Advanced Java Application Development Using JavaEE

University of Colombo School of Computing

UCSC

Agenda

- About Us
- Objectives
- Course Structure
- Overview of Java EE
- Hypertext Markup Language (HTML)
- JavaScript
- Request Response Model
- Servlets

About Us

- Dr. Kasun DeZoysa (Course Coordinator)
- Mr. Manjusri Wickramasinghe
- Mr. Aruna P. Kastoori
- Mr. Chathura Suduwella

Objectives of the Course…(1)

- Understanding the value propositions of JavaEE
- Getting a big picture of JavaEE architecture and platform.
- Getting high-level exposure of APIs and Technologies that constitute JavaEE
- Understanding why JavaEE can be used for as a platform for development and deployment of web services.

Structure of the Course…(1)

- In order to obtain the certificate the students should obtain 80% attendance
  - Please make sure you sign each week
  - 80% is calculated from the first seven(07) weeks
- This course assumes a basic knowledge of JavaSE.
- There will be no class on 24th October, 2015

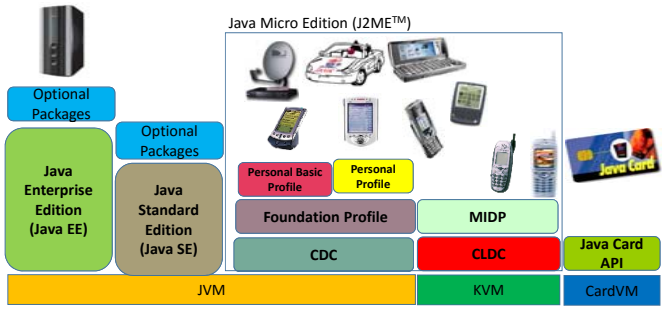
Structure of the Course…(2)

- Rough plan for 8 weeks
  - Week 1 – HTML, JavaScript, Servlets
  - Week 2 - JDBC, Listeners, JSP
  - Week 3 – MVC, EJB Basics
  - Week 4 - EJB, JPA
  - Week 5 – Hibernate
  - Week 6 – Struts
  - Week 7 – Spring
  - Week 8 – Security

Overview of Java EE

The Java™ Platform…(1)

The Java™ Platform…(2)



Why Java EE?

Challenges

Portability
Diverse Environments
Time-to-Market
Core Competence
Assembly
Integration

Key Technologies

Java SE™
Java EE™
JMS
Servlets
JSP
Connector
XML
Data Binding
XSLT

Products

App Servers
Web Servers
Components
Databases
Object to DB Tools

Legacy Systems

Databases
TP Monitors
EIS Systems

What is Java EE? …(1)

- Open and Standard platform for

"developing, deploying and managing n-tier, web-enabled, server-centric, and component based enterprise applications."

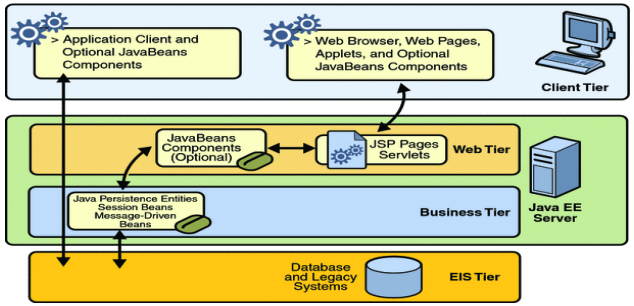
What is Java EE? …(2)

- Use "component and container" model in which container provides system services in a well-defined and as industry standard.
- Java EE standard also provides portability of code because it is based on Java technology and standard-based Java programming APIs.

What is Java EE? …(3)

- Can use any Java EE implementation for development and deployment
  - *Use production-quality standard implementation which is free for development and deployment.*
  - *Use high-end commercial Java EE products for scalability and fault tolerance*
- Can use off-the-shelf 3rd party business components

An Example



Hypertext Markup Language (HTML)

Basics…(1)

- HTML is a scripting language
- Content and styles are intermingled
- Three basic tags
  - `<html>` - Represents the complete HTML document
  - `<head>` - Contains page information.
  - `<body>` - Contains the visible content of your webpage

Basics…(2)

- Files are saved using the `*.html, *.htm` extensions
- Internet media type `"text/html"`
- Interpreted by your web browser
- HTML pages are static pages.

