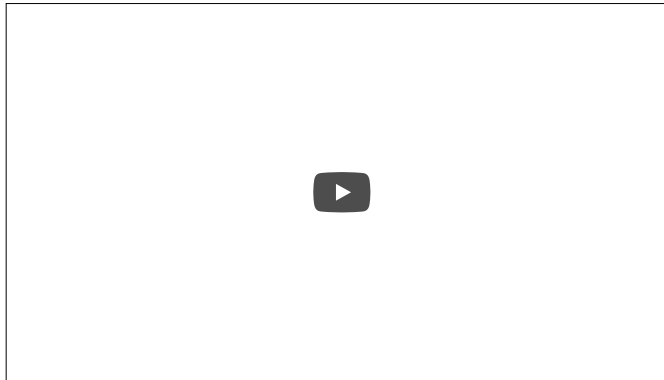


- ✓ 1. Course Intro
- ✓ 2. Client Server Demonstration
- ✓ 3. Ajax Definition & Examples
- 4. APIs
- 5. Create An Async Request with XHR
- 6. The XHR Object
- 7. XHR's .open() method
- 8. XHR's .send() method
- 9. A Full Request
- 10. Project Initial Walkthrough
- 11. Setting a Request Header
- 12. Project Final Walkthrough
- 13. XHR Recap
- 14. XHR Outro



A little history

JavaScript frameworks and Single Page Apps are the way to build today, but let's review where we've come from.

In the *traditional* server-rendered web application, the client computer makes a request for a web page. The server creates and returns a page to the client. Finally, the client loads the new page and displays the information. If they interact with the page, say to add or remove something by submitting a form, they start the cycle all over again. The client will make another request, the server returns a totally new page, the client loads and presents it to the user again.

Up until the mid 2000s, this was basically the only way internet communication occurred. Information would reside on the server, and a client would request that data and refresh the page and display it. This cycle would repeat for each and every new page request.

In the late 90s, the Microsoft Outlook team added the XMLHttpRequest component to Internet Explorer and built a web version of the Outlook mail client. This code was later picked up by other browsers as XMLHttpRequest. This allowed browsers to make HTTP requests from Javascript and update the current page in place without fetching an entire page from the server. Instead of the synchronous model of waiting for a whole page, the user interface could update asynchronously as the user kept working. Most of the data being exchanged used the XML format.

AJAX

In 2005, [Jesse James Garrett coined the term AJAX](#) to mean "Asynchronous Javascript and XML". This is essentially the technique of using [XMLHttpRequest](#) to fetch data and then modify the current page.

AJAX took the web world by storm, spreading far beyond Microsoft Outlook. State-of-the-art web applications like Flickr, GMail, and Google Maps rapidly adopted it. Instead of having to wait for data and have the entire page refresh, these new, near instantaneous applications were incredible.

Browser Inconsistencies

Hold up, though. Ajax wasn't all ponies and rainbows. There were several different, incompatible browser implementations and developers were forced to code for one browser or write complex code for them all. Eventually, JavaScript libraries like jQuery and YUI emerged to reconcile the differences.

AJAX apps were great, but difficult for individual developers to write; as browsers kept changing, and people demanded apps on more devices, the code then became more and more complex and confusing. This challenge led to the rise of standard Javascript frameworks and libraries. JavaScript libraries arose to hide the complex browser differences, JavaScript frameworks made developing complex, powerful applications manageable.





- ✓ 1. Course Intro
- ✓ 2. Client Server Demonstration
- ✓ 3. **Ajax Definition & Examples**
- 4. APIs
- 5. Create An Async Request with XHR
- 6. The XHR Object
- 7. XHR's .open() method
- 8. XHR's .send() method
- 9. A Full Request
- 10. Project Initial Walkthrough
- 11. Setting a Request Header
- 12. Project Final Walkthrough
- 13. XHR Recap
- 14. XHR Outro



Facebook showing new data appearing at the bottom of the page

NEXT



Mentorship
Get support and stay on track