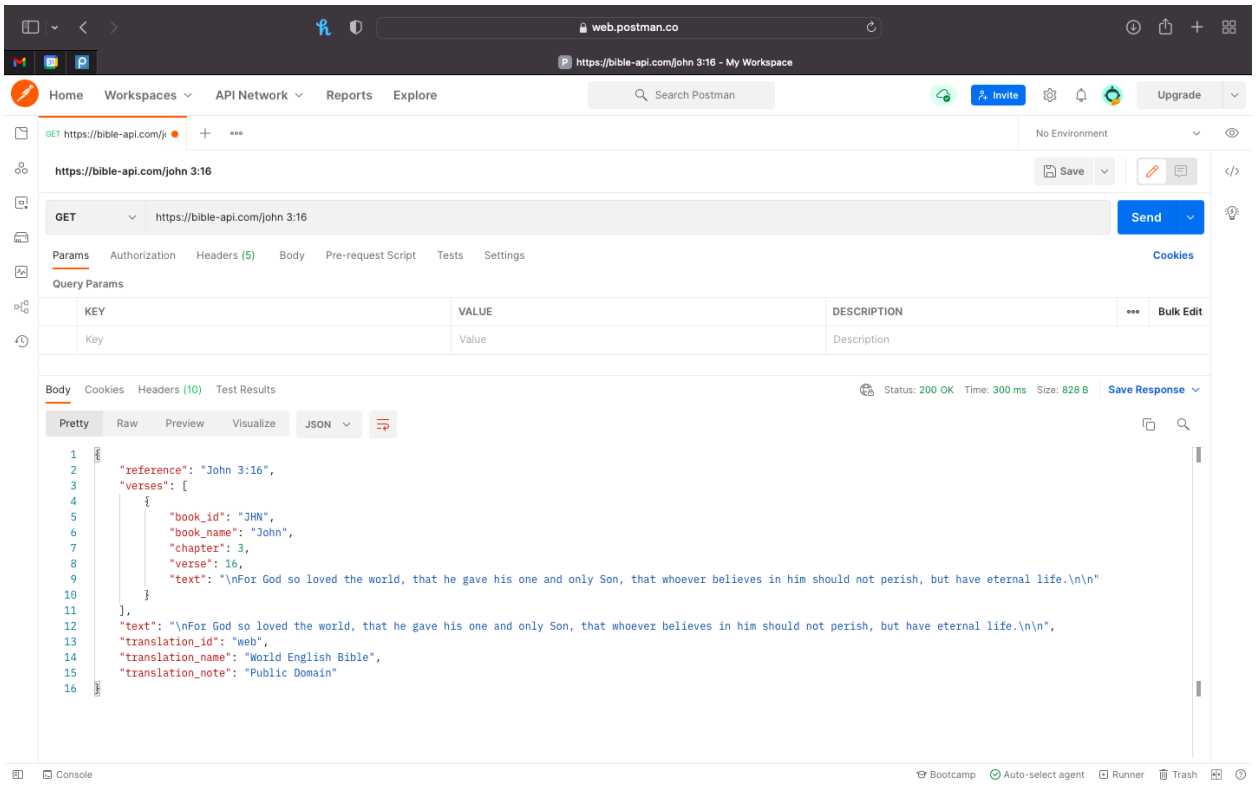
Mattia Danese
CS20 - Web Programming
Professor DiOrio
Assignment 10: API Fun

Online Link
https://mattia-danese.github.io/CS20/hw10/index.html

Questions

- The most challenging aspect of this homework assignment was figuring out I needed to put an `await` statement before converting the response from `fetch` into a JSON object. Before doing so, my code was not working and I was really confused for a while because using `fetch` is fairly trivial.
- The most satisfying aspect of this homework assignment was getting getting the XMLHttpRequest working because the data pattern splits the functionality into clear steps and it was nice to know what is going on at each step/check

Postman Picture



HTML Code

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>API Fun</title>

   <link rel = "stylesheet" href = "./style.css" />
   <script src="./script.js"></script>

</head>
<body onload="initialize();">
   <h1>API Fun</h1>

   <div id="container">
       <div id="info">
           <h3>API Information</h3>

           <div>The API I selected is <a
href="https://bible-api.com">bible-api.com</a>. This API provides easy online
access to all the Bible verses which can also be in different languages.
Using this API will return, in JSON format, all the verses that was specified
in the API call.</div>
           <br>
           <div>I found the API in <a
href="https://github.com/public-apis/public-apis">this</a> GitHub repo that
is a collection of numerous public APIs.</div>
           <br>
           <div>The option I used for the API request was "john 3:16". This
specifies the chapter and verse in that chapter that the API should return.
This is the only option to specify in the API request.</div>
           <br>
           <div>Two applications where this API would be helpful are:
               <ul>
                   <li>a site that when given a Bible verse translates it
into a desired language</li>
                   <br>
                   <li>free digitization of the Bible so having a physical
copy is not needed</li>
               </ul>
           </div>
       </div>
```

```
        <div id="data">
            <h3>Getting JSON with AJAX:</h3>
            <div id="ajax"></div>

            <h3>Getting JSON with Fetch:</h3>
            <div id="fetch"></div>
        </div>
    </div>
</body>
</html>
```

Javascript Code

```
async function setupAJAX(){
   let req = new XMLHttpRequest();
   if(!req){
       console.log("Can't create HTTPRequest object!");
   }

   req.onreadystatechange = function(){
       if(this.readyState == 4 && this.status == 200){
           document.getElementById("ajax").innerHTML = this.responseText;
       }};

   req.open("GET", "https://bible-api.com/john 3:16", true);
   req.send();
}

async function setupFetch(){
   let res = await fetch("https://bible-api.com/john 3:16");
   let data = await res.json();
   document.getElementById("fetch").innerHTML = JSON.stringify(data);
}

function initialize(){
   setupAJAX();
   setupFetch();
}
```

CSS Code

```
body{
   background-color: azure;
}

h1{
   text-align: center;
   text-decoration: underline;
}

#container{
   display: table;
   width: 70%;
   margin: auto;
}

#info, #data{
   display: table-cell;
   border: thin solid black;
   width: 50%;
   margin: 0px;
}

#info div, #info h3, #data div, #data h3{
   margin-left: 30px;
   margin-right: 30px;
}

#ajax, #fetch{
   margin-bottom: 30px;
}

@media only screen and (max-width: 1166px) {
   #container, #info, #data{
       display: block;
   }
   #container{
       width: 80%;
   }

   #info, #data{
       width: 100%;
```

```
    }
    h3{
        text-align: center;
    }
}

@media only screen and (max-width: 670px) {
    #container{
        width: 100%;
    }
}
```