

CSC309 *Programming on the Web*

week 7: jquery, ajax

Amir H. Chinaei, Spring 2017

Office Hours: M 3:45-5:45 BA4222

ahchinaei@cs.toronto.edu

<http://www.cs.toronto.edu/~ahchinaei/>

review

❖ **so far:**

■ **front-end**

- structure & semantic, appearance, behavior
- many design tips

■ **back-end**

- databases
 - structured & semi-structured data

■ **interacting with backend**

- http, rest, node, express

❖ **this week:**

- using jquery/ajax for the interaction

review

- ❖ **http** (underlying protocol for client/server interaction)

GET /allWords.html HTTP/1.1 Host: www.project.org

HTTP/1.1 200 Ok Content-Type: text/html <html><p>Hello</p></html>

- ❖ **rest** (architectural styling/mindset)

www.project.org/allWords

www.project.org/words/computer

- ❖ **node** (asynchronous and event driven)

...

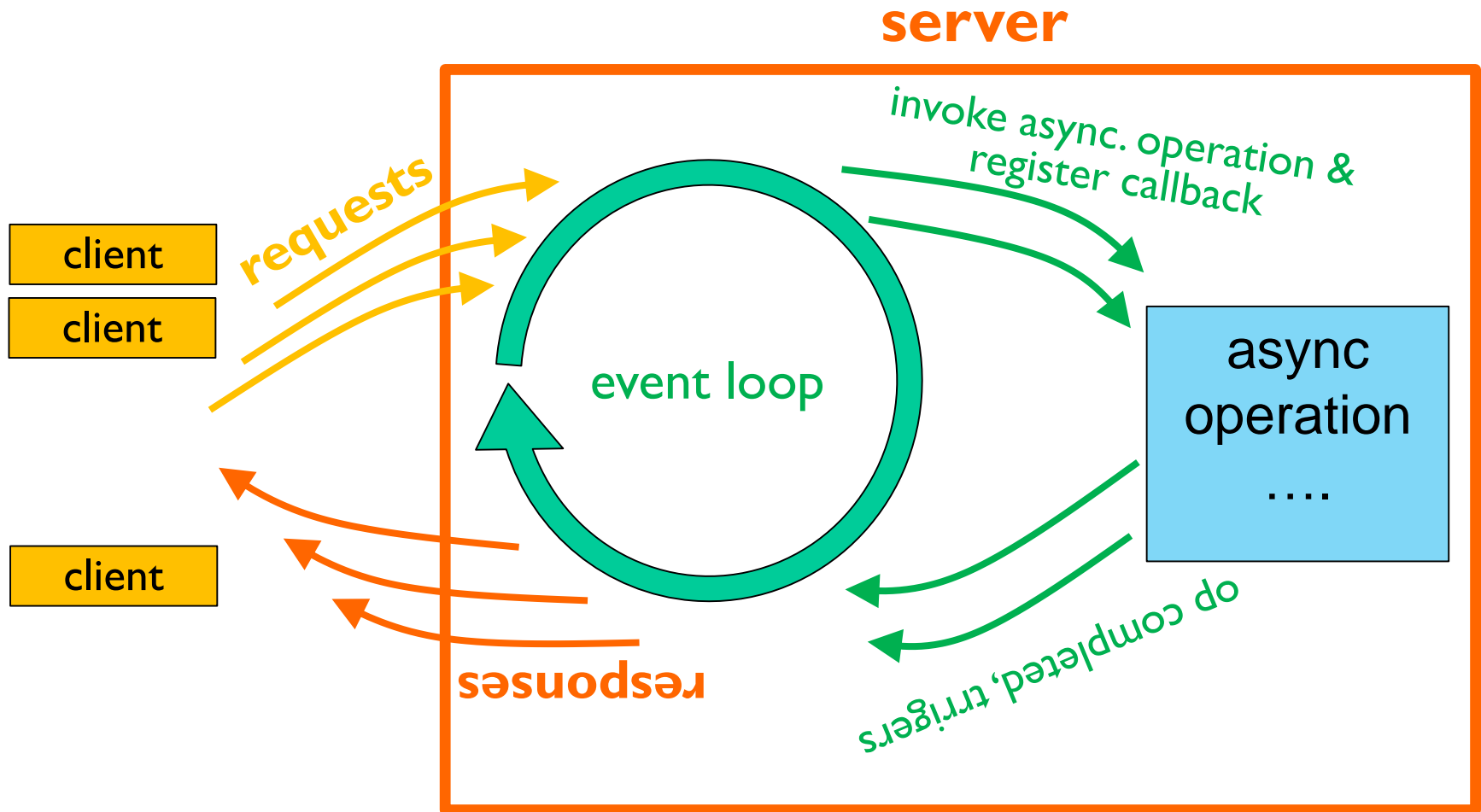
```
fs.readFile( "some.txt", readCompletedCallback);
```

```
//do other tasks ...
```

```
function readCompletedCallback( error, dataBuffer) {  
  response.writeHead(200, {'Content-Type': 'text/plain'});  
  response.end('<html><p>Hello</p></html>');
```

