

EECS
4413

BUILDING e-COMMERCE SYSTEMS

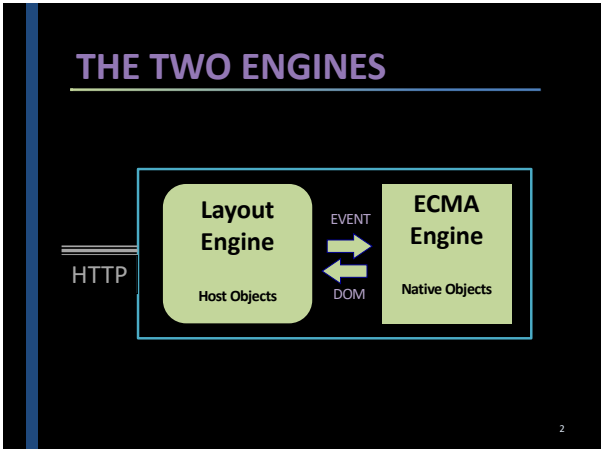

Define the POSSIBLE

CLIENT-SIDE I

SPA

PROF H ROUMANI
Dept. of Electrical Engineering & Computer Science
LASSONDE
SCHOOL OF ENGINEERING

1



2

THE THREE AMIGOS

HTML + CSS + JS

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="StyleSheet" href="a.css" type="text/css"/>
    <script type="text/javascript" src="b.js"></script>
  </head>
  <body>
    ...
  </body>
</html>
```

3

3

SERVING STATIC PAGES

Apache

- Store in document root (on red: ~/www)
- URL on red: www.eecs.yorku.ca/~username

Tomcat

- Store under webapp (WebContent)
- But not in WEB-INF

Node

- Store in a folder X anywhere and then:
- app.use('/serve', express.static(path to X));

4

4

HELLO SPA

Single-Page Applications

- Origin server sends static html/css/js
- JS talks to view via DOM
- View talks to JS via events
- JS uses ajax to fetch resources from services
- One page only (SPA) yet multiple views

5

5

BASIC EVENTS

- onload, onunload
- onclick, ondblclick
- onmousedown, up, over, move, out
- onfocus, onblur
- onkeypress, down, up
- onsubmit, onselect, onchange

6

6

BASIC HTML for SPA

- All views but one are invisible via onload
- One div (with id and class) per view
- The div is populated via ajax
- View elements must have unique id's
- View element events invoke js function
- Can also use anchors with javascript: href

7

7

BASIC DOM for SPA

- `document.getElementById(...)`
- `document.getElementsByClassName(...)`
- `element.style.display = <block | none>`
- `form.elements["<id>"].value`
- `element.innerHTML = <string>`

8

8

BASIC AJAX

- `var http = new XMLHttpRequest()`
- `http.onreadystatechange = function()
 { this.readystate, status, responseText }`
- `http.open(method, url, async, user, psw)`
- `http.send(null or data if POST)`
- `http.setRequestHeader(...)`
- `http.getResponseHeader(...)`
- `http.abort()`

9

9

AJAX EXAMPLE

```
function doAjax(address, handler)
{
    var http = new XMLHttpRequest();
    http.onreadystatechange = function()
    {
        if (http.readyState == 4 && http.status == 200)
        {
            handler(http.responseText);
        }
    }
    http.open("GET", address, true);
    http.send();
}
```

10

10

GET vs POST

Tomcat

- doGet vs doPost
- Can invoke one from the other

Node

- app.get vs app.post (app.use for both)
- For post, you access the via req.body

Ajax

- Specify "get" or "post" in the open method
- For post, send the body via the send method

11

11

SECURITY: SOP and CORS

Same Origin Policy

- Origin = protocol + host + port
- SOP = all resources (img, ajax, ...) same origin

Cross Origin Resource Sharing

- Header: Access-Control-Allow-Origin: origin
- Specify in htaccess in Apache and Tomcat

Node

```
app.use((req, res, next) => { res.header
("Access-Control-Allow-Origin", "*"); next(); });
```

12

12