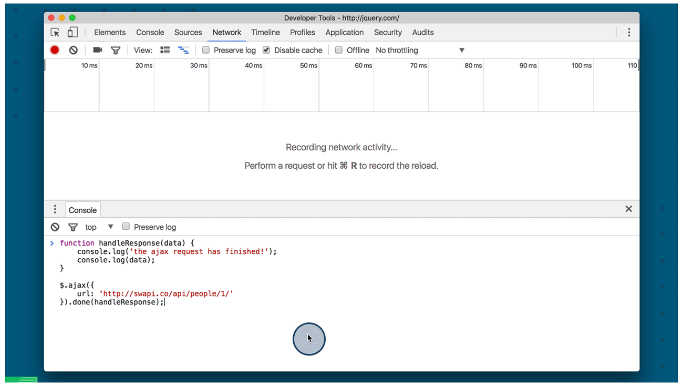
If you recall from setting up an XHR object, the response was handled by a function. It's the same thing with the `.ajax()` method. We can *chain* on to `.ajax()` with a `.done()` method. We pass the `.done()` method a function that will run with the Ajax call is done!

```
function handleResponse(data) {
    console.log('the ajax request has finished!');
    console.log(data);
}

$.ajax({
    url: 'https://swapi.co/api/people/1/'
}).done(handleResponse);
```


Developer Tools - http://jquery.com/

Recording network activity...

Perform a request or hit ⌘ R to record the reload.

```
> function handleResponse(data) {
      console.log('the ajax request has finished!');
      console.log(data);
  }

  $.ajax({
      url: 'http://swapi.co/api/people/1/'
  }).done(handleResponse);
```

Asynchronous call set up with a `done` method to handle the response. The request is made, and then the response is logged to the console.

Let's convert the existing, plain XHR call with jQuery's `.ajax()`. This is what the app currently has:

```
const imgRequest = new XMLHttpRequest();
imgRequest.onload = addImage;
imgRequest.open('GET', `https://api.unsplash.com/search/photos?page=1&query=${searchedForText}`);
imgRequest.setRequestHeader('Authorization', 'Client-ID <your-client-id-here>');
imgRequest.send();
```

A lot of this information is handled behind the scene by jQuery, so here's the first step in the conversion:

```
$.ajax({
    url: `https://api.unsplash.com/search/photos?page=1&query=${searchedForText}`
}).done(addImage);
```

With the jQuery code:

- we do not need to create an XHR object
- instead of specifying that the request is a `GET` request, it defaults to that and we just provide the URL of the resource we're requesting
- instead of setting `onload`, we use the `.done()` method

QUIZ QUESTION

The only change that needs to be made is including the Client ID header along with the request so that Unsplash will verify the request. Why don't you check out the API for the .ajax() method and select the code below that correctly adds an "Authorization" header to the request.

$.ajax({

```
.done(addImage);
```

```
$.ajax({
  url: https://api.unsplash.com/search/photos?page=1&query=${searchedForText}
  setHeader: [['Authorization', 'Client-ID 123abc456def']]
}).done(addImage);
```

```
$.ajax({
  url: https://api.unsplash.com/search/photos?page=1&query=${searchedForText}
  authorization: 'Client-ID 123abc456def'
}).done(addImage);
```

[✓]
```
$.ajax({
  url: https://api.unsplash.com/search/photos?page=1&query=${searchedForText}
  headers: {
    Authorization: 'Client-ID 123abc456def'
  }
}).done(addImage);
```

SUBMIT

The request should send perfectly now. Fantastic work! But there seem to be issues with the response and how it's handled.

NEXT

Mentorship
Get support and stay on track 1