Essentials of Information Technology

PC-CS-305

Introduction to Web Application Technologies

Topic & Structure of the lesson

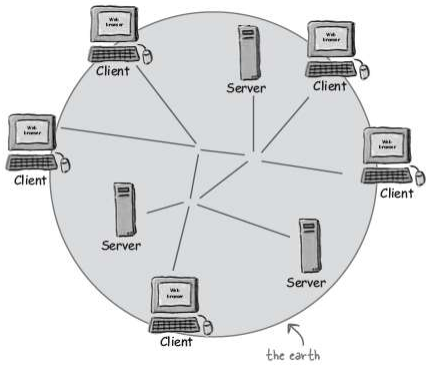
- Describe Internet Services.
- Describe the World Wide Web.
- Distinguish between Web Applications and Web Sites.
- Describe Java servlet technology and list three benefits of this technology compared to traditional common Gateway Interface (CGI) scripting.

Topic & Structure of the lesson

- Describe Java Server Pages technology and list three benefits of JSP technology over rival template page technologies.
- Describe the Java 2 Platform Enterprise Edition.

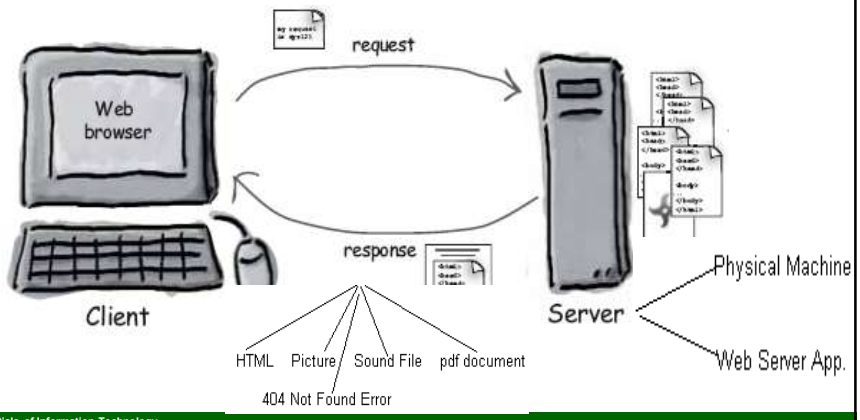
The Internet Is Network of Networks

- The web consist of thousands of clients and servers connected through wires and wireless networks.



Web Server

A webserver takes a client request finds the resource and gives something back to client.



Web Client

Web client let the user request something on server, and show the user the result of request.Client is either the human or browser application.

Browser

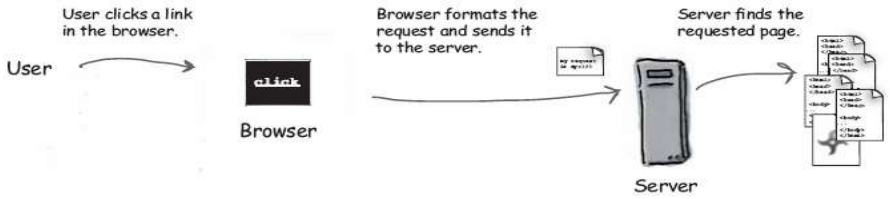
Browser is a piece of software that knows how to communicate with server.

Steps Involved In Client Server Communication

1.User clicks a link in the browser.

2.Browser formats the request and send it to server.

3.Server finds the requested page.

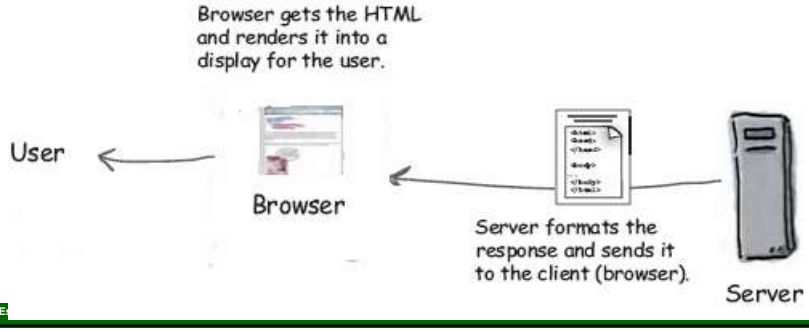


PC-CS-305: Essentials of Information Technology

Steps Involved In Client Server Communication

4.Server format the response and send it to the client.

