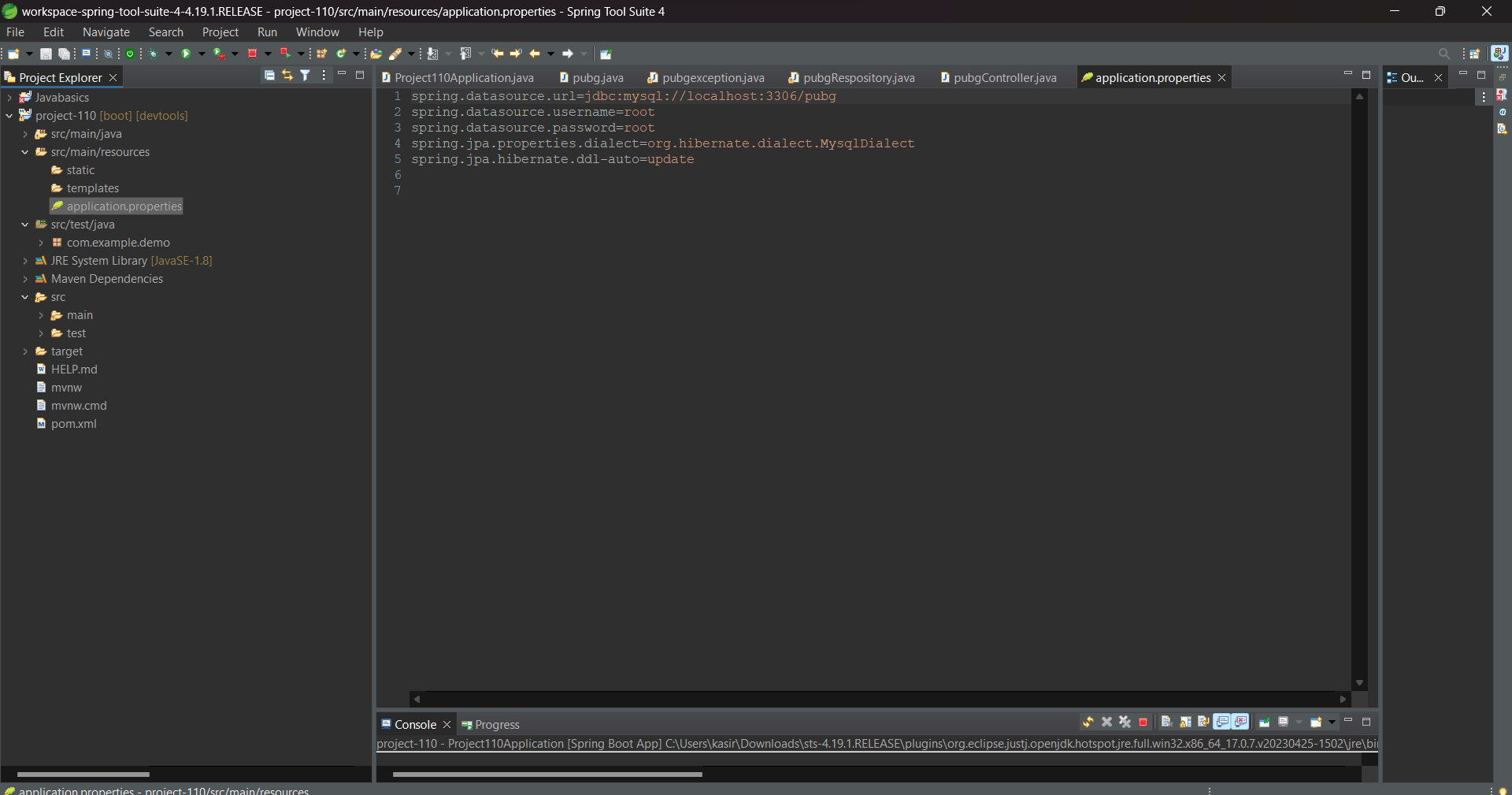
Spring Boot Code:



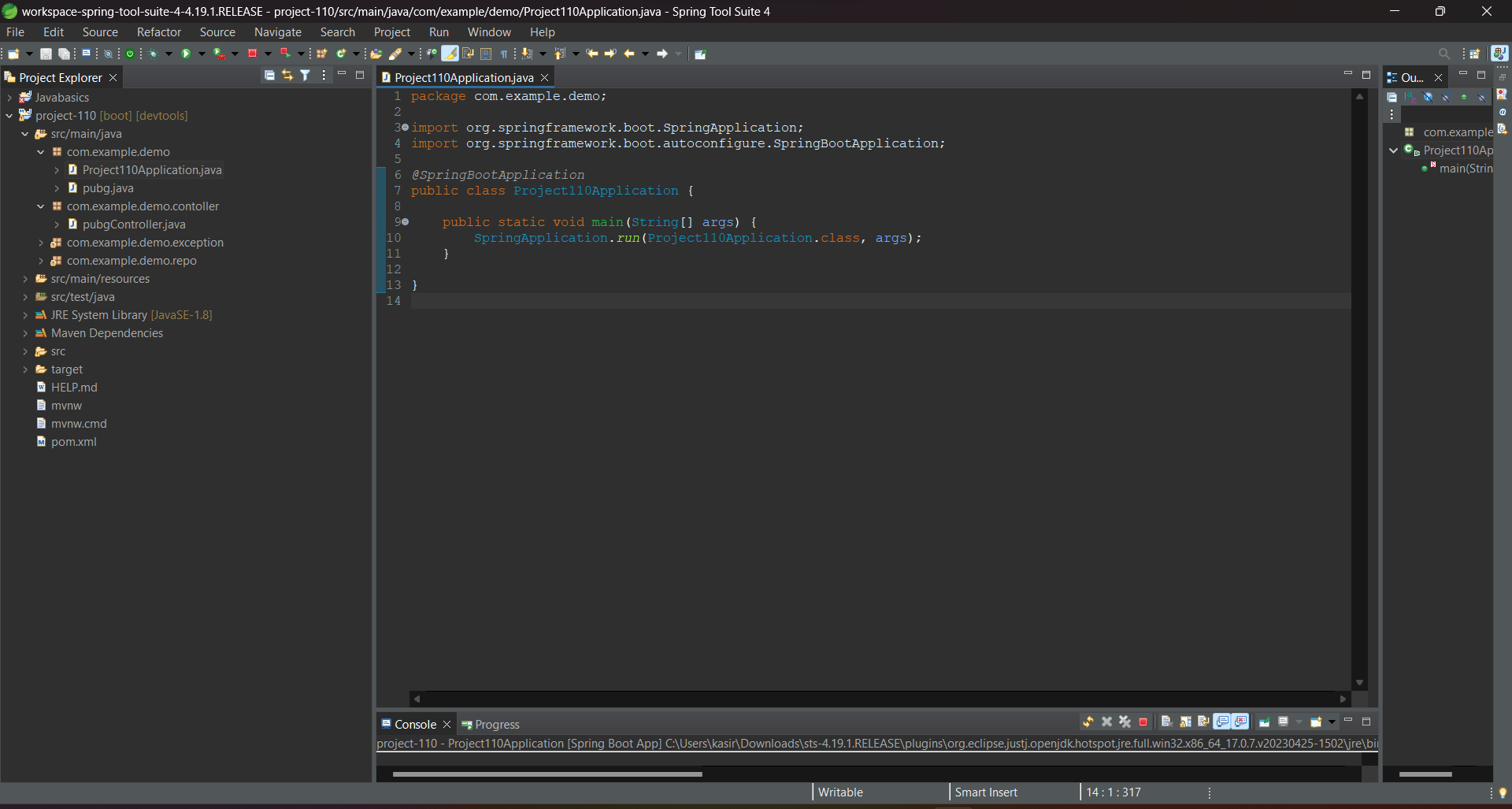
spring.datasource.url=jdbc:mysql://localhost:3306/pubg

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.properties.dialect=org.hibernate.dialect.MysqlDialect

spring.jpa.hibernate.ddl-auto=update



package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@*SpringBootApplication

