Spring Boot:

* Auto-configuration
* Embedded tomcat
* Dependency management
* Starter dependency-pom

IOC container:

1. It will create a bean and configure it and, manages it till it end.
2. It can read data directly from property files, external files.

Dependency Injection:

1. Setter Injection
2. Constructor Injection

A 🡺 B 🡺 C 🡺 D 🡺 tightly coupled 🡺 it can be converted into loosely coupled by using dependency Injection

Controller layer 🡺 All URLS related code

Service Layer 🡺 Business logic

Repository/DAO layer 🡺 Database related queries

Entity/Model classes 🡺 POJO class 🡺 Database table

Exception handling 🡺

Student

Private int Id

Private String Name

Private String College

Private Address address

Address

Id

Plotno

City

State

A picture containing text, sky, screenshot

Description automatically generated

@Controller 🡺 URls

@ResposeBody 🡺 java object to Json and send data to HttpResponse

@RestController 🡺 @Controller + @ ResponseBody

@RequestBody 🡺HttpRequest Json data to java objects

@RequestMapping(value=”/api/student”) 🡺 URLs mappping

@PostMapping(“/save”) 🡺 saving data

@GetMapping 🡺 fetching

@PutMapping 🡺 updating

@DeleteMapping 🡺 deleting data

@Service(“a1”) 🡺 service layer

@Repository 🡺 Repository

@Transactional 🡺

@Component 🡺

@Configuration 🡺

@Bean 🡺 method level

@Autowired

@Qualifier(“a1”)

@Primary

@ExceptionHandler

@controllerAdvice

@RestControllerAdvice

@SpringBootApplication

@EnableAutoConfiguration

@SpringBootConfiguration

@ComponentScan

ResponseEntity:

Lombok

@Setter

@Getter

@Data

Student: id,name,college,mobileNumber

200 ok status success.

Message.

StudentResponse