The Learning House
427 S. 4th Street #300
Louisville KY 40202

Web Apps and Spring Core MVC Unit

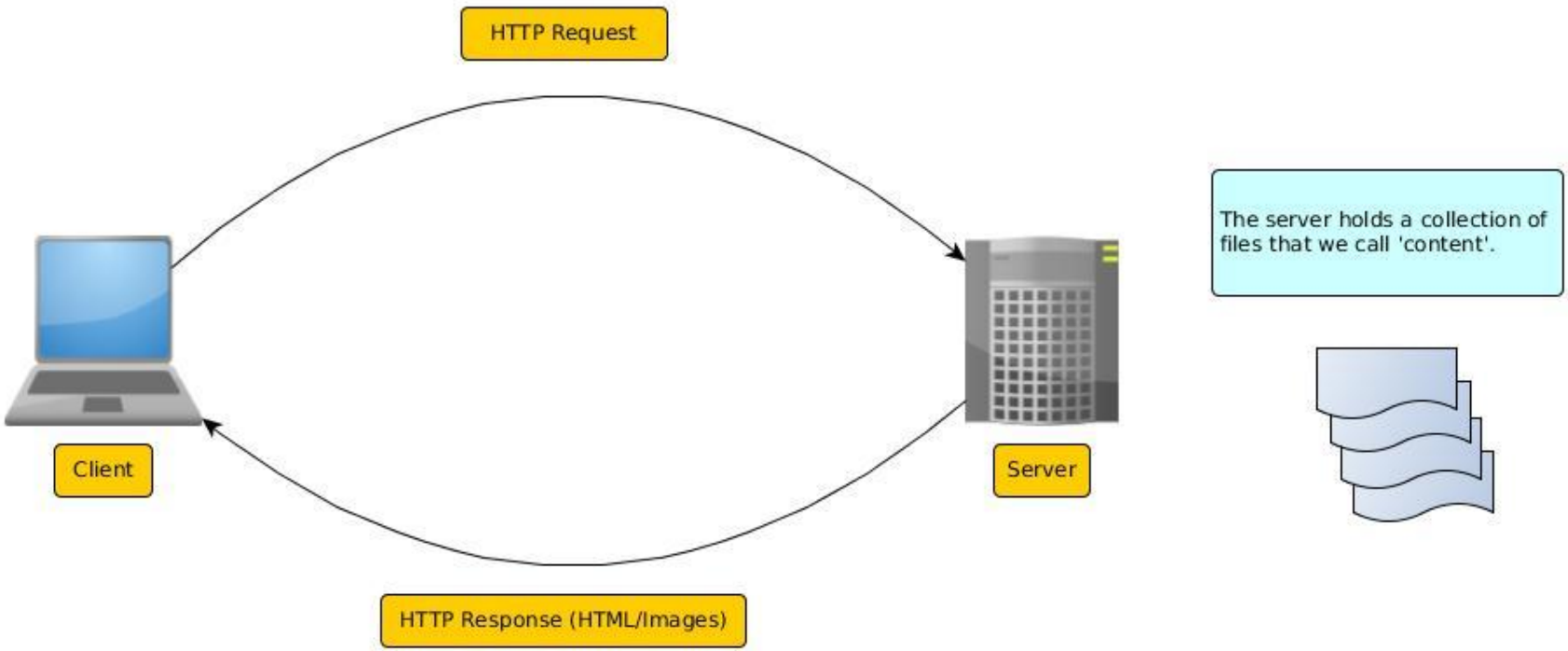
Lesson 1 - Intro to HTTP and Web Applications

