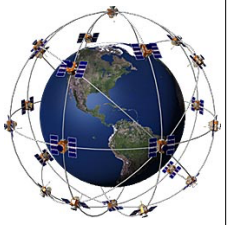
EECS 1012: Introduction to Computer Science

November 7, 2016

Lab 6:GPS

- Almost (all?) modern portable devices have GPS (global positioning system) receivers built into them.
  - Actually they have support for multiple GPS systems (GLONASS, others) but the terms GPS is typically used to describe all such systems.

How basic GPS works



- Constellation of satellites in orbit
- Your device receives signals from them.
- Each satellite broadcasts its position and time
- Your receiver 'solves' its location based on all satellites having identical clock times.

How basic GPS works

- If your device has view of 4 or more satellites then there is a solution to the problem
  - Basic approaches get about an 8m accuracy 95% of the time when 4 or more satellites are in view.
- There exist a wide range of technologies to improve this, and sub meter accuracy is possible with appropriate technology.

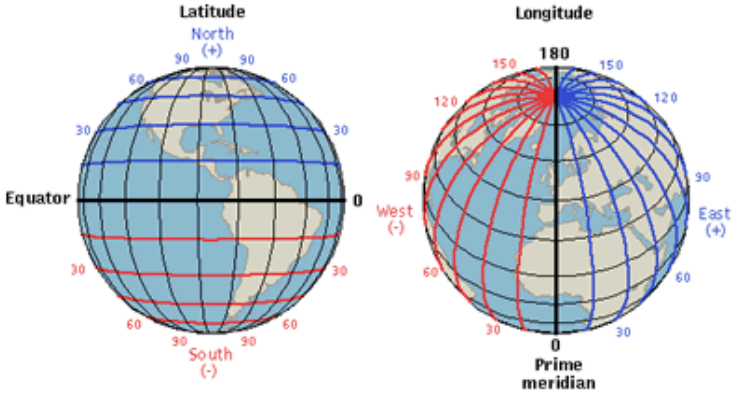
GPS+other…

- Cell phones can also localize themselves with respect to cellular receivers.
- Any device on a WIFI network can be localized by the network.
- From a software point of view, we don't want to have to work out the details of these technologies.
  - Really just want localization information

Warning!!

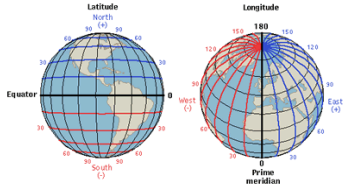
- GPS requires access to the GPS signal
  - Does not work in many places (indoors)
- Geolocation data will provide some estimate without GPS signals, but accuracy can be very poor indeed.

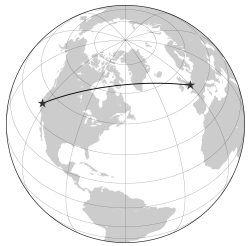
Latitude & Longitude



Local coordinates

- Note that computing distances from (latitude, longitude) measurements is non-trivial in general.





Local coordinates

- For local measurements you can often get away by approximating this computation.
- The lab uses linear interpolation — likely sufficient for the distances you are dealing with.
  - But its important to remember that the linear approximation is only approximate.

Geolocation in Javascript

- navigator.geolocation object provides an interface to geolocation capabilities of the local device.
- Basically provides two interfaces
  - getCurrentPosition(callback)
  - id = watchPosition(callback)

Event-driven process

- It can take some time for the localization system to return
  - Event-driven process is desirable.
- Highly power-intensive to run the GPS chip set
  - **Keep your batteries charged in the lab.**

getCurrentPosition()

```
var gps = document.getElementById("gps");
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    } else {
        gps.innerHTML = "Geolocation is not supported";
    }
}
function showPosition(position) {
    gps.innerHTML = "Latitude: " + position.coords.latitude +
    " Longitude: " + position.coords.longitude;
}
```

getCurrentPosition()

- One-shot geolocation
- If run in your browser, you will be asked if the web page should have access to your location
  - The html2apk enables this automatically — you will not be asked.
- Not designed to be asked repeatedly.
  - Can be very slow for multiple queries.

watchPosition()

```
var gps = document.getElementById("gps");
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.watchPosition(showPosition);
    } else {
        gps.innerHTML = "Geolocation is not supported";
    }
}
function showPosition(position) {
    gps.innerHTML = "Latitude: " + position.coords.latitude +
    " Longitude: " + position.coords.longitude;
}"
```

watchPosition()

- Your callback will be called…repeatedly
  - How often is hardware dependent.
- This is how things like google maps runs
- This will really drain the battery (just like similar apps on your phone).

