J2EE Introduction

LEARNING OBJECTIVES

At the end of this lesson, you will be able to:

- Introduction to web applications
- J2EE Architecture

Web

Consists of clients and servers connected through networks

- **Web client**
  - lets the user request a resource on the server and show the result to the user
  - Resource could be html page, picture, PDF file etc.
  - Browser - Interprets the HTML code and renders the web page for the user
  - Ex: Google Chrome, safari, internet explorer

- Web server
  - Receives a client request
  - Finds the resource
  - Sends the response back to the client
  - Ex: Apache, IIS

Communication

Most of the Clients and Server communicate using HTTP

- HTTP
  - Stands for Hyper Text Transfer Protocol
  - HTTP conversation is a simple **Request/ Response** sequence - Browser *requests* and server *responds*.
  - Used by the server to send the response to the client
  - Stateless protocol
- HTML
  - Used by the server to send a set of instructions to the browser
  - Tells the browser how the content should be presented to the user
  - Can be part of the HTTP response

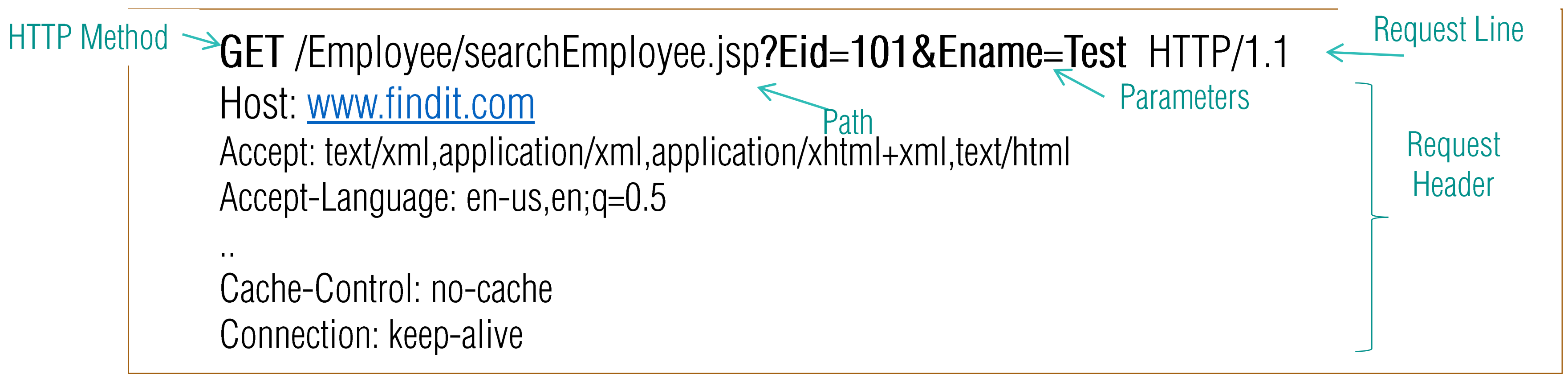
HTTP request and response

- Key elements of HTTP request
  - HTTP method(GET, POST, HEAD, PUT etc)
  - The page to access (request URL)
  - Request Headers
  - Form parameters (location depends on http method)

- Key elements of HTTP response
  - Response Header
    - A status code
    - Content-type (MIME type)
    - Last Modified
  - Content (HTML, image, etc)

HTTP request Methods

- Method tells the web server what kind of request is being performed on a URL
- Most used methods of HTTP protocol are GET and POST
- GET
  - Retrieves data/resource from the server by specifying parameters in the URL
  - Parameters specified by user is appended to the URL



```
GET /Employee/searchEmployee.jsp?Eid=101&Ename=Test  HTTP/1.1
Host: www.findit.com
Accept: text/xml,application/xml,application/xhtml+xml,text/html
Accept-Language: en-us,en;q=0.5
..
Cache-Control: no-cache
Connection: keep-alive
```

- Used with intent of viewing the data and allows url to be bookmarked
- Not used
  - when sensitive data like usernames, passwords are passed
  - Large amount of data needs to be sent as parameters
  - Submitting a form for updating data on server

HTTP request Methods

- POST
  - Utilizes a message body to send data to a web server
  - Used
    - Data to be sent is sensitive
    - When altering state of application

```
POST /Employee/UpdateEmployee.jsp HTTP/1.1
Host: www.findit.com
User-Agent: Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US;
Accept: text/xml,application/xml,application/xhtml+xml,text/
...
Keep-Alive: 300 Connection: keep-alive

Eid=101&Ename=Test
```

