

Luke Talcott – ldt9gn@umsystem.edu

GitHub link:

https://github.com/LukeT11/WebMobileProgramming/tree/main/Web_Lessons/Web_ICP4

Web ICP 4

Object-Oriented JavaScript, jQuery with API's

Introduction

Object-Oriented JavaScript uses all objects with values with properties to be used as object methods. These methods are actions that can be used on objects.

jQuery is a JavaScript Library that makes it easier to use with simple API's with things like HTML documentation traversal and manipulation, event handling, Ajax, etc.

API's is an Application Programming Interface is a type of software interface that services as a connection between other devices and software that allows a bridge for easy access and transfer of information.

Tasks

ICP4 tasks, description and example uses.

ICP4 Tasks (jQuery, JavaScript (OO), API):

- GitHub User Finder
 - Create an App that calls the GitHub API
 - Searched by Username
 - Display's the following user's accounts details:
 - Name of the user
 - Id of the user
 - Profile Picture
 - Link to the user's account

GitHub User Finder

- HTML

```
index.html x style.css x script.js x
1 <!-- Github User Finder html
2 ICP4
3 <!-->
4
5 <!--Head title and included scripts and stylesheets-->
6 <html lang="en">
7 <head>
8 <title>Github User Finder</title>
9 <link href="style.css" rel="stylesheet">
10 <script src="jquery.js" type="text/javascript"></script>
11 <script src="script.js" type="text/javascript"></script>
12 </head>
13 <!--Body for page-->
14 <body>
15 <!--Header title at top-->
16 <header>
17 <h1>Find Me!</h1>
18 </header>
19 <!--Container for page-->
20 <div id="container">
21 <!--Search box GitHub Username-->
22 <div class="search">
23 <input id="username" placeholder="Enter a github username..." type="text"></input>
24 </div>
```

```
index.html x style.css x script.js x
23 <input id="username" placeholder="Enter a github username..." type="text"></input>
24 </div>
25 <!--Header if Username not found-->
26 <div><h2 id="notFound"></h2></div>
27 <!--Profile of Username found with name, avatar and info.-->
28 <div class="profile">
29 <div id="profileUsername"></div>
30 <div id="avatar"></div>
31 <!--Information on User searched-->
32 <div class="information">
33 <div class="infoText" id="profileId"></div>
34 <div class="infoText" id="name"></div>
35 <div class="infoText" id="link"></div>
36 </div>
37 </div>
38 </div>
39 </body>
40 </html>
```

- CSS

```
index.html x style.css x script.js x
1  /*CSS stylesheet for GitHub User Finder*/
2
3  /*Body styles*/
4  body {
5      font-family: Helvetica, Arial, sans-serif;
6      border-top: 5px solid #4c4066;
7      margin: auto;
8      color: #666666;
9      font-size: 16px;
10 }
11
12 /*Header at the top*/
13 header {
14     text-align: center;
15     background-color: #fdfdfd;
16     border-bottom: 1px solid #e5e2e0;
17     font-size: 2vw;
18 }
19
20 #logo {
21     position: relative;
22     top: -2px;
23 }
24
```

```
index.html x style.css x script.js x
25  /* || Container Styles for the rest of the page*/
26  #container {
27      max-width: 1040px;
28      margin: auto;
29      min-height: 490px;
30      padding-bottom: 10px;
31  }
32
33  /*Search Box and Input styles*/
34  .search {
35      text-align: center;
36      margin-top: 50px;
37      margin-bottom: 15px;
38  }
39
40  input[type='text'] {
41      height: 2.2em;
42      font-size: 2em;
43      width: 50%;
44      padding: 5px;
45  }
46
47  /* || Styles for the profile with user info.*/
48  .profile {
49
```

```
index.html × style.css × script.js ×
49     height: auto;
50     max-width: min-content;
51     padding: 20px 50px 0 50px;
52     display: none;
53     background-color: #0de501;
54     text-decoration: none;
55     color: white;
56     margin: auto;
57 }
58
59 .profile:hover {
60     background-color: #077201;
61 }
62
63 /*Username of User styles*/
64 #profileUsername{
65     padding: 0 0 15px 0;
66     font-size: 24px;
67     font-weight: bold;
68     text-align: center;
69 }
70
71 /*Avatar and Avatar Image styles*/
72 #avatar {
```

```
index.html × style.css × script.js ×
73     text-align: center;
74     margin: auto;
75 }
76
77 #avatarImg {
78     max-width: 32vw;
79     height: auto;
80 }
81
82 /*Information about the User styles*/
83 .information {
84     vertical-align: top;
85     display: inline-block;
86     text-align: left;
87     border-top-left-radius: 5px;
88     border-bottom-left-radius: 5px;
89     padding: 20px 0 10px 0;
90     font-size: 20px;
91 }
92
93 .infoText {
94     padding: 0 1px 10px 2px;
95 }
96
```

```
index.html × style.css × script.js ×
88     border-bottom-left-radius: 5px;
89     padding: 20px 0 10px 0;
90     font-size: 20px;
91 }
92
93 .infoText {
94     padding: 0 1px 10px 2px;
95 }
96
97 /*User not found styles*/
98 #notFound {
99     text-align: center;
100 }
```