SOFTWARE GUILD

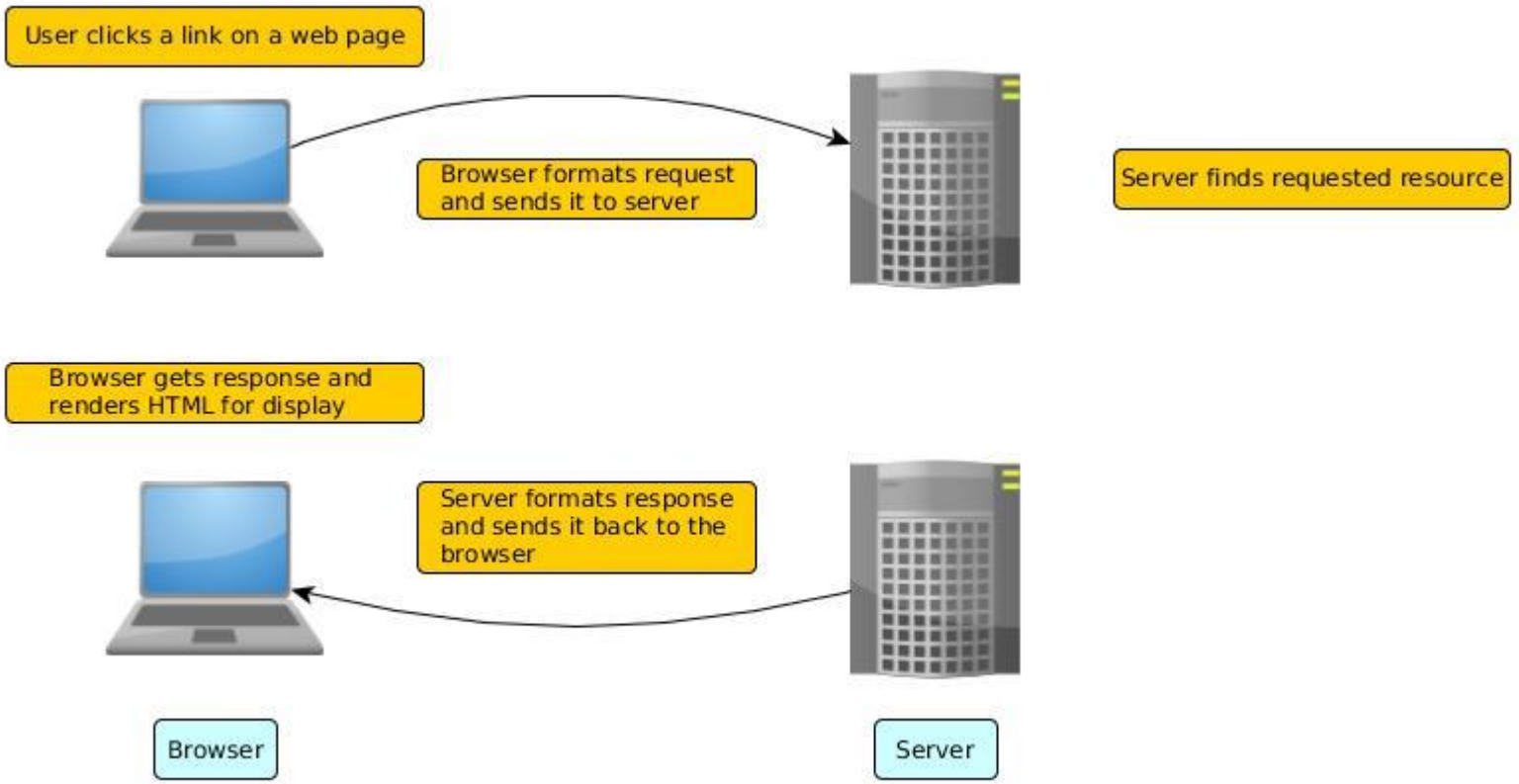
Objectives

- Gain a high level understanding of:
  - What web servers are and how they work
  - What web clients are and what they do
  - How HTML fits into the picture
  - The HTTP Protocol
  - What URL's are
  - The major components of a Java Web Application
- NOTE: Thanks to 'Head First Servlets and JSP' for some of the images.

What does a web server do?