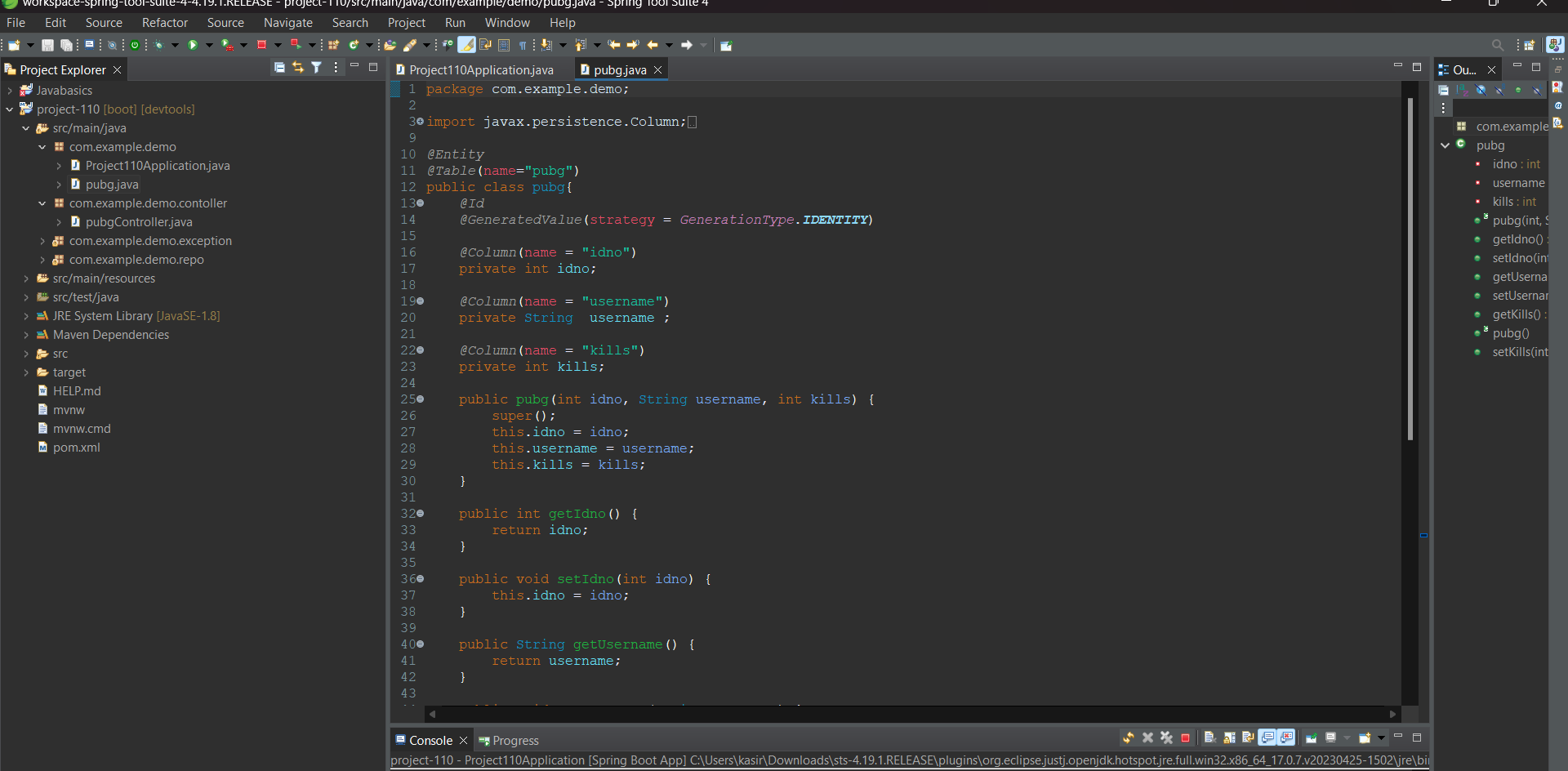
public class Project110Application {

public static void main(String[] args) {

SpringApplication.*run*(Project110Application.class, args);

}

}



package com.example.demo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name="pubg")

public class pubg{

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "idno")

private int idno;

@Column(name = "username")

private String username ;

@Column(name = "kills")

private int kills;

public pubg(int idno, String username, int kills) {

super();

this.idno = idno;

this.username = username;

this.kills = kills;

}

public int getIdno() {

return idno;

}

public void setIdno(int idno) {

this.idno = idno;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public int getKills() {

return kills;

}

public pubg() {

super();

