Lecture 3
Handling the Client Request: Form Data

Lecture Agenda
Applied

1. The role of form data.
2. Creating and submitting HTML forms.
3. Reading individual request parameters.
4. Reading the entire set of request parameters.
5. Handling and missing malformed data.
6. Dealing with incomplete form submissions.
7. Filtering special characters out of the request parameters.

Lecture Agenda
Knowledge

1. The HTTP Protocol.
2. HTTP POST and GET Methods.
3. Advantages and Disadvantages of POST and GET.
4. List of HTTP Methods.
5. HTML / HTML Form Basics.
6. HTTP GET Header / HTTP POST Header.

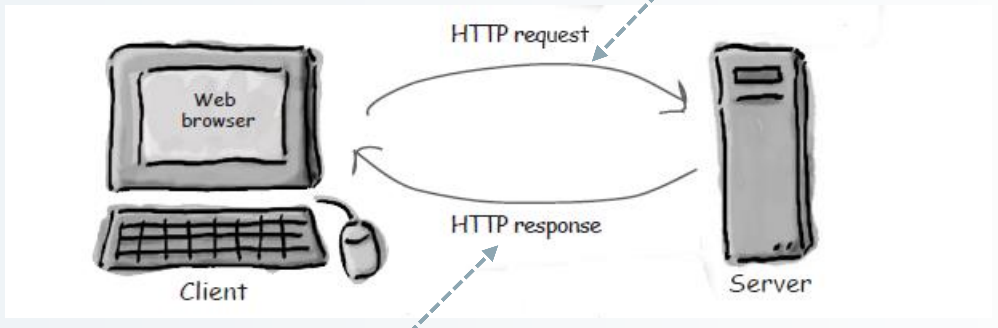
The HTTP Protocol

What is the HTTP Protocol?

Request and Response

Key elements of request stream:

- HTTP Method (the action to be performed).
- The page to access (URL)
- Form parameters (similar to arguments to a method).



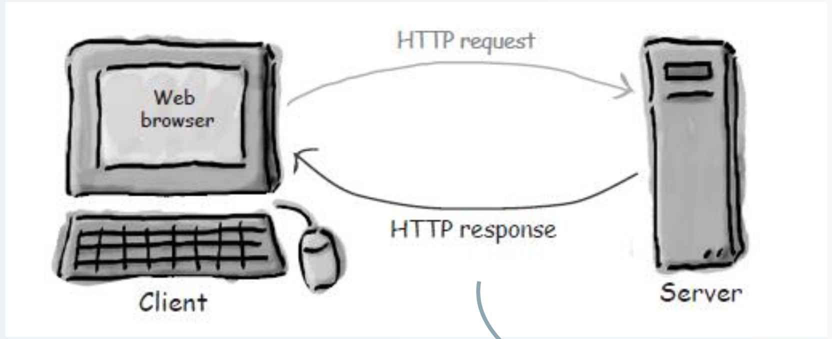
- HTTP protocol runs on top of TCP/IP.
- TCP is responsible for making sure that a file sent from once network node to another ends up as a complete file at the destination.
- request/response sequence; a browser sends a request, and a server responds.

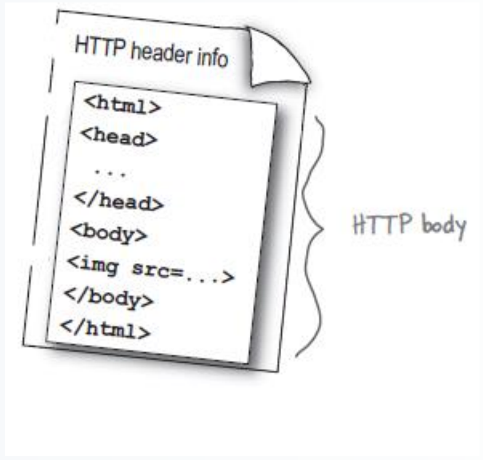
Key elements of response stream:

- A status code (for whether the request was successful).
- Content-type (text, picture HTML, etc.)
- The content (the actual HTML, image, etc.)

The HTTP Response

HTML is part of the HTTP response





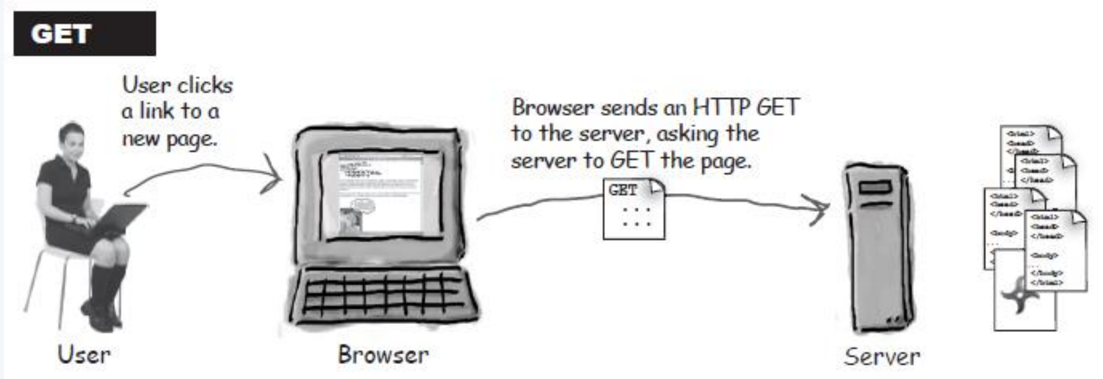
```
<html>
<head>
...
</head>
<body>
<img src=...>
</body>
</html>
```

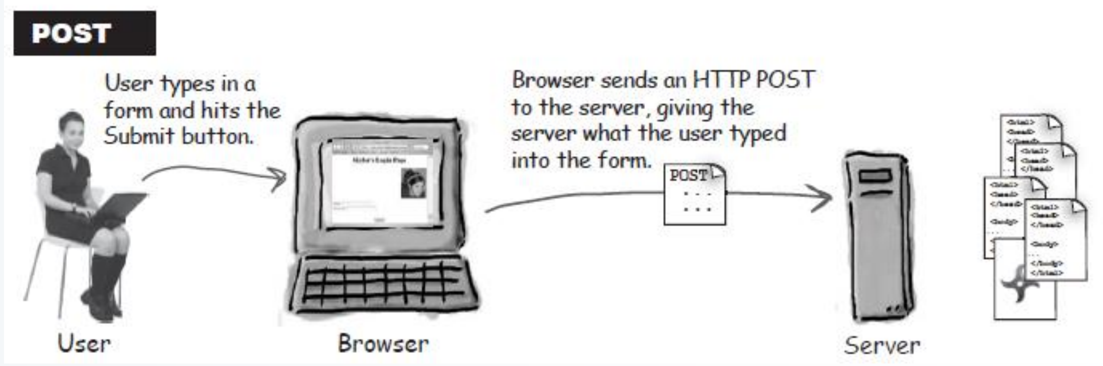
- A HTTP response can contain HTML.
- HTTP adds header information on top of whatever content in the response.
- A browser will use the header information to help process the HTML page.
- Think of HTML content as data pasted inside an HTTP response.

HTTP Methods

HTTP GET/POST Method

- The first thing you’ll find is an HTTP method name.
- The method names are not Java methods, but the idea is similar.
- The method name tells the server the kind of request that’s being made and how the rest of the message will be formatted.
- The HTTP protocol has several methods, but the ones you’ll use the most often are GET and POST.


GET
User clicks a link to a new page.
Browser sends an HTTP GET to the server, asking the server to GET the page.
GET
User
Browser
Server

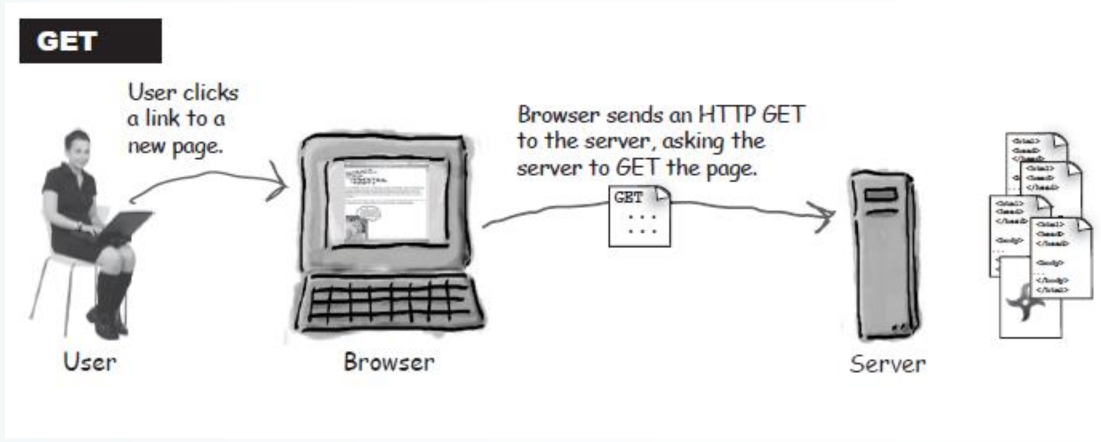

POST
User types in a form and hits the Submit button.
Browser sends an HTTP POST to the server, giving the server what the user typed into the form.
POST
User
Browser
Server

HTTP GET

GET is a simple request.

GET Request Key Elements:

- GET is the simplest HTTP method.
- GET has a main job, to ask the server to ***get*** a resource and send it back to the caller/client.
- The resource might be a HTML page, a jpeg, a PDF etc..
- The point of GET is to get a resource from the server.

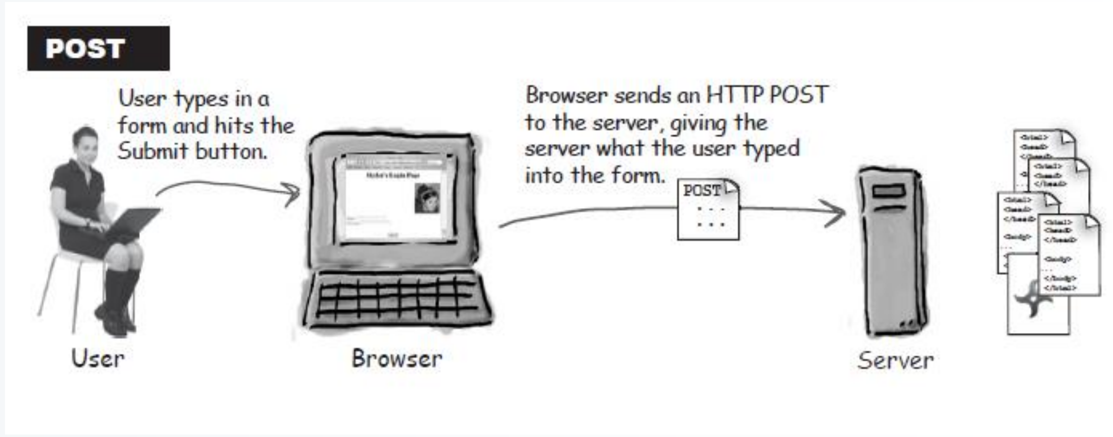

GET
User clicks a link to a new page.
Browser sends an HTTP GET to the server, asking the server to GET the page.
GET
User
Browser
Server

HTTP POST

POST can send user data.

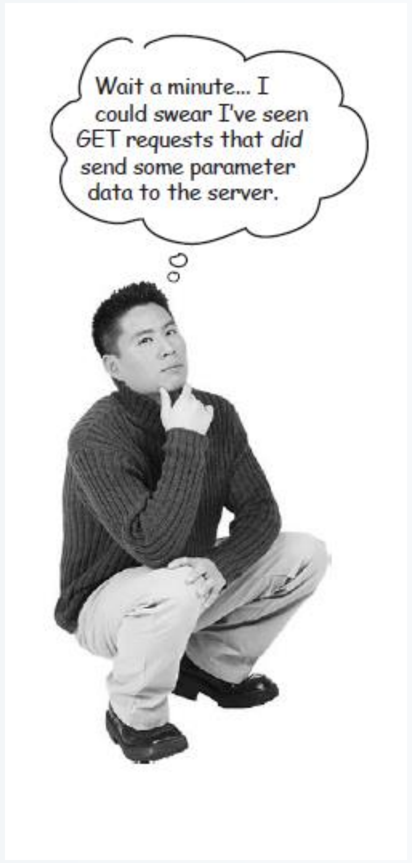
POST Request Key Elements:

- POST is a more powerful HTTP request method.
- POST is like GET + plus added capabilities.
- POST will allow the caller to request something (a resource) and at the same time send form data to the server.


POST
User types in a form and hits the Submit button.
Browser sends an HTTP POST to the server, giving the server what the user typed into the form.
POST
User
Browser
Server

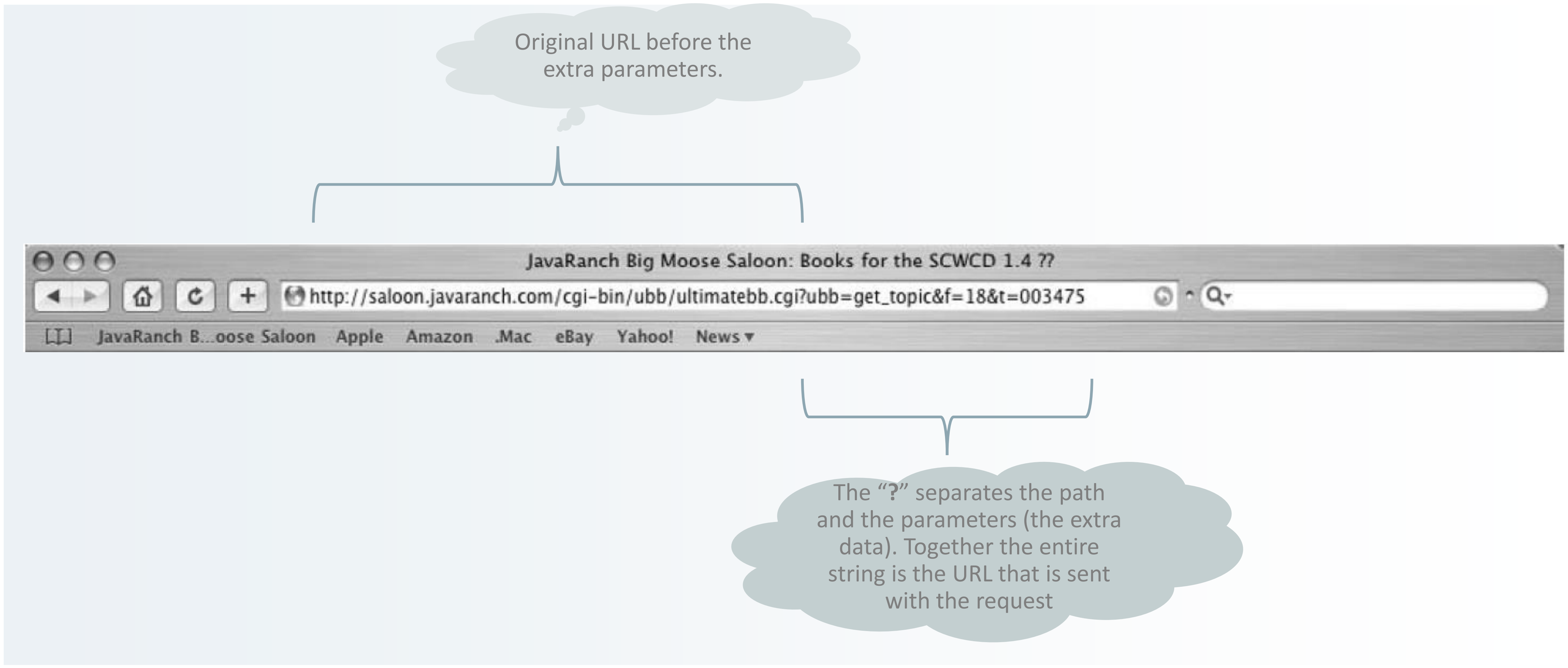
GET requests

Can get request send data?


