CSc 337
AJAX

Benjamin Dicken

Announcements

- PA 7
- PA 8
- Final Project
- Fahmida’s Office Hour Cancelled today
- Updated links on schedule

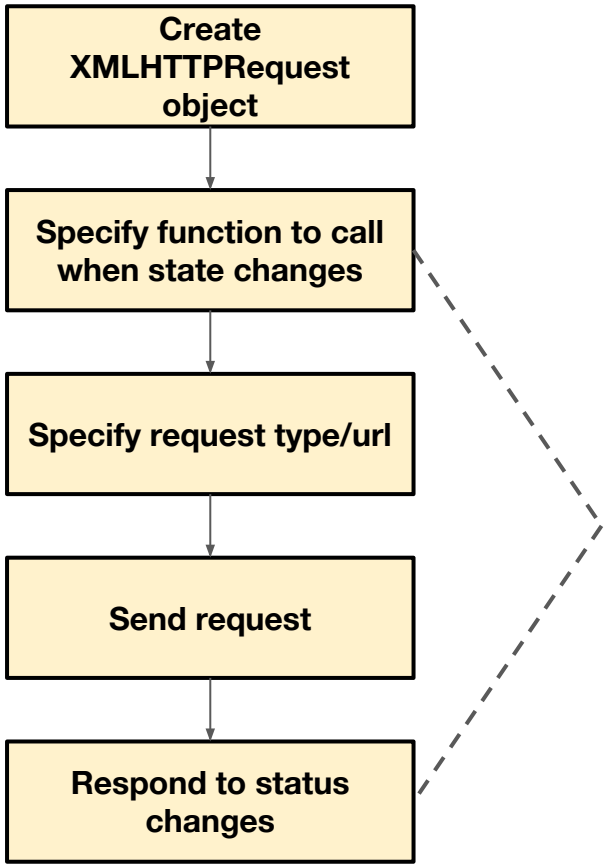
AJAX

- **A**synchronous **J**avascript **A**nd **X**ml
- Use for making asynchronous requests from the client
- Can fetch content from a server without needing to reload entire page, change URL or page, etc
- (Not just for XML)

XMLHTTPRequest (XHR)

- **XMLHTTPRequest** object has the core functionality to make these requests
- Use this object to make request, and specify how to respond to it
- Supported by most modern browsers

AJAX / XHR



5 different statuses to check:
0 / uninitialized
1 / loading
2 / loaded
3 / interactive
4 / complete

Also need to check status code (200 is OK)

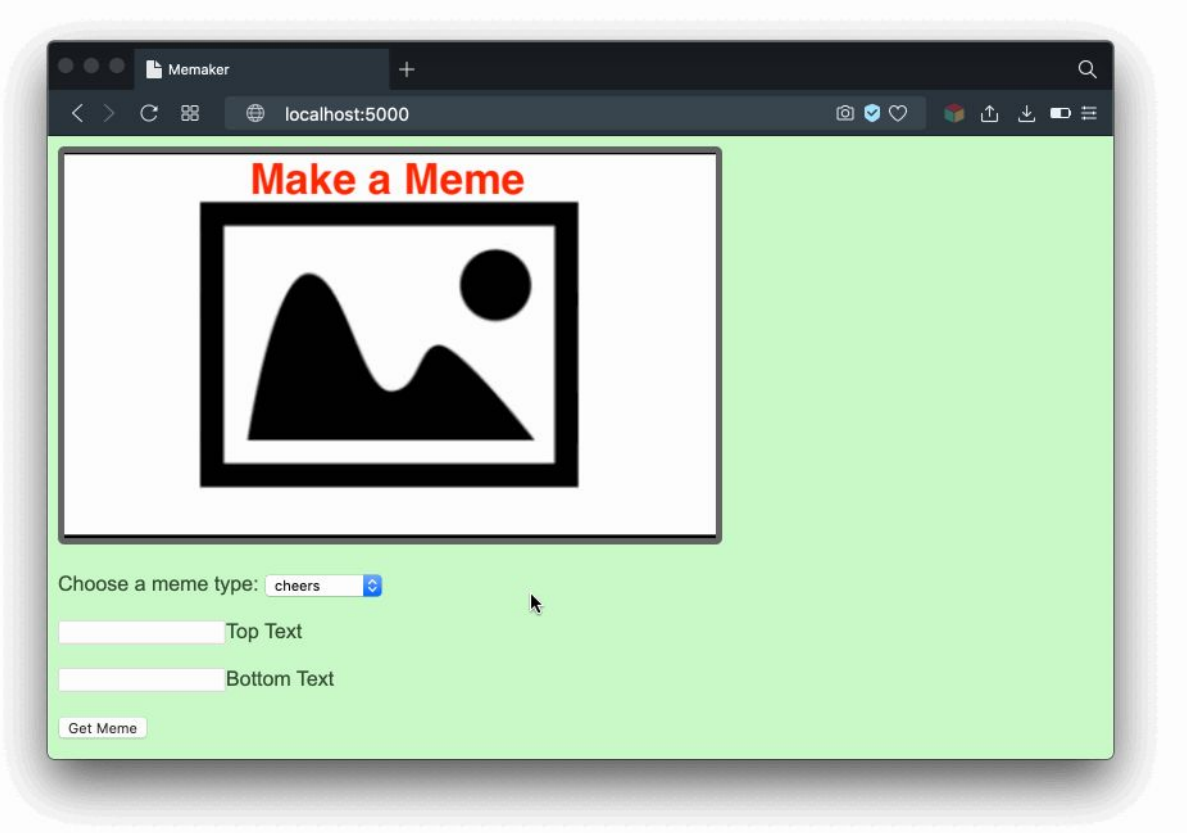
An Example

```
function setupRequest() {
  var httpRequest = new XMLHttpRequest();
  if (!httpRequest) {
    alert('Error!');
    return false;
  }

  httpRequest.onreadystatechange = () => {
    if (httpRequest.readyState === XMLHttpRequest.DONE) {
      if (httpRequest.status === 200) {
        alert(httpRequest.responseText);
      } else { alert('ERROR'); }
    }
  }

  let url = 'http://localhost:3000/some/path/';
  httpRequest.open('GET', url);
  httpRequest.send();
}
```

