Final Project Work

Points possible: 100

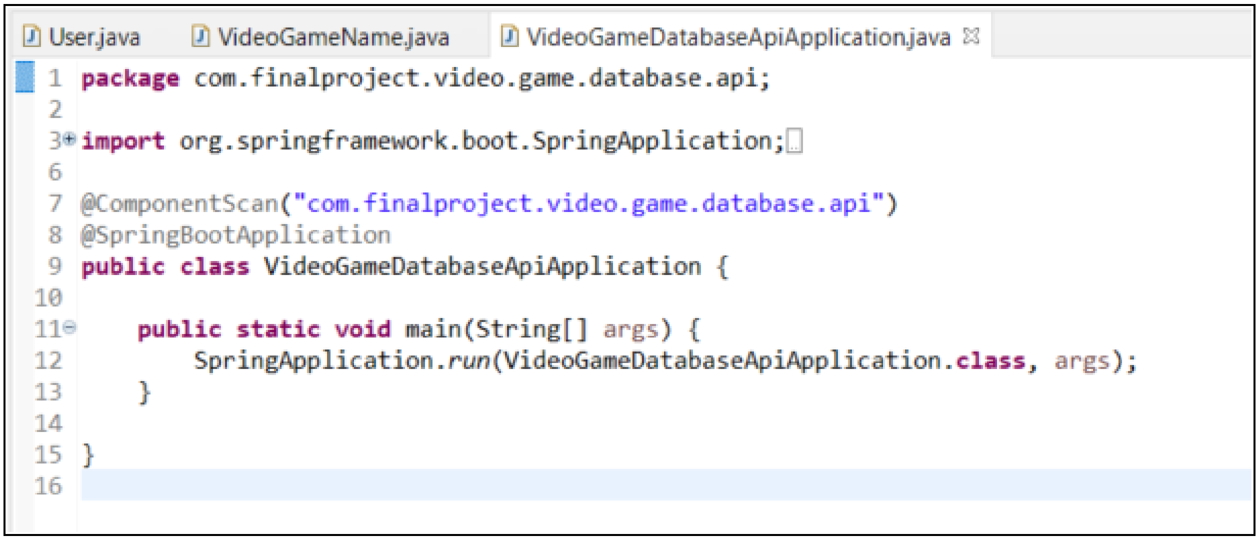
Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete that were committed to for the given week.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Lastly, in the Learning Management System, click the "Add Submission" button and paste the URL to your GitHub repository.

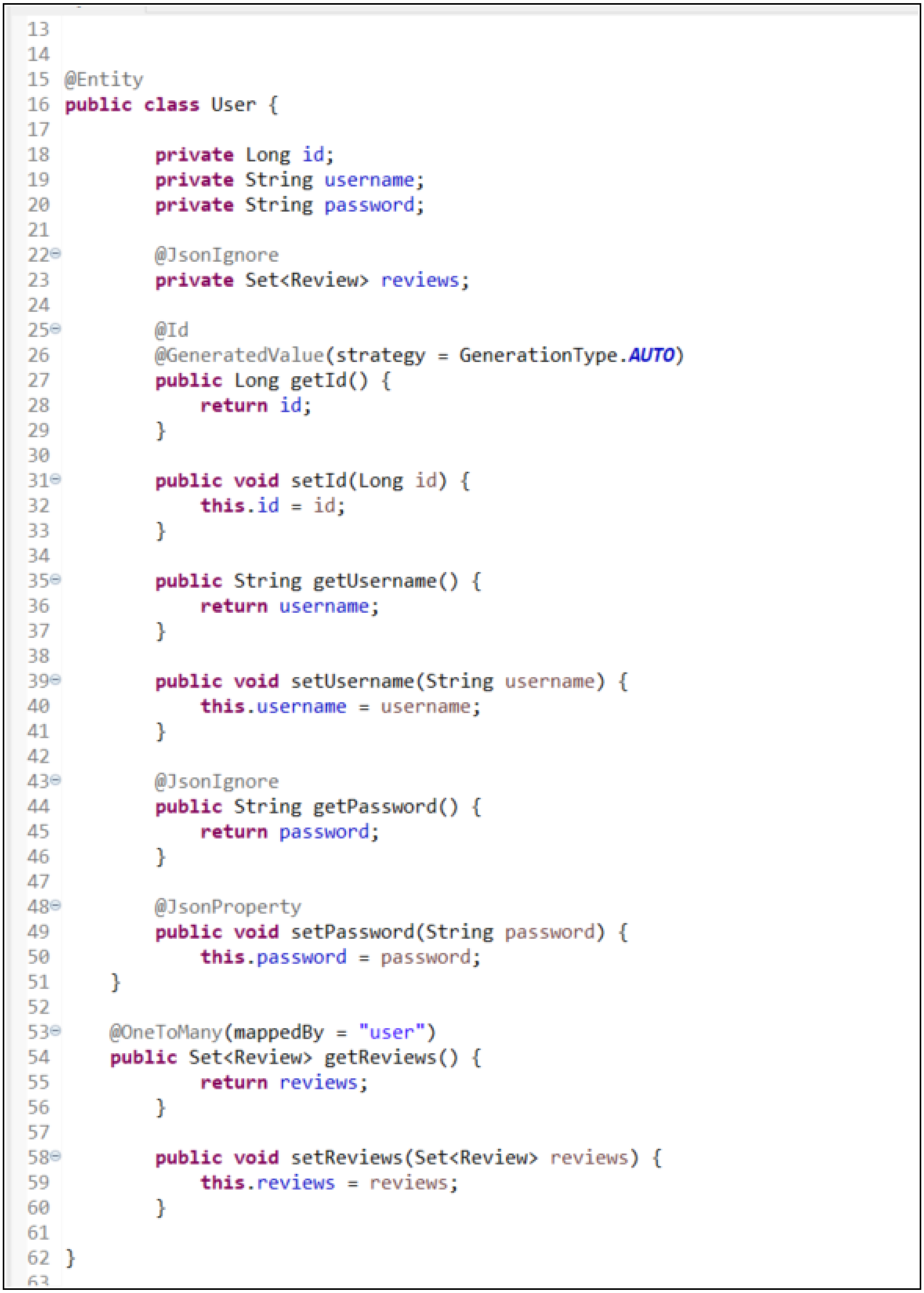
Coding Steps:

Continue contributing to your final project.