Wait a minute... I could swear I've seen GET requests that *did* send some parameter data to the server.

GET requests

Its true … you can send a little data with HTTP GET


Original URL before the extra parameters.
JavaRanch Big Moose Saloon: Books for the SCWCD 1.4 ??
JavaRanch B...oose Saloon Apple Amazon .Mac eBay Yahoo! News
The "**?**" separates the path and the parameters (the extra data). Together the entire string is the URL that is sent with the request

```
http://saloon.javaranch.com/cgi-bin/ubb/ultimatebb.cgi?ubb=get_topic&f=18&t=003475
```

POST vs GET

Reasons you might use POST instead of GET

- The total amount of characters in a GET is limited. For example, if the user types a long passage into a search input box, the GET might not work.
- The data sent with a GET is appended to the URL up in the browser bar, therefore whatever sent is exposed (password or sensitive data).
- The user cannot bookmark a form submission if you use POST instead of GET. Depending on your application and design, you may or may not want users to be able to bookmark the resulting request from a form submission.

GET Method

Advantages and Disadvantages

Advantages

- URLs can be bookmarked safely.
- Pages can be reloaded safely.
- Can bookmark pages.
- Browsers can cache results
- Easier to test interactively

Disadvantages

- Variables are passed through url as name-value pairs. (Security risk).
- Limited number of variables that can be passed (browser based ex. IE limit is 2048characters).

POST Method

Advantages and Disadvantages

Advantages

- Name-value pairs are not displayed in URL.
- Unlimited number of name-value pairs can be passed.
- URL is simpler.
- Special characters can be sent
- Browsers will not cache results

Disadvantages

- Page cannot be bookmarked.
- Performance response degraded

GET vs POST. When to use Which?

General Practice

GET

- Usually used for submitted search requests.
- Usually used where you want the user to be able to pull up the page again.

POST

- Used for higher security requests where data may be passed to alter a database.
- Usually used where you do not want the user to bookmark the page.

The Remaining HTTP methods

HTTP methods beside GET and POST

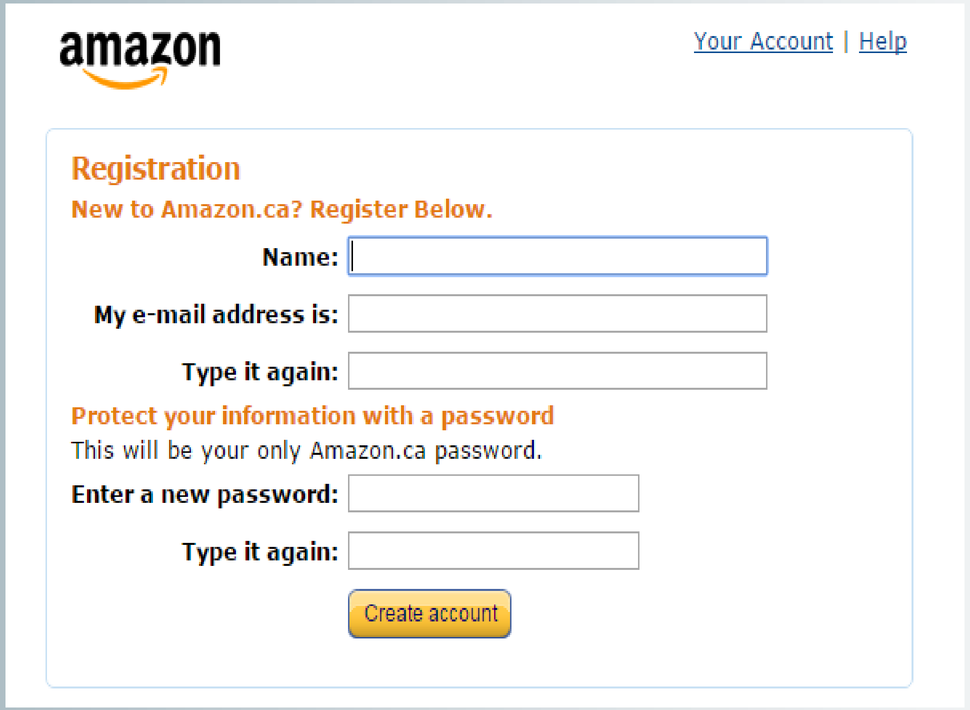
List of HTTP Methods:

1. **GET**
2. **POST**
3. HEAD
4. TRACE
5. PUT
6. DELETE
7. OPTIONS
8. CONNECT

GET and POST are the two big methods . The remaining methods are rarely used.

Form Basics

The Role of Form Data


amazon

Your Account | Help

Registration

New to Amazon.ca? Register Below.

Name:	
My e-mail address is:	
Type it again:	

Protect your information with a password

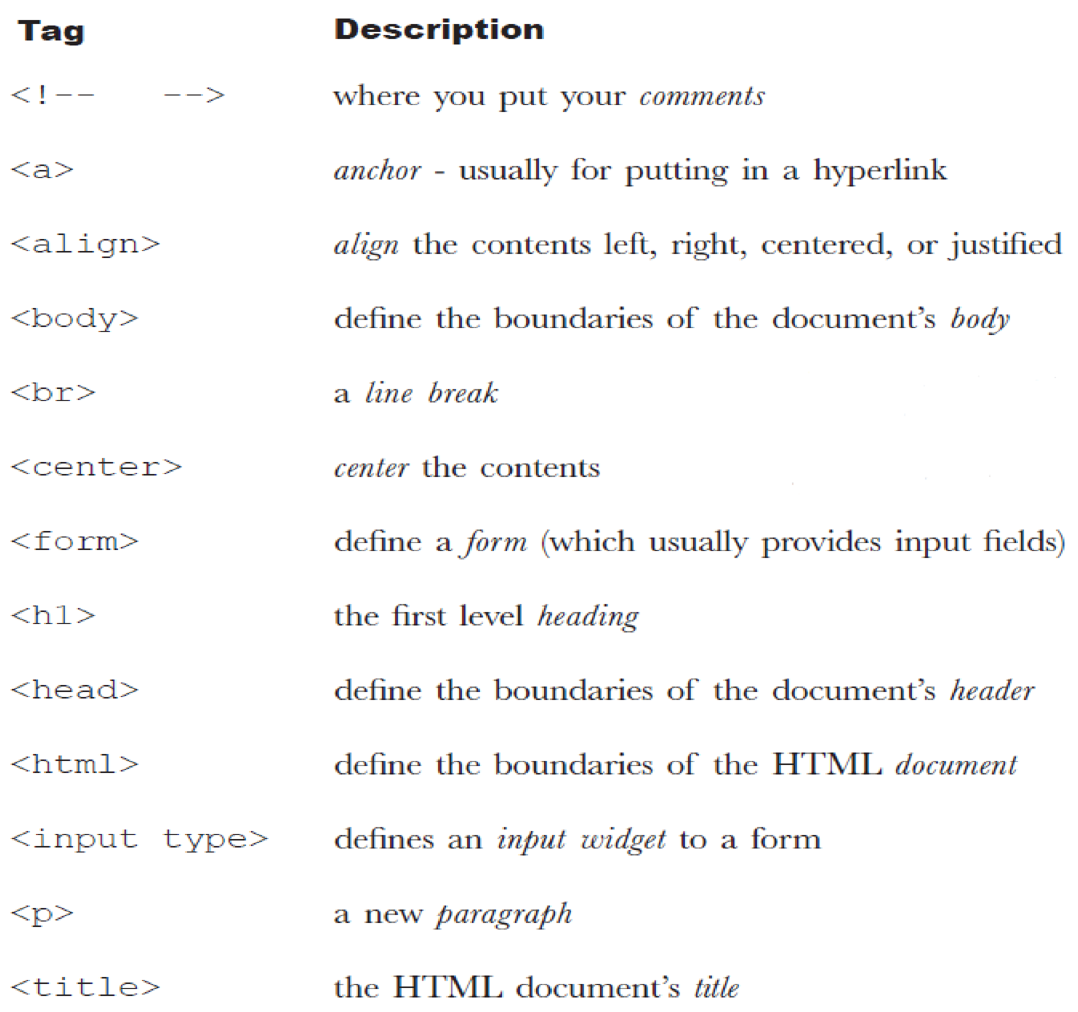
This will be your only Amazon.ca password.

Enter a new password:	
Type it again:	

Create account

HMTL Crash course

Two-minute HTML guide

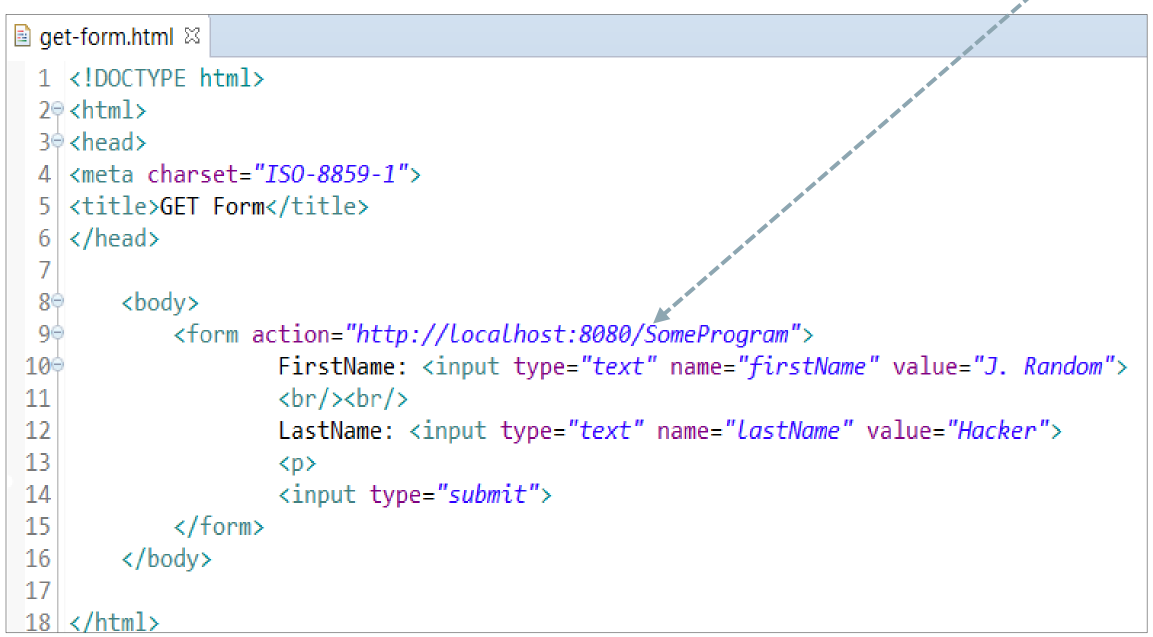


Tag	Description
`<!-- -->`	where you put your *comments*
`<a>`	*anchor* - usually for putting in a hyperlink
`<align>`	*align* the contents left, right, centered, or justified
`<body>`	define the boundaries of the document's *body*
`<br>`	a *line break*
`<center>`	*center* the contents
`<form>`	define a *form* (which usually provides input fields)
`<h1>`	the first level *heading*
`<head>`	define the boundaries of the document's *header*
`<html>`	define the boundaries of the HTML *document*
`<input type>`	defines an *input widget* to a form
`<p>`	a new *paragraph*
`<title>`	the HTML document's *title*

Creating Form Data

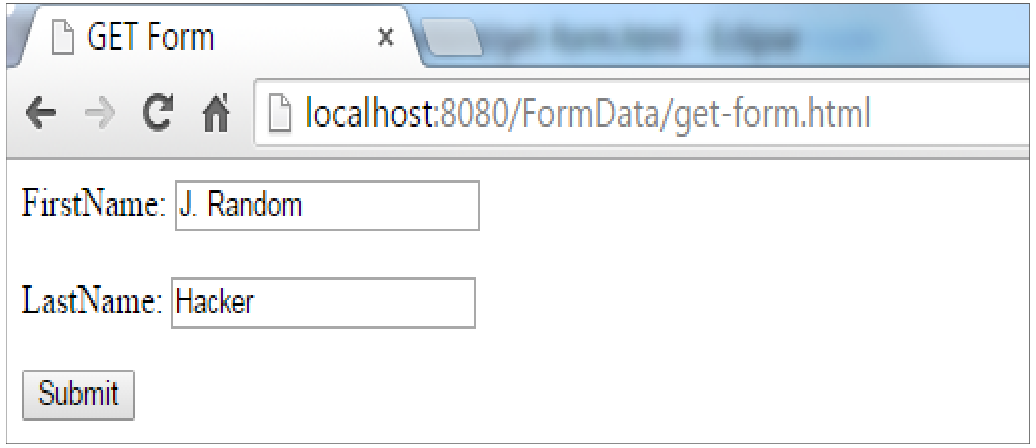
HTML Forms

Normally a relative address is used for "action" not a full address as shown.



get-form.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>GET Form</title>
</head>

    <body>
        <form action="http://localhost:8080/SomeProgram">
                FirstName: <input type="text" name="firstName" value="J. Random">
                <br/><br/>
                LastName: <input type="text" name="lastName" value="Hacker">
                <p>
                <input type="submit">
        </form>
    </body>

