Quick Revisit

* Spring Framework – It’s a Java Based framework to create loosely coupled enterprise Application
* It uses two important design patterns IoC (Inversion Of Control), DI (Dependency Injection)
* Spring framework contains many modules spring core, spring web/mvc, spring aop, spring security, spring data etc.,
* Spring framework is also called as framework of frameworks (It supports other framework as well – Struts, Hibernate etc.,)
* It mainly uses POJO classes (Plain Old Java Object)
* Bean is the back bone of Spring framework.
* Spring framework manages the life cycle of bean (Object creation, object destruction etc.,)
* There should be xml based configuration using which it can configure all of the beans used for the spring application.
* To configure bean, it used xml based configuration or annotation or java based configurations.

Spring Boot

* It’s a simplified way of creating Spring based application
* It’s a opinionated framework which supports text based configuration. (application.properties)
* Spring Boot will automatically configure many things which includes beans, datasource, views, dao etc.,
* Spring boot is used to create Web services, micro services, cloud based services, reactive applications, batch applications etc.,
* Spring boot mainly uses an annotation called @SpringBootApplication which is the combination of @AutoConfiguration, @ComponentScan &
* 3 ways of creating spring boot applications (1. Spring Initializr (<https://www.start.spring.io>) 2. STS (Spring Tool Suite) 3. SpringBoot CLI (Command Line Interface)

JDBC based Application using MySQL

Spring Based CRUD application using MySQL

MAVEN – Java Based Project Management Tool

Maven uses xml configuration.

Maven Configuration file name is pom.xml

Maven Will do

* Build the Project (Converting the source code to byte code .java 🡪 .class)
* Test the Project (Running Junit test cases)
* Package the project (Single jar, war, ear file)
* Deploying the project (Adding this single jar or war in the server)
* Generate source code
* Managing dependency of the project

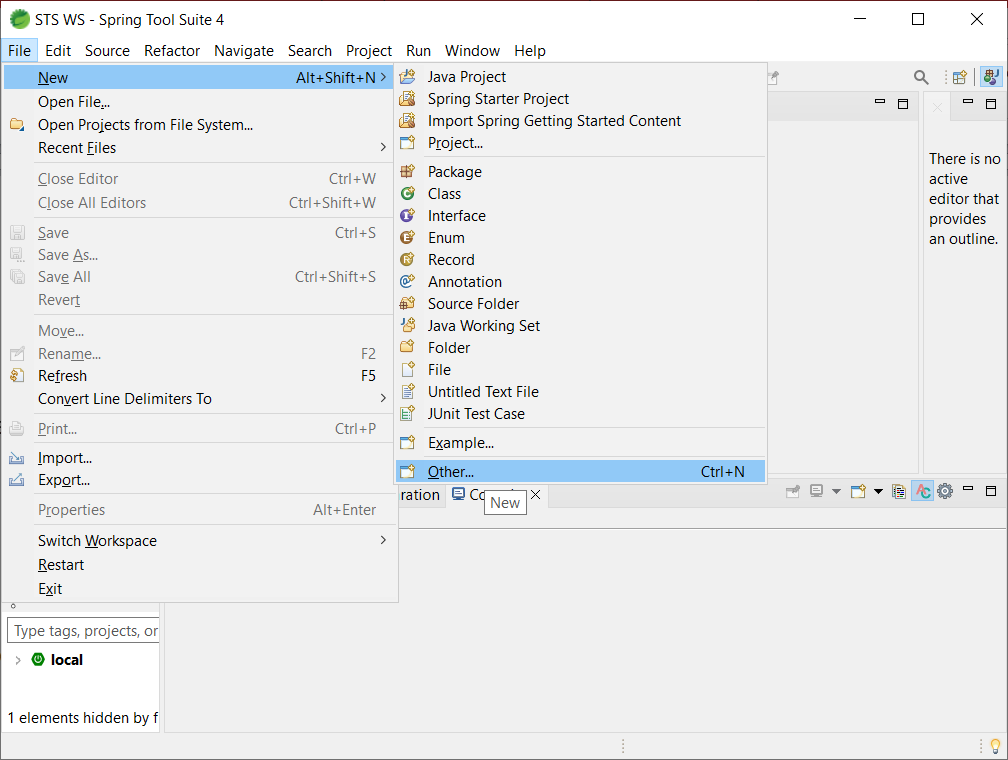
Maven uses two properties to uniquely identify java project

