School of Computer Science

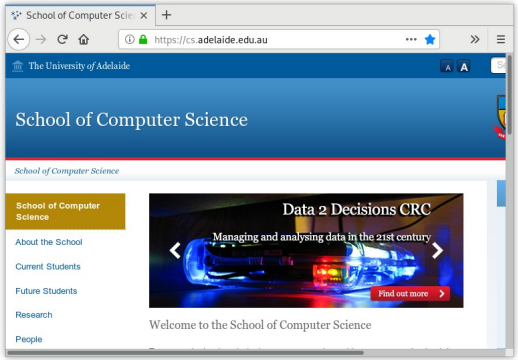
COMP SCI 2207/7207 Web and Database Computing
Lecture 22: Advanced Routes & Sending Data to the Server

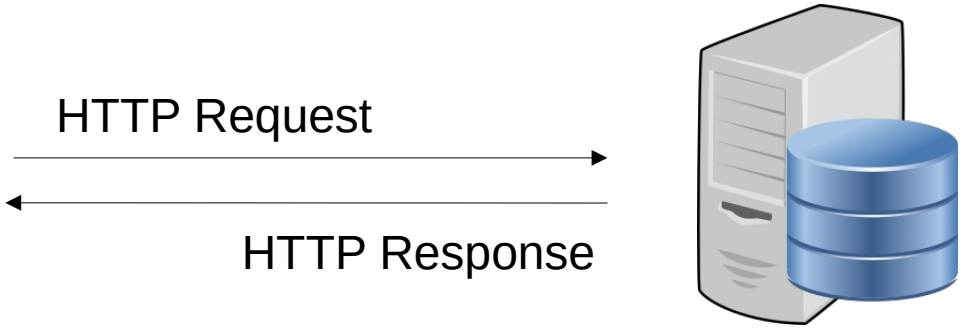
adelaide.edu.au

seek LIGHT

Server Review

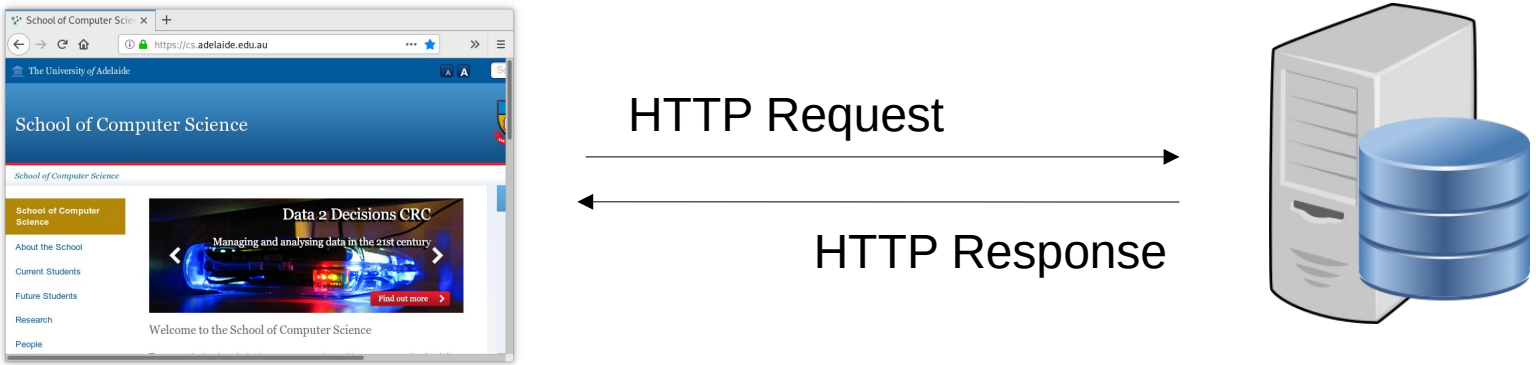
Client-server model revisited





- Clients access services and resources
  - Web browsers usually the clients in a web system.
- Servers provide those services and resources
- Communication between client and server is done using HTTP requests

HTTP & Client-server architecture



HTTP requests and responses are our only method of communication

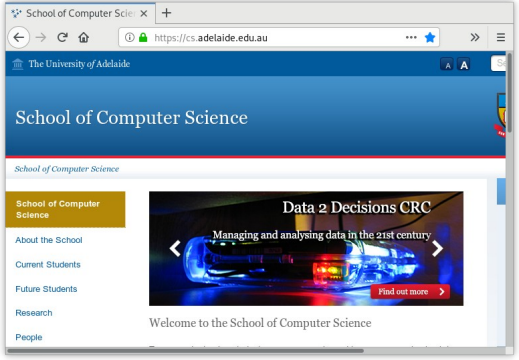
Our server can respond with static or dynamic content