</html>
```



GET Form

localhost:8080/FormData/get-form.html

FirstName: J. Random

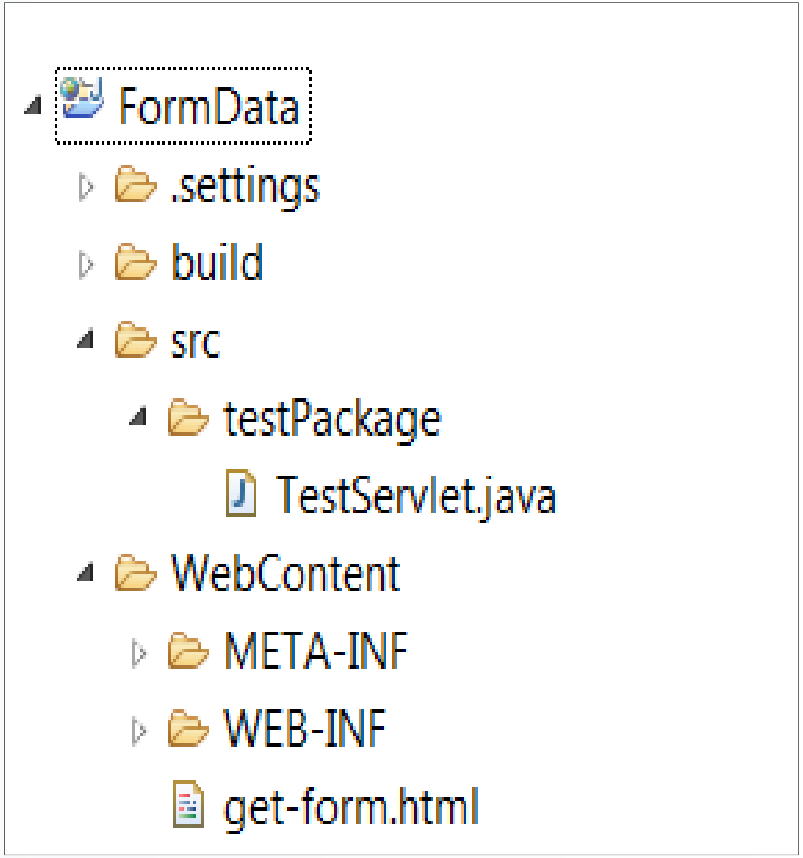
LastName: Hacker

Submit

Installing HTML Files

HTTP GET/POST Method

- HTML file do <u>**not**</u> go in src
  - HTML files go in **WebContent**
    - When deployed **WebContent** beomes the top-level web application directory.
    - In contrast, code under **src** get deployed to the **WEB_INF\classes** folder of the Web app directory.
- Example Eclipse project name: FormData
  - Files
    - WebContent/get-form.html
  - URLs
    - http://localhost/FormData/get-form.html

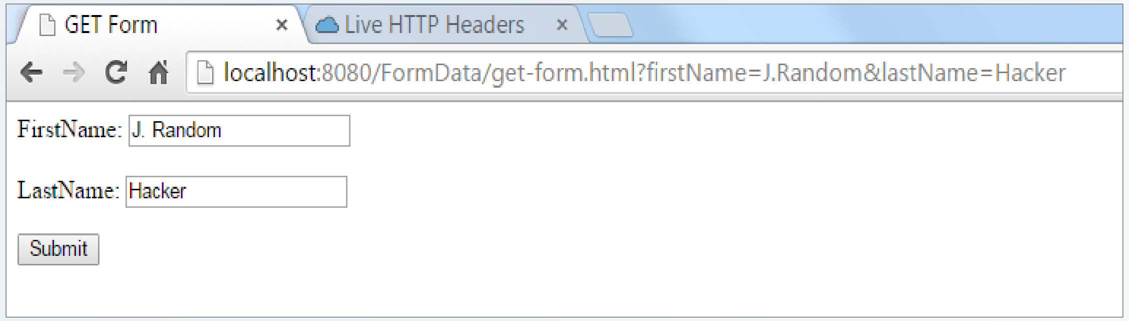


```
FormData
  .settings
  build
  src
    testPackage
      TestServlet.java
  WebContent
    META-INF
    WEB-INF
    get-form.html
```

GET Form

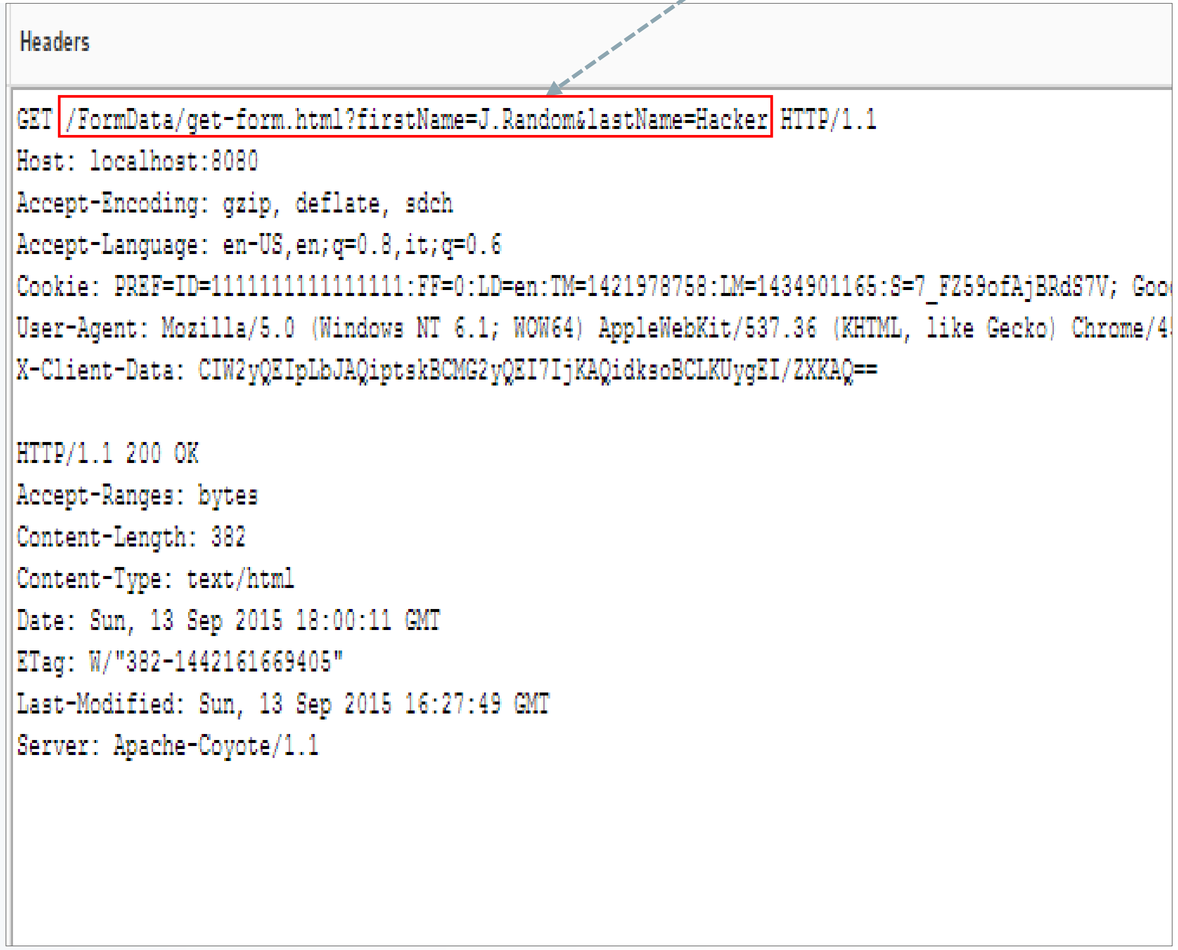
Submission Result (HTTPLive Headers - Chrome)

Name/Value parameter pairs passed.


GET Form
Live HTTP Headers
localhost:8080/FormData/get-form.html?firstName=J.Random&lastName=Hacker
Submit

FirstName: J. Random

LastName: Hacker

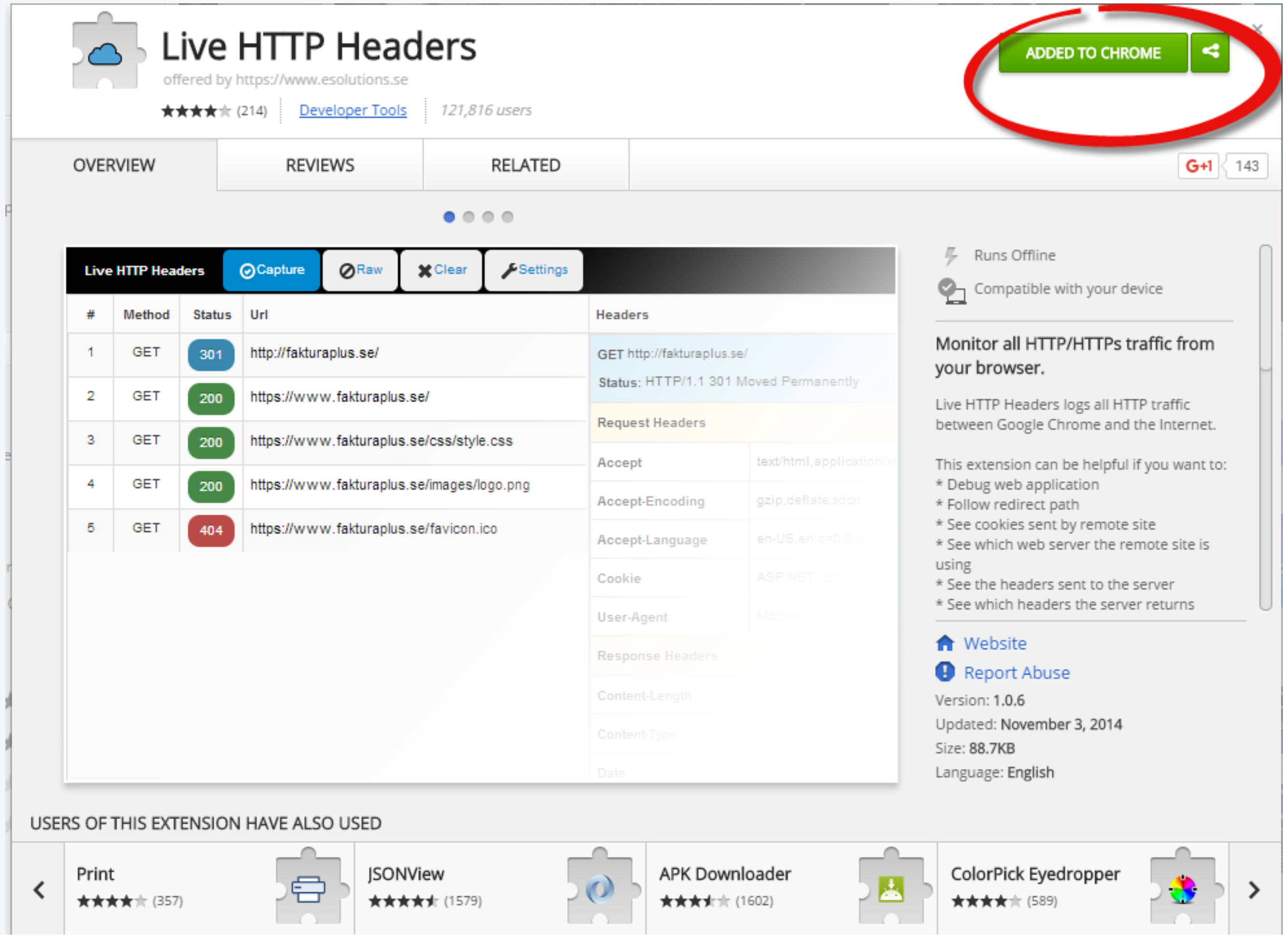


```
GET /FormData/get-form.html?firstName=J.Random&lastName=Hacker HTTP/1.1
Host: localhost:8080
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,it;q=0.6
Cookie: PREF=ID=1111111111111111:FF=0:LD=en:TM=1421978758:LM=1434901165:S=7_FZ59ofAjBRdS7V; Goo
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/4
X-Client-Data: CIW2yQEIpLbJAQiptskBCMG2yQEI7IjKAQidksoBCLKUygEI/ZXKAQ==

HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Length: 382
Content-Type: text/html
Date: Sun, 13 Sep 2015 18:00:11 GMT
ETag: W/"382-1442161669405"
Last-Modified: Sun, 13 Sep 2015 16:27:49 GMT
Server: Apache-Coyote/1.1
```

Live HTTP Headers – Chrome Browser

Tool for Viewing Live HTTP Headers.

- Search for Live HTTP Headers for Chrome
  - Live Headers logs all HTTP traffic between Chrome and the internet.



Live HTTP Headers

offered by https://www.esolutions.se

★★★★☆ (214) | Developer Tools | 121,816 users

#	Method	Status	Url
1	GET	301	http://fakturaplus.se/
2	GET	200	https://www.fakturaplus.se/
3	GET	200	https://www.fakturaplus.se/css/style.css
4	GET	200	https://www.fakturaplus.se/images/logo.png
5	GET	404	https://www.fakturaplus.se/favicon.ico

Headers

GET http://fakturaplus.se/

Status: HTTP/1.1 301 Moved Permanently

Request Headers

Accept	text/html,application/x
Accept-Encoding	gzip,deflate,sdch
Accept-Language	en-US,en;q=0.8,
Cookie	ASP.NET_S
User-Agent	Mozilla

Response Headers

Content-Length

Content-Type

Date

Runs Offline

Compatible with your device

Monitor all HTTP/HTTPs traffic from your browser.

Live HTTP Headers logs all HTTP traffic between Google Chrome and the Internet.

This extension can be helpful if you want to:
* Debug web application
* Follow redirect path
* See cookies sent by remote site
* See which web server the remote site is using
* See the headers sent to the server
* See which headers the server returns

Website

Report Abuse

Version: 1.0.6

Updated: November 3, 2014

Size: 88.7KB

Language: English

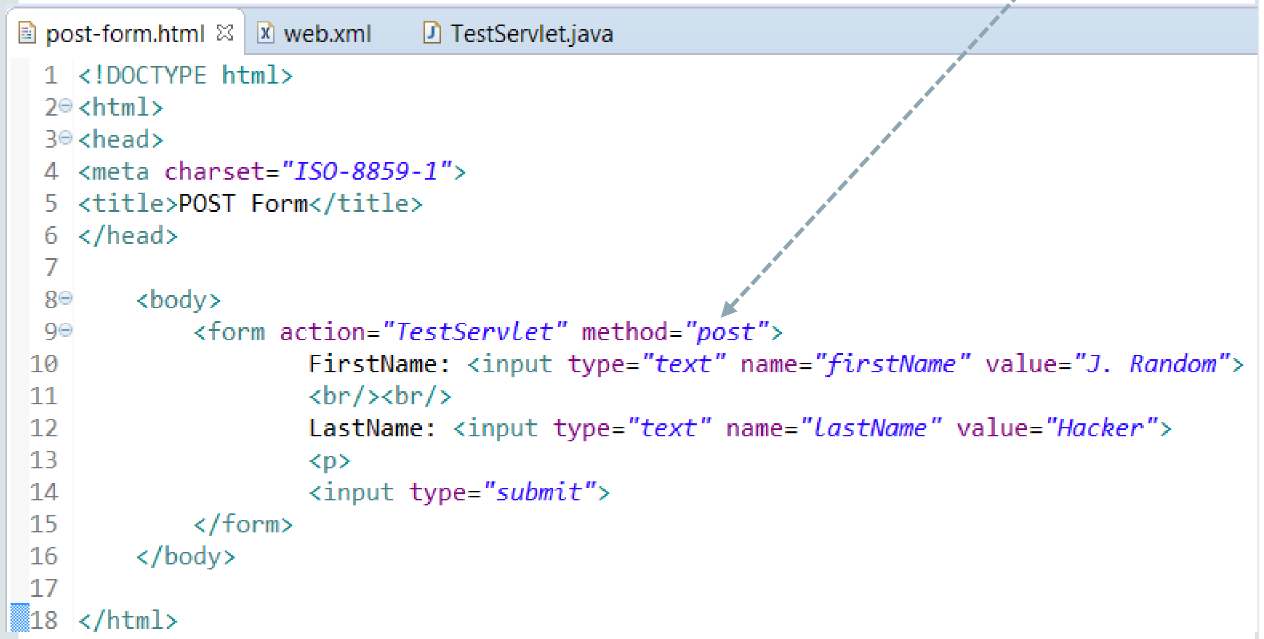
USERS OF THIS EXTENSION HAVE ALSO USED

Print ★★★★☆ (357)	JSONView ★★★★★ (1579)	APK Downloader ★★★★☆ (1602)	ColorPick Eyedropper ★★★★☆ (589)

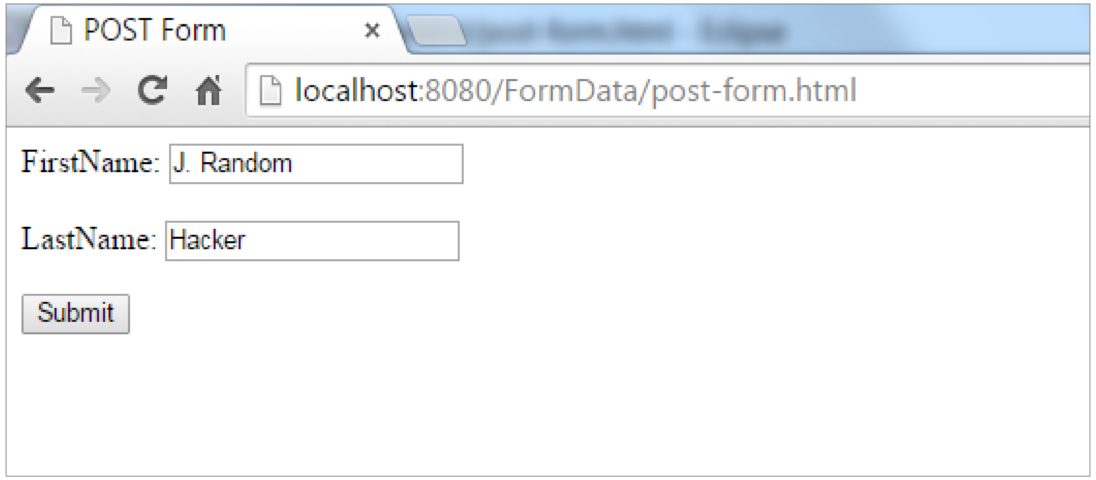
Sending POST Data

HTML Forms

The default method is GET. So, if a form says method="GET" or it has no method at all, GET is used.

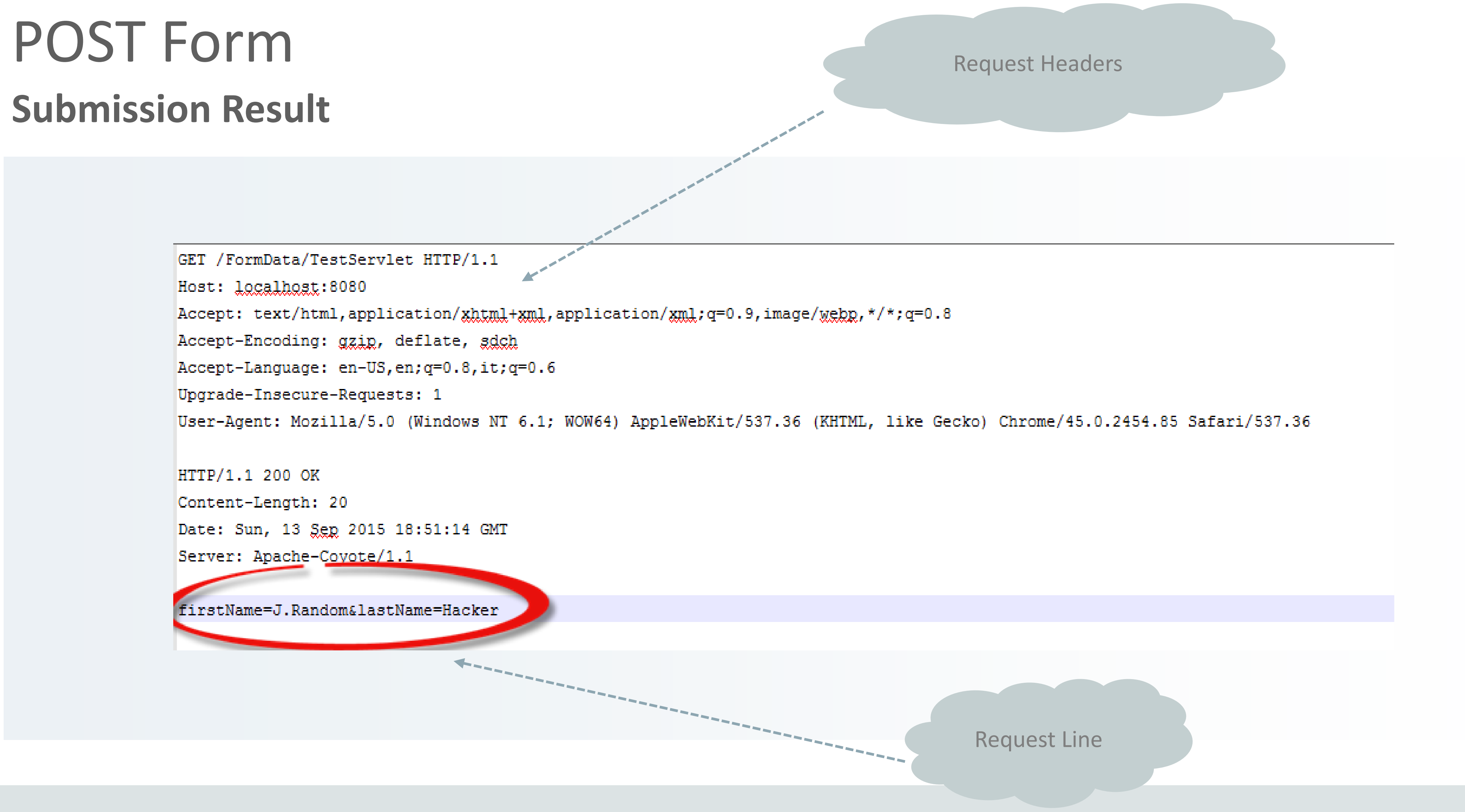


```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>POST Form</title>
</head>

    <body>
        <form action="TestServlet" method="post">
                FirstName: <input type="text" name="firstName" value="J. Random">
                <br/><br/>
                LastName: <input type="text" name="lastName" value="Hacker">
                <p>
                <input type="submit">
        </form>
    </body>

</html>
```



POST Form

Submission Result



```
GET /FormData/TestServlet HTTP/1.1
Host: localhost:8080
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,it;q=0.6
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.85 Safari/537.36

HTTP/1.1 200 OK
Content-Length: 20
Date: Sun, 13 Sep 2015 18:51:14 GMT
Server: Apache-Coyote/1.1

firstName=J.Random&lastName=Hacker
```

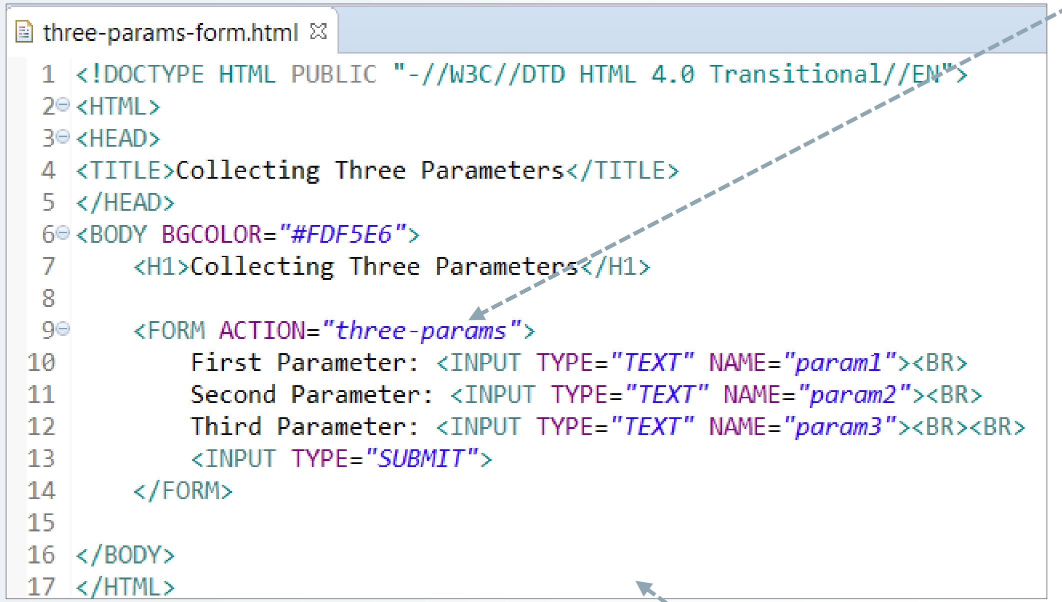
Reading Form Data In Servlets

Methods to read form data

- request.getParameter(“name”)
  - Returns URL-decoded value of first occurrence of name in query string.
  - Works Identically for GET and POST requests
  - Returns null if no such parameter is in query data
- request.getParameterValues(“name”)
  - Returns an array of the URL-decoded values of all occurrence of name in query string.
  - Returns a one-element array if param not repeated
  - Returns null if no such parameter is in query.
- request.getParameterNames() or request.getParameterMaps()
  - Returns an Enumeration of Map of request parameters
  - Usually reserved for debugging.

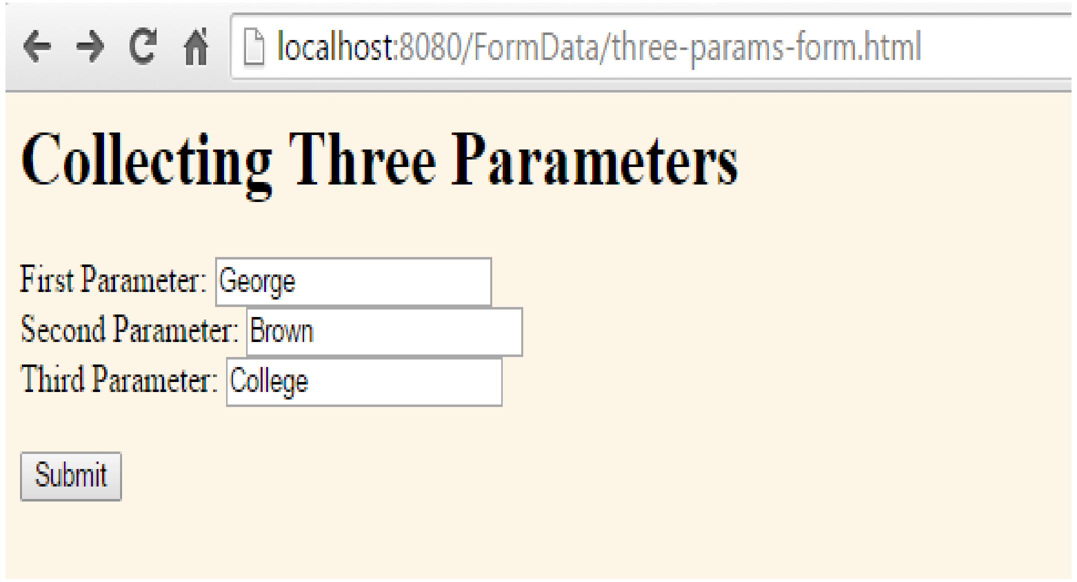
HTML Form with Three Parameters

Code Concepts


three-params-form.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Collecting Three Parameters</TITLE>
</HEAD>
<BODY BGCOLOR="#FDF5E6">
    <H1>Collecting Three Parameters</H1>

    <FORM ACTION="three-params">
        First Parameter: <INPUT TYPE="TEXT" NAME="param1"><BR>
        Second Parameter: <INPUT TYPE="TEXT" NAME="param2"><BR>
        Third Parameter: <INPUT TYPE="TEXT" NAME="param3"><BR><BR>
        <INPUT TYPE="SUBMIT">
    </FORM>

</BODY>
</HTML>
```