5.Browser get the html and render it into display for the user.



PC-CS-305: E

Web Application & Web Sites

- A web site is a collection of static HTML pages
- A web application is a web site with dynamic functionality on the server (or sometimes on the client using applets or other interactive elements.)

Web Application & Web Sites

- Web application use HTML forms as the user interface to code that is running on the server.
  - Data is passed from HTML form to the server using CGI (Common Gateway Interface).
  - The CGI data is sent in the HTTP request stream

Jobs That Web Server Can't Do

1.Serving Dynamic Web pages

The webserver serve only static web pages, to serve dynamic web pages, web server have to communicate with separate helper application.

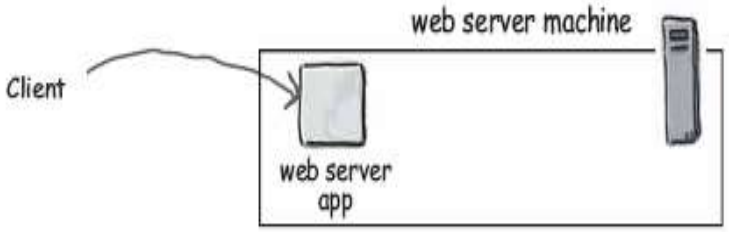
Jobs That Web Server Can't Do

2. Saving data on the server

When user submit data in the form, the webserver hands over the parameter to the helper application and gives the application a way to generate a response to client. The non Java term for another application on server is ***CGI program.***

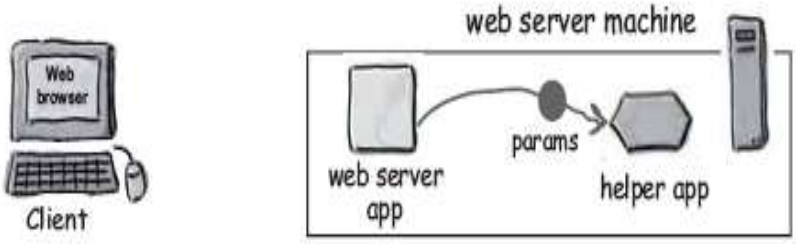
CGI Program on Web Server

- User click a link that has URL to CGI program instead of static page



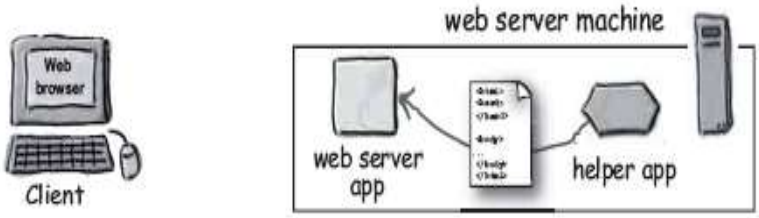
CGI Program on Web Server

2. Web server application sees that the request is for helper program so the webserver launches and run the program. The web server app sends along any parameters from a GET or POST



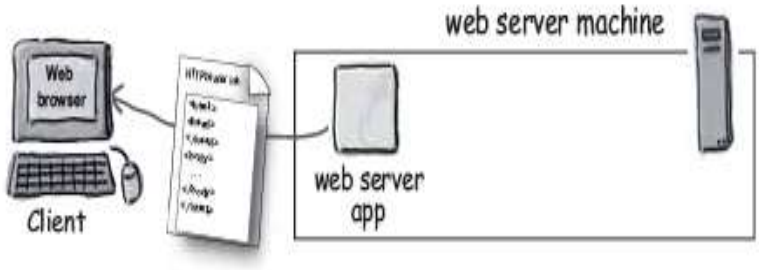
CGI Program on Web Server

3. The helper app constructs the brand new page and send the HTML back to the server. As far as the webserver is concerned, the HTML from the helper app is a static page.



