* It is lightweighted application
* It is lossely coupled application
* It can easily integrates with other frmeworks.
* Mostly it will work on HAS-A Relationship.
* IOC 🡺 Inversion of control: It will creates the bean, it will configure beans and manages bean till its usage
* It will help to read data directly from configuration files or properties file.
* Dependecy Injection

@Autowired

A 🡺 B 🡺 C 🡺 D

* Bean: Bean Is an Object 🡺 by default it is singleton object 🡺 it is used on bean level 🡺 @Bean
* Components 🡺 it is also bean but that make entire class as a single bean 🡺 @Component

Springboot:

1. AutoConfiguration 🡺 @EnableAutoConfiguration
2. ComponenScan 🡺 @ComponentScan
3. @SpringBootConfiguration
4. @SpringBootApplication 🡺 @EnableAutoConfiguration, @ComponentScan, @SpringBootConfiguration

Student student = new Student();

@Autowired

Student studnt;

**Student**

Integer Id

String Name

String CollegeName

Address address

**Address**

Id

City

State

ControllerClass

ServiceClass cls = new ServiceClass();

Cls.methodName();

@Autowired

ServiceClass cls;

Cls.methodName();

Anotations:

@Controller

@ResponseBody 🡺 java 🡺 json 🡺 HttpResponse

@RequestMapping(“/student/api”)

@Restcontroller

@RequestBody 🡺 json request 🡺 java object

@Service(“abc”)

@Transactional

@Repository

@Autowired == injecting data

@Qualifier(“abc”) 🡺

@Bean 🡺 method level

@Component🡺 clss lvl

@Configuration

@Value

@PathVariable 🡺 read data directly from URI/URL ({studentId}) 🡺 http://localhost:8080/student/1

@Requestparam 🡺 read data from Url 🡺 http://localhost:8080/student?studentId=1

@PostMapping 🡺 save data

@GetMapping 🡺 fetch operation

@PutMapping 🡺 update data 🡺 entire object

@DeleteMapping 🡺 delete data

@Patch 🡺 partial update

@EnableAutoConfiguration

@ComponentScan

@SpringbootConfiguration

@SprigbootApplication

@ControllerAdvice+@ResponseBody

@RestControllerAdvice

@ExceptionHandler

@ResponseStatus

Diagram

Description automatically generated