Hibernate Advance Concepts

Caching, AutoScaling, Criteria, Native Query, Named Native Query

HQL,

Javalin & Spring Boot. (Web Services)

Day 29 Revisit – Hibernate

Hibernate is a ORM (Object Relational Mapping) Framework.

Hibernate simplifies the process of interacting with any RDBMS (Oracle/MySQL/MSSQL/DB2/PostGres)

Hibernate is a JPA Implementation (Java Persistence API)

Persistence – Serializing the state of an object to a database table.

They are two ways to use Hibernate

1. Using a xml mapping file (This will have info about the bean class and the database table)
2. Using annotation (This is recommended method – It eliminates xml mapping) [Add annotations to the entity bean class directly]

Hibernate uses a xml configuration file or text based properties file.

This configuration file contains details about the database (URL, username, password, driver\_details, Dialect to interact with that Database)

Name of hibernate configuration file is hibernate.cfg.xml or hibernate.properties

Official Site : <https://hibernate.org>

Hibernate can be downloaded as a zip file or It can be added as a dependency in maven project.

Hibernate can be used in both core [stand-alone] & adv Java (Web based) projects.

Adv of using Hibernate

1. Writing DB independent Queries
2. Caching improves performance
3. Auto Scaling.

To create “Dynamic Web Project” we need Eclipse EE edition.

Eclipse EE Download link -- <https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2022-03/R/eclipse-jee-2022-03-R-win32-x86_64.zip>

Two types of Web Applications

1. Static Web Application ( Data will not change with respect to users and time --- Constant data) – Website about a particular place, particular person – HTML,CSS/JS
2. Dynamic Web Application (Data will change with respect to the users and time --- GMAIL Inbox, Social media dashboard ) – HTML/CSS/JS (Server Side Code – Servlets & JSP)

Types Java Applications

1. Stand-alone (Core-Java) Applications – This is going to run on JVM and produces the result in console – J2SE (CUI/GUI based --- Java Project / Swing/AWT/Java FX based)
2. Web Applications – This is going to run with the help of Container/Server (tomcat/glass fish/weblogic) and produces the result in web browser. [Adv-Java Applications]

Core Java/ Stand alone project --- Starting/Entry point is main() method

Adv Java/ Dynamic Web Project --- Starting/Entry point is web.xml (Deployment Descriptor)

Web Application can also be referred as client – server applications.

1. Client (Web – browser Google Chrome, Mozilla Firefox, Apple Safari, Opera ….)
2. Server (Web Server/Application Server – Tomcat, Weblogic, GlassFish, …..)

Types of Servers

1. Web Server (It will handle JSP, Servlets & JSF codes)
2. Application Server (In addition to Servlets & JSP it also handles EJB, Message Beans, Web Services, JNDI, Message Queues, Mail Service etc.,)
3. Database Server (Backbone of RDBMS) – With the help SQL, it helps to manage the data in a structured format.

Browser contains 3 Engines

1. HTML rendering Engine (Which is capable of producing result for HTML documents)
2. JS Engine (This is responsible for running the JavaScript code)
3. Style/CSS Engine (This is responsible to generate styling to the Web Page)

