



Web Platform APIs

fetch: a modern AJAX API

fetch is a new API available in modern browsers that is more elegant than **XMLHttpRequest**.

```
fetch('https://swapi.info/api/people/1')  
  .then(response => response.json())  
  .then(data => console.log('SUCCESS', data))  
  .catch(error => console.log('ERROR:', error));
```

should first check if response is ok!

fetch is available in all modern browsers.

fetch is built on Promises.

alternative API using **Request**:

```
const request = new Request('https://swapi.info/api/people/1');  
fetch(request)  
  .then(response => response.json())  
  .then(data => console.log('SUCCESS', data))
```

Browser support:

<http://caniuse.com/#search=fetch>

<https://developer.mozilla.org/en-US/docs/Web/API/Window/fetch>

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

Same Origin Policy (SOP)

The same-origin policy restricts how a document or script loaded from one origin can interact with a resource from another origin.

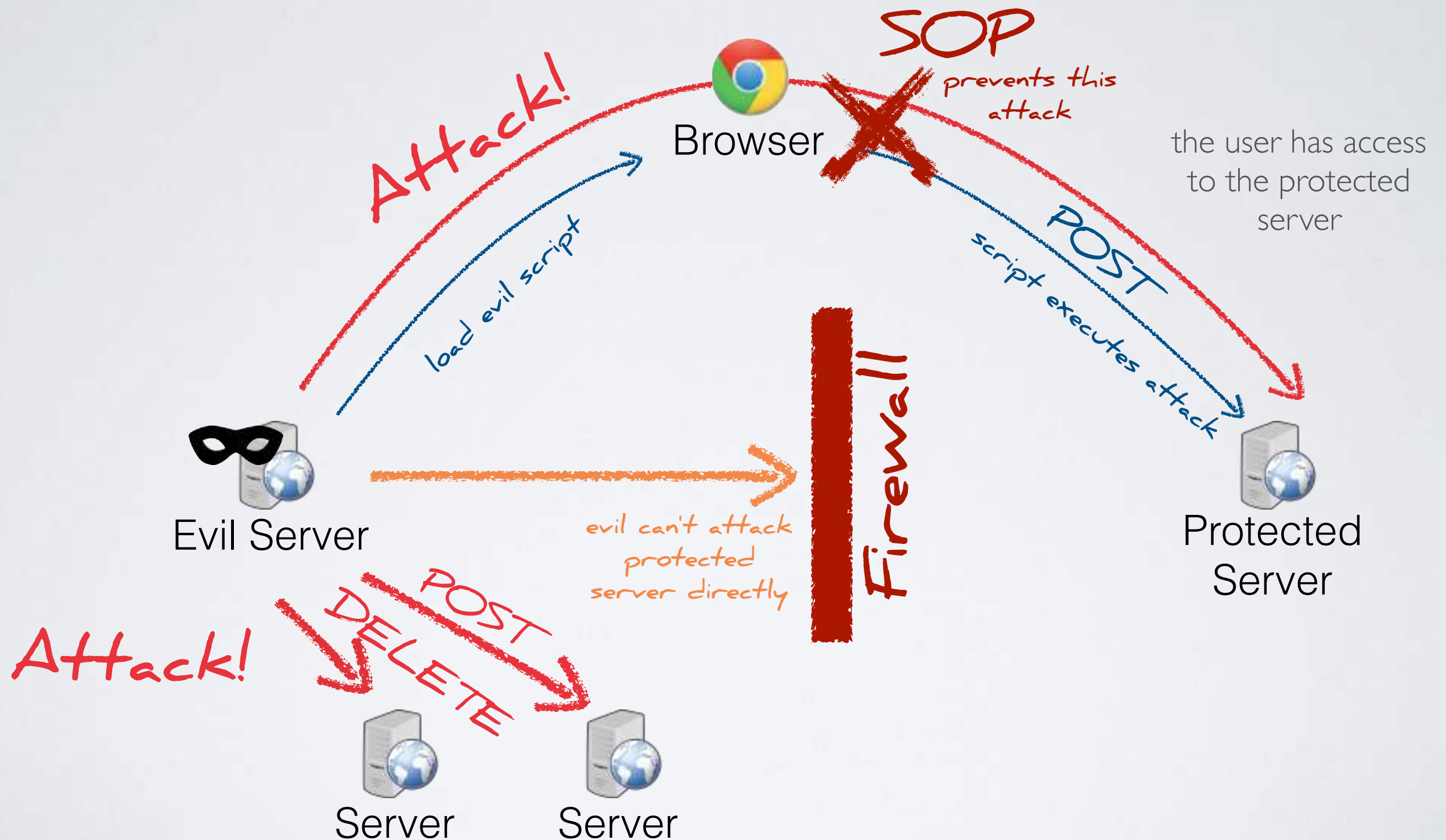
- A page can't access data on another page
- A script can only make AJAX calls to resources from the origin it was loaded from
- Example: A script from "example.com" can't read or write your emails from "gmail.com"

The SOP protects the client not the server!

The SOP prevents "distributed APIs"!

SOP

Same Origin Policy



Cross Origin Resource Sharing (CORS)

CORS allows to circumvent the SOP to implement "distributed APIs".

- A server API can define that its resources can be accessed from other origins.
- Browsers then allow scripts to make AJAX calls to that API.

CORS is "implemented" on the server but affects the behaviour of the browsers.

CORS only affects browser clients. Other clients (i.e. mobile) are not affected.

fetch vs. ky / axios ...

fetch is a modern built-in browser API:

```
const res = await fetch('https://swapi.co/api/people/1/');  
if (!res.ok) {  
  throw new Error('Network response was not ok');  
} else {  
  const data = await res.json();  
  console.log('SUCCESS', data);  
}
```

non-2XX status is
NOT an error!

fetch is available in all modern browsers (<http://caniuse.com/#search=fetch>).

For many projects, it makes sense to use a http library with more features:

- **ky**: <https://github.com/sindresorhus/ky> (modern, based on fetch)
- **up-fetch**: <https://github.com/L-Blondy/up-fetch>
- **axios**: <https://github.com/axios/axios> (legacy, based on XHR)

```
import ky from 'ky';  
const response = await ky.get('http://www.example-api.com');  
const data = await response.json<ResponseType>();
```

Features of http libraries: automatic json data transformation, status handling, http interceptors, response timeout, easily cancellable ...

Web Socket

A bi-directional connection between browser and server.

<https://developer.mozilla.org/en-US/docs/Web/API/WebSocket>

Demo

Streaming

https://developer.mozilla.org/en-US/docs/Web/API/Streams_API

Demo

EXERCISES

A close-up photograph of a pair of hands wearing camouflage-patterned gloves. The hands are using a pair of black pliers to cut a wire. The background is dark and textured, possibly a workbench or a piece of equipment. The word "EXERCISES" is written in large, white, bold, sans-serif capital letters across the top left of the image.

Exercises - fetch, WebSocket, Streaming

JSON

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/
Global_Objects/JSON](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON)

Demo

Forms

Demo

Web Worker

Demo