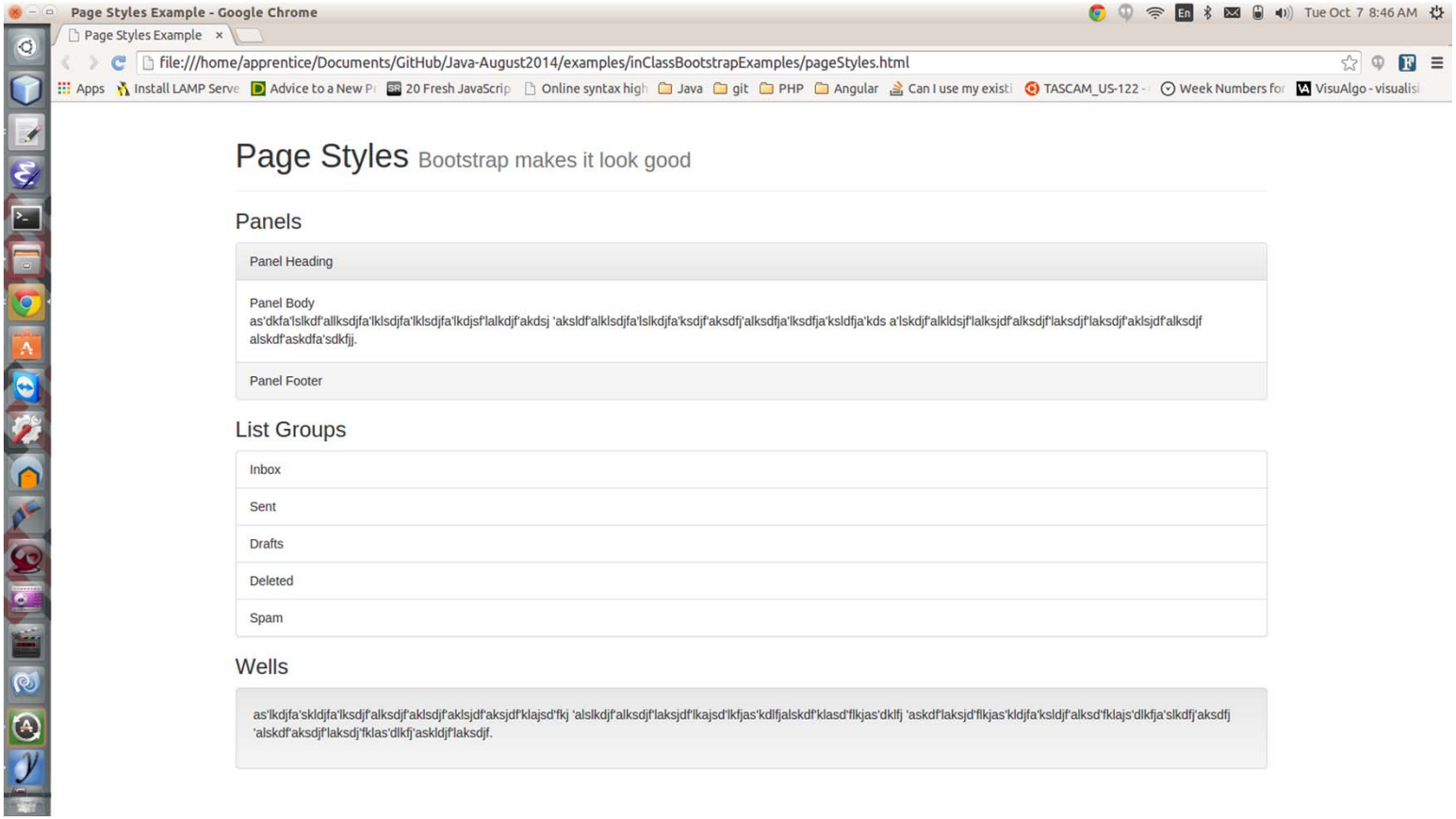
What does a web client do?



