INFM 603: Information Technology and Organizational Context

Session 8: Ajax and Asynchronous Programming

Jimmy Lin
The iSchool
University of Maryland

Thursday, October 25, 2012

RideShare Exercise

- Design a database to match drivers with passengers for ride sharing on long car trips:

  - Drivers post available seats; they want to know about interested passengers
  - Passengers come looking for rides: they want to know about available rides and can make reservations
  - These things happen in no particular order
  - To simplify, passengers don't get to post "rides wanted" ads

- Build a web application to accomplish the above

RideShare Exercise: Tasks

○ Design the tables you will need

  ● First decide what information you need to keep track of
  ● Then design tables to capture this information

○ Design SQL queries

  ● What happens when a passenger comes looking for a ride?
  ● What happens when a driver comes to find out who the passengers are?

○ Role play!

Tables

- Ride: Ride ID, Driver ID, Origin, Destination, Available Seats

- Passenger: Passenger ID, Name, Address, Phone Number

- Driver: Driver ID, Name, Address, Phone Number

- Booking: Ride ID, Passenger ID

Queries

○ Passenger searches for a ride:

  ● Join: Ride, Driver
  ● Where: Origin and Destination match request, Available Seats > 0
  ● Select: Name, Phone Number

○ Passenger "books" a ride:

  ● Assuming successful search above: decrease Available Seats by one
  ● Insert row into Booking table with Ride ID and Passenger ID

○ Driver ready to go: Who are my passengers?

  ● Join: Ride, Passenger, Booking
  ● Where: (Driver) Name, Origin, and Date match
  ● Select: (Passenger) Name, Phone Number

Demo

- We're going to build the RideShare web app…

- Like, right now!

Slight Simplification

- Ride table:

  - RideId
  - Driver (name)
  - Phone
  - Origin
  - Destination
  - Date
  - Seats

- Booking table:

  - RideId
  - Passenger (name)
  - Phone

Today

- More JavaScript!

- Ajax

- JSON

- More PHP!

Synchronous vs. Asynchronous

- Definitions

  - Synchronous: happening, existing, or arising at precisely the same time
  - Asynchronous: not synchronous

- Communications

  - Synchronous

  - Asynchronous

- Programming

  - Synchronous

  - Asynchronous

Ajax

What's Ajax?

- Asynchronous JavaScript and XML

- The only thing you need to learn:

Get this URL

```
var url = "...";
var request = new XMLHttpRequest();
request.open("GET", url);
request.onload = function() {
  if (request.status == 200) {
    // Your code here
  }
};
request.send(null);
```
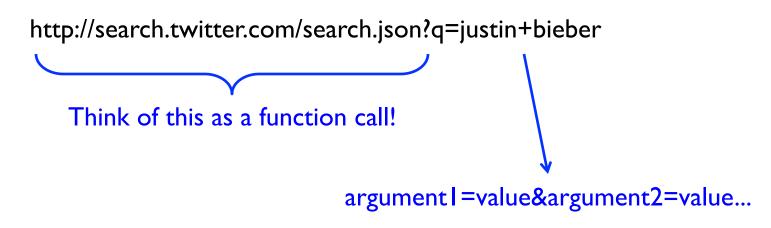
Callback function

What's at the URL?

○ A static file (e.g., JSON)


```
{
    name: "Fido",
    weight: 40,
    breed: "Mixed",
    loves: ["walks", "fetching balls"]
}
```

What's at the URL?

- An application programming interface (API)

http://search.twitter.com/search.json?q=justin+bieber

Think of this as a function call!

argument1=value&argument2=value...

- How do we write APIs?

Got it?