// TODO Auto-generated constructor stub

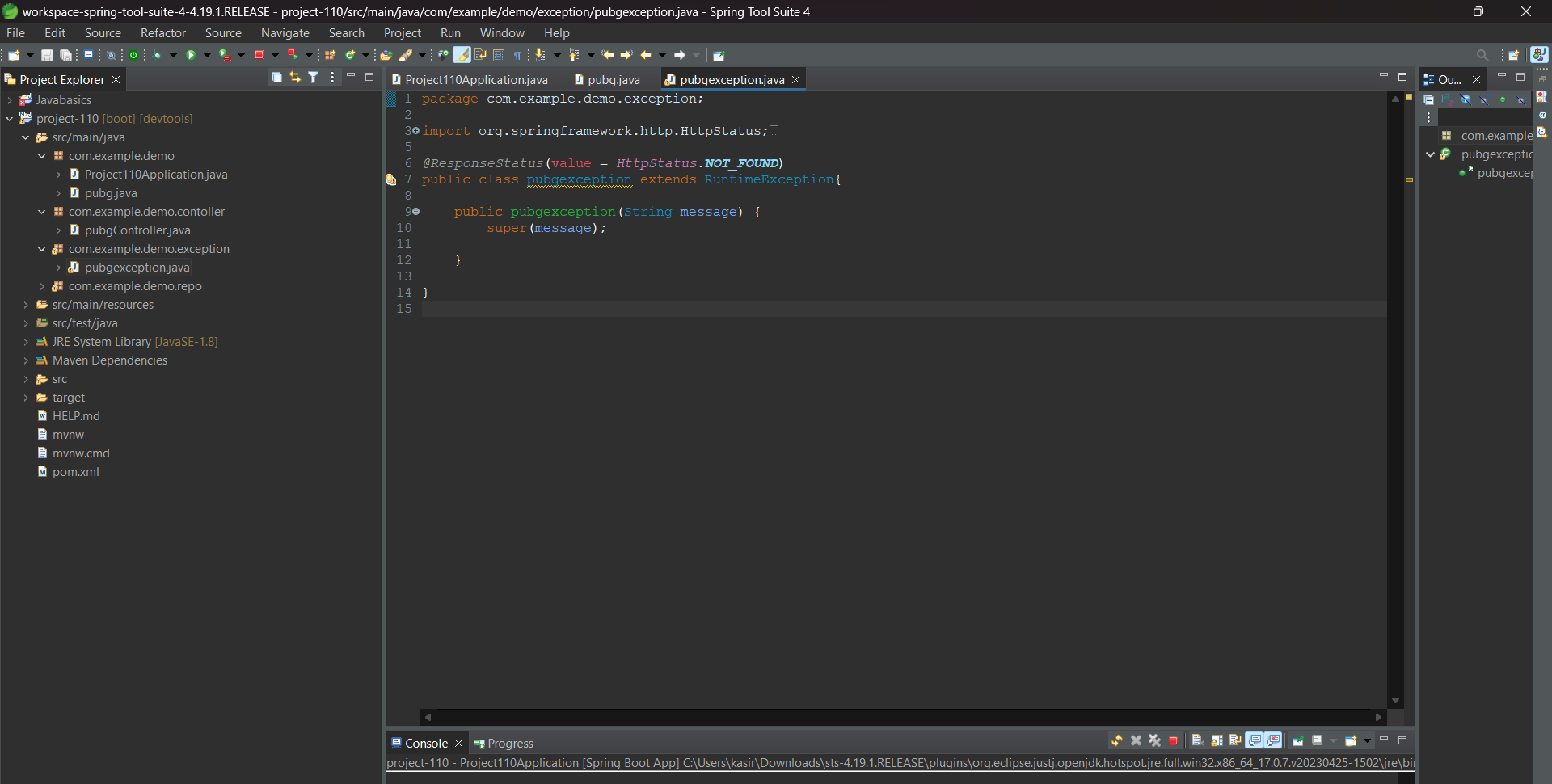
}

public void setKills(int kills) {

this.kills = kills;

}

}



package com.example.demo.exception;

import org.springframework.http.HttpStatus;

import org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(value = HttpStatus.NOT\_FOUND)