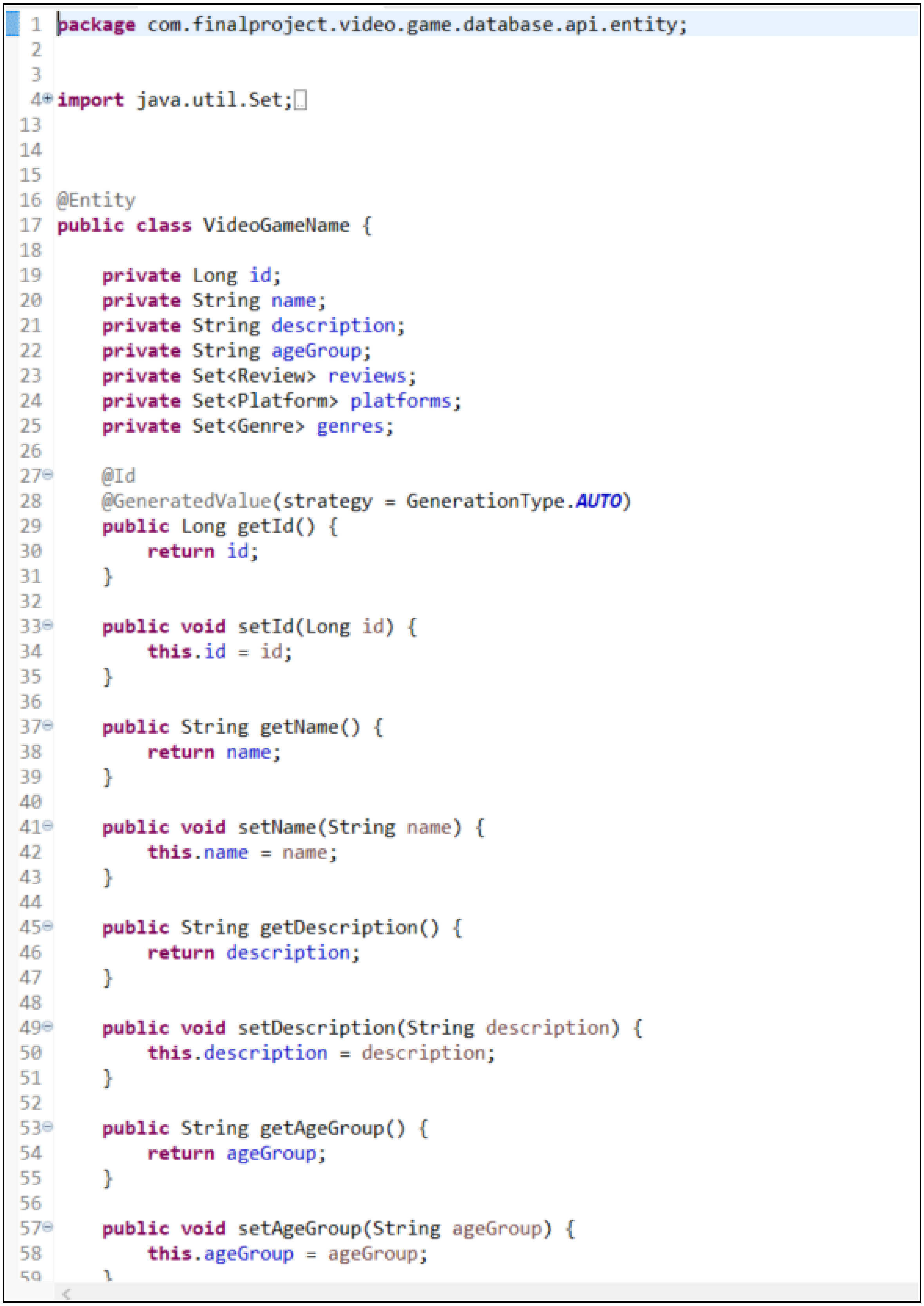
Screenshots of Code:



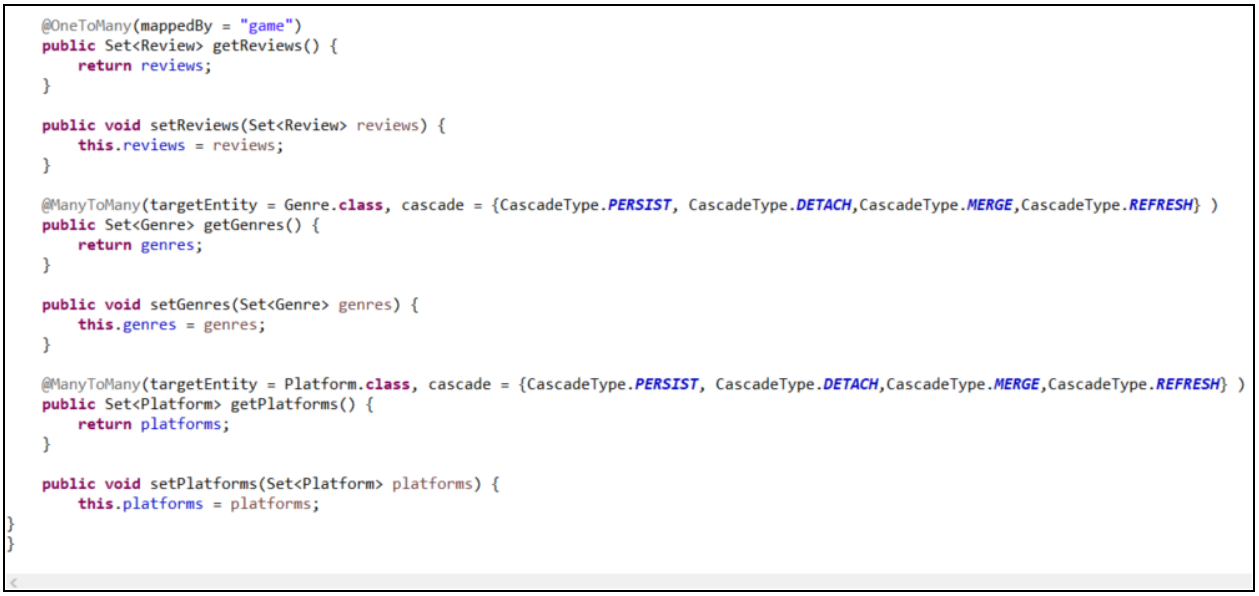
```java
package com.finalproject.video.game.database.api;

import org.springframework.boot.SpringApplication;

@ComponentScan("com.finalproject.video.game.database.api")
@SpringBootApplication
public class VideoGameDatabaseApiApplication {

    public static void main(String[] args) {
        SpringApplication.run(VideoGameDatabaseApiApplication.class, args);
    }

}