You write HTML

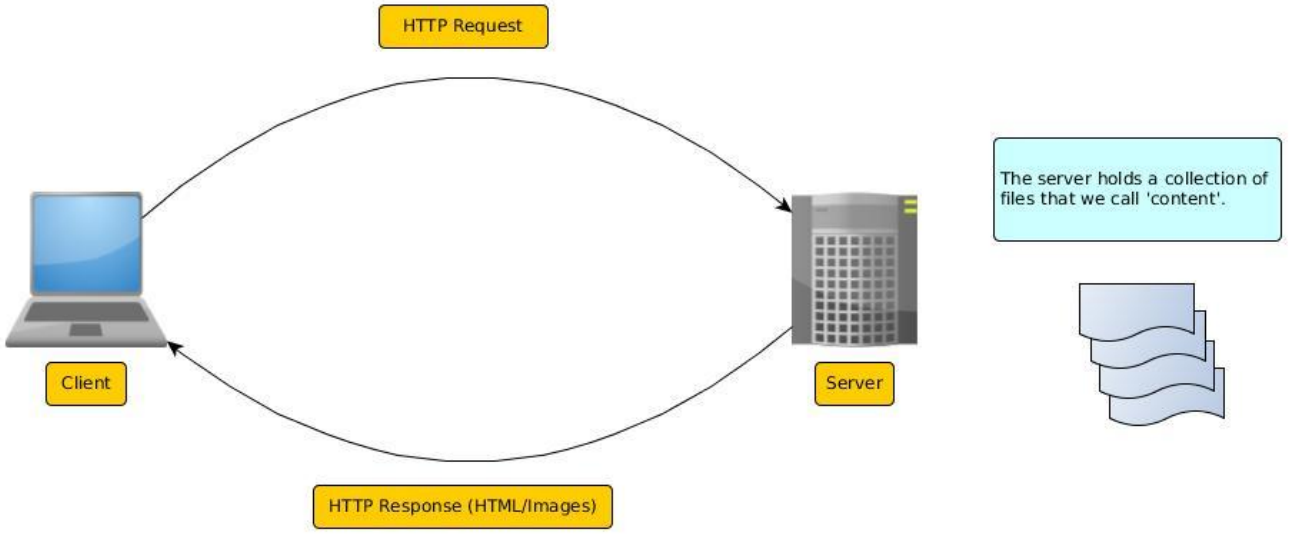
```
<!DOCTYPE>
<html>
    <head>
        <title>Page Styles Example</title>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
        <link rel="stylesheet" type="text/css" href="css/bootstrap-theme.css">
    </head>
    <body>
        <div class="container">
            <div class="page-header">
                <h1>Page Styles <small>Bootstrap makes it look good</small></h1>
            </div>
            <h3>Panels</h3>
            <div class="panel panel-default">
                <div class="panel-heading">
                    Panel Heading
                </div>
                <div class="panel-body">
                    Panel Body<br/>
                    as'dkfa'lslkdf'allksdjfa'lklsdjfa'lklsdjfa'lkdjsf'lalkdjf'akdsj
                    'aksldf'alklsdjfa'lslkdjfa'ksdjf'aksdfj'alksdfja'lksdfja'ksldfja'kds
                    a'lskdjf'alkldsjf'lalksjdf'alksdjf'laksdjf'laksdjf'aklsjdf'alksdjf
                    alskdf'askdfa'sdkfjj.
                </div>
                <div class="panel-footer">
                    Panel Footer
                </div>
            </div>
```

The browser renders



HTTP

- All Web Servers speak HTTP
- HTTP is a request/response protocol

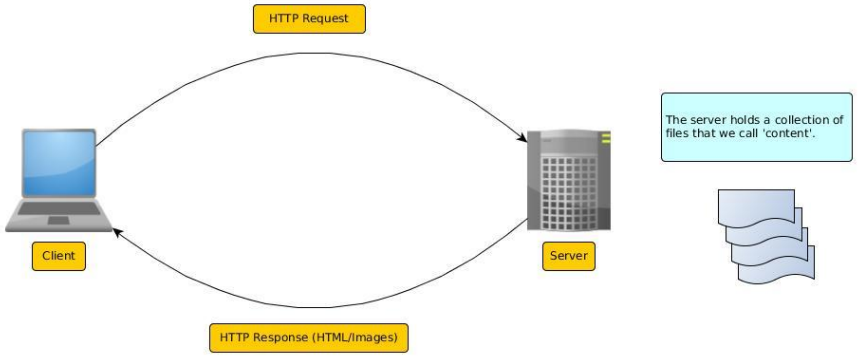


HTTP – Request/Response

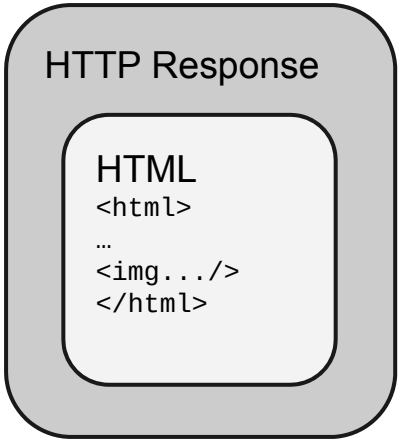
- Request:
  - HTTP method to be performed
  - URL of page to access
  - Parameters (form data - like parameters to a method)
- Response:
  - Status code
  - Content type (text, picture, HTML, PDF, etc.)
  - Content