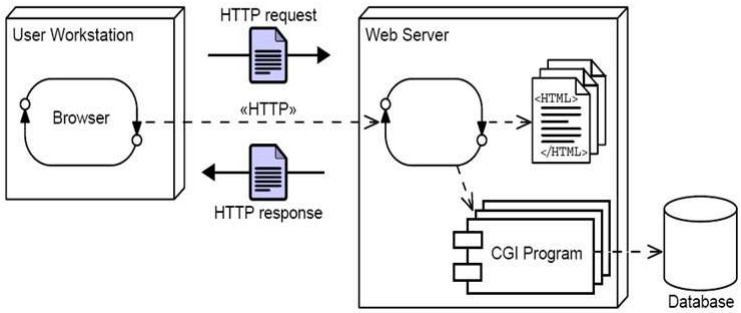
CGI Program on Web Server

4. The helper application is shut down and the client gets back an HTML page that has the current date as part of its now static content.



CGI Programs on the Web Server

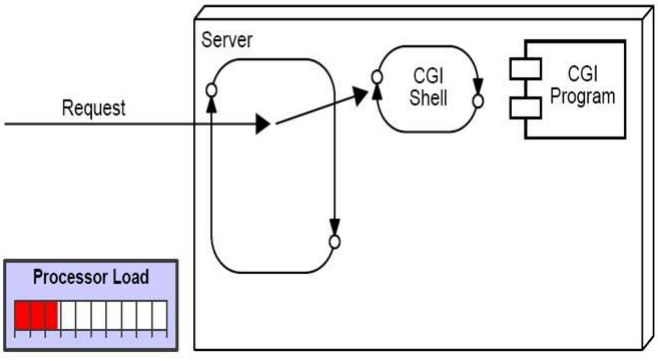
Deployment diagram of a web server with CGI programs:



PC-CS-305: Essentials of Information Technology

Execution of CGI Programs

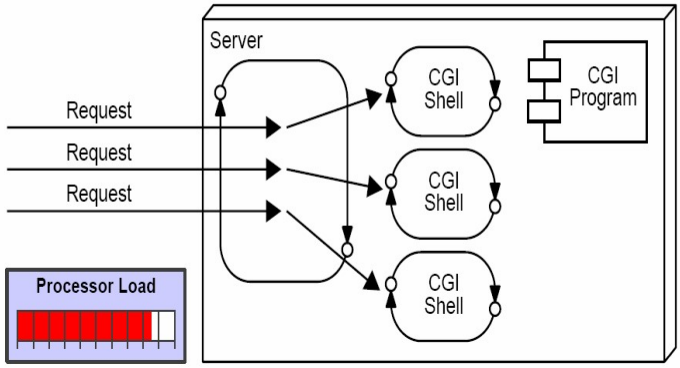
How CGI works with one request:



PC-CS-305: Essentials of Information Technology

Execution of CGI Programs

How CGI works with many requests:



Advantages & Disadvantages of CGI Programs

CGI program advantages

- written in a variety of languages
- relatively easy for a web designer to reference

CGI program disadvantages

- Each shell is heavy weight
- Not scalable.
- CGI processing code is mingled with HTML.
- language is not always secure or object oriented.
- It use platform dependent language.

Java Servlets

A servlet is a Java technology component that executes on the server.