Forwards to servlet named *"three-params"*


localhost:8080/FormData/three-params-form.html

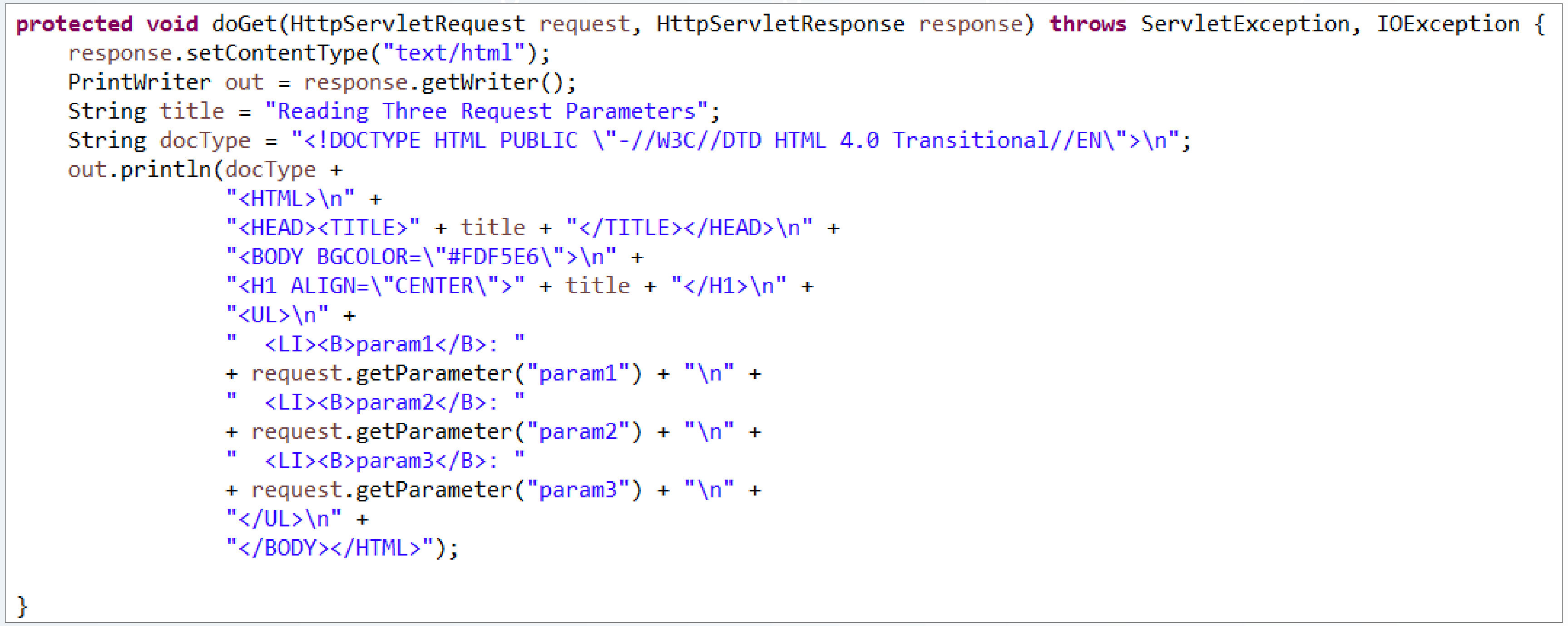
Collecting Three Parameters

First Parameter: George
Second Parameter: Brown
Third Parameter: College

File created in WebContent/three-params-form.html

Reading Three Parameters

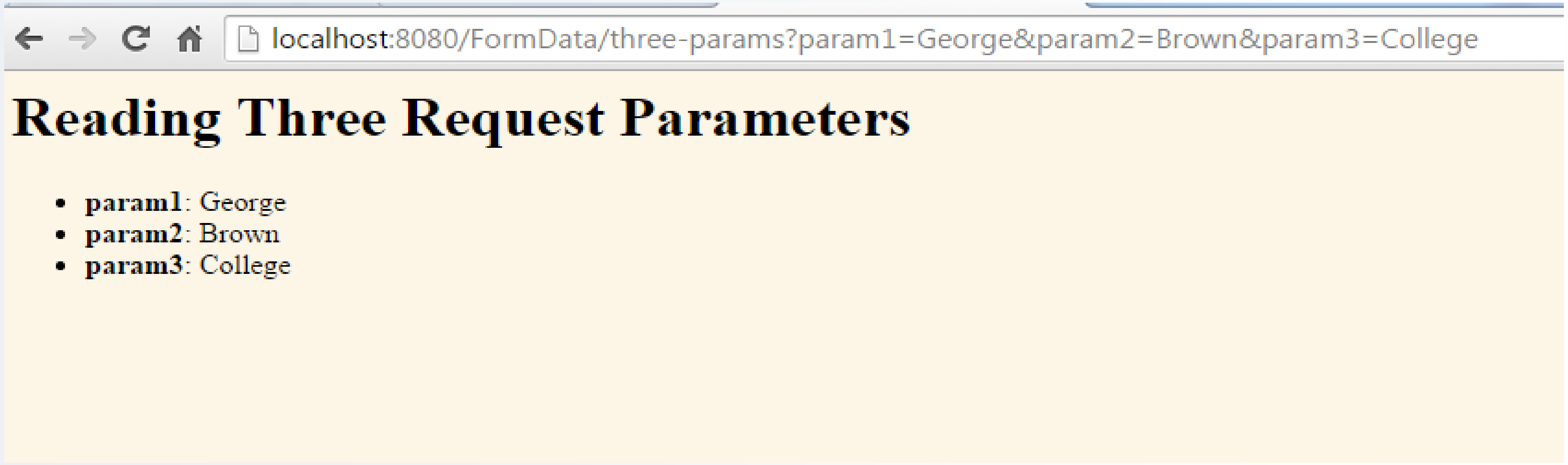
Sample code



```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String title = "Reading Three Request Parameters";
    String docType = "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 Transitional//EN\">\n";
    out.println(docType +
                "<HTML>\n" +
                "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
                "<BODY BGCOLOR=\"#FDF5E6\">\n" +
                "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n" +
                "<UL>\n" +
                "  <LI><B>param1</B>: "
                + request.getParameter("param1") + "\n" +
                "  <LI><B>param2</B>: "
                + request.getParameter("param2") + "\n" +
                "  <LI><B>param3</B>: "
                + request.getParameter("param3") + "\n" +
                "</UL>\n" +
                "</BODY></HTML>");

}
```

Reading Three Parameters Result

Result displayed on screen


localhost:8080/FormData/three-params?param1=George&param2=Brown&param3=College

```
Reading Three Request Parameters

  • param1: George
  • param2: Brown
  • param3: College
```

Reading All Parameters

Sample code

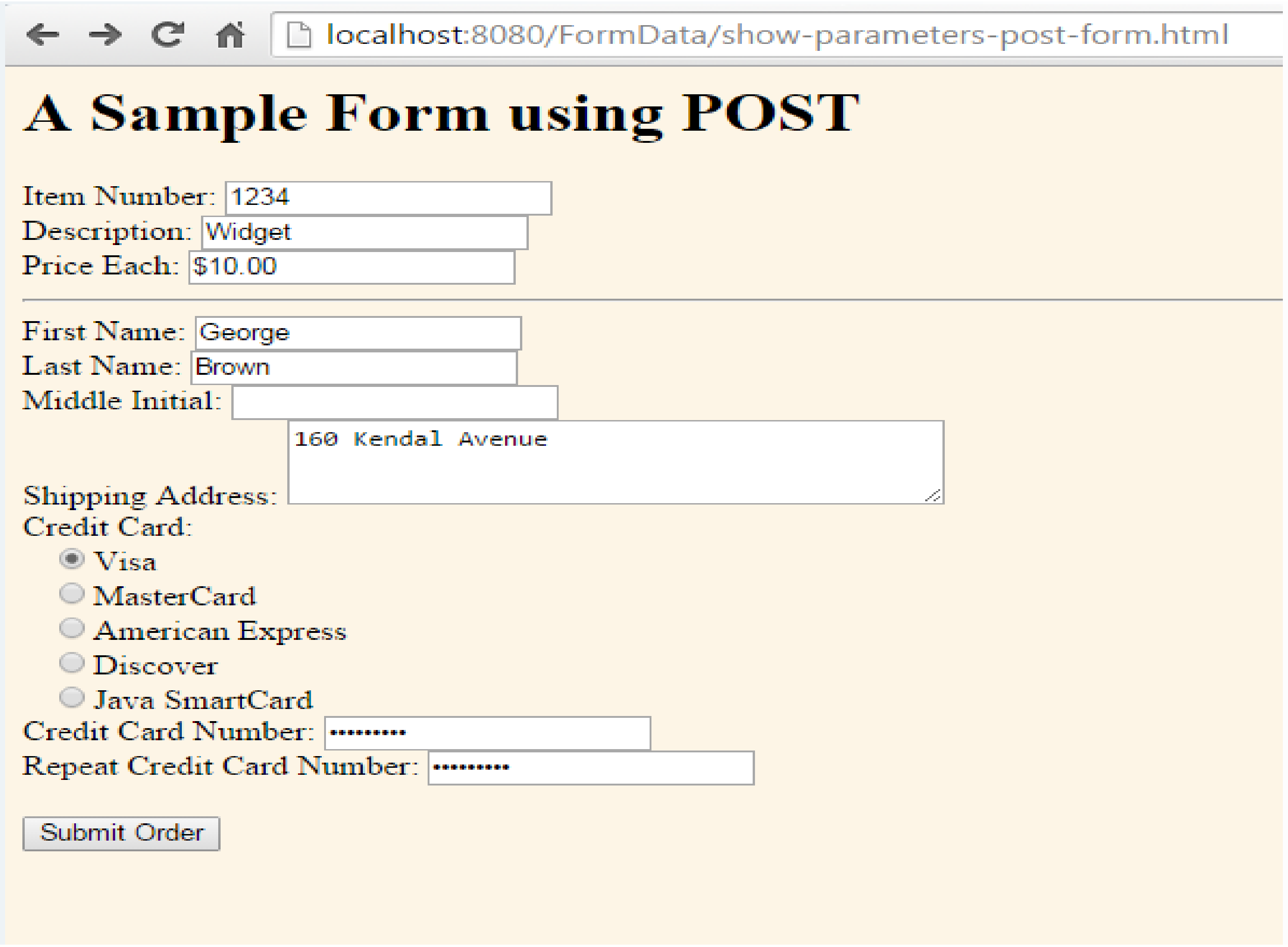
Special Java type used to define a collections of constants

"</BODY></HTML>"

```java
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String docType = "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 " + "Transitional//EN\">\n";
    String title = "Reading All Request Parameters";
    out.println(
            docType + "<HTML>\n" + "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" + "<BODY BGCOLOR=\"#FDF5E6\">\n"
                    + "<H1 ALIGN=CENTER>" + title + "</H1>\n" + "<TABLE BORDER=1 ALIGN=CENTER>\n"
                    + "<TR BGCOLOR=\"#FFAD00\">\n" + "<TH>Parameter Name<TH>Parameter Value(s)");

    Enumeration<String> paramNames = request.getParameterNames();
    while (paramNames.hasMoreElements()) {
        String paramName = paramNames.nextElement();
        out.print("<TR><TD>" + paramName + "\n<TD>");
        String[] paramValues = request.getParameterValues(paramName);
        if (paramValues.length == 1) {
            String paramValue = paramValues[0];
            if (paramValue.length() == 0)
                out.println("<I>No Value</I>");
            else
                out.println(paramValue);
        } else {
            out.println("<UL>");
            for (int i = 0; i < paramValues.length; i++) {
                out.println("<LI>" + paramValues[i]);
            }
            out.println("</UL>");
        }
    }
    out.println("</TABLE>\n</BODY></HTML>");
}
```

Reading All Parameters

Sample Form



localhost:8080/FormData/show-parameters-post-form.html

A Sample Form using POST

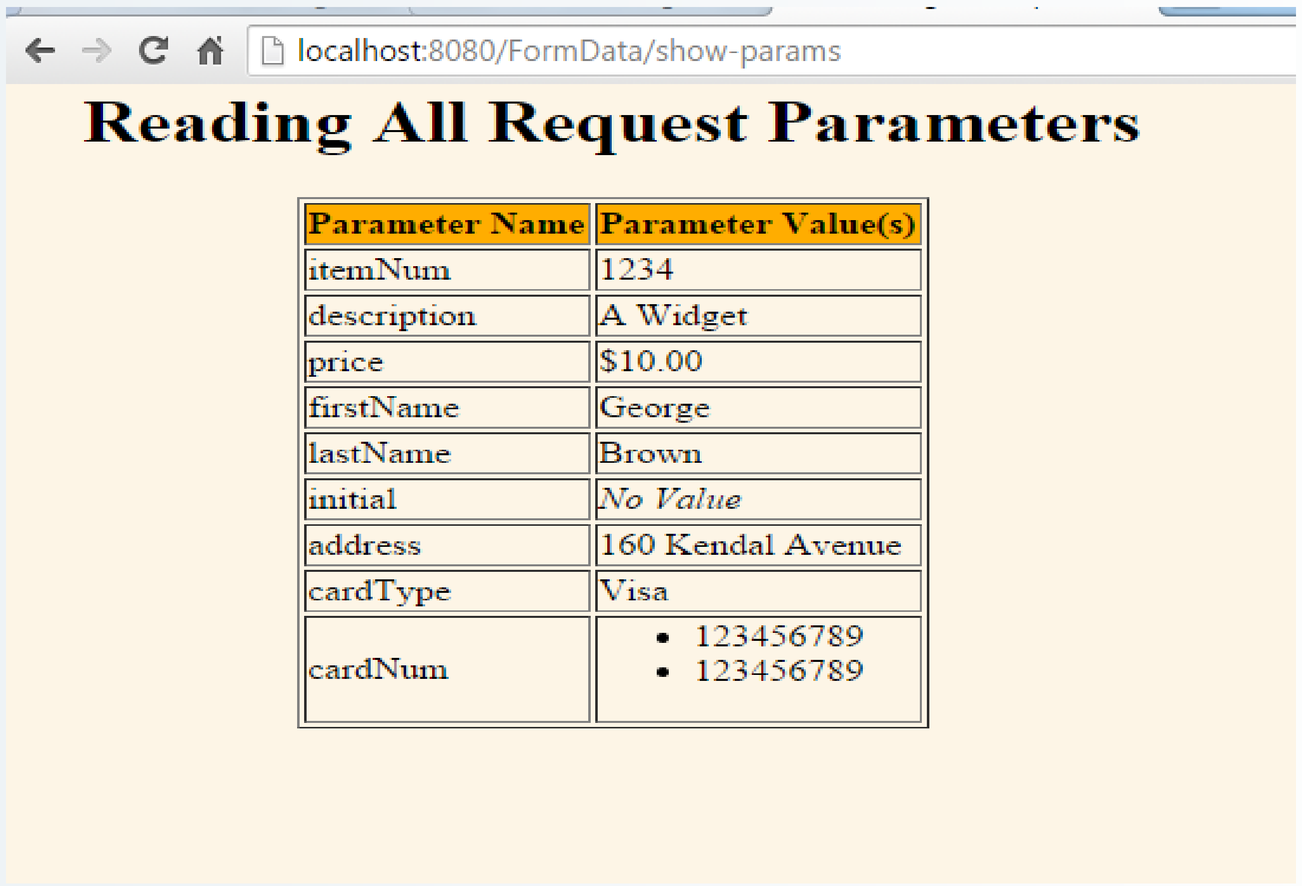
Item Number: 1234
Description: Widget
Price Each: $10.00

First Name: George
Last Name: Brown
Middle Initial:
Shipping Address: 160 Kendal Avenue
Credit Card:
- [✓ Visa]
- MasterCard
- American Express
- Discover
- Java SmartCard

Credit Card Number: ••••••••••
Repeat Credit Card Number: ••••••••••

Reading All Parameters

Result



localhost:8080/FormData/show-params

Reading All Request Parameters

Parameter Name	Parameter Value(s)
itemNum	1234
description	A Widget
price	$10.00
firstName	George
lastName	Brown
initial	*No Value*
address	160 Kendal Avenue
cardType	Visa
cardNum	• 123456789 • 123456789

Handling Missing and Malformed Data

Missing and Malformed Data Validation

Validating Input

- Missing
  - Field missing in form
    - getParameter() returns **null**
  - Field blank when form submitted
    - getParameter() returns an **empty string** (or possibly a **white-space**)
  - Must check for null before checking for empty string

