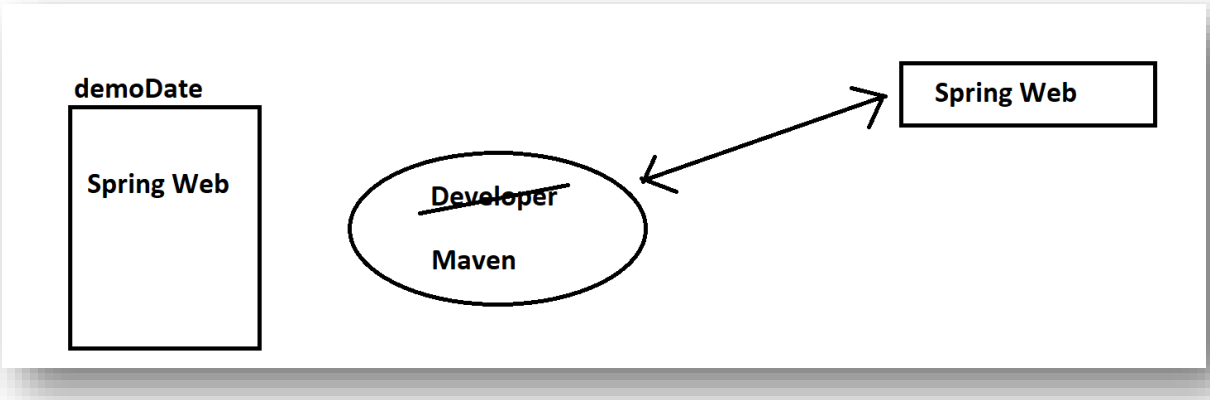
Returning date

```java
package com.flynaut.demoDate.rest;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

import java.time.LocalDate;

@RestController
public class DateController {
    //@RequestMapping(value="/date",method =
RequestMethod.GET)
    @GetMapping("/date")
    public LocalDate date(){
        LocalDate localDate = LocalDate.now();
        return localDate;
    }
}
```

Maven

- Project build tool
- Used for build management and managing dependencies



- Tell maven the projects we are going to work on(dependencies)
- Go out and download the JAR for us

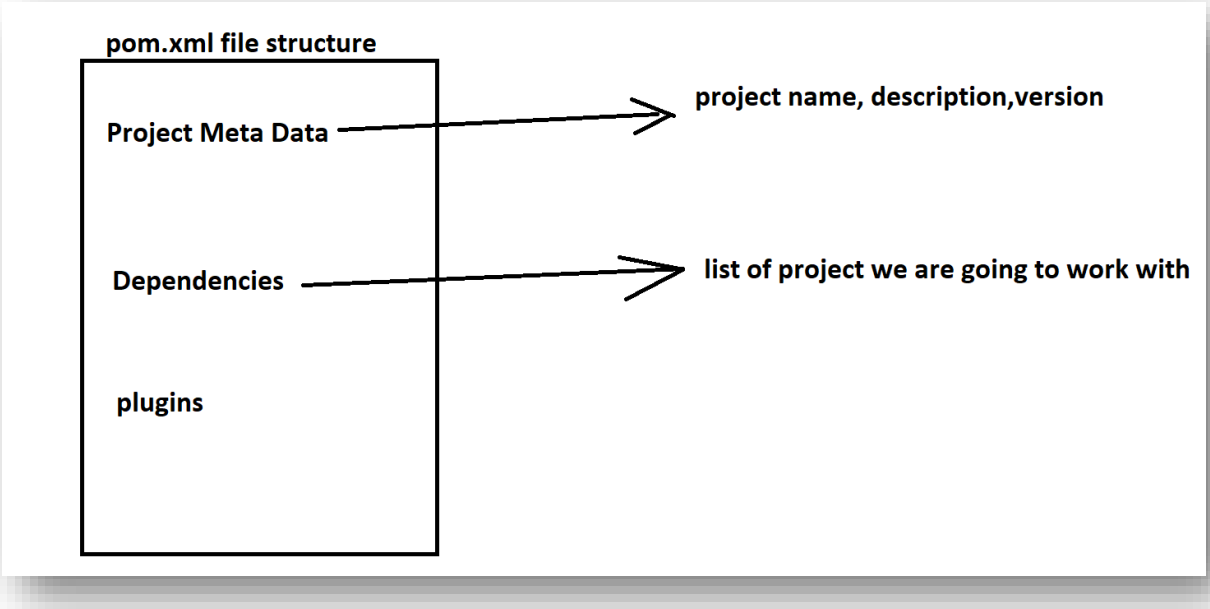
Maven Project Structure

Directory	Description
src/main/java	Our source code
src/main/resources	Properties/ configuration files used by our application
src/test	Unit testing code and properties
target	The destination directory for compiled code(automatically created by maven)
Pom.xml	Maven Configuration File

