From Web Sites to Web Apps

This lesson gives an overview of web development models, synchronous vs asynchronous and cross-domain requests.

We'll cover the following

The Web Development Models

In a traditional web development scenario, when you click a link or submit a form, your browser sends to the server a request that returns a full new web page tailored to your request. This model is subject to longer load times and limited interactivity.

Another web development model aims to avoid transmitting a whole new page for each user action. Here's how things works in that model:

- User actions on the page are intercepted through JavaScript event handlers.
- HTTP requests are sent to the server without interrupting the navigation on the page.
- Only the needed portions of the page are updated with the requests' results.

Albeit more challenging, this web development model can lead to limited resource loads, improved interactivity and a user experience nearly on par with native applications.

The set of technologies enabling the creation of web applications is codenamed **AJAX** (*Asynchronous JavaScript* and XML). An AJAX call is an asynchronous HTTP request made to retrieve or send data from/to a server.

Synchronous vs Asynchronous Requests

In a *synchronous* exchange, the asker waits until he gets the needed info. A telephone call is an example of a synchronous exchange.

On the contrary, the asker in an *asynchronous* exchange can do something else while waiting for the completion of his request. Email is an example of an asynchronous exchange.

The traditional web development model uses synchronous requests: the web client is blocked while waiting for the server to complete its request. The AJAX model uses asynchronous requests: data is fetched when needed in the background.

Cross-Domain Requests

For security reasons, many websites have a conservative policy regarding AJAX requests. This same origin policy states that requests are limited to their origin domain: `"http://mysite"` cannot send a request to `"http://anothersite"`. This prevents some servers to be accessible via AJAX calls.

Enabling cross-domain requests is done by setting on cross-origin resource sharing (CORS) in the server configuration.

> For more information about this topic, check out this MDN article.