Request Header

Parameters in message body

HTTP request Methods

- HEAD Method
  - The HEAD method is like GET, except that the server replies only with a response line and headers, but no entity-body
- PUT, DELETE etc

HTTP Response

Status code

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONID=0AAB6C8DE415E2E5F307CF334BFCA0C1; Path=/testEL
Last-Modified: Tue, 10 Dec 2013 09:18:31 GMT
Date: Wed, 18 Dec 2013 16:02:31 GMT
Content-Type: text/html
Content-Length: 397
Server: Apache-Coyote/1.1
Connection: close

<html>
 ...
</html>
```

Response Header

Content : could be Html or other content

Dynamic Web Application

- Web Server application serves only static pages

- Web Server alone wont be enough
  - To create dynamic pages which don’t exist before request
    - To display Employees list from a database, Google Search
    - To process online order based on user input
  - To write/save data on the server
    - To save data entered in a form to database

- A helper application is needed to fulfill the above

- Web server can communicate with the helper application to achieve the dynamic requirements

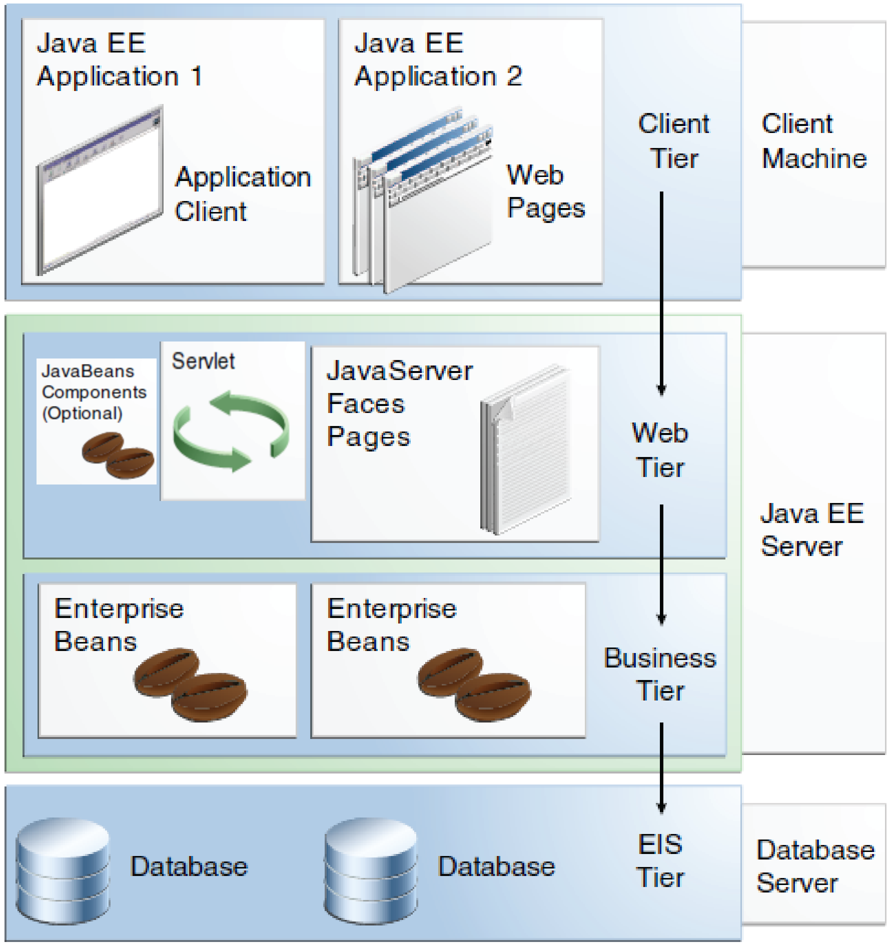
CGI

- Web pages can be made dynamic using CGI (Common Gateway Interface)
- **CGI** (Common Gateway Interface) is a standard that specifies how external programs may be used by web servers
- Programs that adhere to CGI standards are known as CGI programs
- Typically written in Perl
- Disadvantages
  - Each client request makes the server spawn a new process of CGI program, which is an expensive operation
  - Performance degrades with the increase in requests
  - Scripting languages used are platform dependent

JAVA EE Platform

- Java EE provides a powerful set of APIs which
  - reduces development time