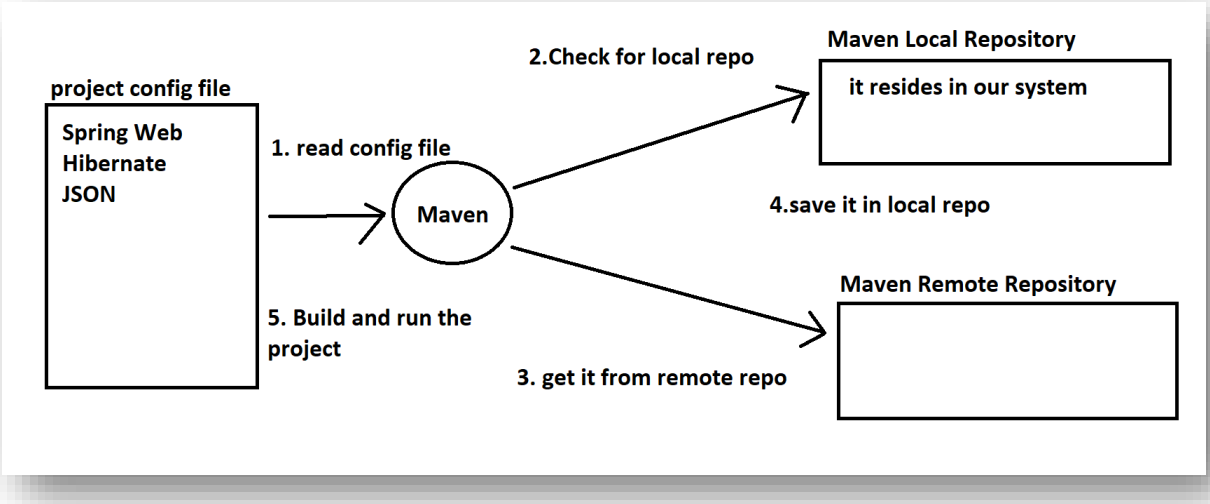
Pom.xml

Project Object Model file

- Configuration file for our project
- Shopping list ☺
- Located at the root position of our project


pom.xml file structure
Project Meta Data
project name, description,version
Dependencies
list of project we are going to work with
plugins

Maven Flow


project config file
Spring Web
Hibernate
JSON
1. read config file
Maven
2.Check for local repo
Maven Local Repository
it resides in our system
4.save it in local repo
Maven Remote Repository
5. Build and run the project
3. get it from remote repo

Project Coordinates:

- To uniquely identify a project
- Similar to GPS coordinates for office - longitude,latitude
- Precise address for our office(city, street, homeno.)

```
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.4.4</version>
```

GroupId: Name of company, group, organization

ArtifactId: Name of our project

Version

If we want to add dependencies manually we need a GAV

Maven Wrapper Files

mvnw

mvnw.cmd (for windows)/ mvnw.sh (for mac/linux)

application.properties file

By default, SB loads properties from this file

- Created by spring initializer
- It is empty at beginning

We can add properties in this file:

Server.port = 9090

We can add our custom properties

```
spring-boot-starter
```

A collection of maven dependencies with compatible version.

```
spring-boot-starter-web
```

- Spring-web
- Spring-wenmvc
- Tomcat
- JSON

Building Spring application is tedious

Why?

- It will be great if there is a list of maven dependencies collected as a group of dependencies- one stop solution
- If we have one stop solution we don’t have to search for each dependency.

Solution- SB Starters

- A curated list of maven dependencies
- It reduces the configuration part

If we are building a Spring app which need web,security

We will simply select dependency in SI

It will add all the appropriate dependencies in pom.xml grouped together in Spring-Boot-Starter

1. spring-boot-starter-web
   - building web apps, Tomcat as an embedded server
2. spring-boot-starter-security
   - To add the security support
3. Spring-boot-starter-jpa
   - Gives database support with Hibernate & JPA

SpringBoot Dev Tools

The Problem:

When we are running the SB app

And If we are making any changes in the source code

Then manually I have to stop the application and restart it ☹

The Solution- SpringBoot Dev Tools ☺

Automatically restarts our application when we make any changes in our source code

How?

Simply add the dependency in pom.xml

Step1: add dependency in pom.xml

```
<dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
      <optional>true</optional>
</dependency>
```

Step2:

File -> settings -> build,execution,deployment -> compiler -> check the box(build project automatically)-> apply -> ok

Step3:

Settings-> advanced settings -> check (allow automate)