We can create spring boot application using:

1. Eclipes Maven

2. start.spring.io(we can use this zipped file both eclipse and STS)

3. Eclipes STS

**ConfigurableApplicationContext, ApplicationContext** is a IOC container that manage the life cycle of bean.

**Point Remeber:**

1. Write All Notations

2 .Main class package(Parent Package) contain all sub packages(important)

3. Identify the starting point of application

4. application.properties file loaded first(then optionally .sql file loaded if and jsp file if)

5. if @Table and @column is not specified then by default data memberName taken(@Entity required)

6. When performing CRUD Operation we can extends following interfaces

a. repository(custom repository)

b. crudRepository(recommended)

c. PagingAndSortingRepository

d.jpaRepository(ties so not recommended)

**Question: My database supports auto-incremented columns. How can I use them as primary keys in Hibernate?**

Ans: JPA and Hibernate support different strategies to generate primary key values. One of them is the **identity strategy** which uses an auto-incremented database column.  
If you want to use this strategy, you have to annotate the primary key attribute @Id and with the @GeneratedValue annotation and set the strategy attribute **to GenerationType.IDENTITY**

Basically IDENTITY use for database that support auto-increment at column level like MySQL,SQLServer, **Postgresetc.**

Auto-increment is no such concept in Oracle, yet there is in MySQL, SQLServer and a few others. Oracle would allow "SEQUENCE", or "TABLE" strategies to be used however

**So for Oracle use GenerationType.SEQUENCE.**

**Question:**

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

@Column(name = "MY\_TABLE\_ID")

private Integer myTableId;

For the first POST after my application starts, I create MyEntity everything works fine, MY\_TABLE\_ID starts with 1 and works as expected.

My issue is, If somebody inserts data manually before I do my POST then I get duplicate key exception as myTableId is entered as 1 which is already present.

Ans:

I had few limitations,

1. I couldn't create database sequence as my database changes were freezed.
2. I didn't want to introduce new Custom IdGenerator class because it would add confusion to other people working with me.

@Id

@GeneratedValue(strategy = GenerationType.AUTO, generator="seq")

@GenericGenerator(name = "seq", **strategy="increment")**

@Column(name = "MY\_TABLE\_ID")

private Integer myTableId;

**increment**

An IdentifierGenerator that returns a long, constructed by counting from the maximum primary key value at startup. Not safe for use in a cluster!

Since, it was incrementing already existing myTableId even if it was *manually* inserted, this resolved my issue.

Simply, @Id: This annotation specifies the primary key of the entity.

@GeneratedValue: This annotation is used to specify the primary key generation strategy to use. i.e Instructs database to generate a value for this field automatically. If the strategy is not specified by default AUTO will be used.

GenerationType enum defines four strategies:

1. Generation Type . TABLE,

2. Generation Type. SEQUENCE(Oracle)

3. Generation Type. IDENTITY(MySQL)

4. Generation Type. AUTO

***AUTO*:** Hibernate selects the generation strategy based on the used dialect.

It is not recommended. Because when we use table relationship then it not maintained order of key between tables.

***IDENTITY*:** Hibernate relies on an auto-incremented database column to generate the primary key(MySQL support Auto Increment column)

***SEQUENCE***: Hibernate requests the primary key value from a database sequence(Oracle not support auto increment).

***TABLE***: Hibernate uses a database table to simulate a sequence.

Practically Difference Between IDENTITY,SEQUENCE and AUTO

<https://ngdeveloper.com/generationtype-identity-vs-generationtype-sequence-vs-generationtype-auto/>

Basic Json when create operation perform using postman

POST: http://localhost:8080/api/tickets/create

{

"passengerName":"Akash",

"bookingDate":"2000-12-12",

"sourceName":"Akola",

"destinationName":"Pune"

,"emailId":"akash@gmail.com"

}