http://localhost:8080/SpringWebDemos/login

Dispatcher Servlet

Controller

RequestMapping

Model

MonoLithic App

View

.jsp

.html

Spring

1. Dependency Management –
2. Spring Configuration for startup—xml
3. Web Container like Tomcat management.
4. Not Production Ready
   1. War file
   2. Upload to tomcat webapps

SpringBoot – BootStrapper for Spring app

Book

SOA -ServiceOrientedArchitecture

Book-producer

Books

Advice

RestController

Application

BookDto

BookServiceI

BookService

ORM – Object Relation Mapping

Hibernate

JPA -Java Persistence API JPA Repository

RDBMS/NOSQL

Domain Driven Design

http://localhost:8081/bookstore/api/books

BookRepo

angular

ios

android

ACID -- Transaction

DB

UserService

NS

US 8080



NS 8091

NOSQL – MOngoDB

Around

BeforeAspect

No Structure

JSON Data (“1”,”sam”} {“2”,”john”,[john@email.com](mailto:john@email.com)}

around

getName

Aspect

Service

getXX()

setXX()

AfterAspect

Client

Get()

Get1()

Abc()

Xyz()

Spring Boot Security

User1,user123,USER – In memory – launched

When launched and hit any endpoint

1. Open the login form
2. Enter user1, user123
3. Authenticate in AuthenticationManager
4. Credentials are set to Principal Object (UserNamePasswordToken)
5. loadUserByUserName(String username) in UserDetailsService
6. Retrieves the User Object – if username is right else null
7. Password encoder – read password from the User Object (existing credential)
8. Checks with the password in the Principal Object

UserDetailsService

User Object

Authenticates

Principal Object

Login Form

Reads the rule that is set & validates

Reads the role from User Object

MyUser

UserRepo

MyUserDetailsService

Login Form

Principal Object

User Object

Authenticates

DB Authentication

1. Create credentials

JWT Authentication

1. Authenticate the user
2. Generate the token – JwtUtil
3. Validate the token