```



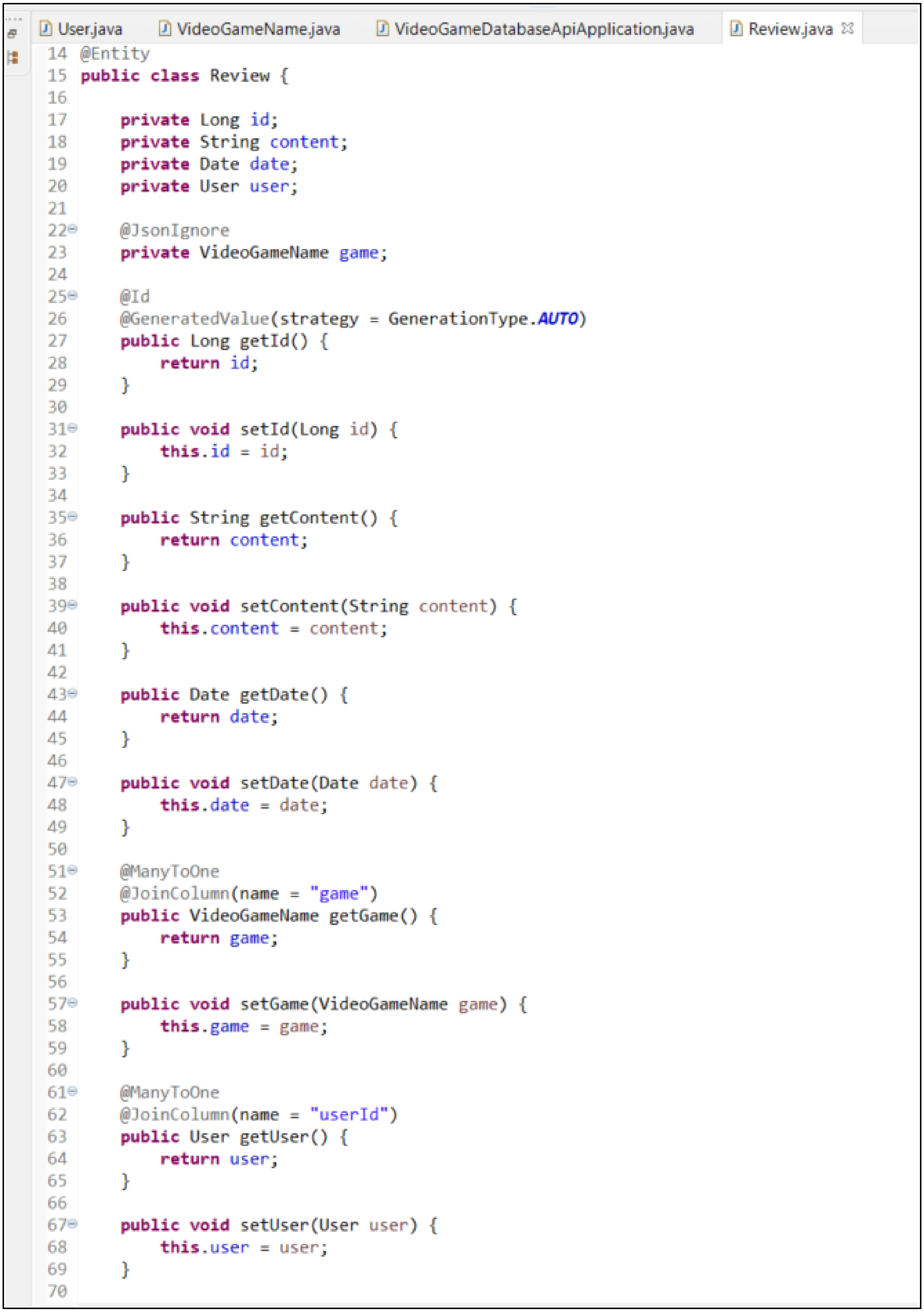
```


@Entity
public class User {

        private Long id;
        private String username;
        private String password;

        @JsonIgnore
        private Set<Review> reviews;

        @Id
        @GeneratedValue(strategy = GenerationType.AUTO)
        public Long getId() {
            return id;
        }

        public void setId(Long id) {
            this.id = id;
        }

        public String getUsername() {
            return username;
        }

        public void setUsername(String username) {
            this.username = username;
        }

        @JsonIgnore
        public String getPassword() {
            return password;
        }

        @JsonProperty
        public void setPassword(String password) {
            this.password = password;
    }

    @OneToMany(mappedBy = "user")
    public Set<Review> getReviews() {
            return reviews;
        }

        public void setReviews(Set<Review> reviews) {
            this.reviews = reviews;
        }

}
```