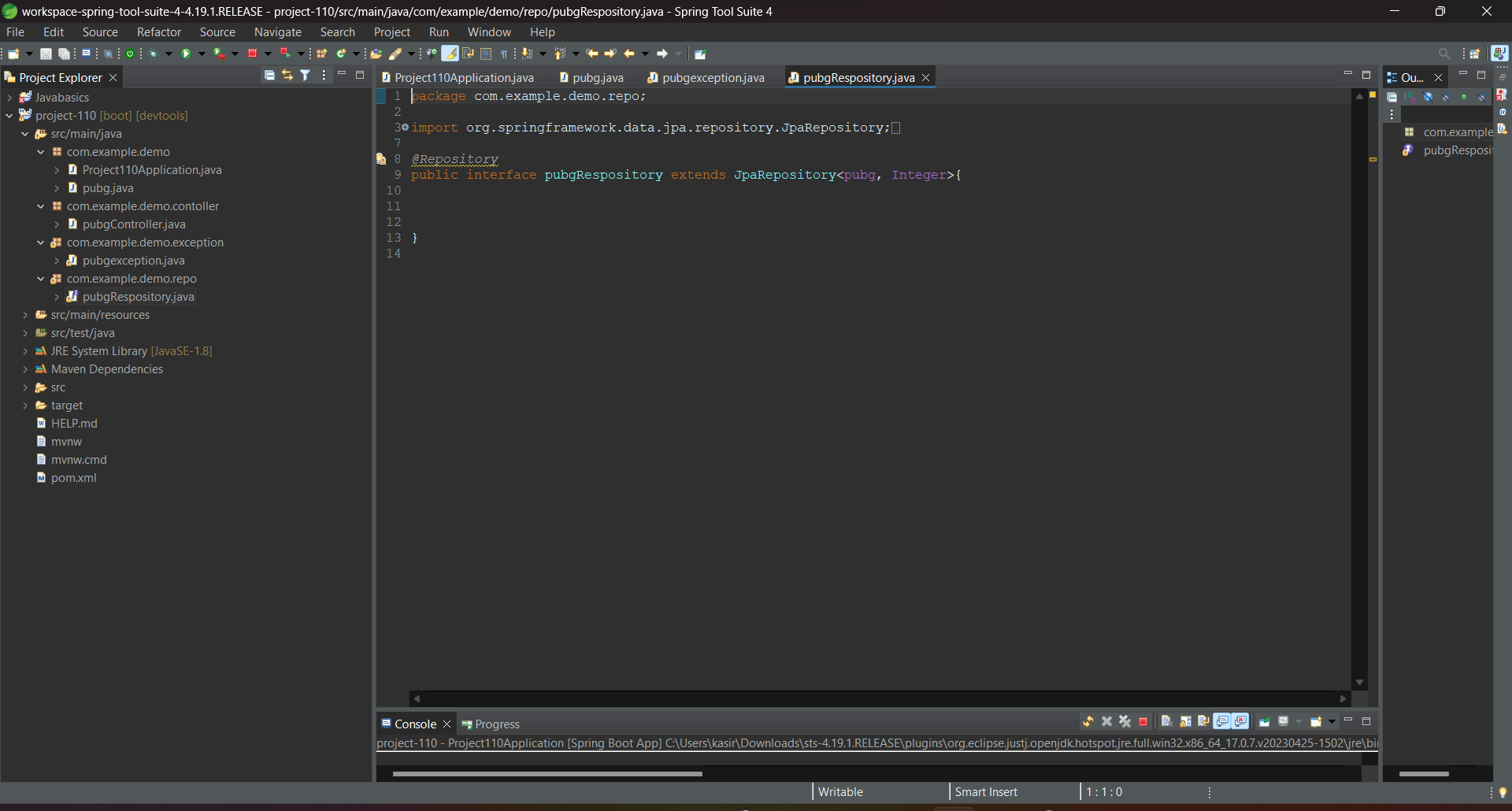
public class pubgexception extends RuntimeException{

public pubgexception(String message) {

super(message);

}

}



package com.example.demo.repo;