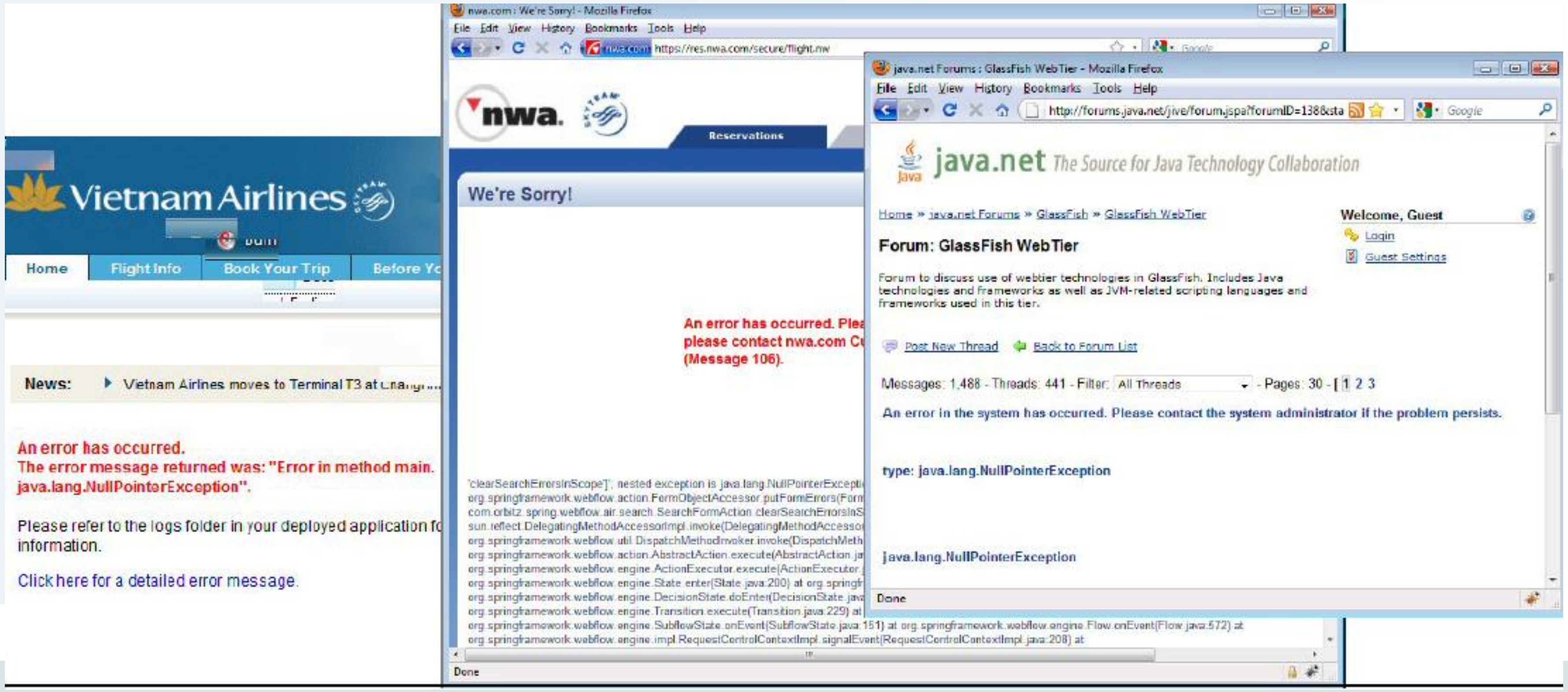
```
String param = request.getParameter("someName");
if( (param == null) || param.trim().equals("")) {
   doSomethingForMissingValues( … );
} else {
   doSomethingWithParameter(param);
}
```

- Malformed
  - Value is **non-empty** string in the **wrong format**.

Malformed Data Validation

Principle and General Rule

- Principle
  - Always assume data could be missing or in wrong format.
  - Users should never see Java error messages


Vietnam Airlines
Home
Flight Info
Book Your Trip
Before Yo
nwa.com : We're Sorry! - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://res.nwa.com/secure/flight.nw
nwa.
Reservations
java.net Forums : GlassFish WebTier - Mozilla Firefox
http://forums.java.net/jive/forum.jspa?forumID=138&sta
java.net The Source for Java Technology Collaboration
Home » java.net Forums » GlassFish » GlassFish WebTier
Welcome, Guest
Login
Guest Settings
Post New Thread
Back to Forum List
Done

News: Vietnam Airlines moves to Terminal T3 at Changi...

An error has occurred.
The error message returned was: "Error in method main. java.lang.NullPointerException".

Please refer to the logs folder in your deployed application fo information.

Click here for a detailed error message.

We're Sorry!

An error has occurred. Plea please contact nwa.com Cu (Message 106).

```
'clearSearchErrorsInScope]'; nested exception is java.lang.NullPointerExcepti
org.springframework.webflow.action.FormObjectAccessor.putFormErrors(Form
com.orbitz.spring.webflow.air.search.SearchFormAction.clearSearchErrorsInS
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccesso
org.springframework.webflow.util.DispatchMethodInvoker.invoke(DispatchMeth
org.springframework.webflow.action.AbstractAction.execute(AbstractAction.ja
org.springframework.webflow.engine.ActionExecutor.execute(ActionExecutor.
org.springframework.webflow.engine.State.enter(State.java:200) at org.springf
org.springframework.webflow.engine.DecisionState.doEnter(DecisionState.java
org.springframework.webflow.engine.Transition.execute(Transition.java:229) at
org.springframework.webflow.engine.SubflowState.onEvent(SubflowState.java:151) at org.springframework.webflow.engine.Flow.onEvent(Flow.java:572) at
org.springframework.webflow.engine.impl.RequestControlContextImpl.signalEvent(RequestControlContextImpl.java:208) at
```

Forum: GlassFish WebTier

Forum to discuss use of webtier technologies in GlassFish. Includes Java technologies and frameworks as well as JVM-related scripting languages and frameworks used in this tier.

Messages: 1,488 - Threads: 441 - Filter: All Threads - Pages: 30 - [1 2 3

An error in the system has occurred. Please contact the system administrator if the problem persists.

type: java.lang.NullPointerException

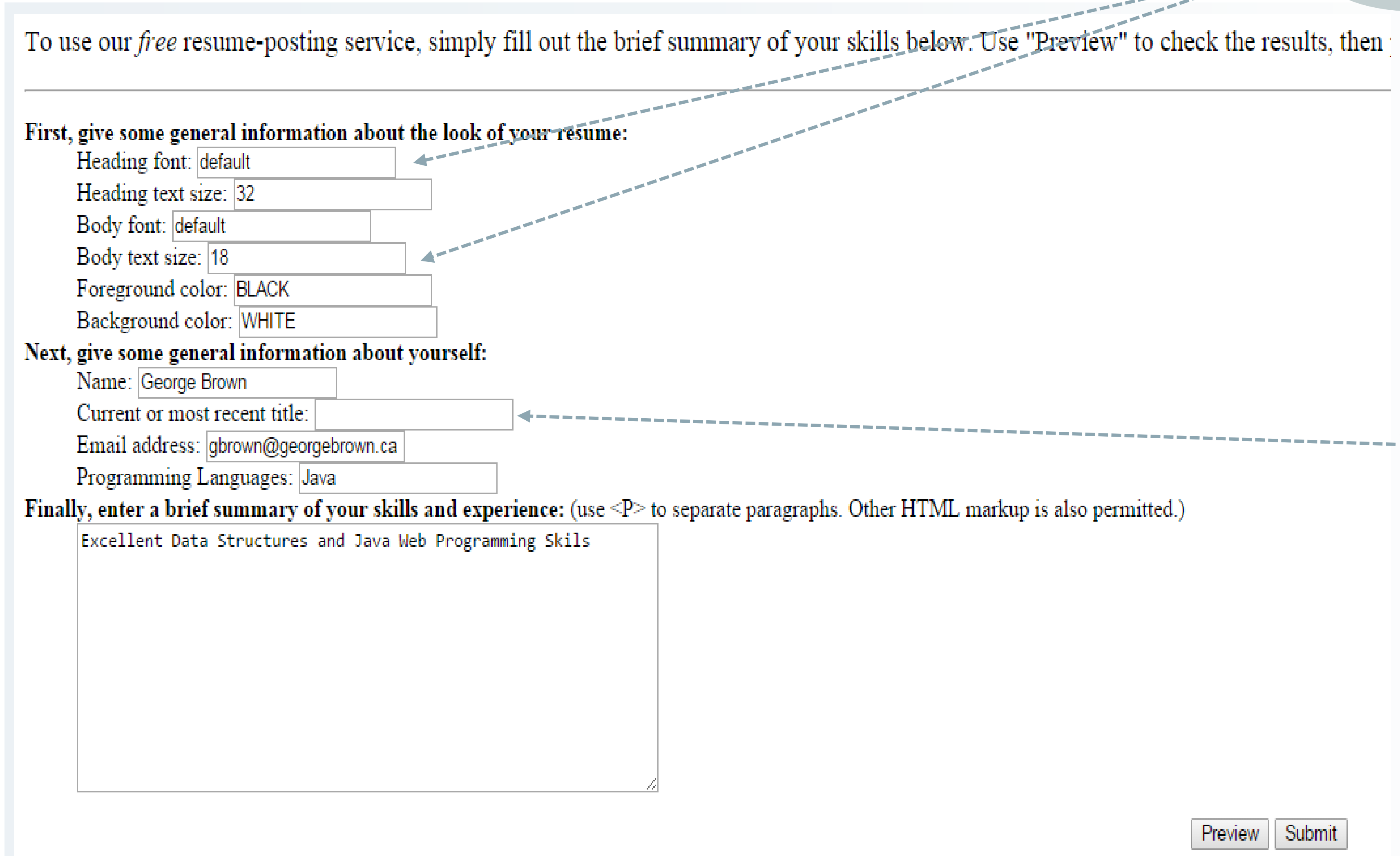
java.lang.NullPointerException

Handling Missing and Malformed Data
Guidelines

- Use default values
  - Replace missing values with application-specific standard values
- Redisplay the Form
  - Show the form again, missing values flagged.
  - Previously-entered values should be preserved.
  - Four options to implement this directly:
    1. Have the same servlet present the form, process the data, and present the results.
    2. Have one servlet present the form; have a second servlet process the data and present the result.
    3. Have one jsp page “manually” present the form; have a servlet or jsp page process the data and present the results
    4. Have a jsp page present the form, automatically filling in the fields with values obtained from a data object. Have a servlet or jsp page process the data and present the results.

Resume-Posting Site

Practical Example



To use our *free* resume-posting service, simply fill out the brief summary of your skills below. Use "Preview" to check the results, then

First, give some general information about the look of your resume:

Heading font:	default
Heading text size:	32
Body font:	default
Body text size:	18
Foreground color:	BLACK
Background color:	WHITE

Next, give some general information about yourself:

Name:	George Brown
Current or most recent title:	
Email address:	gbrown@georgebrown.ca
Programming Languages:	Java

Finally, enter a brief summary of your skills and experience: (use <P> to separate paragraphs. Other HTML markup is also permitted.)

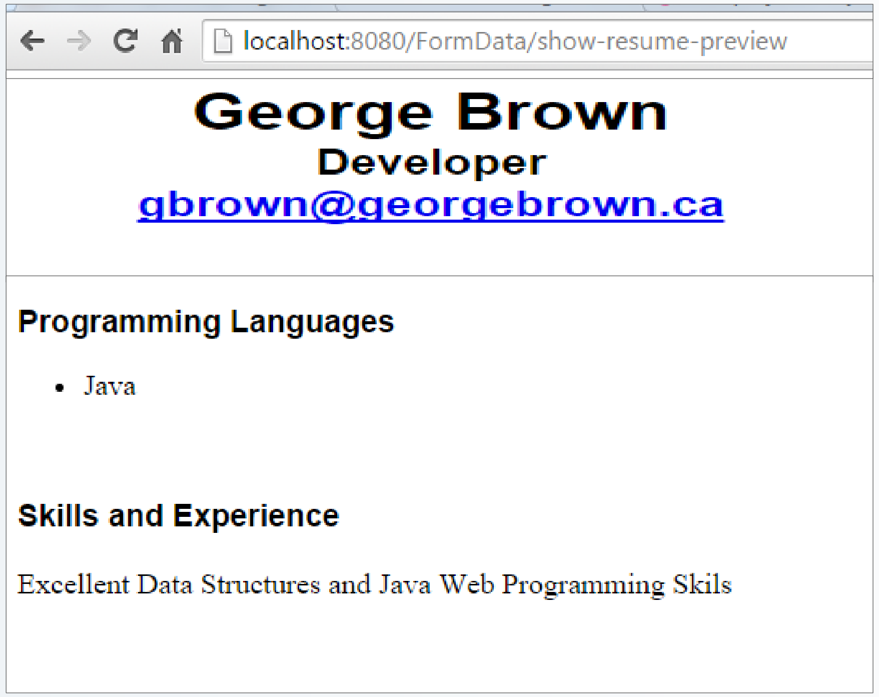
```
Excellent Data Structures and Java Web Programming Skils
```

Some fields are built with default values

Other fields could be left blank

Resume-Posting Site

Result Returned



localhost:8080/FormData/show-resume-preview

George Brown

Developer

gbrown@georgebrown.ca

Programming Languages

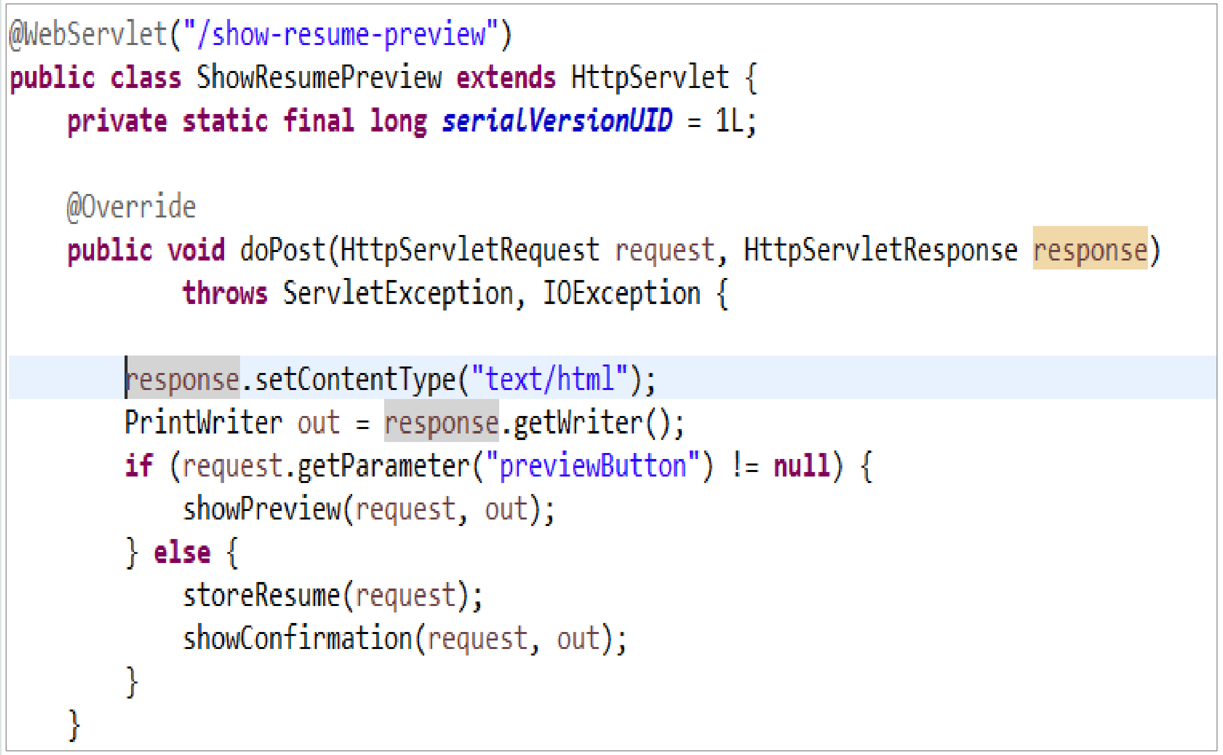
- Java

Skills and Experience

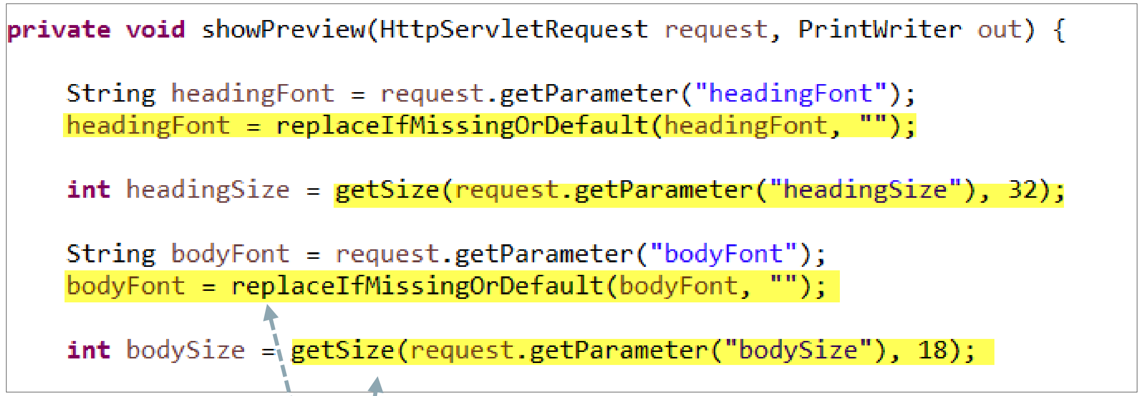
Excellent Data Structures and Java Web Programming Skils