  - reduces application complexity
  - improves application performance
- The Java EE platform uses a distributed multitier application model for enterprise applications
- Application logic is divided into components according to function
- Component is a self-contained functional software unit that is assembled into a Java EE application
- Components communicates with other components
- Application components are installed on various machines depending on the tier

Multitiered Applications



Java EE multitiered applications are generally considered to be three-tiered applications

Client tier Components like Web clients and Application clients are components that run on the client

Web Components like Servlets, JavaServer Faces, and JavaServer Pages (JSP) form the web tier and run on the server

Business components like Enterprise JavaBeans (EJB) form the business tier run on the server

Enterprise Information System Tier handles EIS software, Mainframe transaction processing and database systems

More hardware can be added to specific tier to improve scalability

Changes can be made in one tier without impacting the other

Components

- Web components
  - Provide the dynamic extension capabilities to a web server

  - Servlets are Java classes that dynamically process requests and construct responses

  - JavaServer Pages (JSP) technology allows to put snippets of java code directly into a text-based document.

  - A JSP page contains two types of text
    - Static data, expressed in any text-based format such as HTML or XMl
    - JSP elements, which determine how the page constructs dynamic content

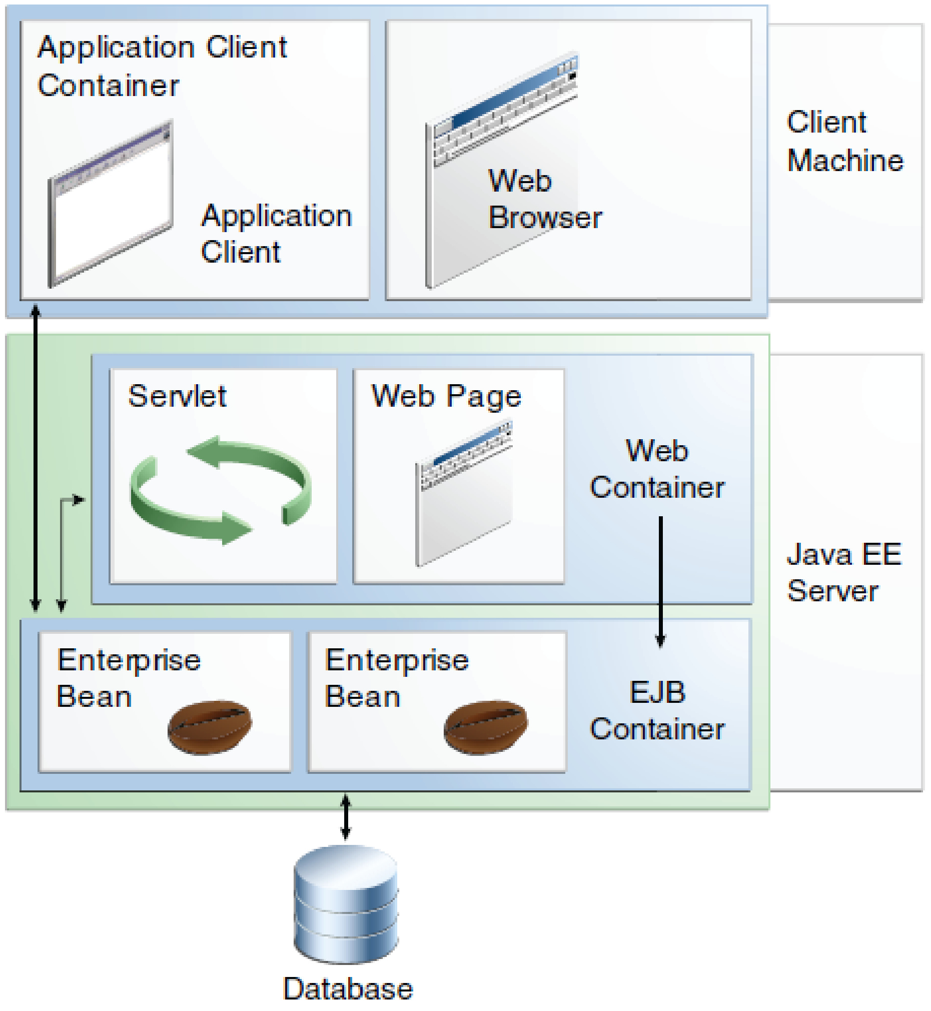
- Business components
  - Enterprise Java beans handle the business code that solves the needs of a particular business domain such as banking or retail

