* Spring is light weighted application.
* Spring is like a loosely coupled application.
* Spring works mostly on HAS-A relationship.
* Spring IOC (inversion of control):

1. This will creates the beans, it will configure them and manages it its usage.
2. It will helps to read data from configuration file, property files.

* Dependency injection:

1. If one object is dependent on other object, this dependency will be removed and will make it as loosely coupling.
2. Using Setter injection and constructor injection.

* Bean: it is an object in spring framework. By default bean is singleton object.
* Bean Factory: group of beans and its configuration is bean factory.
* Dependencies we will be adding in our pom.xml- file.
* Spring boot is auto configure.
* Spring boot has in built embedded tomcat server and maven.
* Spring boot has actuators🡺 Monitoring.
* Controller layer 🡺 All our urls related code will be here.
* Service layer 🡺 All our business logic
* Repository/DAO(Data access object) layer 🡺 Data base related 🡺 Hibernate JPA

JPA 🡺 it is a specification 🡺 Interface

Hibernate is a implementation framework for JPA (Java persistence API).

ORM Tool 🡺 Object relational mapping.

Insert into…

Save(object)

Diagram

Description automatically generated

@EnableAutoConfiguration

@SpringbootConfiguration

@ComponentScan

@SpringBootApplication

@Controller

@ResponseBody 🡺 java to json repose 🡺 Httpreponse

@RequestBody 🡺 json object to java convert

@RestController 🡺 @Controller + @ReponseBody

@RequestMapping(“/student”) 🡺 URl mapping

@Service 🡺 business logic

@Repository 🡺 Database related

@PostMapping 🡺 save

@GetMapping 🡺Fetching

@PutMapping 🡺 Update

@DeleteMapping 🡺 delete

@PathVariable🡺 Reading data directly from URL 🡺 http://local:8080/student/Sai

@RequestParam 🡺 Reading data as key value pair 🡺 http://local:8080/student?name=Sai

@Transactional 🡺 Transactional management

@Autowired 🡺

@Qualifier 🡺to differentiate same bean type

@Bean 🡺 Object , method level

@Component🡺 class level

@Configuration 🡺

@ControllerAdvice + @ResponseBody

@RestControllerAdvice

@ExceptionalHandler