Back End

Java Basics <https://www.journaldev.com/546/difference-jdk-vs-jre-vs-jvm>

* *JVM*
* *JDK, JRE*
* *Exception handling*
  + throw
  + throws
  + try-catch
  + finally
  + resource
  + key take away
    - use specific exceptions
    - throw fast
    - catch late
    - logging
    - closing resources
    - custom exception
    - single catch block for many exceptions
* Thread
* *Scanner*
  + *Scanner class is used to receive inputs and parse them into int, double or String (default)*
  + *It’s utility class used to parse regular expression by generating tokens*
  + *It is used to read CSV file also*
  + *But it’s bit costly in terms of performance*
* Serialization
* Generics
* Collections
* Synchronization
* GC
* Memory Management
* Design patterns
* JAR, WAR
* JDBC <https://www.journaldev.com/2681/jdbc-tutorial>

J2EE

* Servlet
* Http
* Container
* Web.xml (Deployment descriptor)

Spring MVC

* Spring framework (IOC, DI)
* Application.xml
* Dispatcher servlet
* Service
* Component
* Repository
* AOP
* Spring Jars / Dependency

Spring Boot

* RESTful API
* Application Config
* Swagger
* Deployment

Hibernate

* DB Config with hibernate
* JPA
* DAO layer creation
* HQL
* Caching mechanism
* Hibernate JARs/Dependency

ORACLE

* DDL, DML, TCL, DCL
* Constraints
* Joins
* Procedures
* Triggers
* Queues
* Sequences

Tomcat

* Deployment workflow
* Context.xml
* Server.xml
* Libraries
* Properties file config / usage
* JVM in tomcat

APIGEE

* APIGEE architecture
* Authentication techniques
* Producer side changes
* Consumer side changes
* OAS

DB

CREATE TABLE `javatutorials`.`student` (

`id` INT NOT NULL,

`name` VARCHAR(45) NULL,

`department` INT NOT NULL,

`rank` INT NULL,

PRIMARY KEY (`id`));

mvn clean install -P autoInstallPackage -Dmaven.wagon.http.ssl.insecure=true -Dmaven.wagon.http.ssl.allowall=true

<https://www.bezkoder.com/spring-boot-jpa-crud-rest-api/>

create table products (

id bigint not null,

active bit,

created\_date datetime,

description varchar(255),

image\_url varchar(255),

name varchar(255),

sku varchar(255),

updated\_date datetime,

primary key (id)

) engine=MyISAM

<https://www.google.com/search?q=spring+boot+tomcat+mysql+jpa+example&rlz=1C1CHZN_enIN1012IN1012&sxsrf=ALiCzsbW1gC5Rzt3OCJU2K5HCoN0wuDuMw%3A1658067522532&ei=QhrUYqODIM6C4-EPvb-9yA8&ved=0ahUKEwijrMbtjoD5AhVOwTgGHb1fD_kQ4dUDCA4&uact=5&oq=spring+boot+tomcat+mysql+jpa+example&gs_lcp=Cgdnd3Mtd2l6EAMyBAgjECc6BwgAEEcQsANKBAhBGABKBAhGGABQygZY6Qdg6hBoAnABeACAAeoBiAGbA5IBBTAuMS4xmAEAoAEByAEIwAEB&sclient=gws-wiz#kpvalbx=_XRrUYsf2Opfu4-EP_uWrmA017>

postman token

postman://auth/callback?code=4faedc6bec151070c2cb0350eae3514eaa4b7529bfa26196e2878d328efc0a68

{

"sku": "Abc",

"name":"Laptop",

"description":"Laptop Products For Sale",

"active":true

}