HTML is part of the response

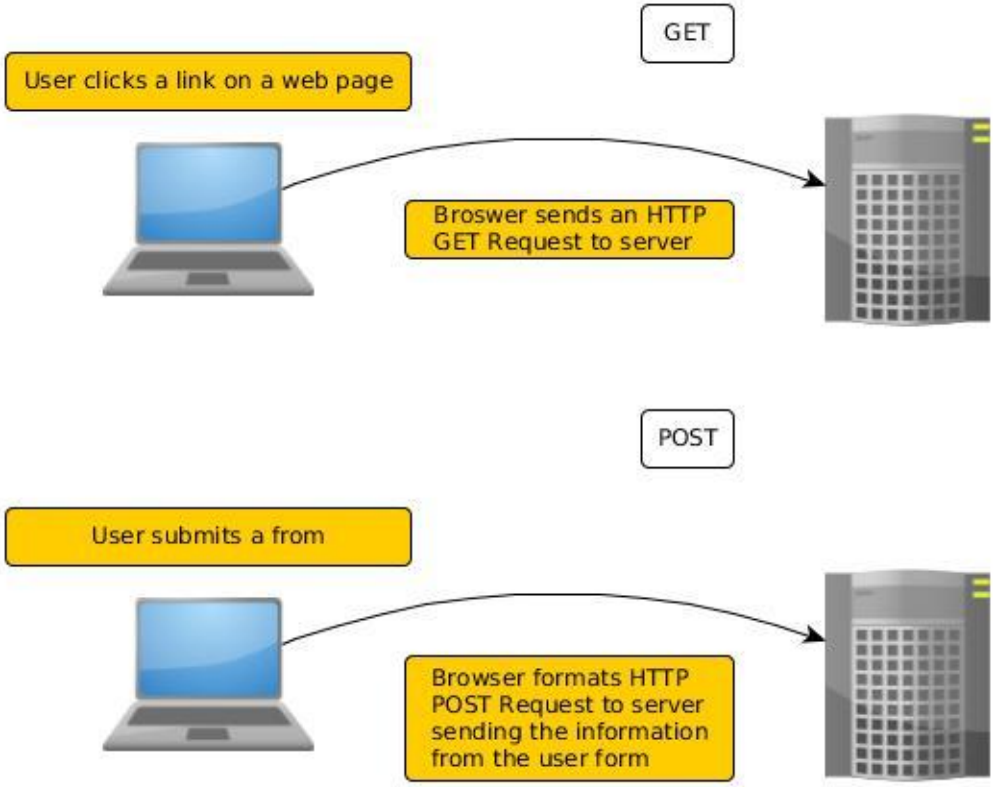
- The browser looks for the opening <html> tag and begins to render the page.



- When the browser encounters an <img> tag it issues another HTTP request to fetch the image.



