



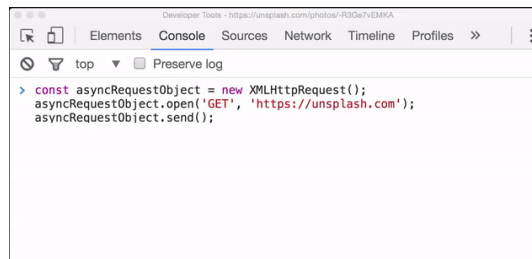
- ✓ 1. Course Intro
- ✓ 2. Client Server Demonstration
- ✓ 3. Ajax Definition & Examples
- ✓ 4. APIs
- ✓ 5. Create An Async Request with XHR
- ✓ 6. The XHR Object
- ✓ 7. XHR's .open() method
- ✓ 8. XHR's .send() method
- 9. A Full Request
- 10. Project Initial Walkthrough
- 11. Setting a Request Header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



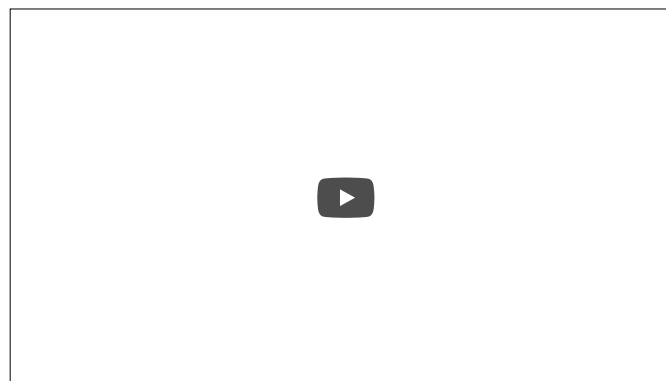
To actually send the request, we need to use the [send method](#):

```
asyncRequestObject.send();
```

Let's check out what happens:



The XHR request is sent, but we don't see anything!



It's a little pointless to make a request for something but then do absolutely nothing with it! Why would you order some cake and then not go to pick it up or not eat it? The horror! We want to eat our cake, too!

Handling Success

To handle the successful response of an XHR request, we set the [onload property](#) on the object to a function that will handle it:

```
function handleSuccess () {  
  // in the function, the 'this' value is the XHR object  
  // this.responseText holds the response from the server  
  console.log( this.responseText ); // the HTML of https://unsplash.com/  
}  
  
asyncRequestObject.onload = handleSuccess;
```

As we just saw, if `onload` isn't set, then the request *does* return...but nothing happens with it.

Handling Errors

You might've picked up that `onload` is called when the response is *successful*. If something happens with the request and it can't be fulfilled, then we need to use the **onerror property**:

<

Lesson 1:
Ajax with XHR

☰

📖

📁

🔍

✓ 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

XHR's .send() method

```
console.log( "An error occurred ☹️" );
}

asyncRequestObject.onerror = handleError;
```

As with `onload`, if `onerror` isn't set and an error occurs, that error will just fail silently and your code (and your user!) won't have any idea what's wrong or any way to recover."

NEXT

🗨️

Mentorship

Get support and stay on track

1