Spring Boot

* It is loosely coupled
* It is Embedded server(tomcat)
* Its mostly uses HAS-A relationship
* Bean: Spring related Object
* IOC container (spring core) 🡺 Inversion of control

1. Creates a bean 🡺 configures that bean 🡺 it manages its usage till its end.
2. We can read data directly config files, property files.

* Dependency Injection:

Without creating an object we can inject data of one class into other. (IOC container).

This achieved using

1. setter Injection.
2. Constructor injection

A 🡺 B 🡺 C 🡺 D

Object new keyword

* Bean Factory: Multiple bean objects.
* By default bean is a singleton.
* @Bean
* annotation to create a bean

Bean Factory: it is like interface

* Controller layer 🡺 all urls/URI related code need to be write
* Service layer 🡺 All business logic
* DAO/Repository layer 🡺 database layer

Dispatcher servlet: this will route to our controller layer.

@Controller

@RequestBody

From httprequest (json body) 🡺 java object

@ResponseBody 🡺 it will converts Java to json and sends httpResponse.

@RestController 🡺 @Controller+@ResponseBody

@RequestMapping(value = “/student”) 🡺 urls related data we will be writing

/student/1

Method 1 🡺 based on id(1)

Method 2 🡺 based on name(“ABC”)

Inbuilt

JPA queries

JPQL

SQL

@Qaulifier(“service2”)

@Autowired

StudentService studentService;

Abstract methods:

Impl class

StudentServiceimpl implements StudentService

A picture containing text, sky, screenshot

Description automatically generated

<http://localhost:8080/student/ABC>

http://localhost:8080/student?studentName=ABC

@PostMapping🡺 save

@GetMapping 🡺 fetching records

@PutMapping 🡺 update

@DeleteMapping 🡺 delete

@PathVariable 🡺 directly reading data from URL

@RequestParam 🡺 data will be like key value pair

@Component 🡺 class level

@Bean 🡺 method level

@Service 🡺 service layer

@Repository 🡺 repository layer

@Transactional 🡺 transactional management

Table A🡺1

Table B🡺 2

@Qualifier 🡺 if there 2 bean same type

@Primary 🡺 to accept multiple same bean type

@Autowired 🡺

@Configuration 🡺 java based configuration

@EnableAutoconfiguration 🡺 to enable autoconfiguration

@SpringbootConfiguration 🡺

@Componentscan 🡺 to scan the components

@SpringbootApplication 🡺 main method

@Value 🡺 read data from property file

@ControllerAdvice + @ResponseBody

@ExceptionHandler

@RestControllerAdvice

Springboot is a autoconfigurer

Springboot has its own internal emebedded tomcat.

Sprinboot has its own maven build tool also

**Actuators 🡺 monitoring and managing like a production**