The Request



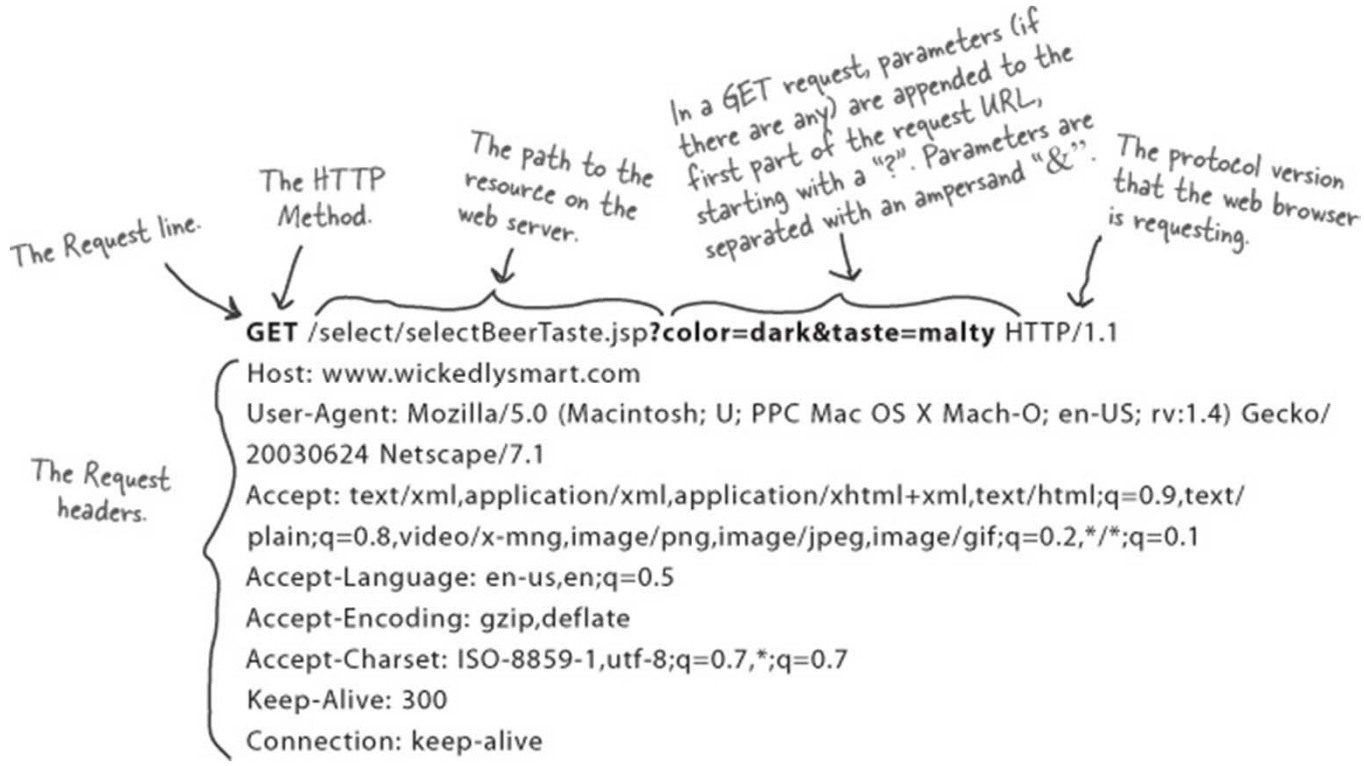
GET and POST

- GET and POST are two of the HTTP methods
- GET is a simple request for a resource on the server
- POST can send user data to the server via a form
- GET can send limited data to the server via URL parameters (i.e. www.site.com?id=foo)

A GET Request

- GET Request consists of:
  - Server (i.e. www.myserver.com)
  - Path to resource (i.e. /addresses/findAddresses.jsp)
  - Parameters (i.e. zipcode = 44311)