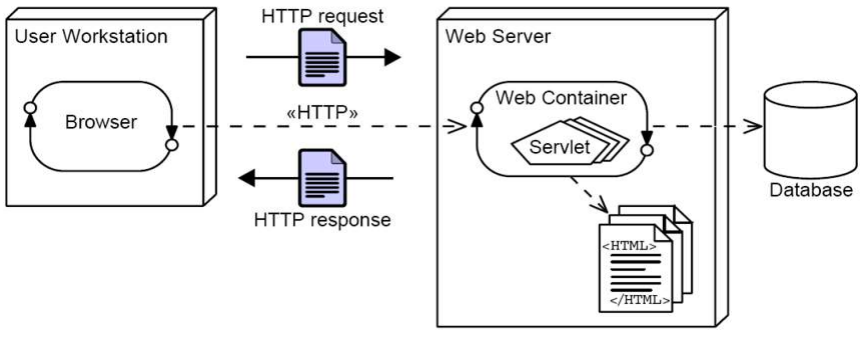
Servlet perform tasks similar to those performed by CGI programs, but servlets execute in a different environment.

Servlets perform the following
- Process the HTTP request
- Generate the HTTP response dynamically.

A web container is a special Java Virtual machine that is responsible for maintaining the life cycle of the servlet as well as issuing threads for each request.

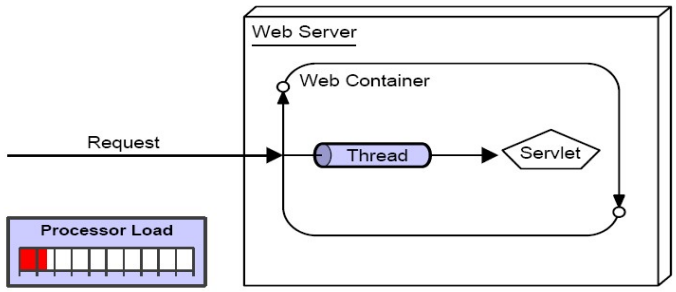
Servlets on WebServer

Deployment diagram of a web server with a web container:



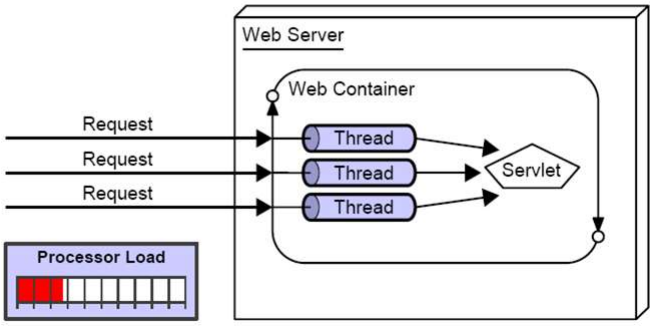
Execution of Java Servlets

How servlets work with one request:



Execution of Java Servlets

How servlets work with many requests:



Advantages & Disadvantages of Java Servlets

Java servlet advantages

- Performances (threads are faster than processes)
- Scalable
- The Java programming language is robust and object oriented
- The Java programming language is platform independent

Java servlet disadvantages

- Separation of concerns : business and presentation logic.
- Concurrency issue.

Servlet

- A servlet is a server-side entity.
- Servlet from several different perspectives.

- Conceptually, a servlet is a piece of code that can be:
  - Plugged into an existing server to extend the server functionality
  - Used to generate the desired output dynamically

Servlet

- *For a servlet container*, a servlet is:
  - A Java class like any other normal Java class
  - A class that implements the javax.servlet.Servlet interface

Servlet

- For a web component developer, a servlet, or specifically an HTTP servlet, is a class that:
  - Extends javax.servlet.http.HttpServlet
  - Resides in a servlet container (such as Tomcat)
  - Serves HTTP requests

Types of Servlet

Two types of servlets

- Generic Servlet (javax.servlet.GenericServlet)
- HttpServlet (javax.servlet.http.HttpServlet)

Generic Servlet

- This class implements the Servlet interface .
- It is an abstract class that provides implementation for all the methods except the service() method of Servlet interface.
- We can extend this class and implement the service() method to write any kind of servlet.

Example http://localhost:9090/genericexample/

Http Servlet

- It is an abstract class that extends GenericServlet.
- It add a new Service() method with signature
  protected void service (HttpServletRequest, HttpServletResponse) throws ServletException,IOException
- This method automatically call the other methods to aid in processing the HTTP based request.