To Create a dynamic web application, we can use any programming Language

In JAVA --- Servlet, JSP, JSF

In .Net --- ASP, .Net Core…., C#

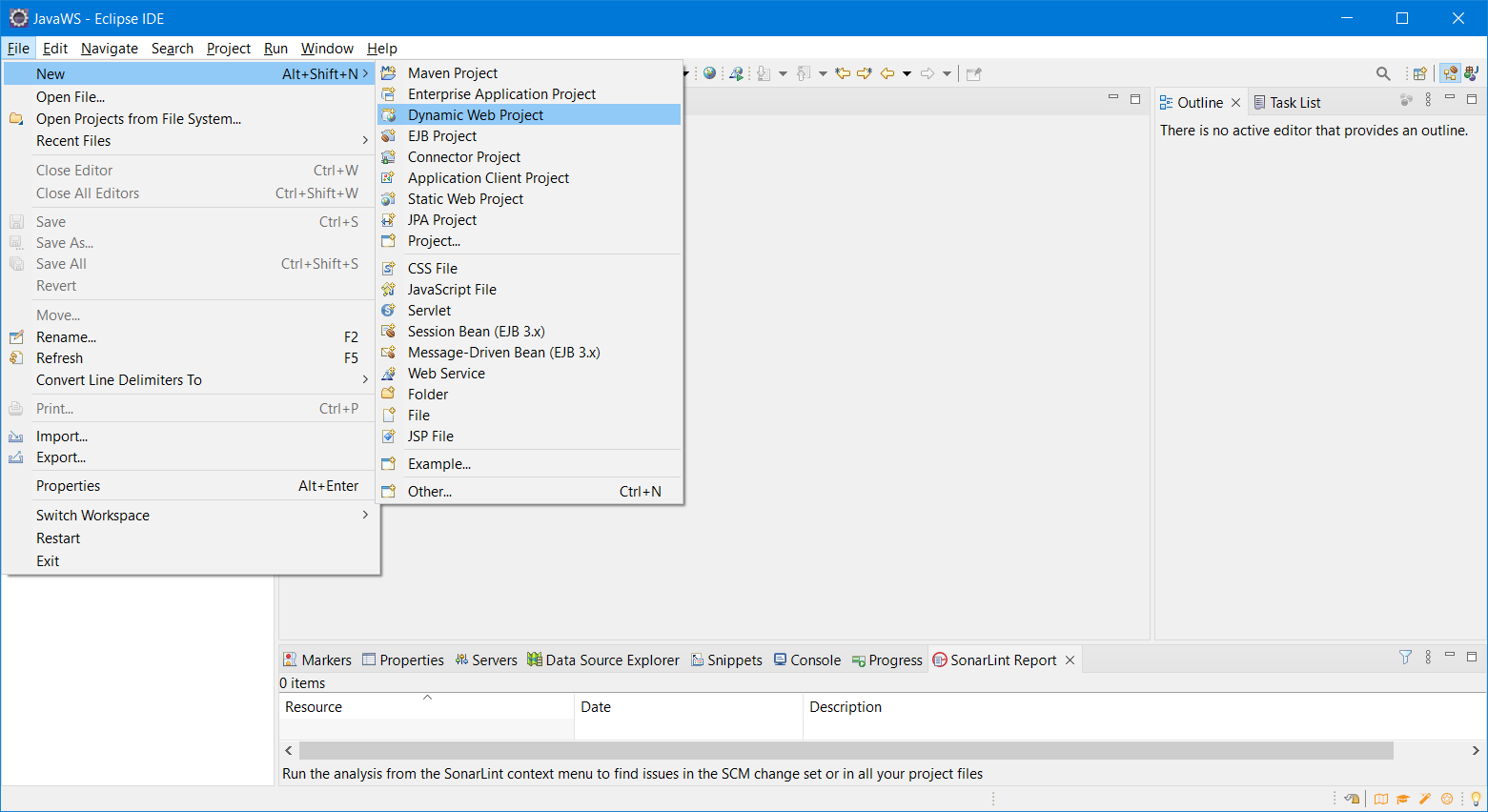
Python, php to create server side code.