- Static content could be a file, or some content that doesn't change
- Dynamic content is anything our server generates on the fly

AJAX & Client-server architecture

AJAX requests are regular HTTP requests made using JavaScript

What the client sees:



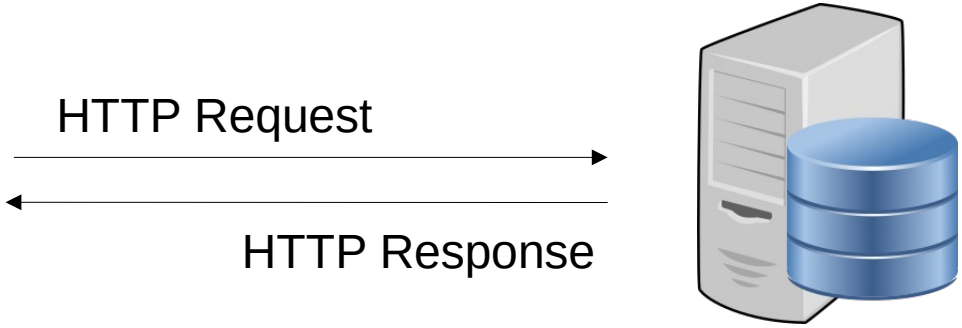
AJAX HTTP Request

AJAX HTTP Response

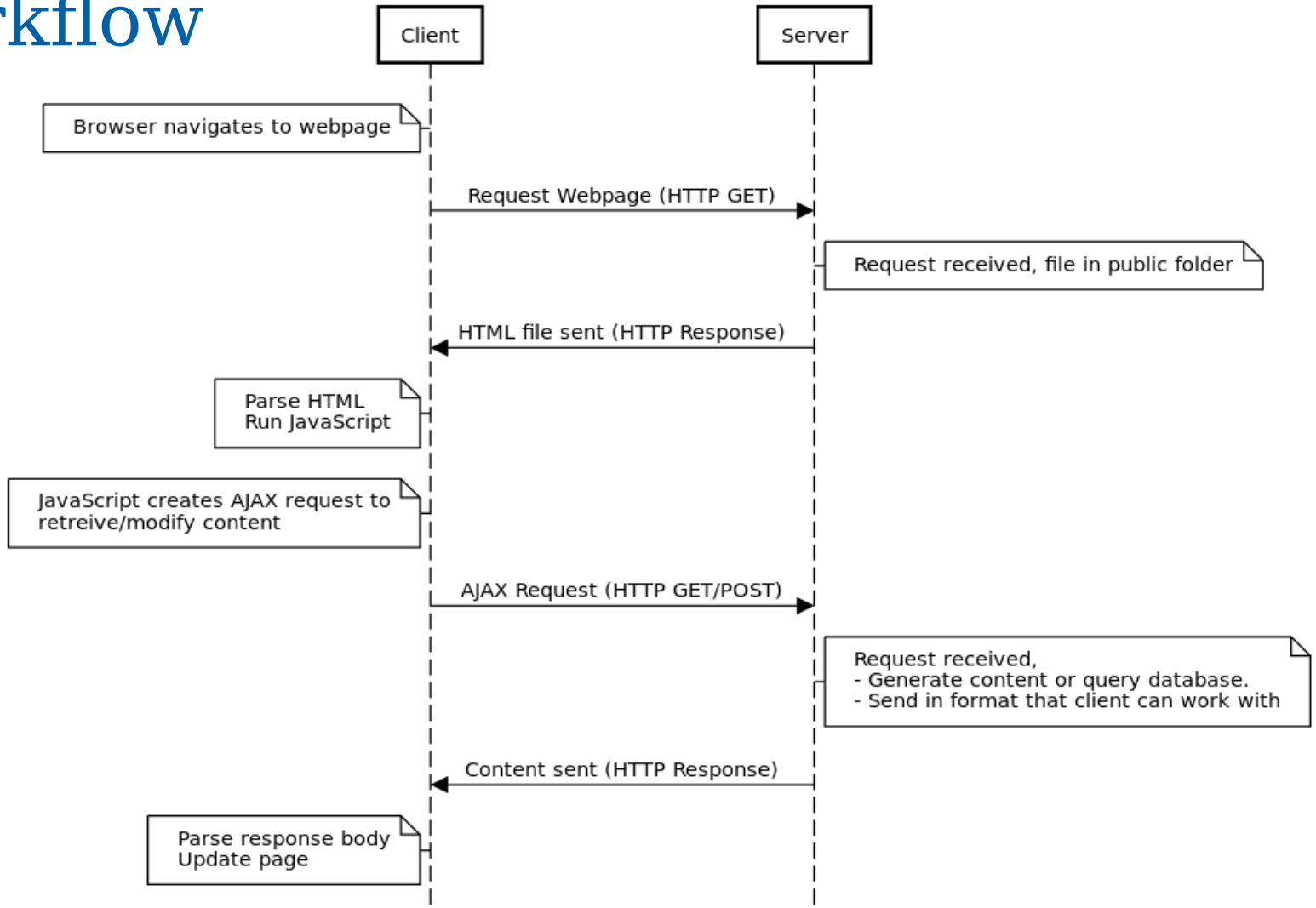
AJAX & Client-server architecture

AJAX requests are regular HTTP requests made using JavaScript

What the server sees:



AJAX workflow



More Complex Requests

Different types of Requests

GET

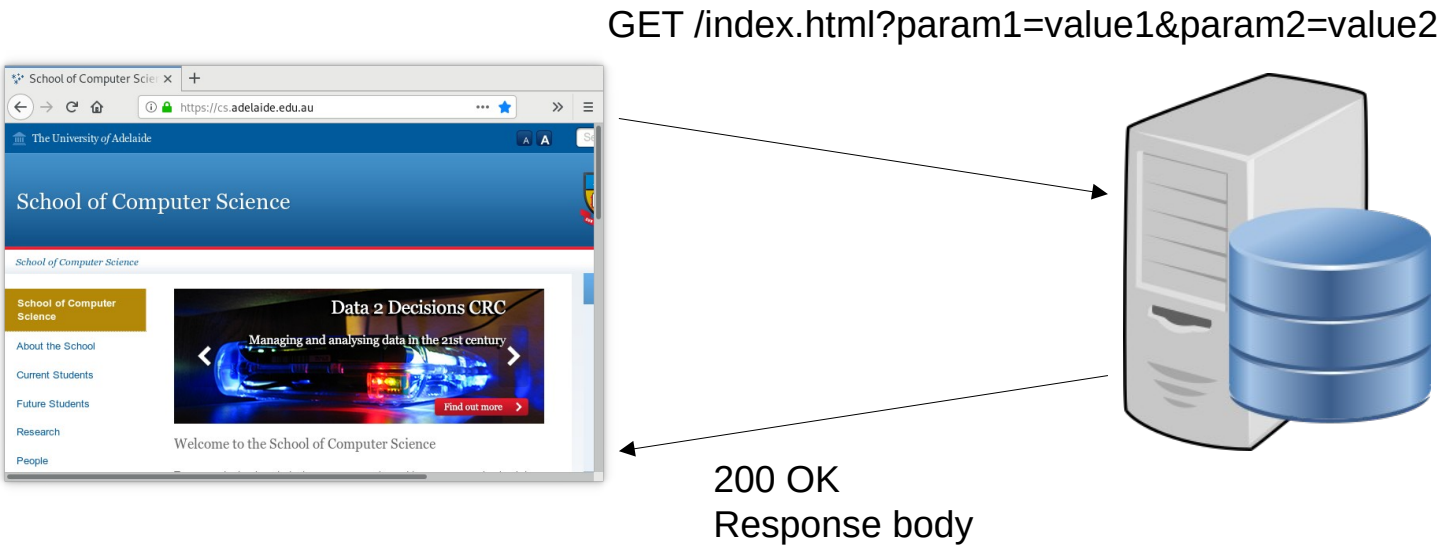
- Used for requesting a resource from a Server.
- No request body
  - Cannot send large amounts of data to server
  - Can have URL parameters for small amounts of data

POST

- Used for sending data to a Server
- Contains a request body

GET request with parameters

- Parameters placed at end of URL
- Must be a valid URL; may require encoding



Sending a GET request with parameters

Through a Form Submit

- A HTML form, when submitted can be used to send a GET request with additional parameters.
- Browser will load response as a new page.

```
<form action="/search.html" method="get">
  <input type="text" name="q" value="Enter your search term here">
  <input type="submit" value="Search">
</form>
```

Enter your search term here | Search

Method	File	Domain	Cause	Type	Transferred	Size	0 ms
GET	search.html?q=searching+for+stuff	www.w3schools.com	subdocument	html	2.73 KB	2.46 KB	293 ms