To turn watch off

- var z = navigator.geolocation.watchPosition(cb);
- …
- navigator.geolocation.clearWatch(z)
  - stops the process (and restores battery performance).

Communicating between processes

- Given two processes (client-server, others) how do we get them to communicate with each other?
  - Not the details of the communication, rather the language that they speak.

XML

- Extensible Markup Language (XML)
- Language much like HTML
- Was designed to store and transmit information
- It was designed to be self-descriptive

```
<note>
  <date>2015-09-01</date>
  <hour>08:30</hour>
  <to>Tove</to>
  <from>Jani</from>
  <body>Don't forget me this weekend!</body>
</note>
```

Sample XML
Very much like HTML
<tag name1="value1" name2="value2"> … </tag>

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>

  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>

  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

XML DOM

- Like HTML, XML has a Document Object Model
  - Parent-child-sibling relationships.
  - Individual nodes have attributes and value.
- Exist standard libraries in many languages to deal with the DOM model (as with HTML).

XML root

- Must have a root element
  - <bookstore> in the previous example
- This helps to enforce the tree structure you saw earlier in HTML.

Like XML HTML is extensible

- Extra tokens/attributes are ignored
- This means that you can “add” to the language and use old code and the new material will be ‘ignored’ (as well as it can be).

Is XML too broad?

- XML can seem to encode almost anything
- That makes it very powerful, but also very difficult
  - How do you know if a given XML document is ‘valid’?
- We’d like to have XML’s flexibility but also the ability to restrict XML to a particular set of rules.

DTD (Document Type Definition)

- Is a specification of what a specific XML file can contain.

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

XML

- Very powerful and well-used language for specifying communication between two software processes.
- One concern with the representation is the representation is complex and adds a lot of material to what must be transferred.
  - This is often a concern for web-based applications where the overhead in transmitting the data and parsing is large.

JSON

- JSON (JavaScript Object Notation)
- Lightweight data-interchange format
- Meets the same ‘needs’ as XML but is generally preferred for lightweight communication

JSON Data

- Written as collection of (name, value) pairs.
- Pairs separated by commas
- Objects defined as { … }
- Arrays defined as []
- JSON names & strings must appear in double quotes

go
x is 123
y is 456
blue is all this

Console HTML CSS Script DOM Net Coo

all json.js {}

```
1  function go()
2  {
3    var output = document.getElementById("output");
4    var s = '{"x" : 123.0, "y" : 456.0, "blue" : "all this"}';
5    var q = JSON.parse(s);
6    output.innerHTML += "x is " + q.x + "<br>";
7    output.innerHTML += "y is " + q.y + "<br>";
8    output.innerHTML += "blue is " + q.blue + "<br>";
9  }
10
```

From JSON string to object

- q = JSON.parse(‘“blue” : “this”’);
  - Then q.blue == “this”

go

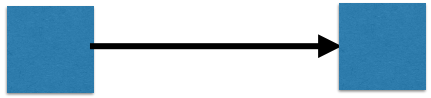
|{"x":123,"y":456,"blue":"all this"}|

Console HTML CSS Script DOM

all json2.js {}

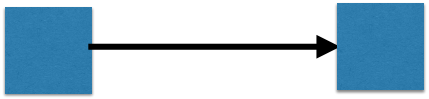
```
1  function go()
2  {
3    var s = {x : 123, y : 456, blue : "all this"};
4    var q = JSON.stringify(s);
5    output.innerHTML = "|"+q+"|<br>";
6  }
7
```