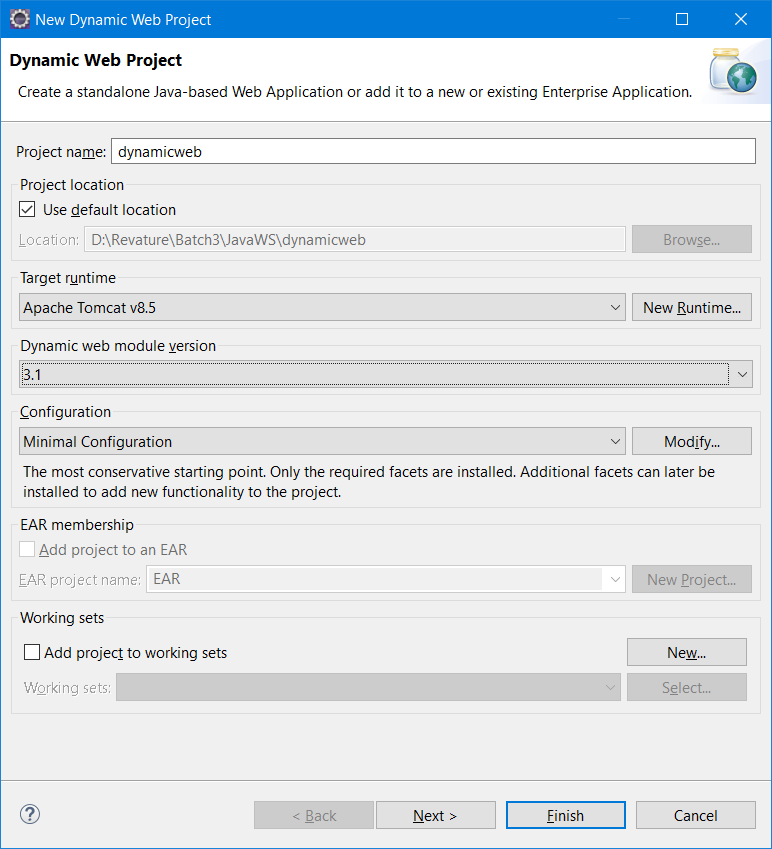
Servlet = **Java** with HTML code ( HTML code is embedded inside Java code)

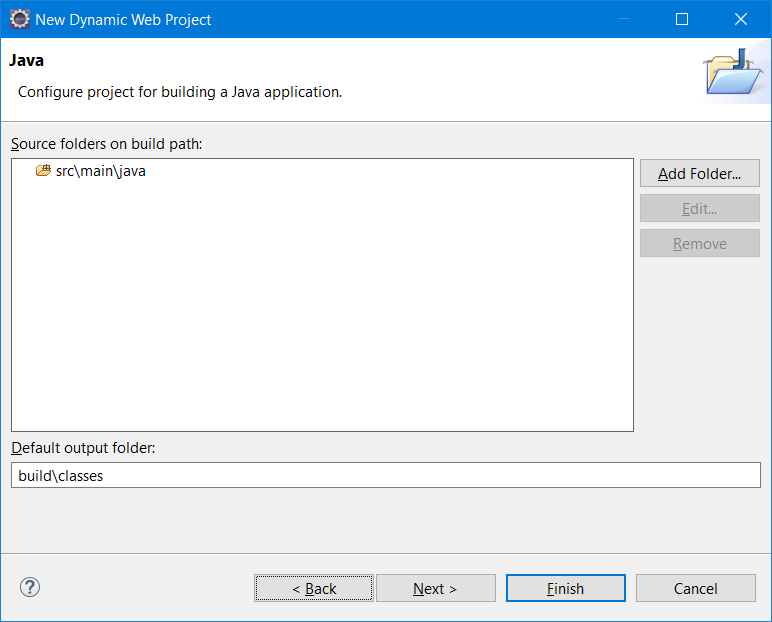
JSP = **HTML** with Java code ( Java code is embedded inside the HTML code)

