**Preparation**

* Use [start.spring.io](http://start.spring.io/) to create a project, the dependencies used include Spring Data JPA and H2 database.
* Other configurations depending on your choice.

**Steps**

* Create 4 packages, namely: entities, repositories, services, and controllers.
* Add this to application.properties

spring.h2.console.enabled=true

* Create a user entity inside an entities package with three fields: id, name, and email.
* package com.CRUD.demo.entities;
* import javax.persistence.\*;
* @Entity
* @Table(name = "user")
* public class UserEntity {
* @Id
* @GeneratedValue(strategy = GenerationType.AUTO)
* private int id;
* private String name;
* private String email;
* public UserEntity() {
* }
* public UserEntity(int id, String name, String email) {
* this.id = id;
* this.name = name;
* this.email = email;
* }
* public int getId() {
* return id;
* }
* public void setId(int id) {
* this.id = id;
* }
* public String getName() {
* return name;
* }
* public void setName(String name) {
* this.name = name;
* }
* public String getEmail() {
* return email;
* }
* public void setEmail(String email) {
* this.email = email;
* }
* }
* Create a user repository inside a repositories package that extends JpaRepository

package com.CRUD.demo.repositories;

import com.CRUD.demo.entities.User;

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

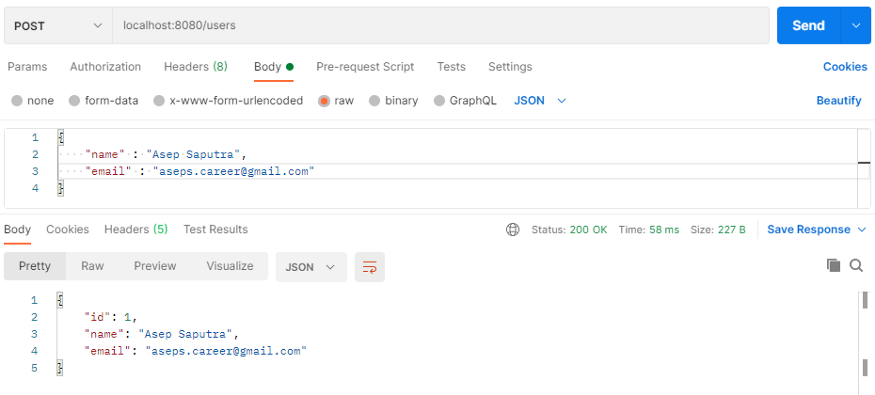
public interface UserRepository extends JpaRepository<User, Integer> {