What this does



{x: 2, y : 3}

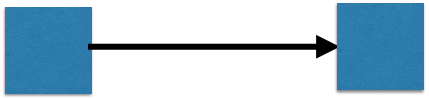
What this does



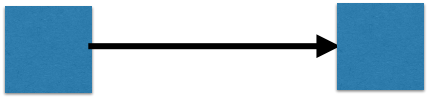
s = JSON.stringify({x: 2, y : 3})

What this does

s = '{"x" : 2, "y" :3}'



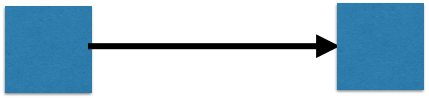
What this does



s = '{"x" : 2, "y" :3}'

q =JSON.parse(s)

What this does



{x: 2, y : 3}

q.x == 2
q.y == 3

What these representations do

- They provide a mechanism for two different software entities to communicate with each other.
- XML is used for larger (more sophisticated) communication.
- JSON is used for lightweight communication.
  - JSON support is built into JavaScript (not surprising given that its Javascript Object Notation).

So lets do something more meaningful

- Suppose we have a source (somewhere) with lots of interesting data accessible as json.
- Then we should be able to query that resource and then convert the data into json and update the user's display based on this.

Hint: we are working towards the last two labs that will use this ability.

Unfortunately…

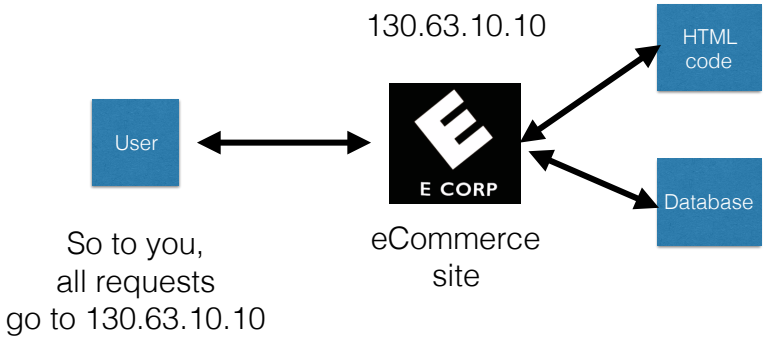
- By default, browsers do not support cross site scripting.

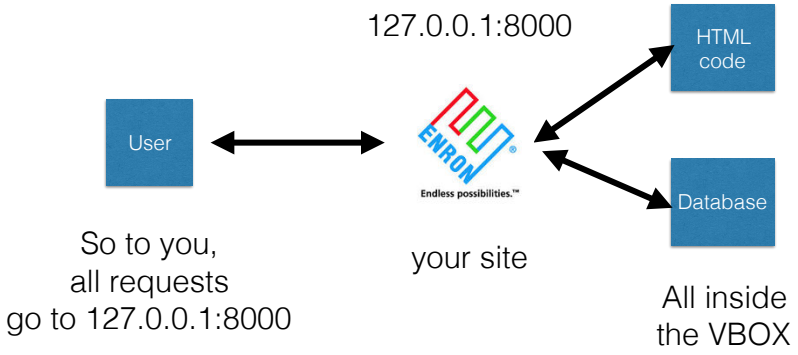
Cross-site scripting (**XSS**) is a type of computer security vulnerability typically found in web applications. **XSS** enables attackers to inject client-side **script** into web pages viewed by other users. A **cross-site scripting** vulnerability may be used by attackers to bypass access controls such as the same-origin policy.

And for good reason…

What this means in practice

- A web page (JavaScript code) is not permitted to access material from another site (cross site) except in very limited ways.
- So this is not a problem in general commercially.