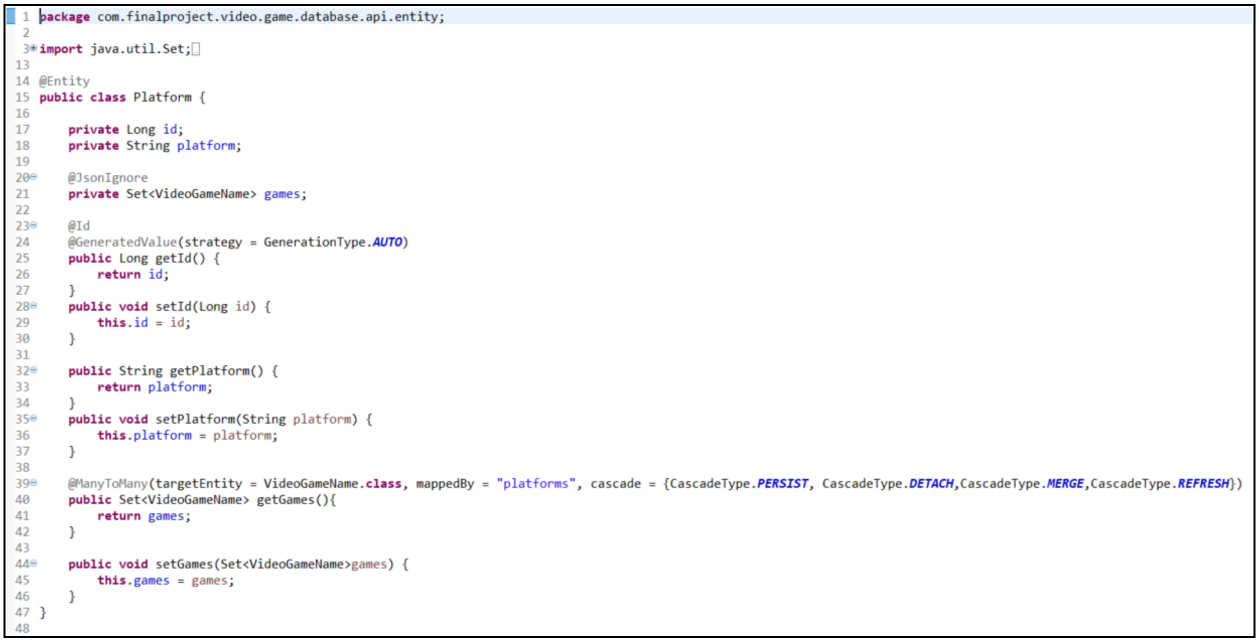
```java
package com.finalproject.video.game.database.api.entity;


import java.util.Set;



@Entity
public class VideoGameName {

    private Long id;
    private String name;
    private String description;
    private String ageGroup;
    private Set<Review> reviews;
    private Set<Platform> platforms;
    private Set<Genre> genres;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getAgeGroup() {
        return ageGroup;
    }

    public void setAgeGroup(String ageGroup) {
        this.ageGroup = ageGroup;
    }
```



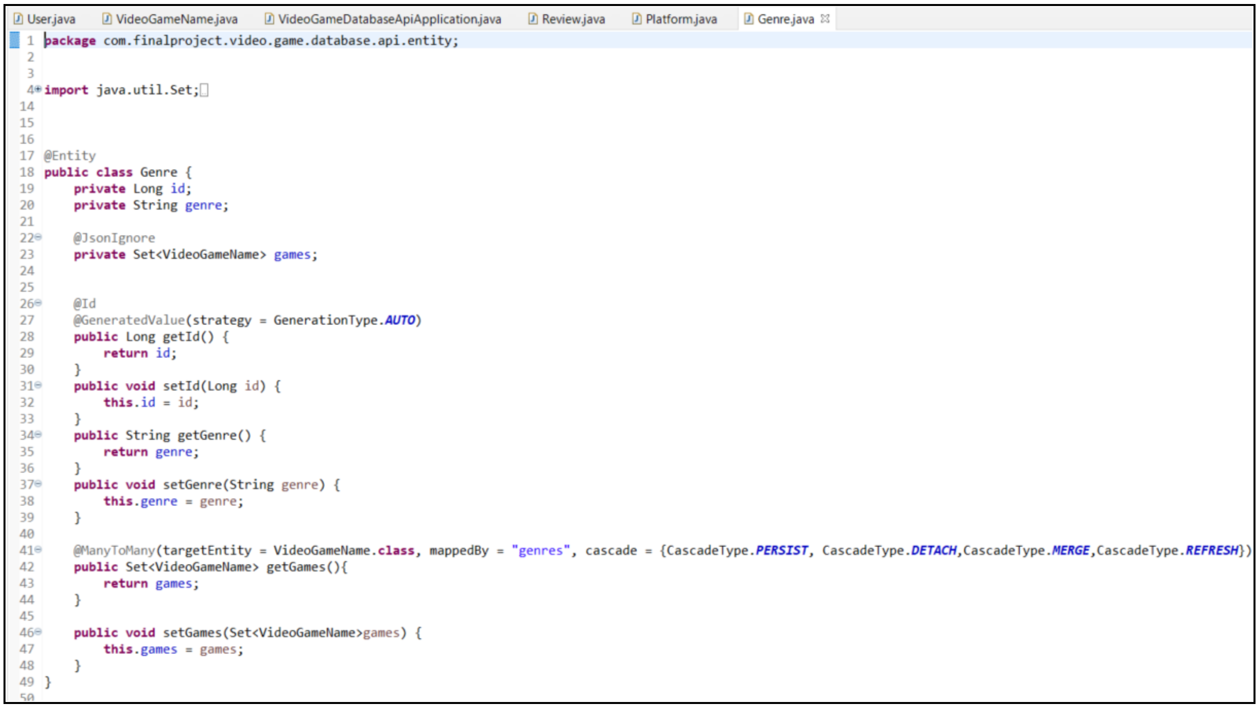
```
    @OneToMany(mappedBy = "game")
    public Set<Review> getReviews() {
        return reviews;
    }

    public void setReviews(Set<Review> reviews) {
        this.reviews = reviews;
    }

    @ManyToMany(targetEntity = Genre.class, cascade = {CascadeType.PERSIST, CascadeType.DETACH,CascadeType.MERGE,CascadeType.REFRESH} )
    public Set<Genre> getGenres() {
        return genres;
    }

    public void setGenres(Set<Genre> genres) {
        this.genres = genres;
    }

    @ManyToMany(targetEntity = Platform.class, cascade = {CascadeType.PERSIST, CascadeType.DETACH,CascadeType.MERGE,CascadeType.REFRESH} )
    public Set<Platform> getPlatforms() {
        return platforms;
    }

    public void setPlatforms(Set<Platform> platforms) {
        this.platforms = platforms;
}
}
```