Simple HTML webpage…(1)

```
<!doctype html>
<html>
     <head>
          <title>First Webpage</title>
     <head>
     <body>
          <h1>Hello World!!!</h1>
     <body>
</html>
```


Simple HTML Webpage…(2)

- Save the code as `hello.html`
- Open `hello.html` in your web browser

HTML Forms … (1)

- Forms are used to collect information from the user
- The `<from>` element defines a HTML form
- Different form elements such as text boxes, text areas, buttons, checkboxes are placed in the HTML form using the `<input>` tag.

HTML Forms … (2)

- Input tag defines different elements based on the value submitted to the type attribute

```
<input type=?>
```

  - `text` -> a textbox
  - `radio` -> radio button
  - `submit` -> button with submit functionality
  - `reset` -> button resets the form to default

HTML Forms…(3)

- The form tag contains number of attributes used for various purposes
- The "`action`" and the "`method`" attributes are the most commonly used attribute.
- The action attribute specifies what needs to be done when the from is submitted
- The method attributes specifies the method of submission

Java Script

Basics…(1)

- High-level interpreted programing language for the web
- Variables are defined using the following syntax (no data type)

```
Variable name can be alphavar <variable_name> =
                          <value>;
```

  - numeric
  - Value can be numbers or strings

Basics…(2)

- Basic arithmetic operations and logoic operations are same as in java.
- JavaScript functions are made up of three parts:
  - Function name
  - Parameters
  - Body

```
var percentage = function(num, denom) {
    var result = (num / denom) * 100;
    return result;
};
```

- Function is called by its name with parameter values specified
  - E.g. `var test = percentage(30,70);`

Basics…(3)

- Can be embedded in HTML files or can be placed in a different file with the `*.js` extension.
  - When embedding in HTML the JavaScript is placed with the `<script>` tags
  - The `<script>` tags are placed in the <head> section of the HTML file.
- To accesses the HTML page elements JavaScript uses the `document` property.
  - `document.form` -> will return all the forms in the HTML document
  - `document.form["Test"]` -> will retun the form with the name "Test" from the set of forms in the HTML document.

Basics…(4)

- Validate function to check for a non-empty user name

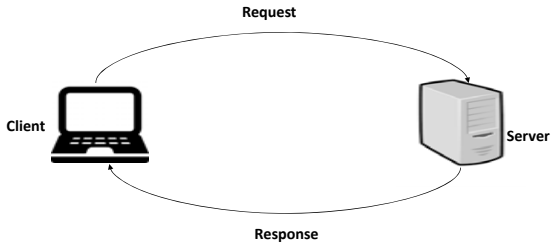
```
function checkUser() {
    var i = document.forms["test"]["user"].value;
    if (i == null || i == "") {
        alert("Username cannot be empty!!!");
        return false;
    }
}
```

- The method is called with the user click the submit button using the "`onsubmit`" attribute in the `form` tag.

```
<form name="test" onsubmit="return checkUser()">
```


Request-Response Model

Request-Response Model



Setting up Environment

- Set up Apache Tomcat 7
- Folder structure of Apache Webserver
- Deploy your HTML files.

Hypertext Transfer Protocol (HTTP)

- HTTP is use to communicated between the client and the server in the request-response model.
- HTTP client will connect to the server sends the request and disconnects.
  - Makes HTTP a stateless protocol
  - Allows other user to connect to the server
- The server responds to the client request with a HTML page.

Anatomy of a HTTP Request

- HTTP method – Specifies what kind of a request is being sent and the format of the message.

Method	Description
GET	Request information from the server
POST	Submits information to the server for processing
DELETE	Delete a specified resource from the server
OPTIONS	Queries the connection options available
HEAD	Exactly like the GET, but only returns the headers no body

- URL
- Form Parameters

Anatomy of a HTTP Response

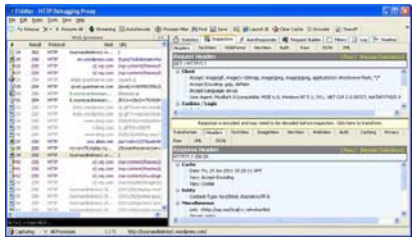
- Status Code – The code returned by the server in response is categorized into five classes based on the first digit.

Code Class	Description
1xx	Informational Messages
2xx	Success Message
3xx	Redirection
4xx	Client Error
5xx	Server Error

- Content Type – e.g. “text/html”
- Content

Activity (15min)

- Observe GET and POST requests using the Fiddler software.