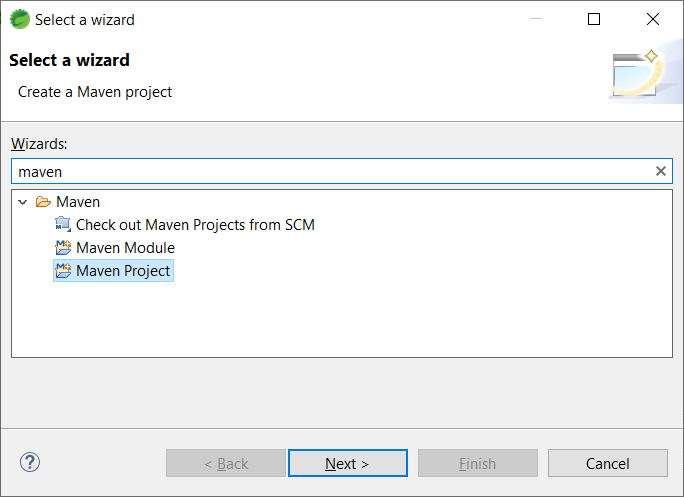
1. groupId (reverse of the company url)
2. artifactiId (name of the application)

The combination of groupId and artifactId will be used as a base package of the application.

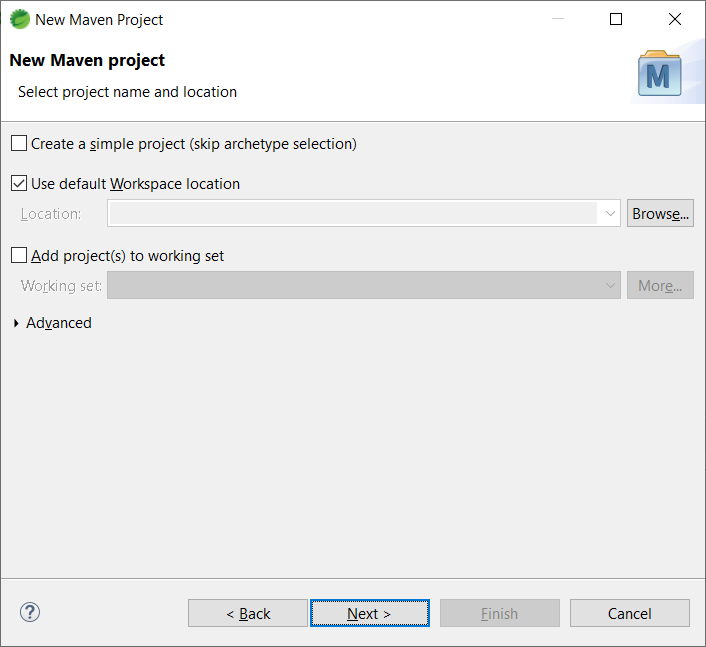
Steps to Create Maven based project

1. Open Eclipse or STS
2. Create a new Maven Project (File 🡪 others 🡪 Maven 🡪 Maven Project)

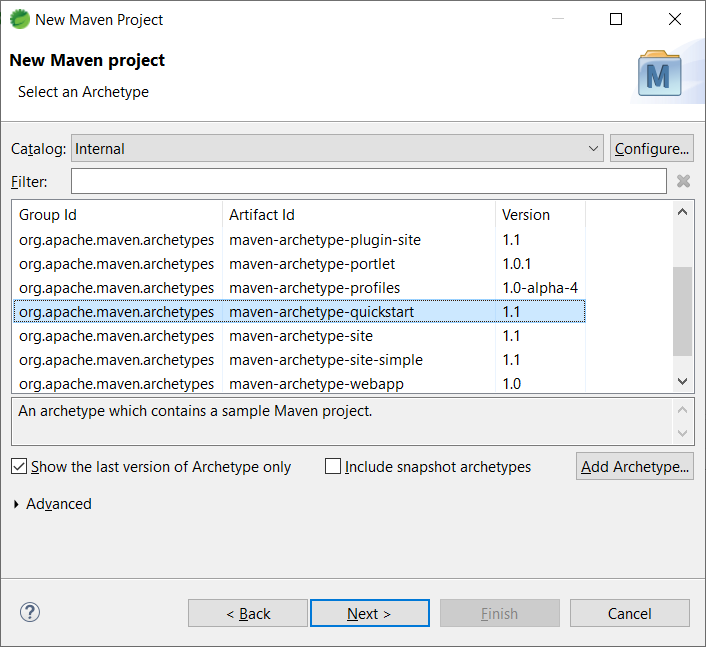




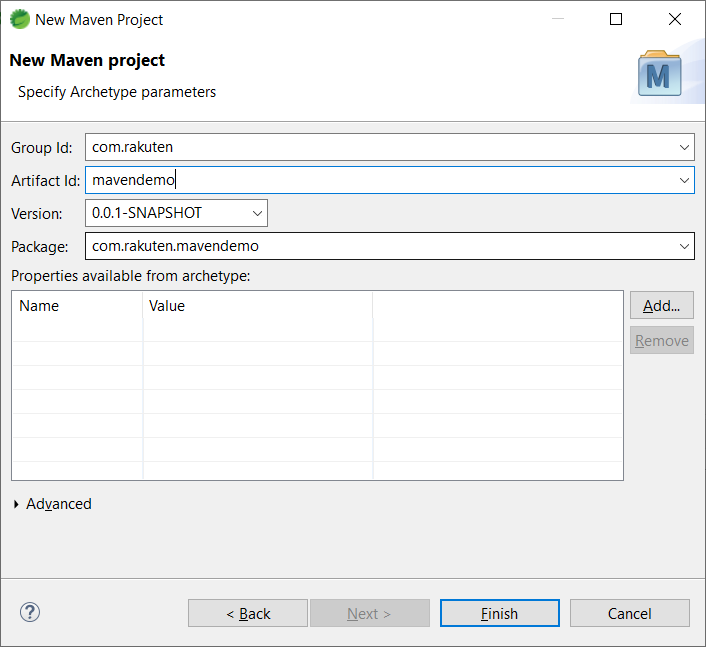