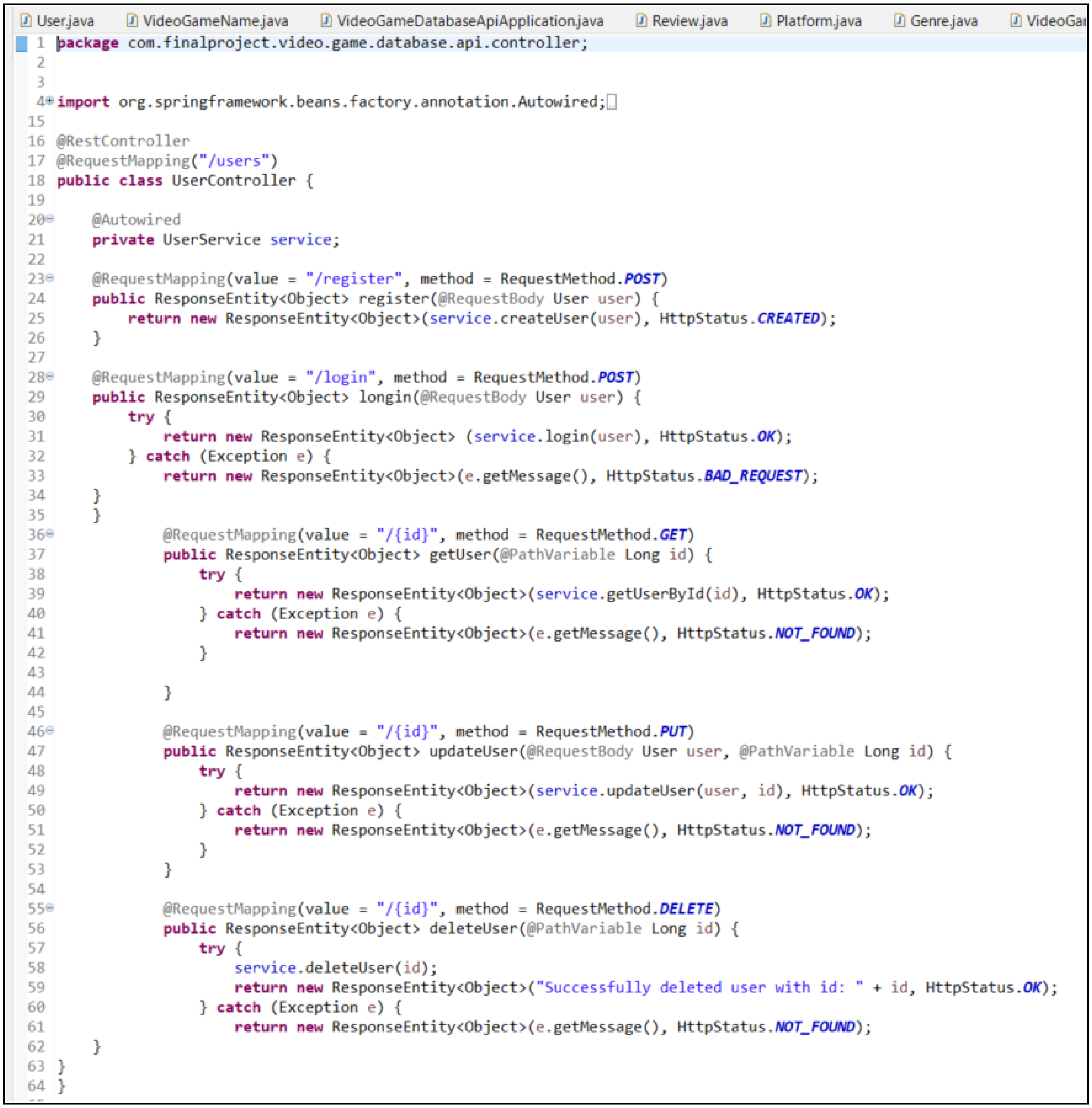
```java
@Entity
public class Review {

    private Long id;
    private String content;
    private Date date;
    private User user;

    @JsonIgnore
    private VideoGameName game;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getContent() {
        return content;
    }

    public void setContent(String content) {
        this.content = content;
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    @ManyToOne
    @JoinColumn(name = "game")
    public VideoGameName getGame() {
        return game;
    }

    public void setGame(VideoGameName game) {
        this.game = game;
    }

    @ManyToOne
    @JoinColumn(name = "userId")
    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