Resume Posting Site

Servlet Code



```
@WebServlet("/show-resume-preview")
public class ShowResumePreview extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Override
    public void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        if (request.getParameter("previewButton") != null) {
            showPreview(request, out);
        } else {
            storeResume(request);
            showConfirmation(request, out);
        }
    }
```

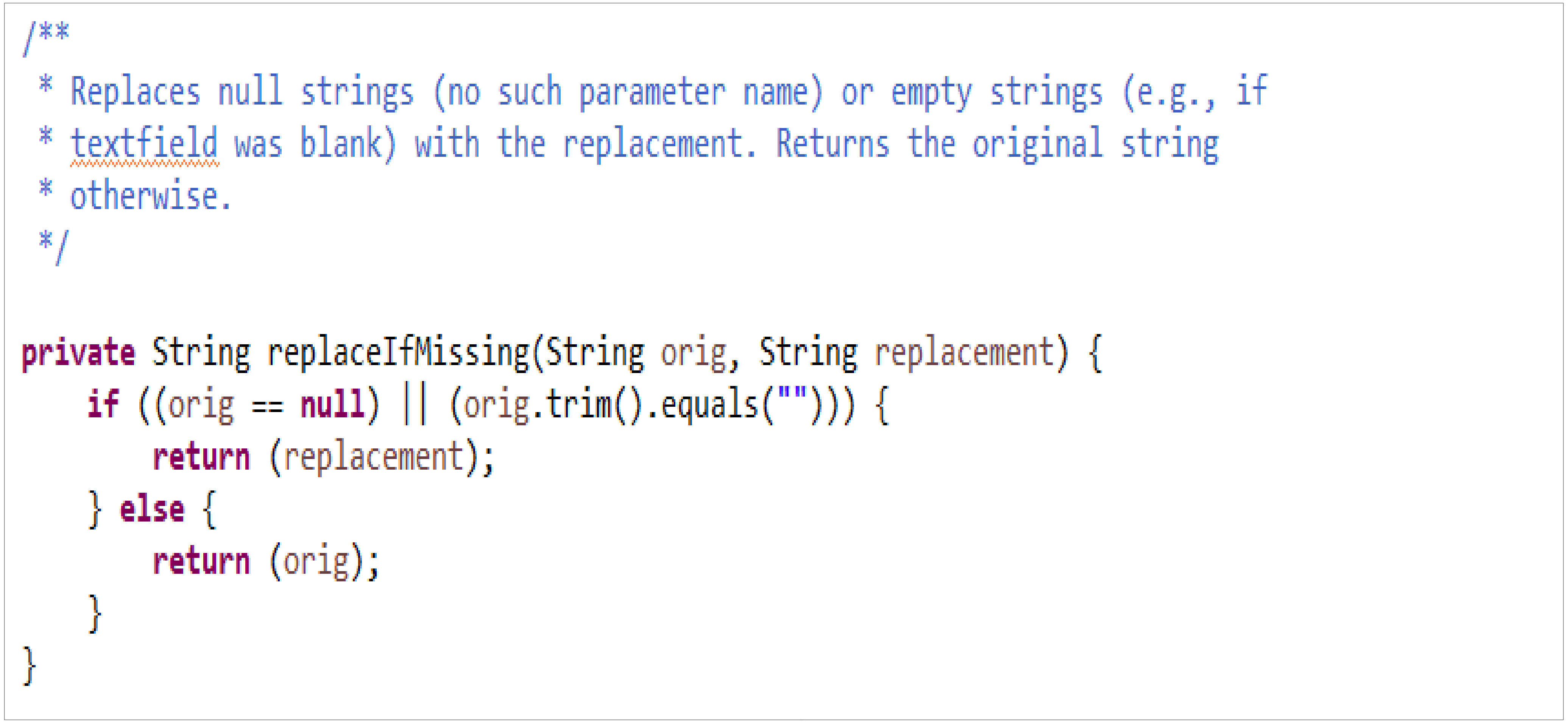


```
private void showPreview(HttpServletRequest request, PrintWriter out) {

    String headingFont = request.getParameter("headingFont");
    headingFont = replaceIfMissingOrDefault(headingFont, "");

    int headingSize = getSize(request.getParameter("headingSize"), 32);

    String bodyFont = request.getParameter("bodyFont");
    bodyFont = replaceIfMissingOrDefault(bodyFont, "");

    int bodySize = getSize(request.getParameter("bodySize"), 18);
```

Method was created
to validate form input.

Resume Posting Site

Servlet Code Continued …

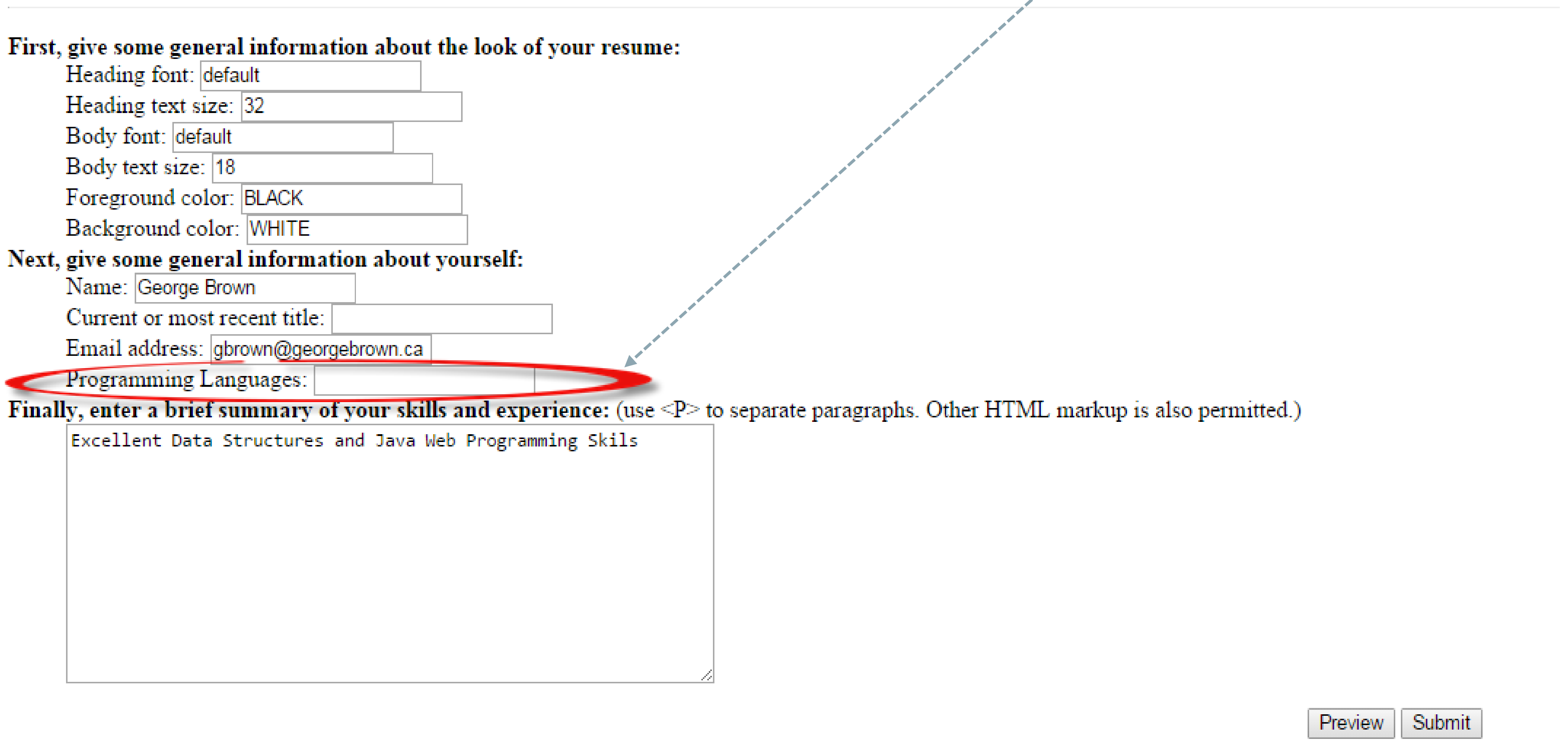


```
/**
 * Replaces null strings (no such parameter name) or empty strings (e.g., if
 * textfield was blank) with the replacement. Returns the original string
 * otherwise.
 */

private String replaceIfMissing(String orig, String replacement) {
    if ((orig == null) || (orig.trim().equals(""))) {
        return (replacement);
    } else {
        return (orig);
    }
}
```

Resume Posting Site

Submitting with missing data

Submit form with missing "Programming Languages".



First, give some general information about the look of your resume:

Heading font:	default
Heading text size:	32
Body font:	default
Body text size:	18
Foreground color:	BLACK
Background color:	WHITE

Next, give some general information about yourself:

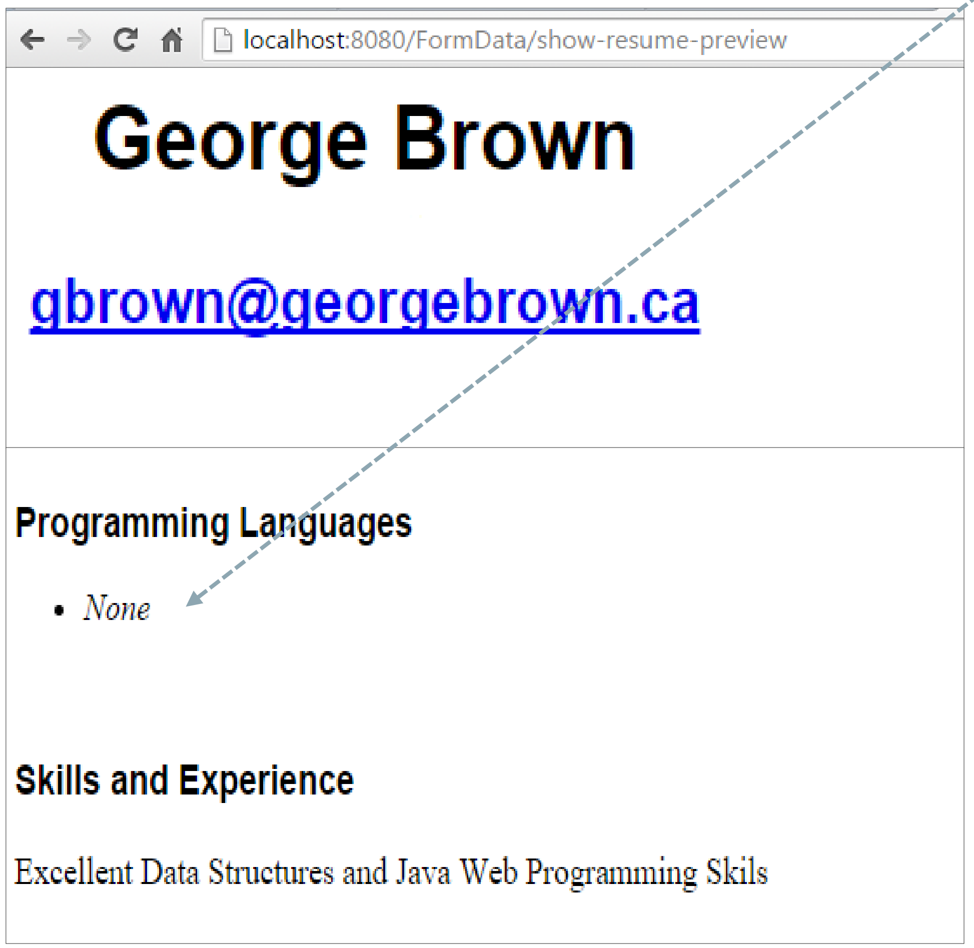
Name:	George Brown
Current or most recent title:	
Email address:	gbrown@georgebrown.ca
Programming Languages:	

Finally, enter a brief summary of your skills and experience: (use <P> to separate paragraphs. Other HTML markup is also permitted.)

```
Excellent Data Structures and Java Web Programming Skils
```

Resume Posting Site

Result after handling missing data.



