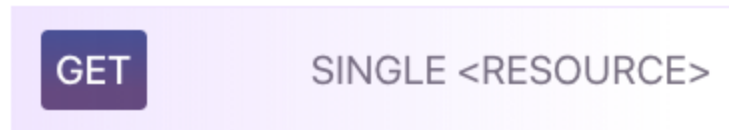




## Exercise 9: JavaScript AJAX/Async/Await

1. Using AJAX concepts, create an HTML page that displays colors from a Web Service:
  - a. To get the colors, use the GET SINGLE <RESOURCE> request from the site [GET REQ/RES](#). On the first page look for:

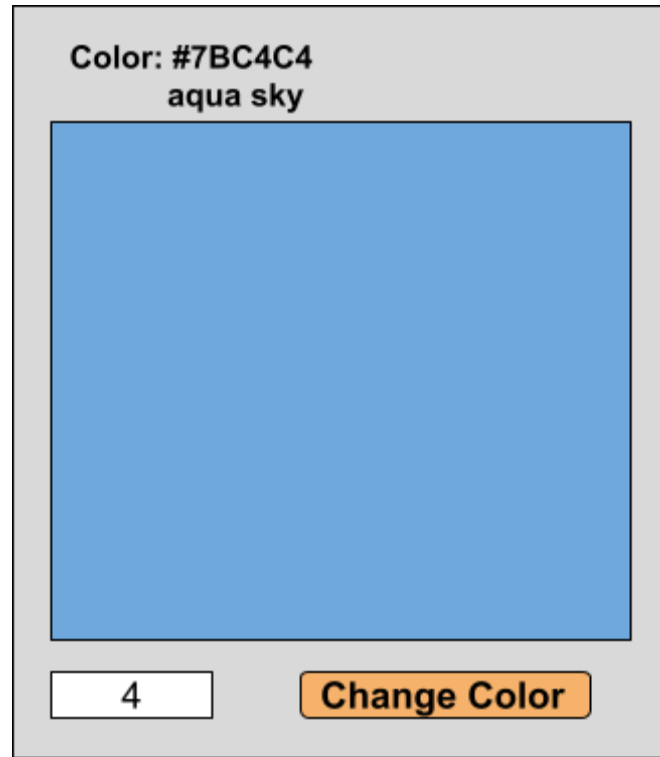


It describes colors using JSON. For instance, to request the unknown color of id 4 use <https://reqres.in/api/unknown/4> and get:

```
{ "data": { "id": 4,
            "name": "aqua sky",
            "year": 2003,
            "color": "#7BC4C4",
            "Pantone_value": "14-4811"
          },
  "support": { "url": "https://reqres.in/#support-heading",
              "text": "To keep ReqRes free, ... appreciated!"
            }
}
```

- b. Create an HTML page (Figure 1) that shows the colors returned by the Web Service. It shows:
  - i. The color name and hex code
  - ii. A square painted with the color
  - iii. A text field to read the color number
  - iv. A Change Color button

When the users click on the button, the app goes to the web service gets the color information, and changes the interface appropriately.



*Figure 1: HTML page showing color information.*

**Tip:** You may need to use the function [JSON.parse\(\)](#).

2. The function below (`getJoeHappiness`) returns a number measuring how happy Joe is. It receives a threshold between 0 - 1 to decide if Joe is happy or sad. If Joe's happiness number (random number from 0 to 1) is greater than this threshold, it returns this number. If not, it generates an error because Joe is sad. Using this function, create a web page (Figure 2) with a button and an input field, to read the threshold that says if Joe is happy or not each time we click on the button. You should use 3 versions of the function:
  - a. The original function with a callback.
  - b. A modified version that uses just Promises to implement the function.
  - c. A modified version using Async/Await.



**SCC0219 - Introduction to Web Development**

Put all 3 versions on the same HTML page (just repeat the HTML and call a different function each time).

```
const getJoeHappiness = (happy, callback) => {  
  setTimeout(() => {  
    const happiness = Math.random();  
    if (happiness > happy) {  
      callback(happiness);  
    } else {  
      callback(new Error("Joe is sad!"));  
    }  
  }, 2000);  
}
```

Threshold:  **Is Joe Happy?**

Yes! By 67 %.

Threshold:  **Is Joe Happy?**

No !!! Joe is sad!

Figure 2: HTML page with two possible values for Joe's happiness

3. In question 1, you had to use an asynchronous function to call the web service API. Rewrite the code of question 1 to use just Promises, if you used Async/Await, or use Async/Await if you used just Promises. If you used a callback function, use now Async/Await. Do not forget to deal with the error conditions.

*Good luck!*