```



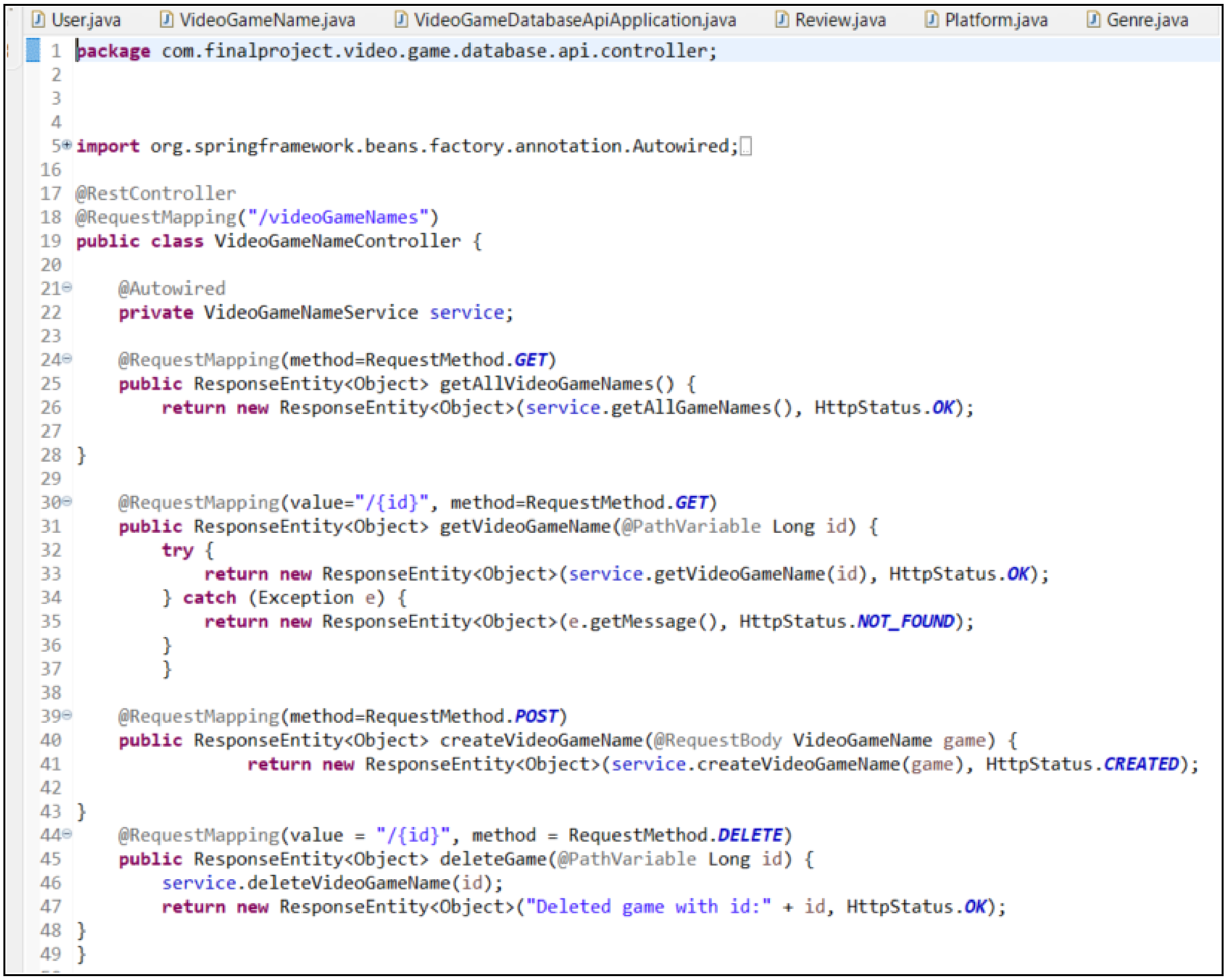
```java
package com.finalproject.video.game.database.api.entity;

import java.util.Set;

@Entity
public class Platform {

    private Long id;
    private String platform;

    @JsonIgnore
    private Set<VideoGameName> games;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }

    public String getPlatform() {
        return platform;
    }
    public void setPlatform(String platform) {
        this.platform = platform;
    }

    @ManyToMany(targetEntity = VideoGameName.class, mappedBy = "platforms", cascade = {CascadeType.PERSIST, CascadeType.DETACH,CascadeType.MERGE,CascadeType.REFRESH})
    public Set<VideoGameName> getGames(){
        return games;
    }

    public void setGames(Set<VideoGameName>games) {
        this.games = games;
    }
}
```



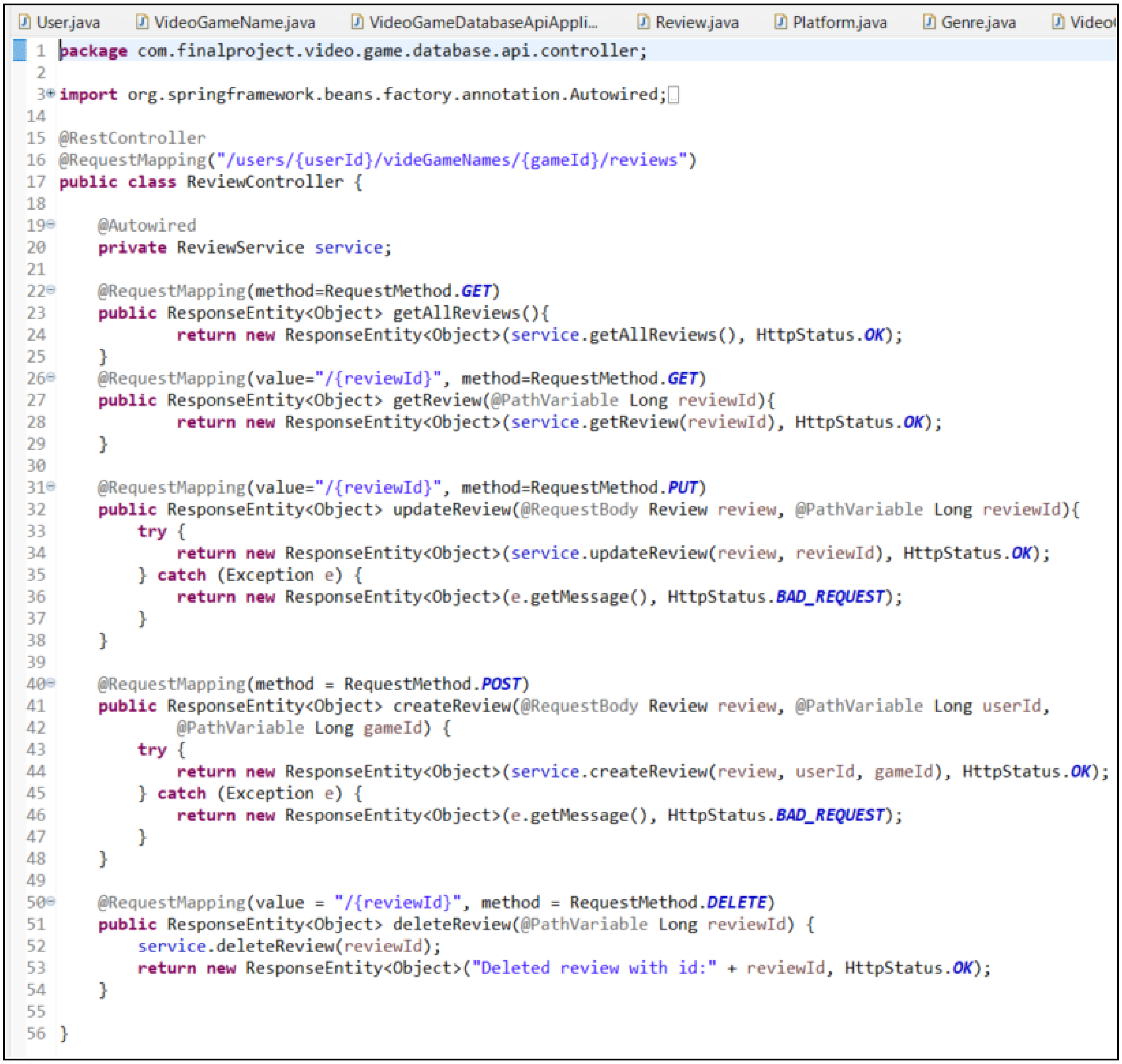
```java
package com.finalproject.video.game.database.api.entity;


import java.util.Set;



@Entity
public class Genre {
    private Long id;
    private String genre;

    @JsonIgnore
    private Set<VideoGameName> games;


    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getGenre() {
        return genre;
    }
    public void setGenre(String genre) {
        this.genre = genre;
    }

    @ManyToMany(targetEntity = VideoGameName.class, mappedBy = "genres", cascade = {CascadeType.PERSIST, CascadeType.DETACH,CascadeType.MERGE,CascadeType.REFRESH})
    public Set<VideoGameName> getGames(){
        return games;
    }

    public void setGames(Set<VideoGameName>games) {
        this.games = games;
    }
}
```