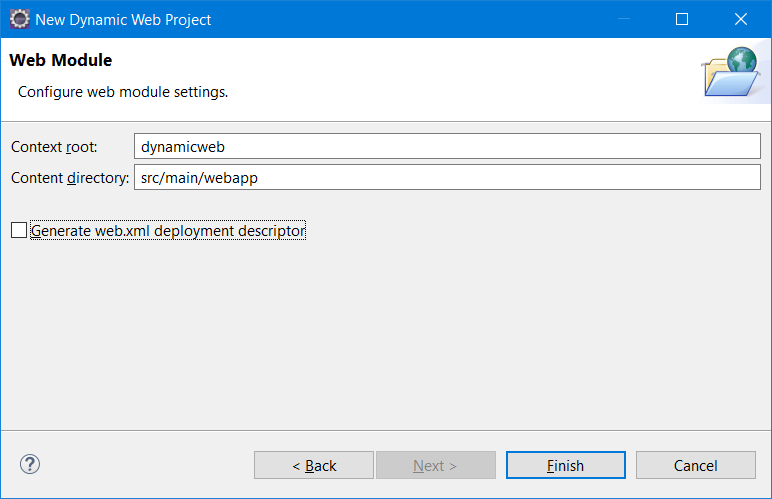
Creating a Dynamic Web Project in Eclipse EE.

* Open Eclipse
* Create a new Dynamic Web Project (File🡪 New🡪Dynamic Web Project or File🡪New🡪Other 🡪Dynamic Web Project (Under Web folder)