- JS

```
index.html × style.css × script.js ×
1  /*JavaScript for GitHub Use Finder*/
2
3  /*Function to get user using GitHub API with XMLHttpRequest and Get request*/
4  function getGithubInfo(user) {
5      //1. Create an instance of XMLHttpRequest class and send a GET request using it.
6      // The function should finally return the object(it now contains the response!)
7
8      const url = "https://api.github.com/users/"+user;
9      const xmlhttp = new XMLHttpRequest();
10
11     xmlhttp.open( method: 'GET', url, async: false);
12     xmlhttp.send();
13     return xmlhttp;
14 }
15
16 /*Function that displays contents for html gathered on user from the GitHub API*/
17 function showUser(user) {
18     //2. set the contents of the h2 and the two div elements in the div '#profile' with the user content
19     $('#profileUsername').text(user.login);
20     $('.profile').css('display', 'block');
21     $('#avatar').html('');
22     $('#profileId').text("ID: " + user.id);
23     if (user.name === null)
24     {
```

```
index.html × style.css × script.js ×
25     $('#name').text("Name: N/A");
26   }
27   else {
28     $('#name').text("Name: " + user.name);
29   }
30
31   $('#link').html('<span>User Link: </span><a id="html_url" href="' + user.html_url + '" target="_blank" >' + user.html_url + '</a>');
32 }
33
34 /*Function for displays for html that the searched user couldn't be found*/
35 function noSuchUser(username) {
36   //3. set the elements such that a suitable message is displayed
37   $('#notFound').text("* User '" + username + "' Not Found");
38 }
39
40 /*Function that takes typed user to be searched using JSON and jQuery*/
41 $(document).ready(function () {
42   $(document).on('keypress', '#username', function (e) {
43     //check if enter(i.e. return) key is pressed
44     if (e.which == 13) {
45       //Clear previous searched user info.
46       $('.profile').css('display', 'none');
47       $('#profileUsername').text("");
48       $('#profileId').text("");

```

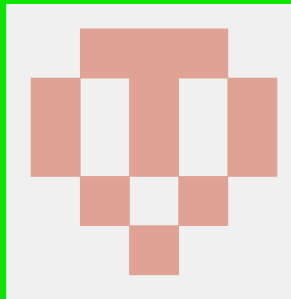
```
tHubUserFinder script.js
index.html × style.css × script.js ×
49   $('#avatar').html("");
50   $('#name').text("");
51   $('#html_url').text("");
52   $('#notFound').text("");
53   //get what the user enters
54   username = $(this).val();
55   //reset the text typed in the input
56   $(this).val('');
57   //get the user's information and store the response
58   response = getGithubInfo(username);
59   //if the response is successful show the user's details
60   if (response.status == 200) {
61     showUser(JSON.parse(response.responseText));
62     //else display suitable message
63   } else {
64     noSuchUser(username);
65   }
66 }
67 })
68 });
69
```

- Output

Find Me!

Find Me!

example



ID: 57936

Name: N/A

User Link: <https://github.com/example>

JavaScript JSON

Format to store and move data from a server to a web page.

```
59      //if the response is successful show the user's details
60      if (response.status == 200) {
61          showUser(JSON.parse(response.responseText));
62          //else display suitable message
63      } else {
64          noSuchUser(username);
65      }
```

XMLHttpRequest

Performs an API or HTTP request exchange data with a web server.

```
3      /*Function to get user using GitHub API with XMLHttpRequest and Get request*/
4      function getGithubInfo(user) {
5          //1. Create an instance of XMLHttpRequest class and send a GET request using it.
6          // The function should finally return the object(it now contains the response!)
7
8          const url = "https://api.github.com/users/"+user;
9          const xmlhttp = new XMLHttpRequest();
10
11          xmlhttp.open( method: 'GET', url, async: false);
12          xmlhttp.send();
13          return xmlhttp;
14      }
```

Example jQuery Selectors

Selectors to match elements like in a html document to show data collected from a web server.

```
16      /*Function that displays contents for html gathered on user from the GitHub API*/
17      function showUser(user) {
18          //2. set the contents of the h2 and the two div elements in the div '#profile' with the user content
19          $('#profileUsername').text(user.login);
20          $('.profile').css('display', 'block');
21          $('#avatar').html('');
22          $('#profileId').text("ID: " + user.id);
23          if (user.name === null)
24          {
25              $('#name').text("Name: N/A");
26          }
27          else {
28              $('#name').text("Name: " + user.name);
29          }
30
31          $('#link').html('<span>User Link: </span><a id="html_url" href="' + user.html_url + '" target="_blank" >' + user.html_url + '</a>');
32      }
```


Contributors

I worked independently, so I am the sole contributor.

Conclusion

I used Object-Oriented JavaScript, jQuery, and the GitHub API to create a web application that displays a valid GitHub user's account details that's searched by their username such as their name, avatar, profile picture and link to their GitHub account. I used the Object-Oriented part of JavaScript to use functions to perform actions on other objects. I called the GitHub API to find and collect information on a GitHub user. I used jQuery and its selectors to pass elements of user that is called by the GitHub API to show information from that user. Lastly, I used XMLHttpRequest to call and get information from the GitHub API.