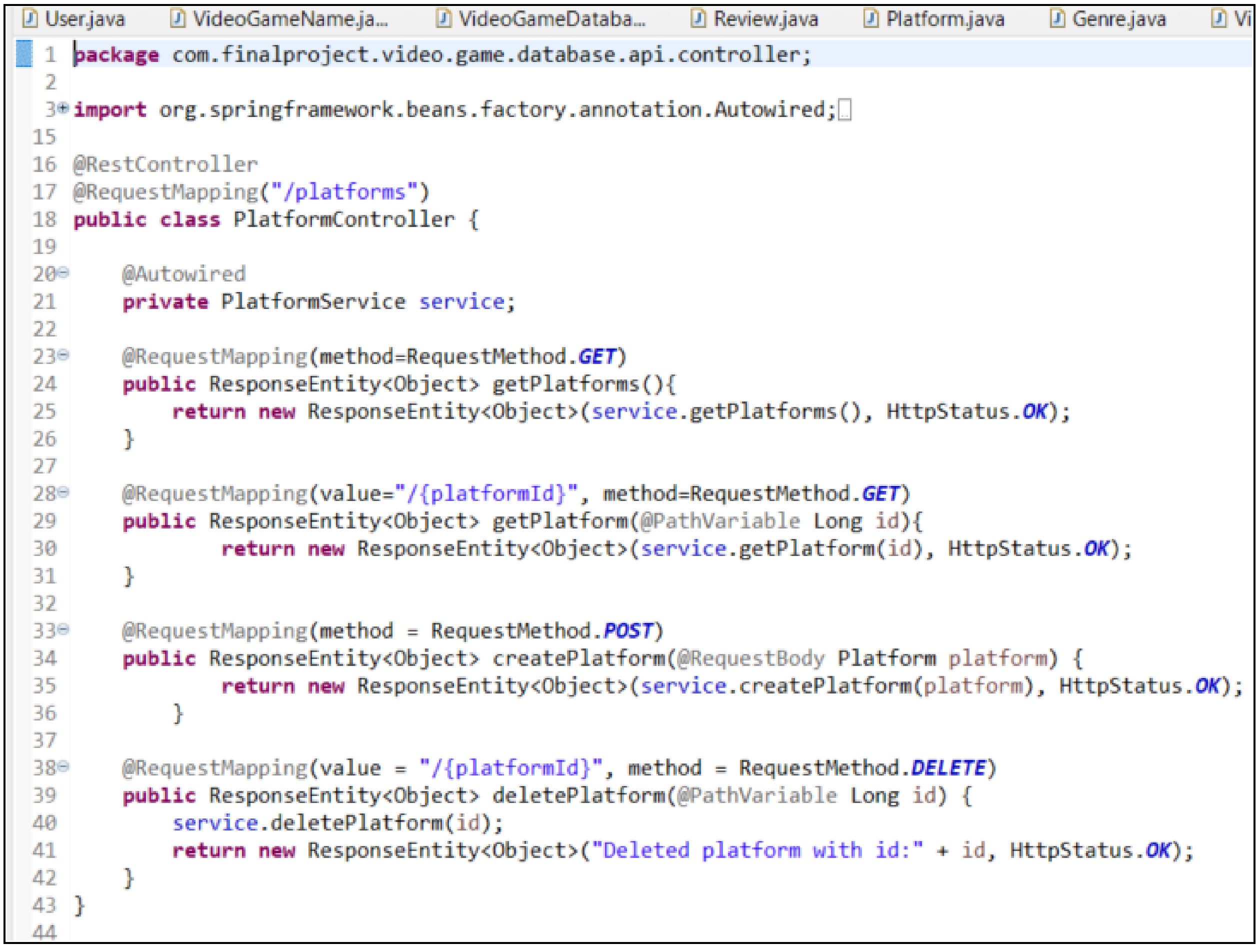
```java
package com.finalproject.video.game.database.api.controller;


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

@RestController
@RequestMapping("/users")
public class UserController {

    @Autowired
    private UserService service;

    @RequestMapping(value = "/register", method = RequestMethod.POST)
    public ResponseEntity<Object> register(@RequestBody User user) {
        return new ResponseEntity<Object>(service.createUser(user), HttpStatus.CREATED);
    }

    @RequestMapping(value = "/login", method = RequestMethod.POST)
    public ResponseEntity<Object> longin(@RequestBody User user) {
        try {
            return new ResponseEntity<Object> (service.login(user), HttpStatus.OK);
        } catch (Exception e) {
            return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
    }
    }
            @RequestMapping(value = "/{id}", method = RequestMethod.GET)
            public ResponseEntity<Object> getUser(@PathVariable Long id) {
                try {
                    return new ResponseEntity<Object>(service.getUserById(id), HttpStatus.OK);
                } catch (Exception e) {
                    return new ResponseEntity<Object>(e.getMessage(), HttpStatus.NOT_FOUND);
                }

            }

            @RequestMapping(value = "/{id}", method = RequestMethod.PUT)
            public ResponseEntity<Object> updateUser(@RequestBody User user, @PathVariable Long id) {
                try {
                    return new ResponseEntity<Object>(service.updateUser(user, id), HttpStatus.OK);
                } catch (Exception e) {
                    return new ResponseEntity<Object>(e.getMessage(), HttpStatus.NOT_FOUND);
                }
            }

            @RequestMapping(value = "/{id}", method = RequestMethod.DELETE)
            public ResponseEntity<Object> deleteUser(@PathVariable Long id) {
                try {
                    service.deleteUser(id);
                    return new ResponseEntity<Object>("Successfully deleted user with id: " + id, HttpStatus.OK);
                } catch (Exception e) {
                    return new ResponseEntity<Object>(e.getMessage(), HttpStatus.NOT_FOUND);
    }
}
}
```