Servlets

Basics…(1)

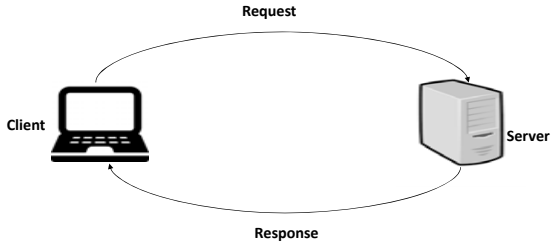
- Servlets are the Java platform technology of choice for extending and enhancing web applications.
- Web applications are helper applications that resides at web server and build dynamic web pages.
- Robust, scalable and secured
- Servlets can also access a library of HTTP specific calls and receive all the benefits of the Java language, including portability, performance and reusability

Basics…(2)

- Servlets have access to the entire family of Java APIs including the JDBC™ API to access enterprise databases.
- Servlets can also access a library of HTTP specific calls and receive all the benefits of the Java language, including portability, performance and reusability.
- This also provide better alternative to Common Gateway Interface (CGI), Netscape Server Application Programming Interface (NSAPI), Internet Server Application Programming Interface (ISAPI) etc.

Request-Response Model Revisited…

- Where does servlets fit in?



Servlet API

- Servlet API consist of two packages that contains all the important classes and interfaces
  - javax.servlet
  - Javax.servlet.http

Servlet Interface

- Contained in the `javax.servlet` package
- Contains five methods
  - Three methods are life cycle methods
    - `service(ServletRequest, ServletResponse)`
    - `init(ServletConfig)`
    - `destroy()`
  - Two are general purpose methods
    - `getServletConfig()`
    - `getServeltInfo()`

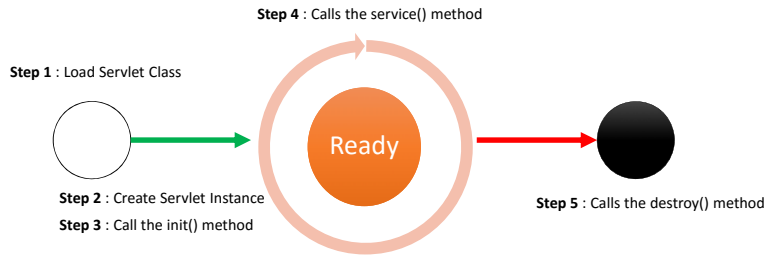
Generic Servlet Class

- This is an abstract class available in the `javax.servlet` package.
- Implements most of the important methods associated with servlets.
- Class implements `Servlet, ServletConfig` and `Serializable` interfaces.
- This class can handle any type of servlet request. Hence it is protocol independent.

HttpServlet Class

- HttpServlet is abstract class that extends the GenericServlet class.
- Provides HTTP specific methods.
- The `service()` method listens to the HTTP methods from the request stream and invokes them accordingly.
- The `service()` method is generally not overridden.

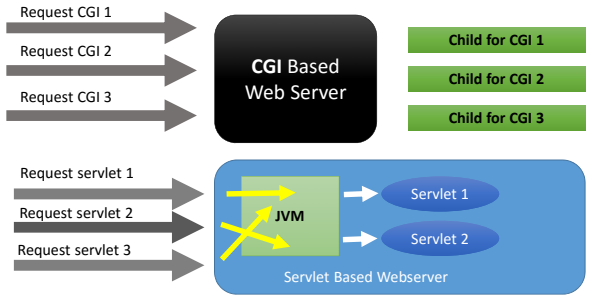
Servlet Life Cycle…(1)



Servlet Life Cycle…(2)

- Loading the Servlet Class:- A servlet class is loaded when the first request for the servlet is received by the web container.
- Servlet Instance Creation:- After being loaded the web container creates an instance of the servlet
- `init()`:- Is invoked when the web container initializes the servlet instance
- `service()`:- Listens to the HTTP requests and handle them accordingly.
- `destroy()`:- Web container all the `destroy()` method before removing the servlet instance.

Servlet vs CGI …(1)



Servlets vs CGI …(2)

- Running Servlets usually has significantly less overhead.
- Servlets can inherit processing state among invocations.
- Furthermore, because they are implemented by threads. They can use concurrently control mechanisms in java for threads to control the sharing of state of the server.
- Servlets compared to CGI program are slower only when initially loaded and are generally faster to run when loaded.
- Servlets can open database connections when initially loaded and those connections can subsequently shared among subsequent servlet accesses. By contrast CGI progress have to renew database connections each time they are run.

Servlet vs CGI …(3)

- By contrast CGI programs lack a common address space to share state information easily with each other. They could write such information to and from files but this would be very inefficient.
- Servlets are also more complex to write, handle and configure than CGI processes.