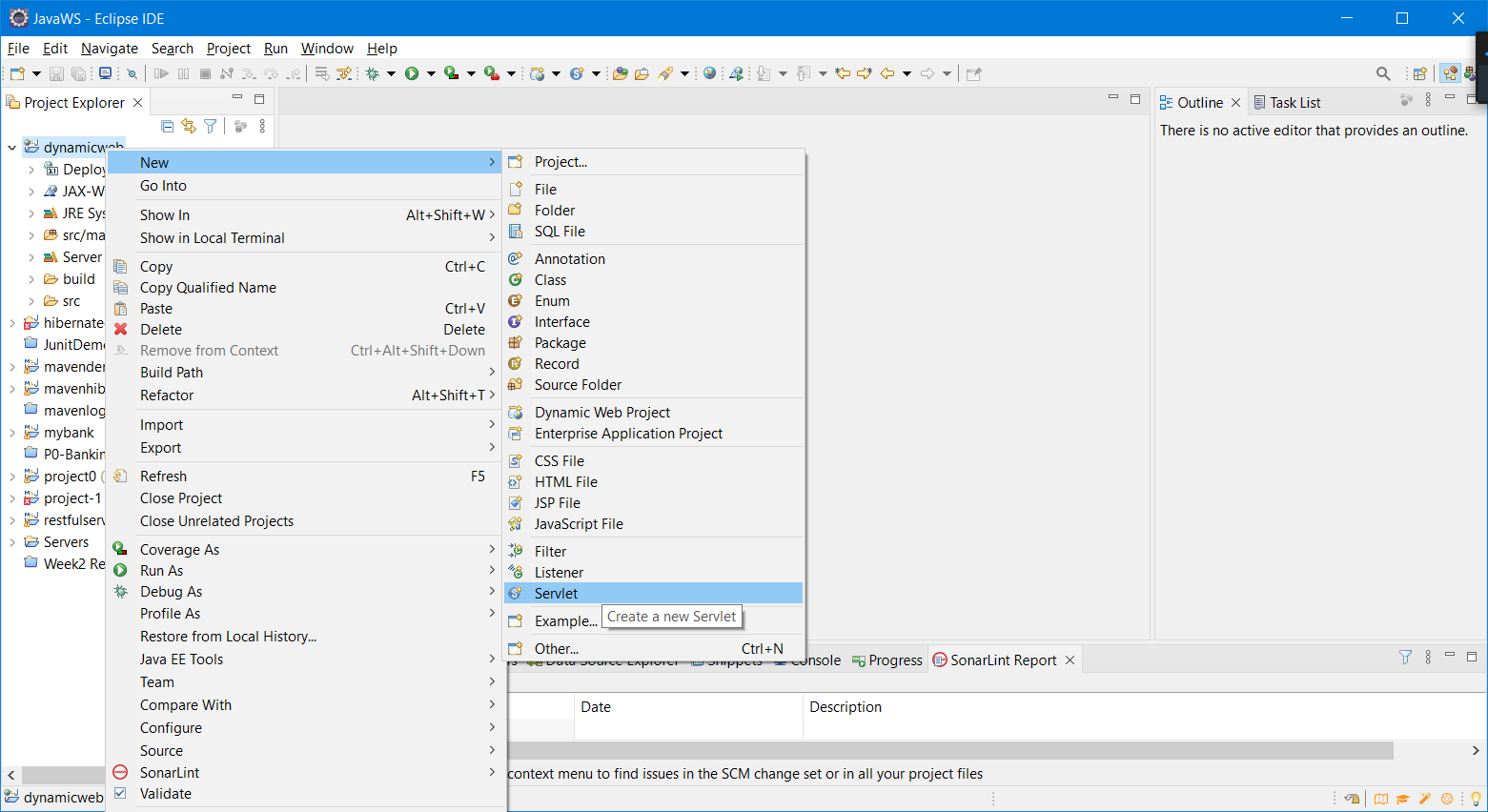


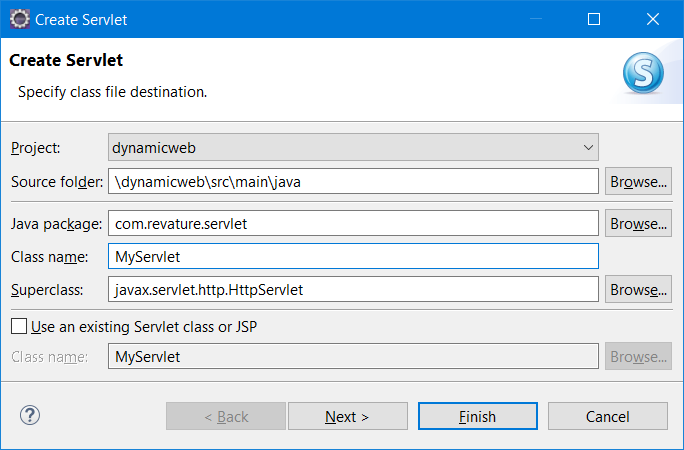


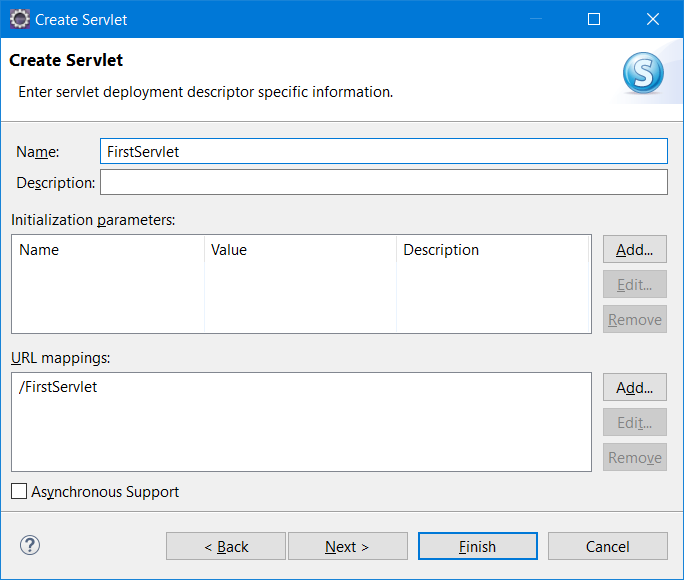


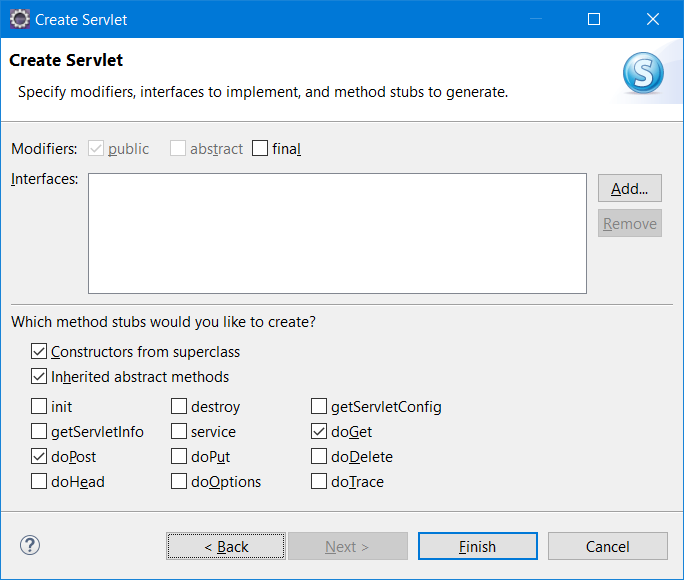


Creating a Servlet Class in dynamic web project









META-INF == Meta Information (Data about Data – What type of Web App, What is the main content of it)

WEB-INF == Web Information (How to start this web project, what will be the starting point, What is the landing/home page name – web.xml)

WEB-INF/lib == this path is called as “Deployment Assembly”

Web.xml == is called as “Deployment Descriptor” – What is the landing page, what are all URL Mappings/ What are all Servlets we have it/ Any Initial Configuration) (web-app is the root element)

JSP – Java Server Pages (Server Side Coding)

Servlet – It’s a Java Class that gets executed in a server and produces result (response) for the browser.

Two Types of Servlet

1. Generic Servlet [It supports all the protocol like http,ftp,smtp,pop etc.,]
2. HttpServlet [It support only http protocol] --- Specific protocol servlet

Servlet Life cycle

1. Initialisation 2) Service 3) Destroy

Service Methods a) doGet() b) doPost() c) doPut() etc.,

JSP (HTML + Java = Java code embedded inside the html)

HTML is a tag based language

JSP is also a tag based language developed by java developers.

Types of Tags in JSP

1. Action Tag <jsp:action> </jsp:action>
2. Scriptlet Tag <% %> --- Multi line tag
3. Expression Tag <%= %> --- Single Line Tag
4. Directive Tag <%@ %>

JSTL – JSP Standard Tag Library --- Format, Function, SQL, XML – JSTL.jar

