

Day 10 WebServers, Servlets, JSP

Day 9 Revisit

SOLID Principles

Design Patterns

SonarQube – Demo

NFR

DRY PRINCIPLES

- S → Single Responsibility (SRP)
- O → Open /Closed (OCP)
- L → Liskov Substitution (LSP)
- I → Interface Segregation (ISP)
- D → Dependency Inversion (DIP)

Writing Efficient, Maintainable, Clean & Secure

DRY – Don't Repeat Yourself (Reuse the existing resources as much as possible)

Design Patterns – Proven/efficient way of solving problems

- Creational (Singleton, Factory, Builder, Prototype, Object Pool, DI)
- Structural (Decorator, Front Controller, Bridge, Proxy, Module)
- Behavioural (Chain of Responsibility, State, Iterator, Specification)
- Concurrency (Join, Lock, Thread Pool, Scheduler)

NFR – Non-Functional Requirements (Performance, Scalability, Security, ....)

JMeter , Selenium

SonarQube ( Min JDK should be 11) – Static Code Analysis (Code Smell)

- Security/Vulnerabilities
- Helps to write clean code
- Scans for any other threats and generate reports
- It also consider the overall test coverage

Run with Coverage (In IDE)

Sonarlint – Sonar plugin for IDE

## Design Principle

YAGNI - YAGNI principle ("You Aren't Gonna Need It") is a practice in software development which states that features should only be added when required.

## Day 10.

### Types of Servers (Client & Server Concepts)

- Web Server ( Helps to send and receive data using web protocols) Tomcat (Servlets,JSP), IIS,
- Application Server ( WebLogic, WebSphere, JBoss ... )
- Database Server ( Mongo/MySQL)

- 1) Request Object (Client sends request to server)
- 2) Response Object ( For every valid request, server creates a response and send it back to client)

- Jenkins Server (Automation Server)
- MongoDB/MySQL
- SonarQube

### Types of Java Application

- Java Stand Alone Applications (CLI based/GUI based with a Starter class [class with main method]) {core Java Applications} – JAR (Java Archive)
- Web/Enterprise Application {Adv Java Applications/ JEE Application} – WAR/EAR (Web/Enterprise Archive)

### Entry point

- For Stand alone application – Starter class (Class with main method)
- For Web/Enterprise Application – web.xml (if it is present) [web.xml is optional]

Web.xml → Deployment Descriptor

Flavours of Java

- J2SE → Java 2 Standard Edition
- J2EE → Java 2 Enterprise Edition
- J2ME → Java 2 Micro Edition

Creating first Web based application in Eclipse EE IDE

Open Eclipse EE IDE

- 1) File → New → Other
- 2) Select “Dynamic Web App” under web folder
- 3) Provide the name of the application and configure target runtime.

**New Dynamic Web Project**

**Dynamic Web Project**

Create a standalone Java-based Web Application or add it to a new or existing Enterprise Application.

Project name: firstweb

Project location

☒ Use default location

Location: E:\Sapient\EclipseWS\firstweb Browse...

Target runtime

Apache Tomcat v8.5 New Runtime...

Dynamic web module version

3.1

Configuration

Default Configuration for Apache Tomcat v8.5 Modify...

A good starting point for working with Apache Tomcat v8.5 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name: EAR New Project...

Working sets

☐ Add project to working sets New...

Working sets: Select...

? < Back Next > Finish Cancel

- 4) Configure tomcat port number and run the app on server
- 5) Create first servlet

What is Servlet – Servlet is a adv Java Class which extends either Http/Generic Servlet

Types of Servlet

- 1) Generic Servlet
- 2) Http Servlet

Protocol – Set of Rules

Servlet – HTML code is embedded inside java code

JSP – Java Server Pages – (Java code embedded inside html code)

Every browser will have 3 engines

- 1) HTML Rendering Engine
- 2) Styling Engine
- 3) JS Engine

Static Web – Content won't change wrt time &/ user

Dynamic Web – Content will change wrt time &/ user

<% %> = JSP Scriptlet (Multi line tag)

<%@ %> = JSP Directive (Single Line tag) [page,include,taglib]

<jsp:dataSource> </jsp:dataSource> = JSP Action Tags (Single line)

<%= %> - JSP Expression Tag (Single line Tag) (out.print)

<%! %> -JSP Declaration Tag

<!-- --> == HTML Comments

<%-- --> = JSP Comments (Multi line)

9 Implicit Object in JSP

Out

Response

Request

Session

Config

Page

Exception

PageContext

Application

JSTL – JSP Standard Tag Library

`${}` = Expression Language Syntax