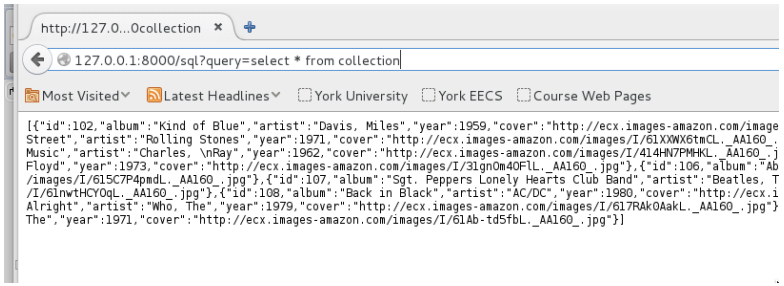


Services provided

- http://127.0.0.1:8000/zip
  - Takes a serve directory -> zip file to html2apk
- http://127.0.0.1:8000/serve
  - Just view the serve directory
- http://127.0.0.1:8000/ip
  - Get your machine's ip address
- http://127.0.0.1:8000/sql
  - access a database (see lab 6, 7 and the next slide)

So, lets go query that database

- Lets not care about how to query it (yet)
  - Neither how to ask the database, nor how to really get the data in JavaScript
- Treat those aspects as magic for now

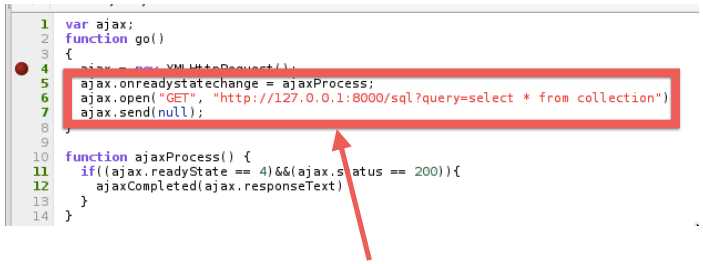


query=….any valid sqlite3 query
but
return is json

XMLHttpRequest

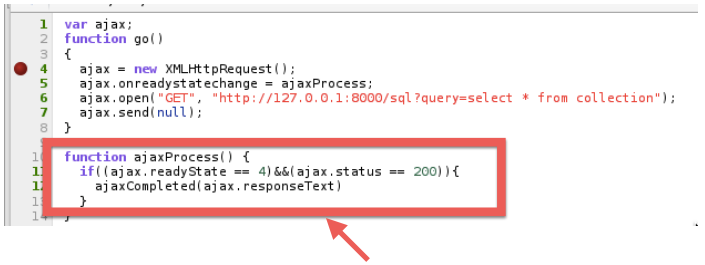
XMLHttpRequest (**XHR**) is an API available to web browser scripting languages such as JavaScript. It is used to send HTTP or HTTPS requests to a web server and load the server response data back into the script.[1] Development versions of all major browsers support URI schemes beyond http and https, in particular, blob URLs are supported.[2]

Data from the response can be used to alter the current document in the browser window without loading a new web page, and despite the name of the API, this data can be in the form of not only XML,[3] but also JSON,[4] HTML or plain text.[5] The response data can also be evaluated by client-side scripting. For example, if it was formatted as JSON by the web server, it can be converted into a client-side data object for further use.



XMLHttpRequest

You specify the URL, and the asynchronous method to be called when the communication state changes



state change callback

You specify the URL, and the asynchronous method to be called when the communication state changes

state: 0 not initialized, 1 setup, 2 sent, 3 in progress, 4 completed
status 200 - ok, 404 not found

```
function ajaxCompleted(text) {
  var output = document.getElementById("output");
  var data = JSON.parse(text);
```

```
function ajaxCompleted(text) {
  var output = document.getElementById("output");
  var data = JSON.parse(text);
  var i;
  for(i=0;i<data.length;i++) {
    output.innerHTML += data[i].album + "<br>";
  }
```

```
function ajaxCompleted(text) {
  var output = document.getElementById("output");
  var data = JSON.parse(text);
  var i;
  for(i=0;i<data.length;i++) {
    output.innerHTML += data[i].album + "<br>";
  }
  for(i=0;i<data.length;i++) {
    var p = document.createElement("img");
    p.src =data[i].cover;
    output.appendChild(p);
  }
}
```

Communicating between processes

- Given the right infrastructure (a data source that communicates in an easily interpretable format) and mechanisms to take that data and incorporate it into your web page, you can write quite sophisticated software easily.
- You can see the effect of this in everything from ‘Find of Friend’ apps (e.g., IOS) to E-commerce storefronts.