Http Servlet

- Like for handling HTTP GET request it will automatically call
- protected void doGet(HttpServletRequest req, HttpServletResponse res)
  throws ServletException,IOException

Java Server Pages Technology

- Template pages look like static HTML pages, but with embedded code to perform dynamic generation of data and HTML.
- Example:

```
<table border='1' cellspacing='0' cellpadding='5'>
<tr><th>number</th><th>squared</th></tr>
<% for ( int i=0; i<10; i++ ) { %>
<tr><td><%= i %></td><td><%= (i * i) %></td></tr>
<% } %>
</table>
```

Table of numbers squared:

number	squared
0	0
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81

Java Server Pages Technology

- JSP pages are translated into Java servlet classes, are compiled, and are treated just like servlets in the web container.
- If designed well, JSP pages focus on the presentation logic, not on the business logic.
- In JSP pages, custom tags and JSP Expression Language provide for reusable code and separation of concerns.
- In a Java technology web application, JSP pages are often used in conjunction with servlets and business objects in a Model-View-Controller pattern.

Advantages & Disadvantages of Java Server Pages

Advantages of JSP Technology

JSP technology has all of the advantages of servlet technology: high performance, high scalability, platform independent and can use the Java language as its scripting language.

Disadvantages of JSP technology

If JSP pages are used in isolation, then the scripting code which perform business and control logic can become cumbersome in the JSP pages. JSP pages are also difficult to debug.

URI,URL,URN

Uniform Resource Identifier.

- A URI is a string that identifies any resource.
- Identifying the resource may not necessarily mean that we can retrieve it.

URI,URL,URN

Uniform Resource Identifier

Example

files/sales/report.html

- Is a URI, because it identifies some resource.
- However, it is not a URL because it does not specify how to retrieve the resource.
- It is not a URN either, because it does not identify the resource uniquely.

URI,URL,URN

Uniform Resource Locator.

- URIs that specifies common Internet protocols such as HTTP, FTP and mailto are also called URLs.
- URL is an informal term and is not used in technical specifications.

URI,URL,URN

Uniform Resource Locator

Example

http://www.manning.com/files/sales/report.html

Is a URL because it also specifies how to retrieve the resource.

URI,URL,URN

Uniform Resource Name.

- A URN is an identifier that uniquely identifies a resource but does not specify how to access the resource.
- URNs are standardized by official institutions to maintain the uniqueness of a resource.

URI,URL,URN

Uniform Resource Name

Example

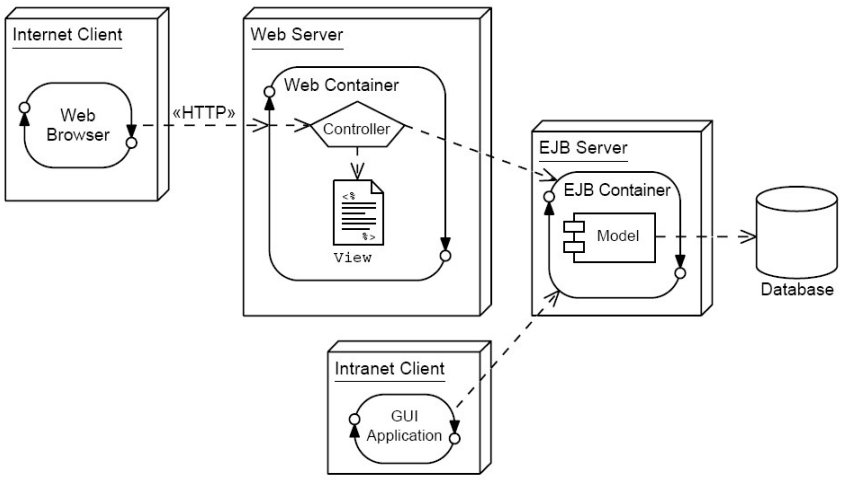
ISBN:1-930110-59-6

Is a URN because it uniquely identifies this book,but it is not a URL because it does not indicate how to retrieve the book.