1. Click “Next”



1. Click “Next”



1. Select “maven-archetype-quickstart” and click Next.
2. Enter groupId, artifactId for the maven project



1. Click “Finish” button

@SpringBootApplication = @SpringBootConfiguration + @EnableAutoConfiguration+@ComponentScan

Dependency

Java Program

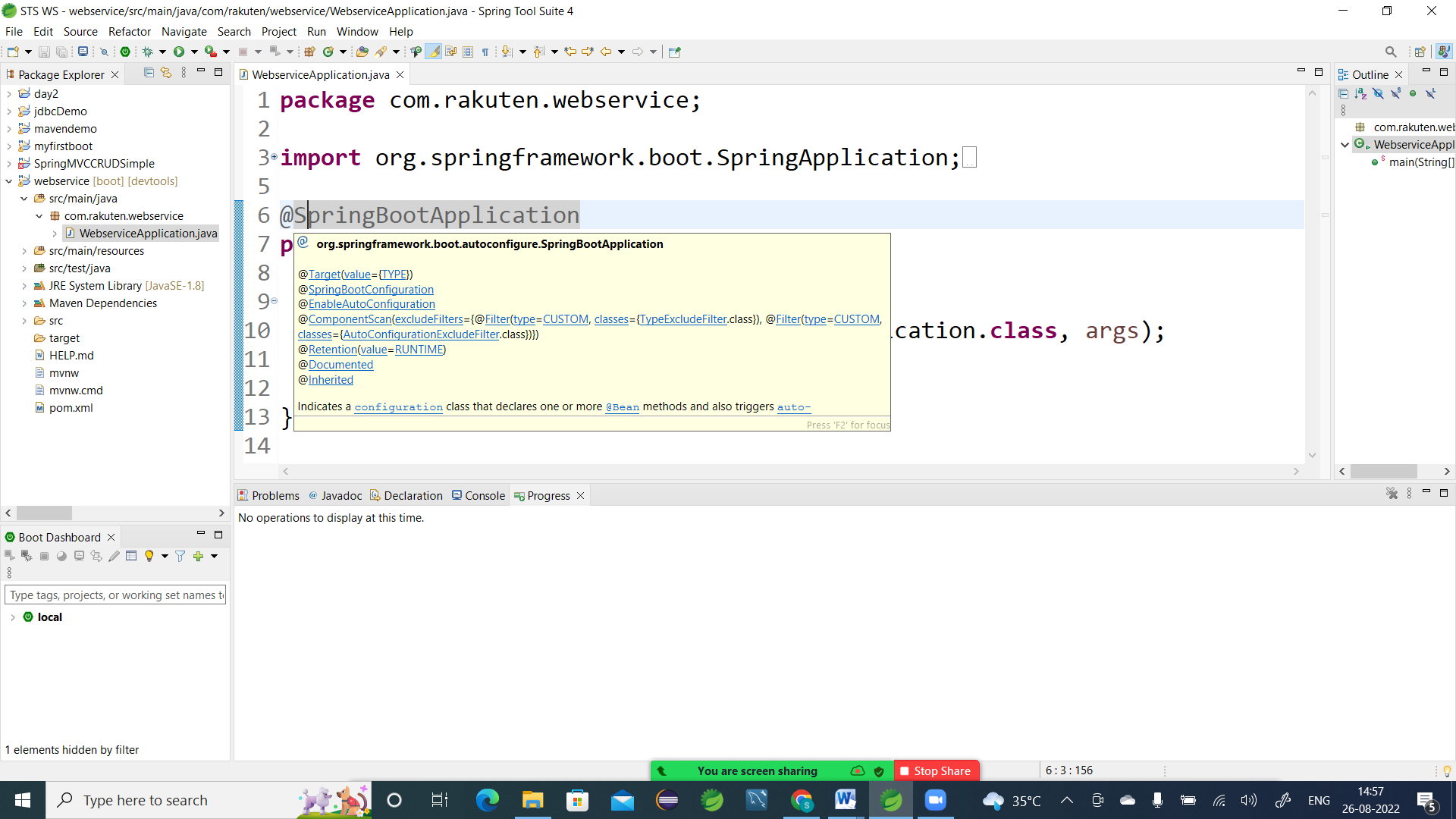
JAVA Programming Lang ( Print some data in the console, you can get input from user, you can read files, you can create multiple-threads)