localhost:8080/FormData/show-resume-preview

George Brown

gbrown@georgebrown.ca

Programming Languages

- *None*

Skills and Experience

Excellent Data Structures and Java Web Programming Skils

Missing "Programming Language" result in "None" parameter default

Filtering Strings for HTML Specific Characters

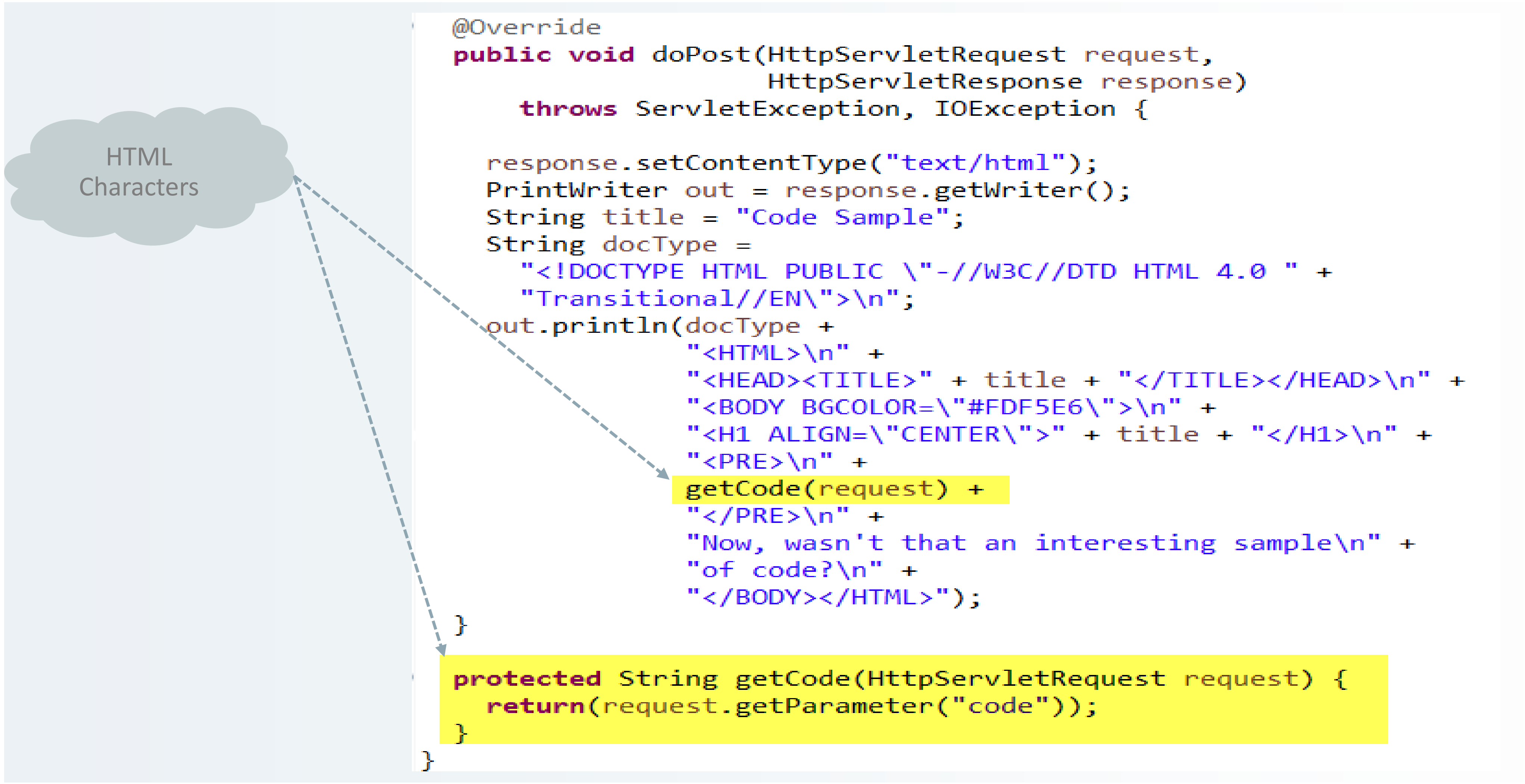
HTML and Special Characters

```java
public static String filter(String input) {
  if (!hasSpecialChars(input)) {
    return(input);
  }
  StringBuilder filtered = new StringBuilder(input.length());
  char c;
  for(int i=0; i<input.length(); i++) {
    c = input.charAt(i);
    switch(c) {
      case '<': filtered.append("&lt;"); break;
      case '>': filtered.append("&gt;"); break;
      case '"': filtered.append("&quot;"); break;
      case '&': filtered.append("&amp;"); break;
      default: filtered.append(c);
    }
  }
  return(filtered.toString());
}
```

HTML Characters

Servlet that Displays Code to Screen

Samples: No Filtering

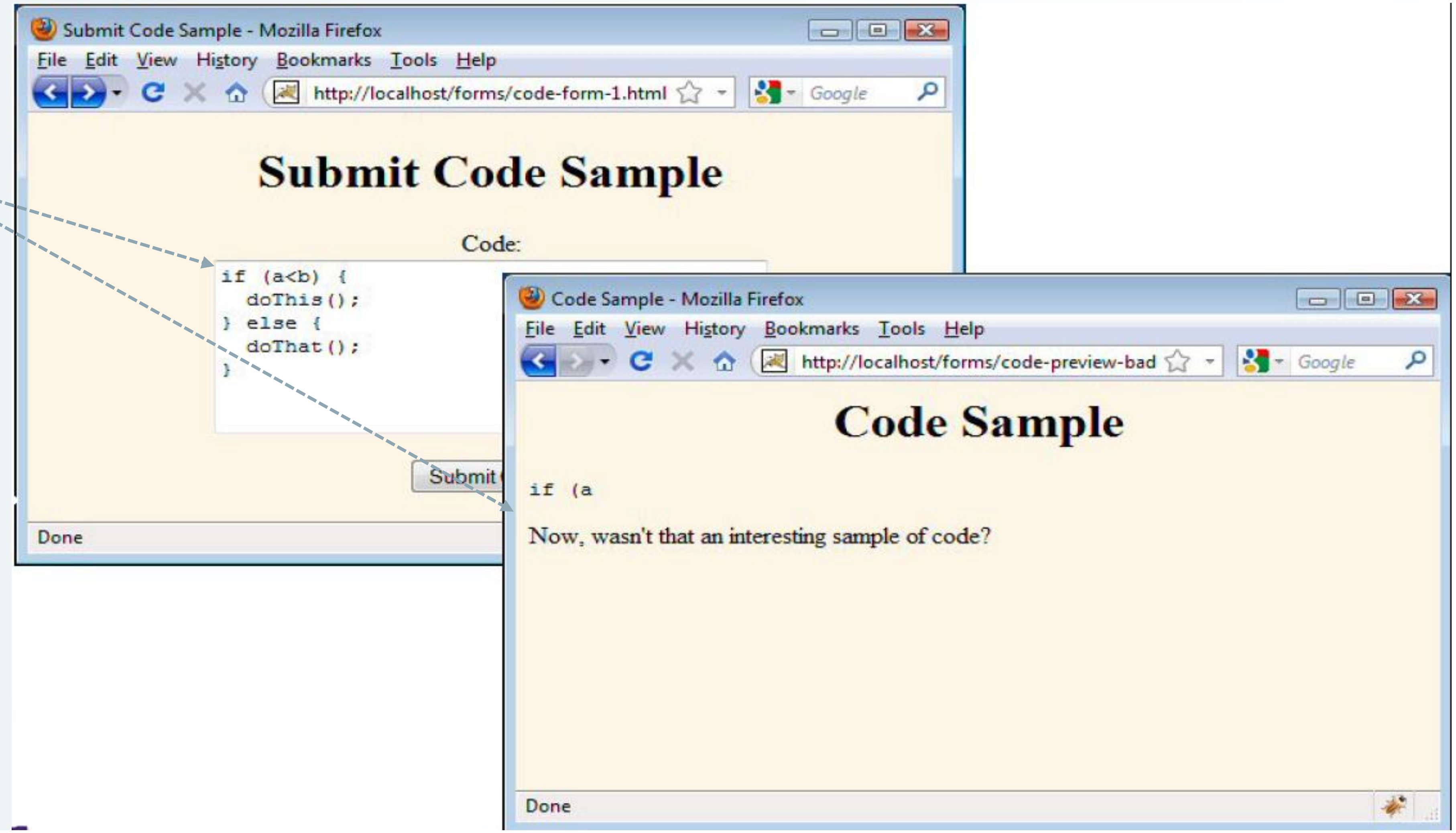


```java
@Override
public void doPost(HttpServletRequest request,
                   HttpServletResponse response)
    throws ServletException, IOException {

  response.setContentType("text/html");
  PrintWriter out = response.getWriter();
  String title = "Code Sample";
  String docType =
    "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 " +
    "Transitional//EN\">\n";
  out.println(docType +
              "<HTML>\n" +
              "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
              "<BODY BGCOLOR=\"#FDF5E6\">\n" +
              "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n" +
              "<PRE>\n" +
              getCode(request) +
              "</PRE>\n" +
              "Now, wasn't that an interesting sample\n" +
              "of code?\n" +
              "</BODY></HTML>");
}

protected String getCode(HttpServletRequest request) {
  return(request.getParameter("code"));
}
}
```

Display code to Screen

A Servlet that does not display code to Screen Continued

Fails to display code to screen. Characters are not translated


Submit Code Sample - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/forms/code-form-1.html
Google

Submit Code Sample

Code:

```
if (a<b) {
  doThis();
} else {
  doThat();
}
```

Submit
Done
Code Sample - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/forms/code-preview-bad
Google

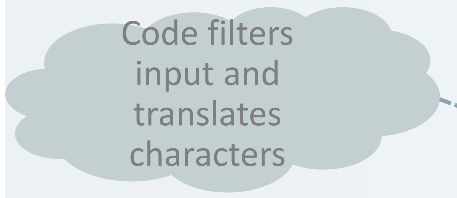
Code Sample

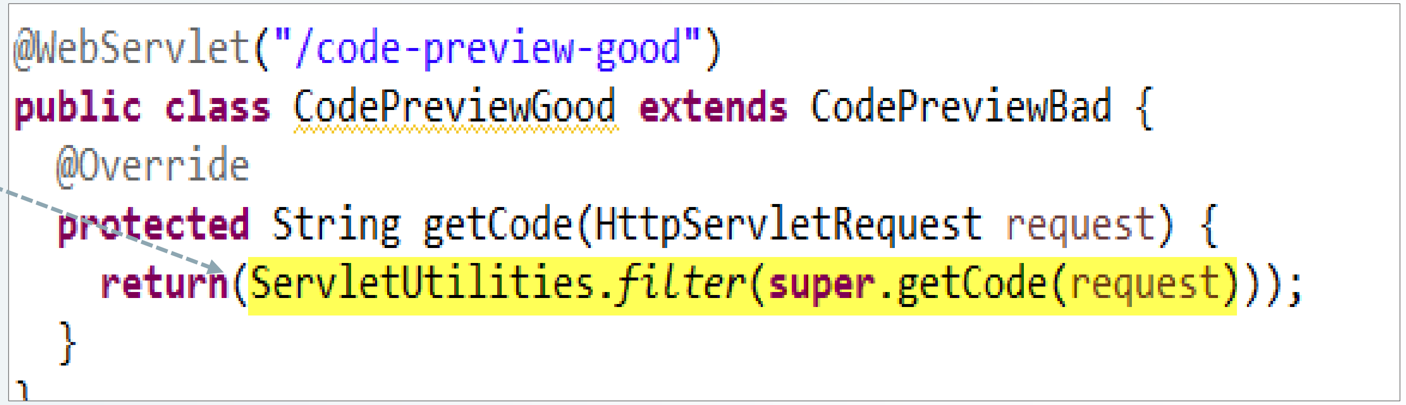
```
if (a
```

Now, wasn't that an interesting sample of code?

Servlet that Displays Code to Screen

Samples: Filtering



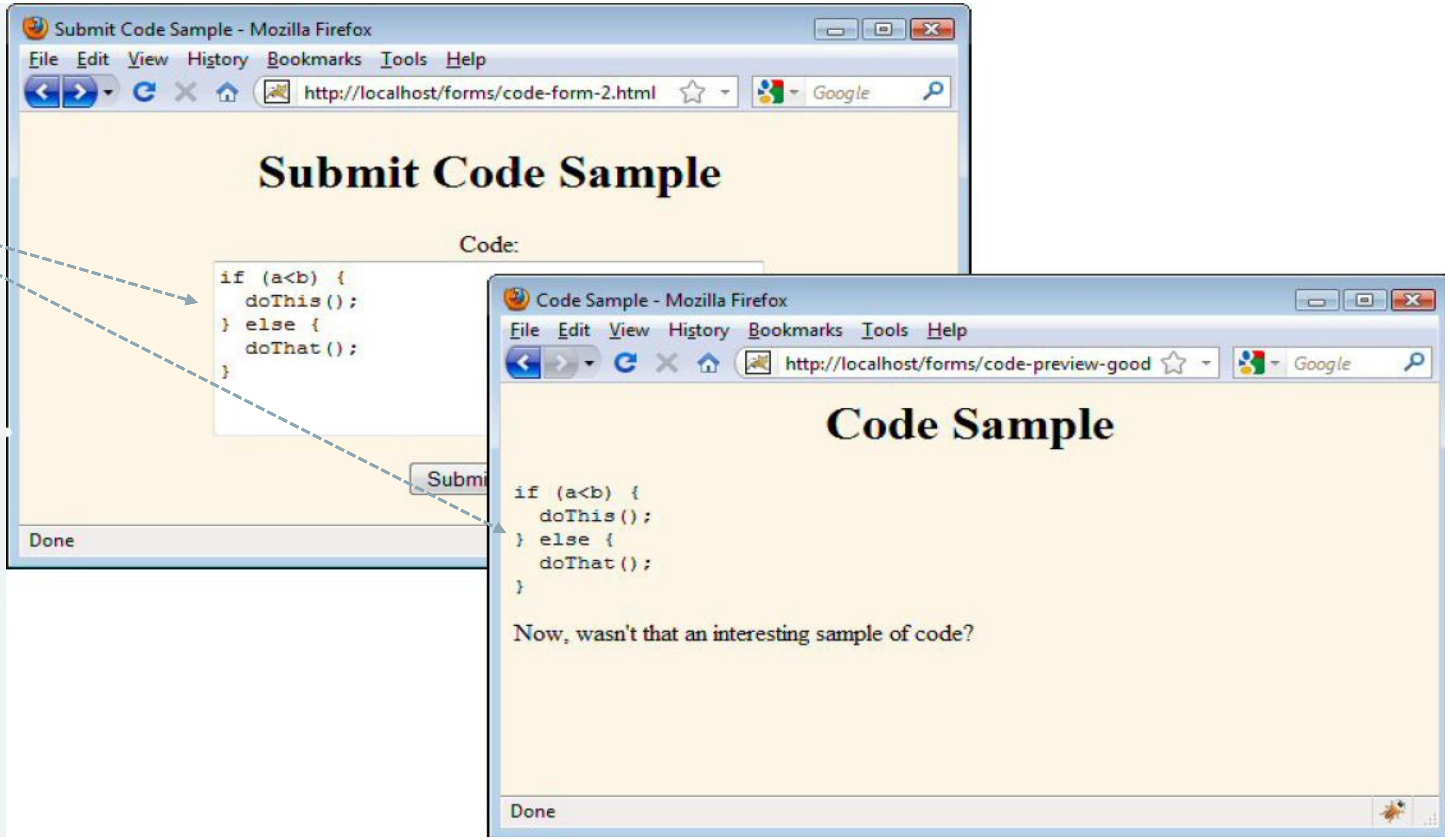


```
@WebServlet("/code-preview-good")
public class CodePreviewGood extends CodePreviewBad {
  @Override
  protected String getCode(HttpServletRequest request) {
    return(ServletUtilities.filter(super.getCode(request)));
  }
```

Display code to Screen

A Servlet that displays code to Screen Continued

submit code and display to screen. Characters are translated


Submit Code Sample - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/forms/code-form-2.html
Google
Submit Code Sample
Code:
Submi
Done
Code Sample - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/forms/code-preview-good
Google
Code Sample
Done

```
if (a<b) {
  doThis();
} else {
  doThat();
}
```

```
if (a<b) {
  doThis();
} else {
  doThat();
}
```

Now, wasn't that an interesting sample of code?