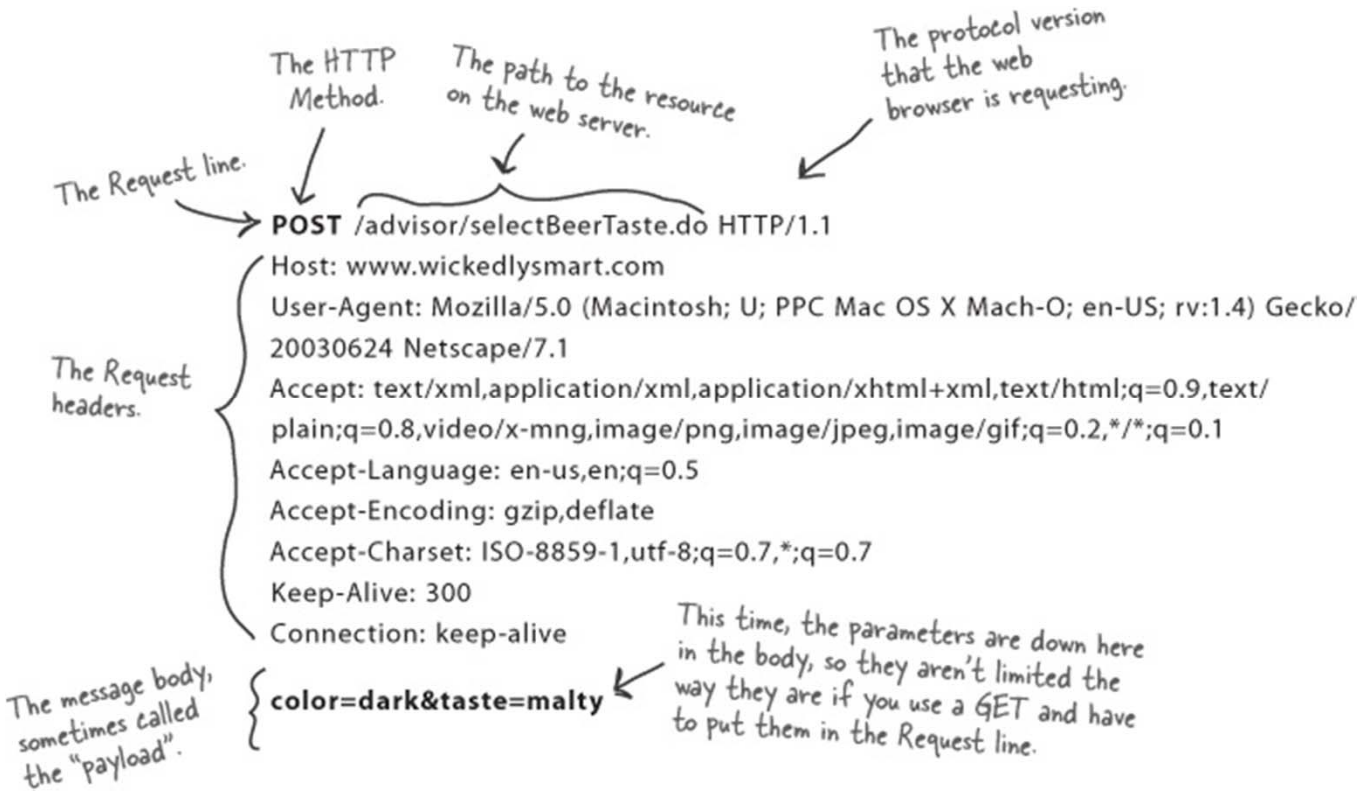
Details of a GET Request



A POST Request

- POST Request consists of:
  - o Server (i.e. www.myserver.com)
  - o Path to resource (i.e. /addresses/findAddresses.jsp)
  - o Parameters (i.e. zipcode = 44311)
- Differs from GET Request because parameters are part of the message body, not part of the URL

Details of a POST Request



HTTP Response

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 8259
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
Vary: Accept-Encoding
Server: Microsoft-IIS/8.0
X-AspNetMvc-Version: 4.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Tue, 07 Oct 2014 13:37:39 GMT
```

URL

- Stands for Uniform Resource Locator

  http://www.myserver.com:80/addresses/findAddress.jsp

Protocol (http://): tells the server which protocol we want to use

Server (www.myserver.com): name of server you want to contact

Port (80): Usually not present in web requests because it defaults to 80 which is the standard port for web servers

Path (/addresses): Path on server to requested resource

Resource (findAddress.jsp): name of the content you are requesting

Dynamic vs. Static Content

- Everything we have discussed works for static content
- Web servers are great at delivering this type of content
- Web servers alone cannot:
  - Server dynamic content (i.e. the current time)
  - Save data to the server (i.e. in a database)
- Dynamic content requires a web application

Java Web Applications

- We will use Servlets and JSP to create our web application
- Servlets do not have a main() method like our previous projects
- Servlet are under the control of a Servlet Container (Tomcat in our case)
- A Java web app consists of one or more Servlets and JSPs that handle HTTP requests

Container Benefits

- Protocol and communication support
  - o Socket/network connections
  - o HTTP protocol
- Lifecycle management
  - o Controls the life and death of servlets
- Multithreading support
  - o Handles multiple requests automatically
- Security
- JSP Support

Container Request Handling

- Our ‘container’ will be Tomcat
- There are others
  - o Jetty
  - o Resin
  - o WebLogic
  - o JBoss

Container Request Handling (2)

- Container looks at request and routes it to appropriate Java web application (i.e. to the set of Servlets and JSPs that we have created to handle certain requests)
- Container creates an HttpServletRequest and HttpServletResponse object and passes them to our Servlet
  - These objects represent the raw HTTP request and response information

Container Request Handling (3)

- After container hands us the request and response objects, our code looks at the request to see what it should do (i.e. look up addresses from the 44311 zipcode)
- Our code does its work and the formats the data (i.e. the list of addresses) into HTML, modifies the request and response appropriately and returns control to the container