

6.5.4 Display the API Data on the Page

Remember that even though the fetch is successful, we still need to format that response before we display it on the page.

Let's do that now. Add a `then()` method and log the response to the console. Make sure to add error handling for bad responses. The code should look something like this:

```
var getFeaturedRepos = function(language) {  
  var apiUrl = "https://api.github.com/search/repositories?q=" + language;  
  
  fetch(apiUrl).then(function(response) {  
    if (response.ok) {  
      console.log(response);  
    } else {  
      alert("Error: " + response.statusText);  
    }  
  });  
};
```

You'll need to call `getFeaturedRepos("javascript");` from the DevTools console again to see the `response` object. Note that the response is the entire HTTP response, not the JSON. Just like we did previously, we need

to use a method to extract the JSON from the response, so let's update `getFeaturedRepos()` to parse the response and log the data to the console, as follows:

```
var getFeaturedRepos = function(language) {  
  var apiUrl = "https://api.github.com/search/repositories?q=" + language;  
  
  fetch(apiUrl).then(function(response) {  
    if (response.ok) {  
      response.json().then(function(data) {  
        console.log(data);  
      });  
    } else {  
      alert("Error: " + response.statusText);  
    }  
  });  
};
```

Next, in the `json()` method's callback function, pass `data.items` and the `language` parameter's value into `displayRepos()`, shown here:

```
var getFeaturedRepos = function(language) {  
  var apiUrl = "https://api.github.com/search/repositories?q=" + language;  
  
  fetch(apiUrl).then(function(response) {  
    if (response.ok) {  
      response.json().then(function(data) {  
        displayRepos(data.items, language);  
      });  
    } else {  
      alert("Error: " + response.statusText);  
    }  
  });  
};
```

Test the `displayRepos()` function by calling `getFeaturedRepos("javascript");` again in the DevTools console,

verifying that the featured repos appear on the page this time.

This testing of the function before we add the buttons that will control the page is vital now that we're making requests, because many more things could go wrong in the process of retrieving information from the server. Catching these bugs before working on the rest of the front end enables us to more easily narrow down what went wrong if things break.

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.