A Simple example

Random number generation website

Use AJAX to change the meme on the page



Callback Hell

```
jimp.read(inputFileName,
  function(err, loadedImage){
    if (err) { return console.log(err); }
    jimp.loadFont(jimp.FONT_SANS_64_WHITE,
      function(err, font){
        if (err) { return console.log(err); }
        loadedImage
          .print(font, 10, 10, topText)
          .print(font, 10, 326, bottomText)
          .write('./public_html/' + outputFileName ,
          function (err, data) {
            if (err) { return console.log(err); }
            fs.readFile('./public_html/index.html', 'utf8',
              function (err, data) {
                if (err) { return console.log(err); }
                var result = data.replace(/.\/default.png/g, '/' + outputFileName);
                res.send(result);
              });
          });
      });
  });
```

Promises

```
jimp.read(inputFileName)
  .then(function (image) {
    loadedImage = image;
    return jimp.loadFont(jimp.FONT_SANS_64_WHITE);
  })
  .then(function (font) {
    loadedImage
      .print(font, 10, 10, topText)
      .print(font, 10, 326, bottomText)
      .write('./public_html/' + outputFileName);
  })
  .then(function (font) {
    fs.readFile('./public_html/index.html', 'utf8', function (err,data) {
      if (err) { return console.log(err); }
      var result = data.replace(/.\/default.png/g, '/' + outputFileName);
      res.send(result);
    });
  })
  .catch(function (err) {
    console.error(err);
  });
```

Promises

- An object representing a task that may or may not yet be complete
- Can Provide a function that will get called when task is finished

  `.then()`
- Can Provide a function that will get called if task fails

  `.catch()`

fetch API

```
var type = document.getElementById('memeType').value;
 var topText = document.getElementById('top').value;
 var bottomText = document.getElementById('bottom').value;
 let url = 'http://localhost:5000/' + type + '/' +topText + '/' + bottomText;

 fetch(url)
   .then((response) => {
     return response.text();
   })
   .then((text) => {
     document.getElementById('image').src = text;
   })
   .catch( (error) => {
     console.log('THERE WAS A PROBLEM');
     console.log(error);
   });
```

https://www.javascripttutorial.net/javascript-fetch-api/

Promises

```
jimp.read(inputFileName)
  .then(function (image) {
    loadedImage = image;
    return jimp.loadFont(jimp.FONT_SANS_64_WHITE);
  })
  .then(function (font) {
    loadedImage
      .print(font, 10, 10, topText)
      .print(font, 10, 326, bottomText)
      .write('./public_html/' + outputFileName);
  })
  .then(function (font) {
    fs.readFile('./public_html/index.html', 'utf8', function (err,data) {
      if (err) { return console.log(err); }
      var result = data.replace(/.\/default.png/g, '/' + outputFileName);
      res.send(result);
    });
  })
  .catch(function (err) {
    console.error(err);
  });
```