**API End Points and Communication**

**Day 3: 28 Feb. 28**

**Spring JDBC :** Spring JDBC provided one of the pre defined API

Ie JdbcTemplate :JdbcTemplate wrap core JDBC and provided more functionality to improve JDBC code in DAO layer.

In place of DataSource DI we need to do DI for JdbcTemplate

In beans.xml file

<bean class="org.springframework.jdbc.datasource.DriverManagerDataSource" id="ds" scope="singleton">

<property name="driverClassName" value="com.mysql.cj.jdbc.Driver"></property>

<property name="url" value="jdbc:mysql://localhost:3306/mvc"></property>

<property name="username" value="root"></property>

<property name="password" value="root@123"></property>

</bean>

JDBC Template

<bean class="org.springframework.jdbc.core.JdbcTemplate" id="jdbcTemplate" scope="singleton">

<property name="dataSource" ref="ds"></property>

</bean>

**JdbcTemplate methods**

1. update() : this method takes 2 parameter,

1st parameter is query and 2nd parameter is var argument means multiple value for those ?.

This update method mainly uses for DML Operation.

1. queryForList() : this method take query and return type of this method is List of Map in Map key must be string and value must object type.

This method internally convert each records as one map object.

1. query method: this method takes 2 parameter

1st parameter query

2nd parameter RowMapper.

RowMapper is a type of Functional interface which contains one abstract method which help to map each record to employee or user defined objects.