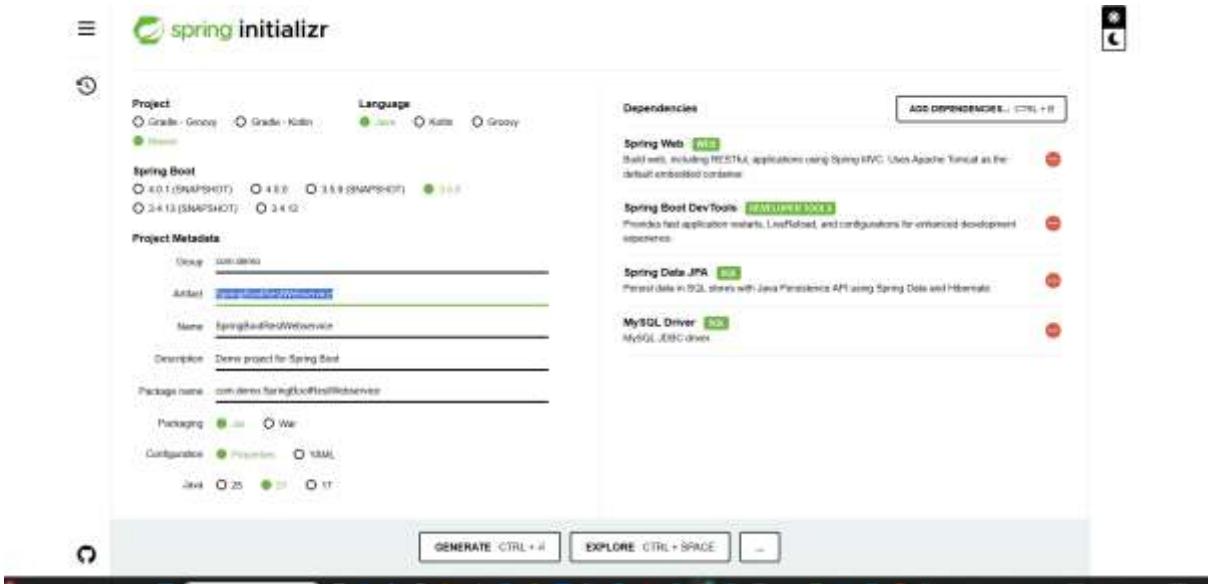


1. Reuse code from SpringWebservice, and conver it to SPringBootWebService
2. Go to starter.spring.io and select the following



3. In resource folder open application.properties file and paste following properties

```

spring.application.name=SpringBootRestWebservice
server.port=8181

#to add viewresolver
#spring.mvc.view.prefix=/WEB-INF/views/
#spring.mvc.view.suffix=.jsp

#spring datasource
spring.datasource.url = jdbc:mysql://localhost:3306/spring?useSSL=false
#&serverTimezone=UTC&useLegacyDatetimeCode=false&zeroDateTimeBehavior=CONVERT_TO_NULL
spring.datasource.username=root
spring.datasource.password=root123
#spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

#hibernate properties
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true

```

4. Create controller package as a subpackage where main function exists
5. Add ProductController class
6. Add Dao package and add ProductDao interface write following code

```

@Repository
public interface ProductDao extends JpaRepository<Product, Integer>{

```

```

        @Query(value="select * from product where price between :lprice and
:lprice",nativeQuery = true)
        //@Query(value="select p from Product p where price between :lpr
and :hpr")
        List<Product> findByPrice(double lprice, double hprice);

        @Procedure(procedureName = "getCount")
        public int getCount();

}

```

7. Add service layer and write ProductService interface and ProductServiceImpl class, add following code in ProductServiceImpl

```

package com.demo.SpringBootRestWebservice.service;

import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.demo.SpringBootRestWebservice.beans.Product;
import com.demo.SpringBootRestWebservice.dao.ProductDao;
import com.demo.SpringBootRestWebservice.dto.ProductDto;
import com.demo.SpringBootRestWebservice.mapper.ProductDtoMapper;

@Service
public class ProductServiceImpl implements ProductService{
    @Autowired
    ProductDao pdao;

    public List<ProductDto> getAllProducts() {
        List<Product> plist= pdao.findAll();
        List<ProductDto> plist1=plist.stream()
            .map(prod-
>ProductDtoMapper.mapToProductDto(prod))
            .collect(Collectors.toList());
        return plist1;
    }

    public boolean addproduct(ProductDto p) {
        //convert ProductDto to Product
        Product prod=ProductDtoMapper.mapToProduct(p);
        Product p1=pdao.save(prod);
        return p1!=null;
    }
}

```

```

    }

    public ProductDto getById(int pid) {
        Optional<Product> p= pdao.findById(pid);
        if(p.isPresent()) {
            ProductDto
            prod=ProductDtoMapper.mapToProductDto(p.get());
            return prod;
        }

        return null;
    }

    public boolean updateproduct(ProductDto p){
        // TODO Auto-generated method stub
        //convert ProductDto to Product
        Product prod=ProductDtoMapper.mapToProduct(p);
        Optional<Product> op=pdao.findById(prod.getPid());
        if(op.isPresent()) {
            Product p2=op.get();
            p2.setPname(p.getPname());
            p2.setQty(p.getQty());
            p2.setPrice(p.getPrice());
            pdao.save(p2);
            return true;
        }
        return false;
    }

    public boolean deleteById(int pid) {
        Optional<Product> op=pdao.findById(pid);
        if(op.isPresent()) {
            pdao.delete(op.get());
            return true;
        }
        return false;
    }

    @Override
    public List<ProductDto> getByprice(double lprice, double hprice) {
        List<Product> plist=pdao.findByPrice(lprice,hprice);
        if(plist.size()>0) {
            List<ProductDto> plist1=plist.stream()
                .map(prod-
                    >ProductDtoMapper.mapToProductDto(prod))
                .collect(Collectors.toList());
        }
    }
}

```

```
        return plist1;
    }
    return null;
}

}
```

8. Add ProductDto class , ProductDetoMapper class
9. Add Controller ProductController
10. Add Product class