Containers

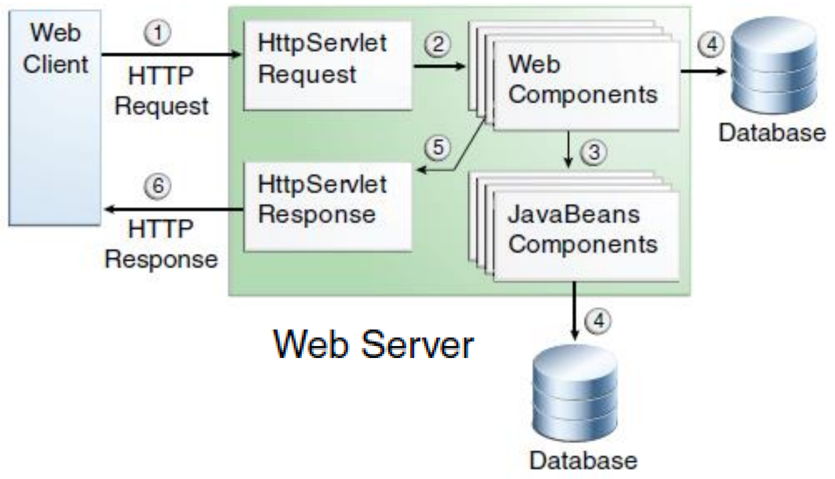
- Java EE server provides following services in the form of a Container for every component type
  - Lifecycle management
  - Communication support
  - Multithreading support
  - Security
  - Transaction and state management etc
- Developer can concentrate on developing business logic
- Containers are the interface between a component and the low-level platform-specific functionality
- The components must be packaged in to a module and deployed in to the container for execution

Container Types

- **Java EE Application server:**
  - A Java EE server provides EJB and web containers
  - Ex - JBoss, IBM Websphere
- **Enterprise JavaBeans (EJB) container:**
  - Manages the execution of enterprise beans
- **Web container**
  - Manages the execution of JSP pages, servlets
  - Ex - Tomcat



Web Container

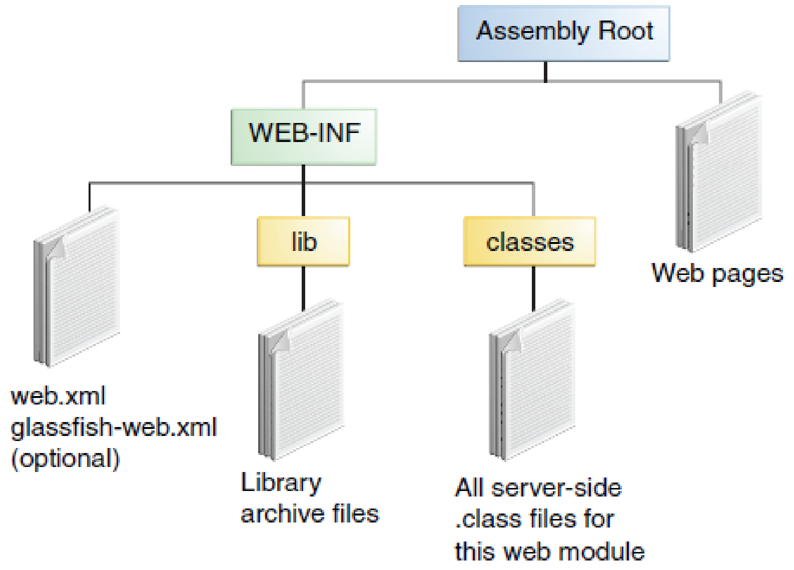


- Web Container : **Ex. Tomcat**
  - Web container supports the Web components by providing the services of a runtime platform
  - Provides services such as request dispatching, security, concurrency, and life-cycle management
  - Gives web components access to APIs

- The client sends an HTTP request to the web server
- Web server forwards request to Web container
- Web container sees the request is for a web component. It creates a request and response objects and sends it to the Web component
- Web component can interact with JavaBeans components or a database to generate dynamic content
- Web component can then generate a dynamic response or it can pass the request to another web component
- The web container converts the response object to an HTTP response and returns it to the client.

DEMO