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

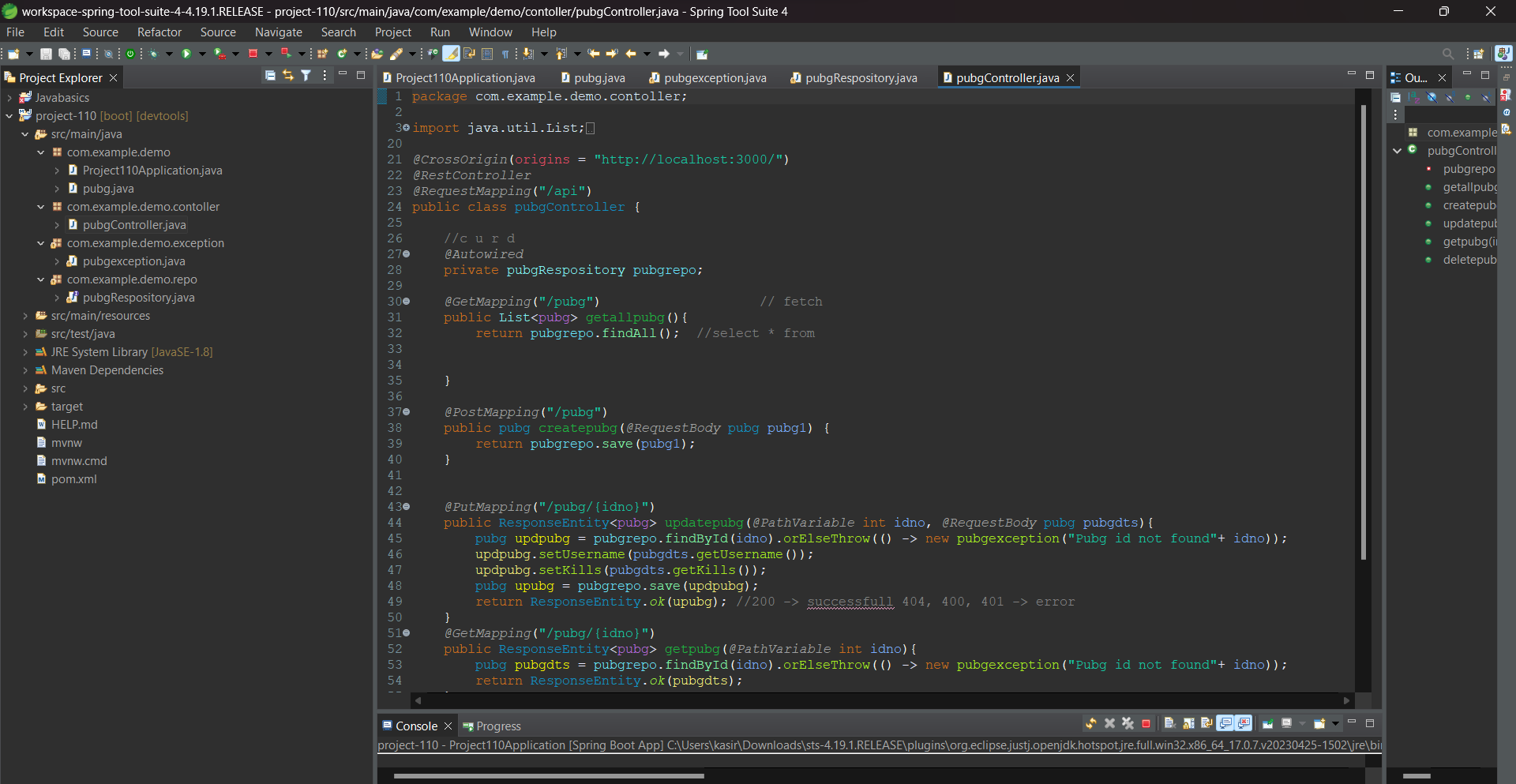
import org.springframework.stereotype.Repository;

import com.example.demo.pubg;

@Repository

public interface pubgRespository extends JpaRepository<pubg, Integer>{

}



package com.example.demo.contoller;

import java.util.List;

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

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

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

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

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

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

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

import com.example.demo.pubg;

import com.example.demo.exception.pubgexception;

import com.example.demo.repo.pubgRespository;

@CrossOrigin(origins = "http://localhost:3000/")

@RestController

@RequestMapping("/api")

public class pubgController {

//c u r d

@Autowired

private pubgRespository pubgrepo;

@GetMapping("/pubg") // fetch

public List<pubg> getallpubg(){

return pubgrepo.findAll(); //select \* from

}

@PostMapping("/pubg")

public pubg createpubg(@RequestBody pubg pubg1) {

return pubgrepo.save(pubg1);

}

@PutMapping("/pubg/{idno}")

public ResponseEntity<pubg> updatepubg(@PathVariable int idno, @RequestBody pubg pubgdts){

pubg updpubg = pubgrepo.findById(idno).orElseThrow(() -> new pubgexception("Pubg id not found"+ idno));

updpubg.setUsername(pubgdts.getUsername());

updpubg.setKills(pubgdts.getKills());

pubg upubg = pubgrepo.save(updpubg);

return ResponseEntity.ok(upubg); //200 -> successfull 404, 400, 401 -> error

}

@GetMapping("/pubg/{idno}")

public ResponseEntity<pubg> getpubg(@PathVariable int idno){

pubg pubgdts = pubgrepo.findById(idno).orElseThrow(() -> new pubgexception("Pubg id not found"+ idno));

return ResponseEntity.ok(pubgdts);

}

@DeleteMapping("/pubg/{idno}")

public void deletepubg(@PathVariable int idno) {

pubgrepo.deleteById(idno);

}

}