

Background

- Historically, creating **web applications** that need **bidirectional communication** between a **client** and a **server** has required an **abuse of HTTP** to **poll** the server for updates while sending upstream notifications as distinct HTTP calls.

bidirectional means the server and the client can send data to each other at any time

2024-06-01

WebSocket
└─ Background

└─ Background

Background

- Historically, creating **web applications** that need **bidirectional communication** between a **client** and a **server** has required an **abuse of HTTP** to **poll** the server for updates while sending upstream notifications as **distinct HTTP calls**.

bidirectional means the server and the client can send data to each other at any time

2024-06-01

WebSocket
└─ Background
 └─ HTTP polling
 └─ HTTP polling

HTTP polling

Check whether the server is changed in a while, thereby performing incremental updates.



A client can send data and ask for data at the same time. But, if client has no data and server has no data, a request and response will still be generated with all the HTTP headers and thus wasting resources. no real-time data because while the client waits, an event could occur and the client will know about it only when the timeout expires.

2024-06-01

WebSocket

Background

HTTP polling

HTTP polling

HTTP polling

Check whether the server is changed in a while, thereby performing incremental updates.

► How often to query?



A client can send data and ask for data at the same time. But, if client has no data and server has no data, a request and response will still be generated with all the HTTP headers and thus wasting resources. no real-time data because while the client waits, an event could occur and the client will know about it only when the timeout expires.

WebSocket

└ Background

└ HTTP polling

└ HTTP polling

HTTP polling

Check whether the server is changed in a while, thereby performing incremental updates.



- How often to query?
- Continuously short interval requests will be washed away the server.

A client can send data and ask for data at the same time. But, if client has no data and server has no data, a request and response will still be generated with all the HTTP headers and thus wasting resources. no real-time data because while the client waits, an event could occur and the client will know about it only when the timeout expires.

WebSocket

└ Background

└ HTTP polling

└ HTTP polling

HTTP polling

Check whether the server is changed in a while, thereby performing incremental updates.



- How often to query?
- Continuously **short interval** requests will be **washed away** the server.
- **Long interval** will require more time to reach the client, **no real-time data**.

A client can send data and ask for data at the same time. But, if client has no data and server has no data, a request and response will still be generated with all the HTTP headers and thus wasting resources. no real-time data because while the client waits, an event could occur and the client will know about it only when the timeout expires.

2024-06-01

WebSocket

└ Background

└ HTTP long polling

└ HTTP long polling

HTTP long polling

When a client sends a data request, the server will block the request until there is data transfer or timeout before returning.



No bidirectional because the client may only send data the first time but then it will only receive until a timeout and another request is made. In the normal polling we could have bidirectional because the interval was shorter.

2024-06-01

WebSocket

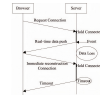
└ Background

└ HTTP long polling

└ HTTP long polling

HTTP long polling

When a client sends a data request, the server will block the request until there is data transfer or timeout before returning.



► Solve the short polling frequency to access the server.

No bidirectional because the client may only send data the first time but then it will only receive until a timeout and another request is made. In the normal polling we could have bidirectional because the interval was shorter.

WebSocket

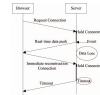
Background

- HTTP long polling

- HTTP long polling

HTTP long polling

When a client sends a data request, the server will block the request until there is data transfer or timeout before returning.



- ▶ Solve the short polling frequency to access the server.
- ▶ **No bidirectional communication, server push data.**

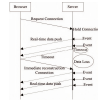
No bidirectional because the client may only send data the first time but then it will only receive until a timeout and another request is made. In the normal polling we could have bidirectional because the interval was shorter.

```

WebSocket
├── Background
│   └── Streaming
│       └── Streaming

```

Iframe embed a hidden frame in an HTML page, then set it as a long connection request, thus the server can send data to the clients constantly.



Because the server need to keep the connections alive.

2024-06-01

WebSocket

- Background
 - Streaming
 - Streaming

Streaming

Iframe embed a hidden frame in an HTML page, then set it as a long connection request, thus the server can send data to the clients constantly.

- It can sends multiple events from a single request.



Because the server need to keep the connections alive.

2024-06-01

WebSocket

- Background
- Streaming
- Streaming

Streaming

Iframe embed a hidden frame in an HTML page, then set it as a long connection request, thus the server can send data to the clients constantly.



► It can send multiple events from a single request.

► But, it increases the burden on the server, causing the server performance degradation, or even collapse.

Because the server need to keep the connections alive.

2024-06-01

WebSocket

- Background
- Streaming
- Streaming

Streaming

Iframe embed a hidden frame in an HTML page, then set it as a long connection request, thus the server can send data to the clients constantly.



- It can sends multiple events from a single request.
- But, it increase the burden on the server, causing the server performance degradation, or even collapse.
- No bidirectional communication.

Because the server need to keep the connections alive.