Sending a GET request with parameters

Using AJAX

- Use string concatenation to construct the URL.
- May need to encode values before sending

```
// Create new AJAX request
var xhttp = new XMLHttpRequest();

// Define behaviour for a response
xhttp.onreadystatechange = function() {

    if (this.readyState == 4 && this.status == 200) {
        // do stuff if successful
    }
};

// Initiate connection
xhttp.open("GET", "search.html?param1="+encodeURIComponent(value1)+
                                "&param2="+encodeURIComponent(value2), true);

// Send request
xhttp.send();
```

Handling a GET request w/ parameters in Express

Use the request object

- req.query is automatically generated as an object who's keys are the request parameters

app.js

```
...

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());

...
```

routes/index.js

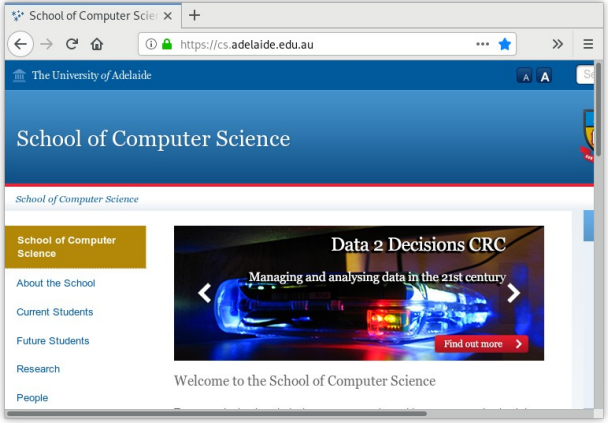
```
...

router.get('/search.html', function(req, res) {
    var q = req.query.param1;
    res.send('You searched for '+q);
});

...
```

POST request

- Can contain large amounts of data
- Held in request body



POST /index.html
{'param1':'value1','param2':'value2'}

200 OK
Response body

Sending a POST request

Through a Form Submit

- Similar to GET
- Search terms won't appear in URL
- Preferred method for passwords

```
<form action="/search.html" method="post">
  <input type="text" name="q" value="Enter your search term here">
  <input type="submit" value="Search">
</form>
```

Enter your search term here | Search

Method	File	Domain	Cause	Type	Transferred	Size	0 ms
POST	search.html	www.w3schools.com	subdocument	html	2.73 KB	2.46 KB	

Sending a POST request

Using AJAX

- Put content in send() call

```
// Create new AJAX request
var xhttp = new XMLHttpRequest();

// Define behaviour for a response
xhttp.onreadystatechange = function() {

    if (this.readyState == 4 && this.status == 200) {
        // do stuff if successful
    }
};

// Initiate connection
xhttp.open("POST", "search.html", true);

// Send request
xhttp.send("Some text that is the body of my request");
```

Handling a POST request in Express

Use the request object

- req.body is a string containing the request body.

routes/index.js

```
...

router.post('/search.html', function(req, res) {
    var text = req.body;
    res.send('You sent '+text);
});

...
```

Sending a JSON POST request

Using AJAX

- Put JSON string in send()
- ALWAYS set content type (in general, not only JSON)

```
// Create new AJAX request
var xhttp = new XMLHttpRequest();

// Define behaviour for a response
xhttp.onreadystatechange = function() {

    if (this.readyState == 4 && this.status == 200) {
        // do stuff if successful
    }
};

// Initiate connection
xhttp.open("POST", "search.html", true);

// Set content type to JSON
xhttp.setRequestHeader("Content-type", "application/json");

// Send request
xhttp.send(JSON.stringify({param1:value1, param2:value2}));
```

Handling a JSON POST request in Express

Use the request object

- req.body is a string containing the request body.
- If the content type is set correctly, the JSON middleware will automatically parse the JSON to an object.

app.js

```
...

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());

...
```

routes/index.js

```
...

router.post('/search.html', function(req, res) {
    var q = req.body.param1;
    res.send('You searched for '+q);
});

...
```

Parameterised URLs in Express

Allows for custom URL paths

- e.g. a request to /user/bob could be handled by the same router method as /user/alice
- Use `:paramName` in the path, access with `req.params.paramName`

routes/index.js

```
...

router.get('/user/:id', function(req, res) {
  res.send('user ' + req.params.id);
});

...
```

Demo

THE UNIVERSITY
of ADELAIDE

What’s happening?

- Prac Exercise 6 – DUE Tonight 11:59pm
- Prac Exercise 7 – Available
  - DUE Monday 13th
  - Websub coming soon