

Day 2: Spring | Spring Boot :: Course Plan

①

- ⇒ Spring | Spring boot framework are given as the collection of modules (30+ modules)
- ⇒ In the development of Spring | Spring boot Projects we can use only those modules which are really required. This helps to reduce the size of the project.
- ⇒ For every module Spring | Spring boot specific APIs libraries are given in the form of JAR files. So we need to add only those JAR files to the project that are required as part of the project.

Development :

- ⇒ modules are core, aop (Aspect Oriented Programming), data JPA, data MongoDB, MVC, Security, Cloud module, and etc.

(Every module is built on top of core module)

⇒ If any module is used in Spring style then it is called Spring module.

Eg.: Spring core, Spring mvc, spring data spa and etc.

⇒ If any module is used in Springboot Style then it is called Springboot module.

Eg.: Spring boot core, Springboot mvc, Spring boot data spa and etc--.

→ Course Plan

=

⇒ Introduction about Languages, Technologies and Frameworks
Diff categories of Java programming)

⇒ Container vs Service (For automating the execution of Comps)

⇒ Spring Core (Basics of Spring Programming)

- ⇒ Spring boot Core (Basic of Spring boot Programming)
 - ⇒ Spring boot data (For interacting with SQL DB like Oracle, mysql, and etc...)
 - ⇒ Spring boot data mongoDB (For interacting with mongoDB Db which is NoSQL Db like)
 - ⇒ Springboot Scheduling (To run the Jobs/tasks by Enabling Scheduling)
 - ⇒ Springboot mailing (To performing mailing operations from the Springboot projects)
 - ⇒ Springboot Mvc (To develop web applications (websites) Using Springboot)
 - ⇒ Spring boot Rest (MVCTT) (To develop distributed Apps Using Spring boot)
- More to come.

- ⇒ In core Java we study Syntaxes (Rules) and Semantics (Structure) of the Java programming.
- ⇒ In Spring core module, we study the Syntaxes (Rules) and Semantics (Structure) of spring programming
- ⇒ In Springboot core module, we study the Syntaxes (Rules) and Semantics (Structure) of the Springboot programming
- ⇒ If app data is having fixed structure | schema | attributes and also deals formatted data then prefer working with SQL DB Slws. Eg.: Oracle, MySQL, etc.
- ⇒ If app data is having dynamically growing structure | schema | attributes and also deals with unformatted data then prefer working with NoSQL DB Slws
Eg.: MongoDB, Cassandra, neo4j, and etc.

⇒ To store product details of e-commerce app then prefer working with NoSQL DB. like

⇒ Scheduling the job task is nothing but executing the job repeatedly or 1 time, on the given time schedule.

like daily, monthly, weekly and etc.

⇒ Web Application / website is client-server app (B2C)

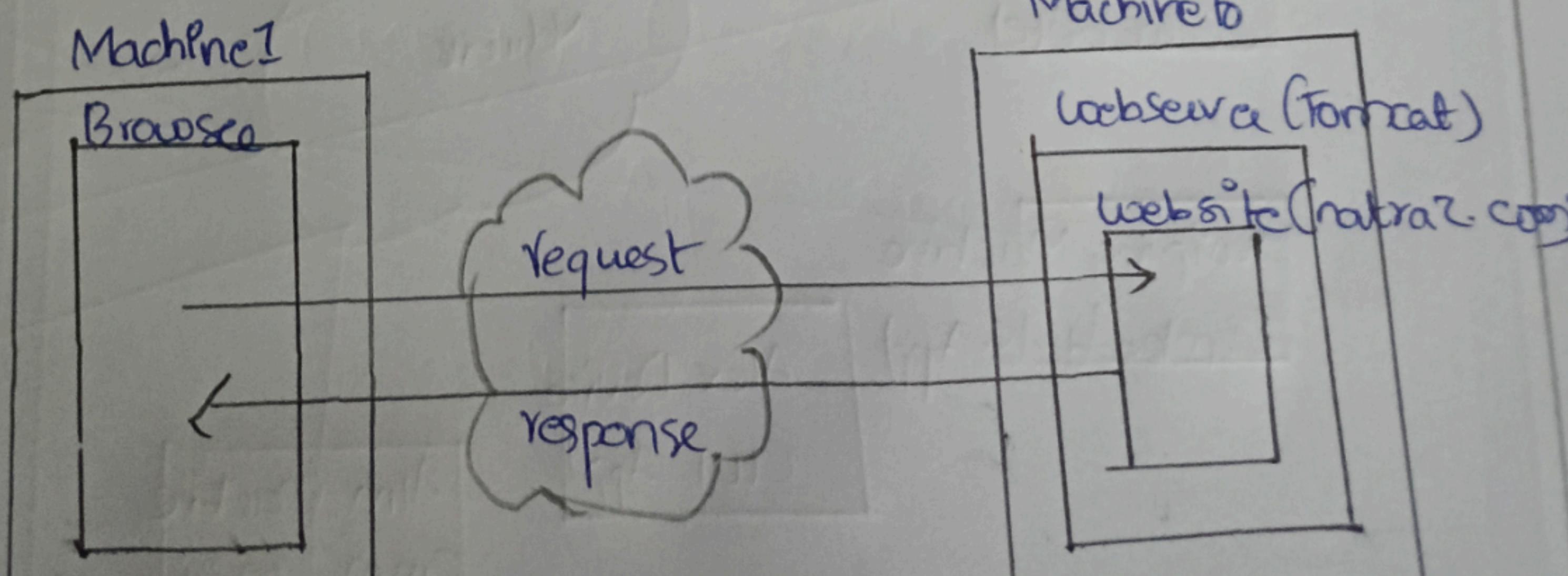
where client is browser and server is software (website)

running the automated environment

like tomcat. here the client server will be interaction

Using request-response model.

Eg: nareshit.com, flipkart.com



⇒ The Application whose Services can be consumed either locally or remotely from diff. types of other client apps is called Distributed APP--. Distributed APPs are B2B applications having Application to Application Interaction.

Eg:: ICC Core Comp, BSE Comp, Weather Report Comp, PhonePe Server App and etc-

ICC :: International Cricket Council

BSE :: Bombay Stock Exchange.