JSTL Jar download url --- <http://www.java2s.com/Code/Jar/j/Downloadjstl12jar.htm>

CREATE TABLE `p1`.`users` (

`id` INT NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(45) NOT NULL,

`password` VARCHAR(45) NOT NULL,

`email` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `username\_UNIQUE` (`username` ASC) VISIBLE,

UNIQUE INDEX `email\_UNIQUE` (`email` ASC) VISIBLE);

<%@ taglib prefix=*"sql"* uri=*"http://java.sun.com/jsp/jstl/sql"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<sql:setDataSource var=*"conn"* user=*"root"* password=*"root"* driver=*"com.mysql.cj.jdbc.Driver"* url=*"jdbc:mysql://localhost:3306/p1"* />

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib prefix=*"sql"* uri=*"http://java.sun.com/jsp/jstl/sql"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Validate Login</title>

</head>

<body>

<sql:setDataSource var=*"conn"* user=*"root"* password=*"root"*

driver=*"com.mysql.cj.jdbc.Driver"* url=*"jdbc:mysql://localhost:3306/p1"* />

<sql:query var=*"rs"*>select \* from users where username='${param.username}' and password='${param.pwd}';</sql:query>

<c:if test=*"*${rs.rows}*"*>

<c:out value=*"Login is Successfull!!!"*></c:out>

</c:if>

<c:otherwise>

<c:out value=*"Error While Loggin In. Pls Check the Credentials!!!"*></c:out>

</c:otherwise>

</body>

</html>

P1 Front End Choices

1. HTML/CSS/JS (Bootstrap –optional)
2. Servlet/JSP/JSTL
3. Angular

P1 Database Choices

1. MySQL (Local/RDS)
2. PostGres (Local/RDS)

P1 BackEnd

1. JAVA – Hibernate
2. Javalin / Spring Boot Web Service

<http://javainsimpleway.com/hibernate-crud-operations-with-java-web-application/>