}

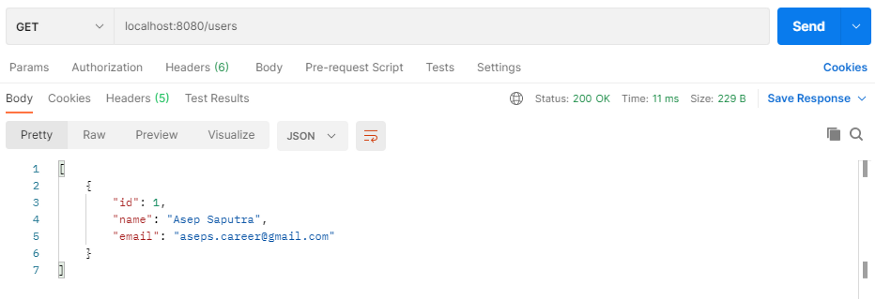
* Create a user service inside a services package
* package com.CRUD.demo.services;
* import com.CRUD.demo.entities.UserEntity;
* import org.springframework.stereotype.Component;
* import java.util.List;
* @Component
* public interface UserService {
* UserEntity Post(UserEntity params);
* List<UserEntity> Get();
* UserEntity Get(int id);
* UserEntity Update(UserEntity params, int id);
* String Delete(int id);
* }
* Create a user service implement inside a services package
* package com.CRUD.demo.services;
* import com.CRUD.demo.entities.UserEntity;
* import com.CRUD.demo.repositories.UserRepository;
* import org.springframework.stereotype.Service;
* import java.util.List;
* @Service
* public class UserServiceImpl implements UserService {
* private final UserRepository userRepository;
* public UserServiceImpl(UserRepository userRepository) {
* this.userRepository = userRepository;
* }
* @Override
* public UserEntity Post(UserEntity params) {
* userRepository.save(params);
* return params;
* }
* @Override
* public List<UserEntity> Get() {
* return userRepository.findAll();
* }
* @Override
* public UserEntity Get(int id) {
* return userRepository.findById(id).get();
* }
* @Override
* public UserEntity Update(UserEntity params, int id) {
* UserEntity user = userRepository.findById(id).get();
* user.setName(params.getName());
* user.setEmail(params.getEmail());
* return userRepository.save(user);
* }
* @Override
* public String Delete(int id) {
* userRepository.deleteById(id);
* return "User(" + id + ")" + " has been deleted!";
* }
* }
* Create a user controller inside a controllers package
* package com.CRUD.demo.controllers;
* import com.CRUD.demo.entities.UserEntity;
* import com.CRUD.demo.repositories.UserRepository;
* import com.CRUD.demo.services.UserService;
* import org.springframework.web.bind.annotation.\*;
* import java.util.List;
* @RestController
* public class UserController {
* private final UserService userService;
* public UserController(UserService userService, UserRepository userRepository) {
* this.userService = userService;
* }
* // Add
* @PostMapping(value = "/users")
* public UserEntity Post(@RequestBody UserEntity params) {
* return userService.Post(params);
* }
* // Get
* @GetMapping(value = "/users")
* public List<UserEntity> Get() {
* return userService.Get();
* }
* // Get By ID
* @GetMapping(value = "/users/{id}")
* public UserEntity Get(@PathVariable int id) {
* return userService.Get(id);
* }
* // Update
* @PutMapping(value = "/users/{id}")
* public UserEntity Update(@PathVariable int id, @RequestBody UserEntity params) {
* return userService.Update(params, id);
* }
* // Delete
* @DeleteMapping(value = "/users/{id}")
* public String Delete(@PathVariable int id) {
* return userService.Delete(id);
* }
* }

**Testing**

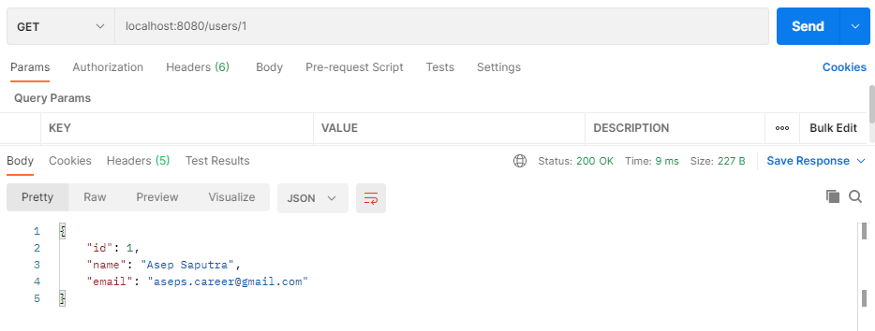
* POST



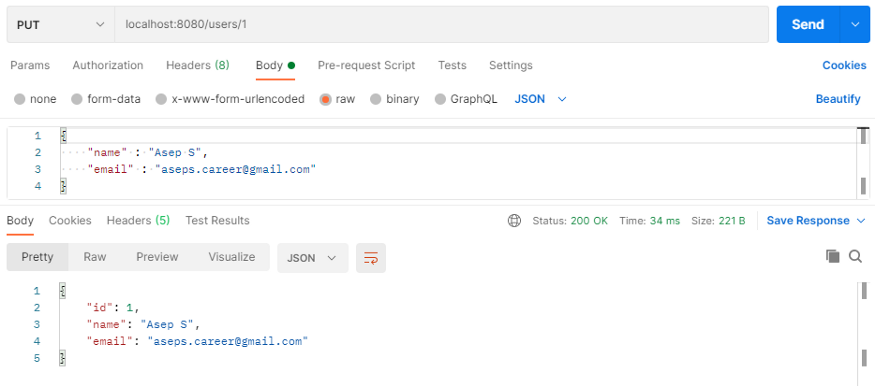
* GET



* GET (id)



* UPDATE



* DELETE