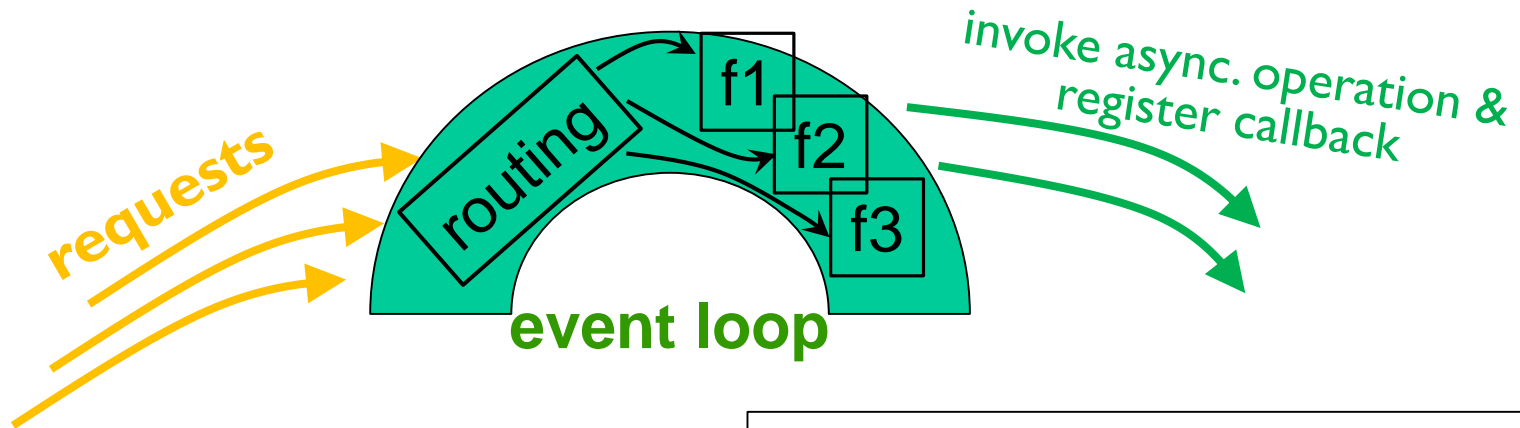
```
...
```

review



review

❖ **express** (robust routing)



❖ **routing** in lab 6:

```
app.get('/name', getName);  
app.post('/name', setName);  
app.get('/logout', logout);  
app.post('/addmsg', addMessage);  
app.get('/messages', getMessages);
```

```
function getName(req, res) {  
  if (req.session.name) {  
    return res.json({name: req.session.name});  
  }  
  else {  
    return res.json({name: ' '});  
  }  
}
```

jquery

❖ a **js** library for

- **traversing** and **manipulating** dom
- **event** handling

```
<html>
<head>
  <title>jquery example</title>
  <script type = "text/javascript"
    src = "https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">
  </script>
  <script type = "text/javascript">
    $(function){document.write("Hello World!"); };
  </script>
</head>
<body>
  <p>this is dummy</p>
</body>
</html>
```

\$(selector).event(action())

dom selector

```
window.onload = function(){...}
```

cf slide 4.14

```
getElementsByTagName("p")
```

```
getElementById("123")
```

css selector (slides 3.4, 3.8, 3.9)

```
p, div {...}
```

```
div p {...}
```

```
div > p {...}
```

```
.welcome {...}
```

```
[type] {...}
```

```
a[target="_blank"]
```

jquery selector

```
$(document).ready(function(){...});
```

or

```
$(function(){ ... });
```

```
$("p")
```

```
$("#123")
```

```
$("p, div")...
```

```
$("div p")...
```

```
$("div > p")...
```

```
$(".welcome")...
```

```
$("[type]")...
```

```
$("a[target='_blank']")...
```

traverse

`$("p").parent("div")...`
`$("p").parents("div")...`
`$("p").parentsUntil("div")...`

`$("div").siblings("p")...`
`$("div").next("p")...`
`$("div").nextAll("p")...`
`$("div").nextUntil("p")...`

`$("p:nth-of-type(3)")...`

`$("p").filter(".welcome")...`
`$("p").not(".welcome")...`

`$("div").children("p")...`
`$("div").children("p").first()...`
`$("div").children("p").last()...`
`$("div p:nth-child(2)")...`
`$("div").find("p")...`

`$("div").prev("p")...`
`$("div").prevAll("p")...`
`$("div").prevUntil("p")...`

`$("p").eq(2)...`
`$("p").first()...`
`$("p").last()...`

effect

```
$("#p").slideDown(speed, easing, callback)
```

```
$("#p").animate(param, speed, callback)  
             {left: "250px"}, "slow", myFunc()
```

❖ queueing

```
$("#p").animate({left: "250px"}, "slow", myFunc());  
$("#p").animate({top: "400px"}, "slow", aFunc());
```

❖ chaining

```
$("#p").animate({left: "250px"}, "slow", myFunc())  
      .animate({top: "400px"}, "slow", aFunc());
```

```
$('a').parent('li').siblings().find('a');
```

misc.

❖ implicit loop

```
$("#p").addClass("intro");
```

❖ add/remove styling

```
$("#p").eq(2).removeClass("intro");
```

```
$("#p").eq(2).toggleClass("intro");
```

❖ explicit loop

```
$("#p").each(function(index){  
    $(this).prepend(index+1);  
});
```