Database (External Application) JDBC Api (JAVA)

RDBMS – MySQL, MSSQL, Oracle ….

Dependencies (It’s a Jar file = Group of Java class)

Driver jars. Spring framework (spring core - )



Web Service – Service offered through web (with the help of internet)

Machine to Machine (Electronic Devices) communication using https or any protocol.

Why Web Service:

ABC Bank Debit Card in ABC ATM (JAVA) - Linux

ABC Bank Debit Card XYZ ATM (.Net) – Mac

Types of Web Service

1. SOAP - Simple Object Access Protocol – XML files – WSDL (Wisdel) – WebService Definition/Description Language
2. REST – REpresentational State Transfer – (It uses JSON files and already available http methods get, post, put,delete to Communicate)

JSON & XML – Platform,Language & Architecture independent way of representing/sharing data.

Parser - Is a software used to extract data from JSON/XML file

Employees employees = [ {“id”:100,”name”:”ABC”,”email”:”[abc@gmail.com](mailto:abc@gmail.com)”}, {“id”:101,”name”:”xyz”,”email”:”[xyz@gmail.com](mailto:xyz@gmail.com)” }];

XML

<employees>

<employee>

<id>100</id>

<name>ABC</name>

<email>abc@gmail.com</email>

</employee>

<employee>

<id>101</id>

<name>xyz</name>

<email>xyz@gmail.com</email>

</employee>

</employees>

JSON

Employees = [

{

“id”:100,

“name”:”ABC”,

“email”:”abc@gmail.com”

},

{

“id”:101,

“name”:”xyz”,

“email”:”xyz@gmail.com”

}

];

Web Application GMAIL – URL

<https://www.gmail.com> --- URL (Address bar of browser)

URL – Uniform Resource Locator (Web Address) – Address of website

Web Service – End points – URI

URI – Uniform Resource Identifier api/v2/courses api/v2/employees (address bar of browser)

@RestController – This annotation converts a POJO class to a RestController Class

@RequestMapping – It maps a URI with a method in Controller or RestController

MVC is very popular Web Design Pattern

Web Applications will be divided into three different layers

1. Model (Data – Entity Beans, database table)
2. View (Template/ User Interface – HTML, JS based framework )
3. Controller (Back End code – It can be any server side code)

Web Service – Calling a method written in any programming language using a URI

@RequestMapping -- It maps to the http get method by default.

CRUD operation – Database

Create/Insert Operation

Read Operation (ReadOne, ReadAll)

Update Operation

Delete Operation

REST web Services uses http methods

|  |  |  |
| --- | --- | --- |
| **Sl No** | **http Method** | **Database Operation** |
| 1 | get() | Read Operation (One/All) |
| 2 | post() | Create/Insert |
| 3 | Put() | Update |
| 4 | Delete() | Delete |

Spring Data JPA

JPA – Java Persistence API (Specification)

JDBC (Specification) – Only interfaces are available. Implementation is provided by different 3rd parties.

JDBC Driver is the implementation of JDBC API by database developers

JPA – specifies a simplified way of interacting with any RDBMS

JPA implementation

* Hibernate (ORM – Object Relational Mapping) -- JAVA RDBMS Mapping [ORM Framework]
* iBatis
* eclipseLink
* Spring Data JPA

Serialization – Is the process of storing the state(properties) of an object in a flat file. (.txt, .rtf,.doc)

Persistence – Is the process of storing the state (properties) of an object in DB table

JPA Annotations

@Entity (It represents a database Table) = Used to create Entity Bean object in Spring Boot

@Table (It provides a name to the data base table)[Class Annotations]

@Id [Column level or property annotation] – It represents the primary key column of the table

@Column [Used to provide meta data to column] – To specify column data type, range, null , default value

@OneToMany

@ManyToMany

@ManyToOne

DAO – Data Access Object

findAll();

findById(int id)

save(Object obj)

update (int id, Object obj)

delete(int id);