1. TOMCAT CONTAINER
2. ADDING SERVER IN ECLIPSE
3. CREATING DYNAMIC WEB PROJECT

Web Module - .war files



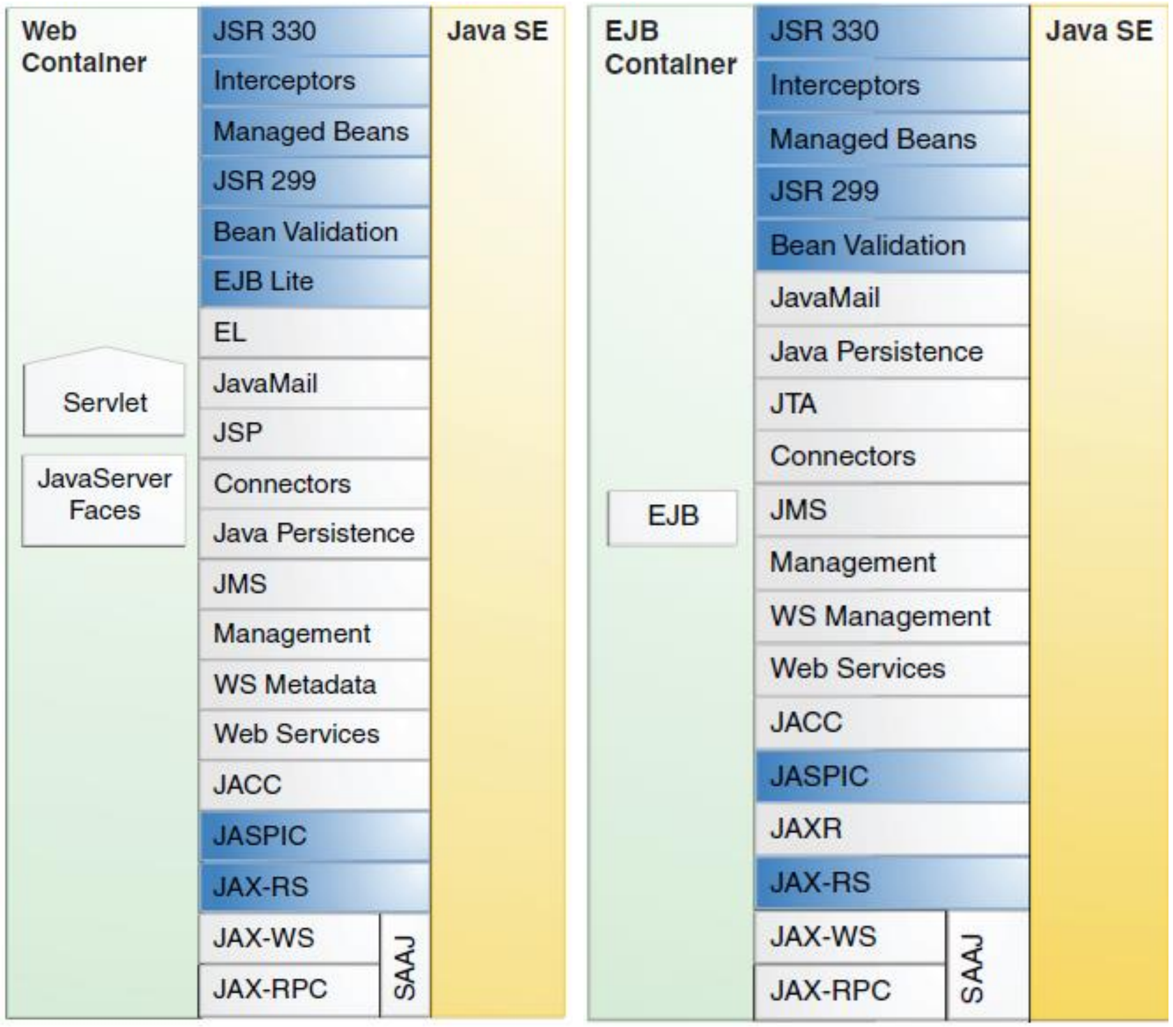
➢ *Web module*

- Smallest deployable and usable unit of web resources
- Consists of Web Components, Static pages, images, files etc

Web module has a specific structure

- The top-level directory is the *document root /context root*
- It contains JSP pages, static web resources, such as images are stored
- It contains subdirectory WEB-INF, which contains the following files and directories
  - *web.xml: The web application deployment descriptor*
  - *Tag library descriptor files*
  - classes: A directory that contains *server-side classes*: servlets, utility classes and JavaBeans components
  - tags: directory that contains tag files
  - lib: A directory that contains JAR files like third party jar file etc

JAVA EE API



Java Persistence API (JPA) is a Java standards-based solution for persistence

Java Transaction API (JTA) provides a standard interface for demarcating transactions

The JavaMessage Service (JMS) API is a messaging standard that allows Java EE application components to create, send, receive, and read messages

The JavaDatabase Connectivity (JDBC) API lets you invoke SQL commands from Java programming language methods.

The JavaNaming and Directory Interface (JNDI) API provides naming and directoryfunctionality, enabling applications to access multiple naming and directory services

The Java Authentication and Authorization Service (JAAS) provides a way for a Java EE application to authenticate and authorize a specific user or group of users to run it.

SUMMARY

J2EE Introduction

SUMMARY

In this lesson, you've learned to:

- Introduction to web applications
- J2EE Architecture