Server Side Include

- The methodology of embedding programming language within HTML is called the ***server side include.***
- The programming language that is embedded within HTML is called the ***scripting language.***
- JSP is a technology that provide a standard specification for combining Java as scripting language with HTML.

J2EE Architecture



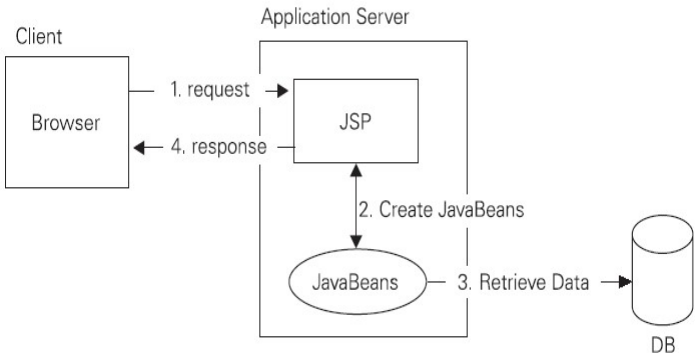
Job Roles

- The modularity of a J2ee architecture clearly distinguishes several job roles:
  - **Web Designer** - Creates view elements
  - **Web Component Developer** - Creates Controller elements
  - **Business Component Developer** - Creates model elements
  - **Data Access Developer** - Creates database access elements.

Approaches for building web applications

- Model 1 Architecture
- Model 2 Architecture

Model 1 Architecture



Model 1 Architecture

- Here the target of every request is JSP page.This page is completely responsible for doing all the task required for fulfilling the request.

Model 1 Architecture

Drawbacks

1. There is no central component that control the work flow of application.

2. It requires embedded business logic using big chunks of Java code into JSP page.

This create problem for web page designers who are not comfortable with server side programming.

Model 1 Architecture

Drawbacks

3. Does not promote reusability of application components.For example code written in JSP page for authenticating a user cannot be reused in other JSP page.

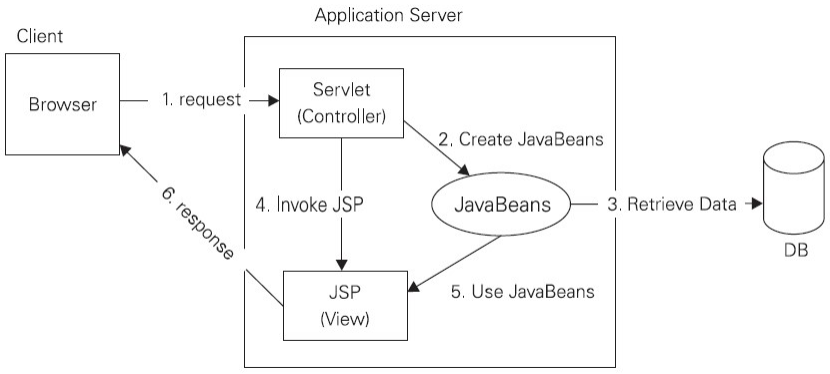
Model 2 Architecture

This architecture follows the MVC design pattern.

In this architecture the target of all the request are servlets that act as the controller for the application.

They analyze the request and collect the data required to generate a response into JavaBeans objects which act as a Model for application.

Model 2 Architecture



Model 2 Architecture

- Finally controller servlets dispatch the request to JSP pages.These use the data stored in Java beans to generate presentable response.

Summary of Main Teaching Points

- Described Internet Services.
- Described the World Wide Web.
- Distinguished between Web Applications and Web Sites.
- Described Java servlet technology and list three benefits of this technology compared to traditional common Gateway Interface (CGI) scripting.
- Described Java Server Pages technology and list three benefits of JSP technology over rival template page technologies.
- Described the Java 2 Platform Enterprise Edition.

References

- Basham, B., Sierra, K., & Bates, B. (2004). *Head First Servlets and JSP: Passing the Sun Certified Web Component Developer Exam*. " O'Reilly Media, Inc.".

Question and Answer Session

Q & A