Spring Boot API’S

I18 Internalization Rest API:

Response:

<http://localhost:9898/api/v1/labels/english>

{

    "address": "Address",

    "State": "State",

    "name": "Name",

    "Country": "Country"

}

<http://localhost:9898/api/v1/labels/hindi>

{

    "address": "पता",

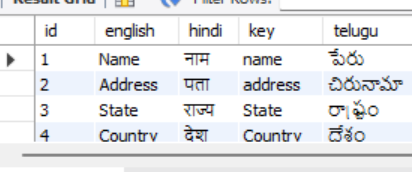
    "State": "राज्य",

    "name": "नाम",

    "Country": "देश"

}

Data Base:



Entity:(Projo Class):

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

**import** lombok.Data;

@Entity

@Data

@Table

**public** **class** Language {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** Long id;

@Column(name = "KEY")

**private** String languageKey;

@Column(name = "ENGLISH")

**private** String english;

@Column(name = "TELUGU")

**private** String telugu;

@Column(name = "HINDI")

**private** String hindi;

@Column(name = "SPANISH")

**private** String spanish;

@Column(name = "GERMAN")

**private** String german;

@Column(name = "CHINESE\_MANDARIN")

**private** String chineseMandarin;

@Column(name = "FRENCH")

**private** String french;

@Column(name = "ITALIAN")

**private** String italian;

@Column(name = "POLISH")

**private** String polish;

@Column(name = "PORTUGESE")

**private** String portugese;

@Column(name = "BRAZILIAN\_PORTUGESE")

**private** String brazilianPortugese;

@Column(name = "LATAM\_SPANISH")

**private** String latamSpanish;

@Column(name = "SWEDISH")

**private** String swedish;

@Column(name = "TAMIL")

**private** String tamil;

@Column(name = "FRENCH\_CANADIAN")

**private** String frenchCanadian;

@Column(name = "TURKISH")

**private** String turkish;

@Column(name = "DUTCH")

**private** String dutch;

}

Dto:

**import** lombok.Data;

@Data

**public** **class** LanguageDto {

**private** Long id;

**private** String languageKey;

**private** String english;

**private** String telugu;

**private** String spanish;

**private** String german;

**private** String chineseMandarin;

**private** String french;

**private** String hindi;

**private** String italian;

**private** String polish;

**private** String portugese;

**private** String brazilianPortugese;

**private** String latamSpanish;

**private** String swedish;

**private** String tamil;

**private** String frenchCanadian;

**private** String turkish;

**private** String dutch;

}

Repository:

import org.springframework.data.jpa.repository.JpaRepository;

import com.example.entity.Language;

public interface LangugaeRepository extends JpaRepository<Language, Long>{}

Service:

**import** java.util.Map;

**public** **interface** LanguageService {

**public** Map<String, String> getChangeLanguage(String language);

}

Service Impl:

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** org.modelmapper.ModelMapper;

**import** org.modelmapper.TypeToken;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.example.dto.LanguageDto;

**import** com.example.entity.Language;

**import** com.example.repository.LangugaeRepository;

**import** com.example.service.LanguageService;

@Service

**public** **class** LanguageServiceImpl **implements** LanguageService {

@Autowired

**private** LangugaeRepository langugaeRepository;

**private** ModelMapper modelMapper = **new** ModelMapper();

@Override

**public** Map<String, String> getChangeLanguage(String language) {

List<Language> allLabels = langugaeRepository.findAll();

List<LanguageDto> allLabelsDTO = modelMapper.map(allLabels, **new** TypeToken<List<LanguageDto>>() {

}.getType());

Map<String, String> labelsPair = **new** HashMap<>();

**for** (LanguageDto labels : allLabelsDTO) {

String key = labels.getLanguageKey();

String value = "";

**switch** (language) {

**case** "english":

value = labels.getEnglish();

**break**;

**case** "telugu":

value = labels.getTelugu();

**break**;

**case** "hindi":

value = labels.getHindi();

**break**;

**case** "german":

value = labels.getGerman();

**break**;

**case** "chinesemandarin":

value = labels.getChineseMandarin();

**break**;

**case** "french":

value = labels.getFrench();

**break**;

**case** "italian":

value = labels.getItalian();

**break**;

**case** "spanish":

value = labels.getSpanish();

**break**;

**case** "polish":

value = labels.getPolish();

**break**;

**case** "portugese":

value = labels.getPortugese();

**break**;

**case** "brazilianportugese":

value = labels.getBrazilianPortugese();

**break**;

**case** "latamspanish":

value = labels.getLatamSpanish();

**break**;

**case** "swedish":

value = labels.getSwedish();

**break**;

**case** "tamil":

value = labels.getTamil();

**break**;

**case** "frenchcanadian":

value = labels.getFrenchCanadian();

**break**;

**case** "turkish":

value = labels.getTurkish();

**break**;

**case** "dutch":

value = labels.getDutch();

**break**;

**default**:

value = labels.getEnglish();

**break**;

}

labelsPair.put(key, value);

}

**return** labelsPair;

}

}

Controller :

**import** java.util.Map;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

**import** com.example.service.LanguageService;

@RestController

@RequestMapping("/api/v1")

**public** **class** LanguageController {

@Autowired

**private** LanguageService languageService;

@GetMapping("/language/{language}")

**public** ResponseEntity<Map<String, String>> getLabelsByLocale(@PathVariable("language") String language) {

**return** **new** ResponseEntity<>(languageService.getChangeLanguage(language), HttpStatus.***OK***);

}

}