Success

```
// A function which successfully uses a Promise to return a result
let error = false;
const getDataSuccessfully = () => {
  let promise = new Promise((resolve, reject) => {
    setTimeout(() => {
      console.log("Data returned from server!");

      if (!error) {
        resolve("Lots of data Success");
      }
      reject(" We have an error");
    }, 5000);
  });
  return promise;
};

// Execute the getDataSuccessfully() function
const promise = getDataSuccessfully();
// THEN, when it is finished, use the data it returns in your JS
// Note, now we are waiting for the data to be ready, not trying to use it too
early!
promise
  .then((data) => {
    console.log(data);
  })
  .catch((error) => {
    console.log("Error: ", error);
  });
```

Unsuccess

```
// A function which successfully uses a Promise to return a result
let error = true;
const getDataSuccessfully = () => {
  let promise = new Promise((resolve, reject) => {
    setTimeout(() => {
      console.log("Data returned from server!");

      if (!error) {
        resolve("Lots of data Success");
      }
      reject(" We have an error");
    }, 5000);
  });
  return promise;
};

// Execute the getDataSuccessfully() function
const promise = getDataSuccessfully();
// THEN, when it is finished, use the data it returns in your JS
// Note, now we are waiting for the data to be ready, not trying to use it too
early!
promise
  .then((data) => {
    console.log(data);
  })
  .catch((error) => {
    console.log("Error: ", error);
  });
```