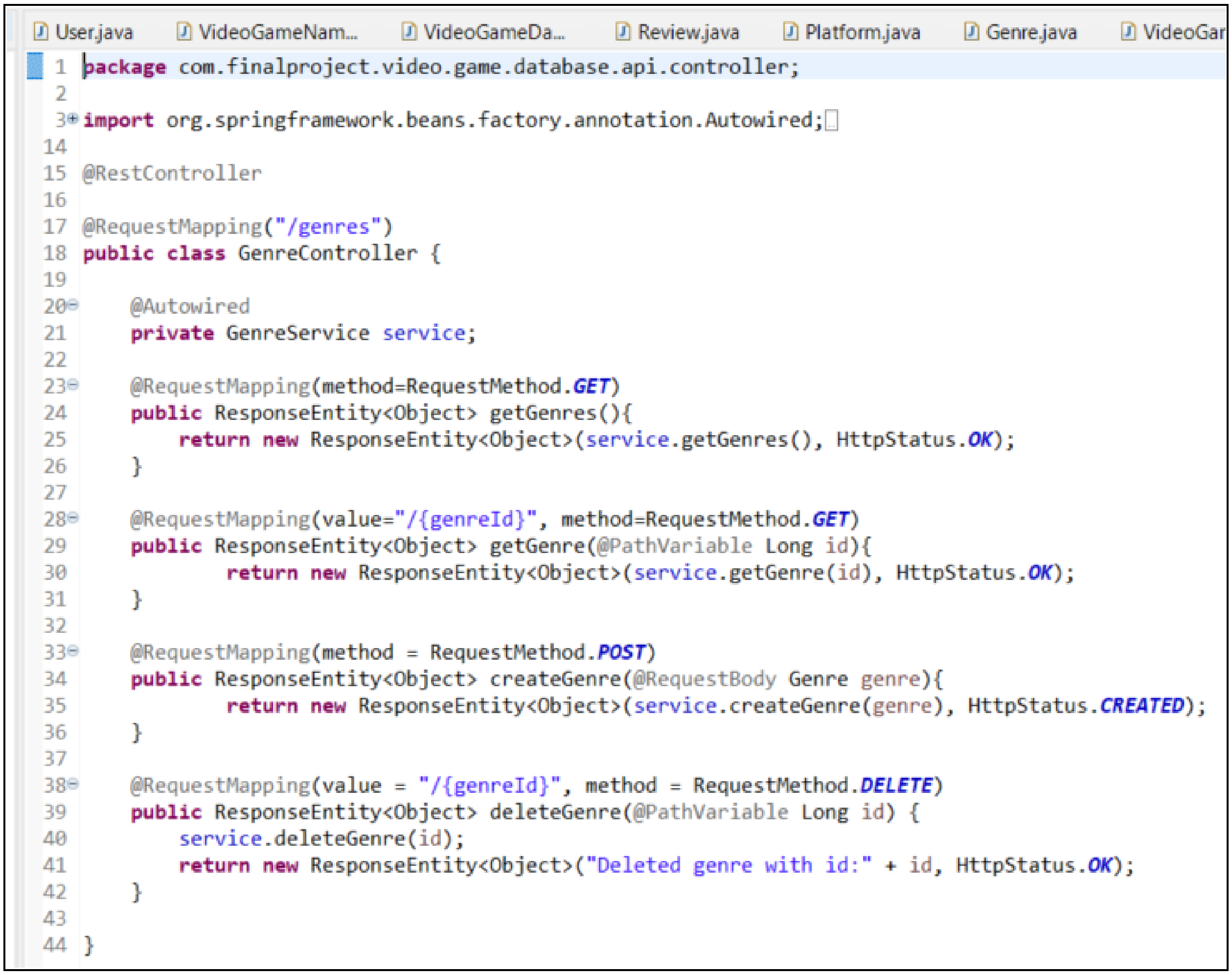
```java
package com.finalproject.video.game.database.api.controller;



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

@RestController
@RequestMapping("/videoGameNames")
public class VideoGameNameController {

    @Autowired
    private VideoGameNameService service;

    @RequestMapping(method=RequestMethod.GET)
    public ResponseEntity<Object> getAllVideoGameNames() {
        return new ResponseEntity<Object>(service.getAllGameNames(), HttpStatus.OK);

}

    @RequestMapping(value="/{id}", method=RequestMethod.GET)
    public ResponseEntity<Object> getVideoGameName(@PathVariable Long id) {
        try {
            return new ResponseEntity<Object>(service.getVideoGameName(id), HttpStatus.OK);
        } catch (Exception e) {
            return new ResponseEntity<Object>(e.getMessage(), HttpStatus.NOT_FOUND);
        }
        }

    @RequestMapping(method=RequestMethod.POST)
    public ResponseEntity<Object> createVideoGameName(@RequestBody VideoGameName game) {
                return new ResponseEntity<Object>(service.createVideoGameName(game), HttpStatus.CREATED);

}
    @RequestMapping(value = "/{id}", method = RequestMethod.DELETE)
    public ResponseEntity<Object> deleteGame(@PathVariable Long id) {
        service.deleteVideoGameName(id);
        return new ResponseEntity<Object>("Deleted game with id:" + id, HttpStatus.OK);
}
}
```



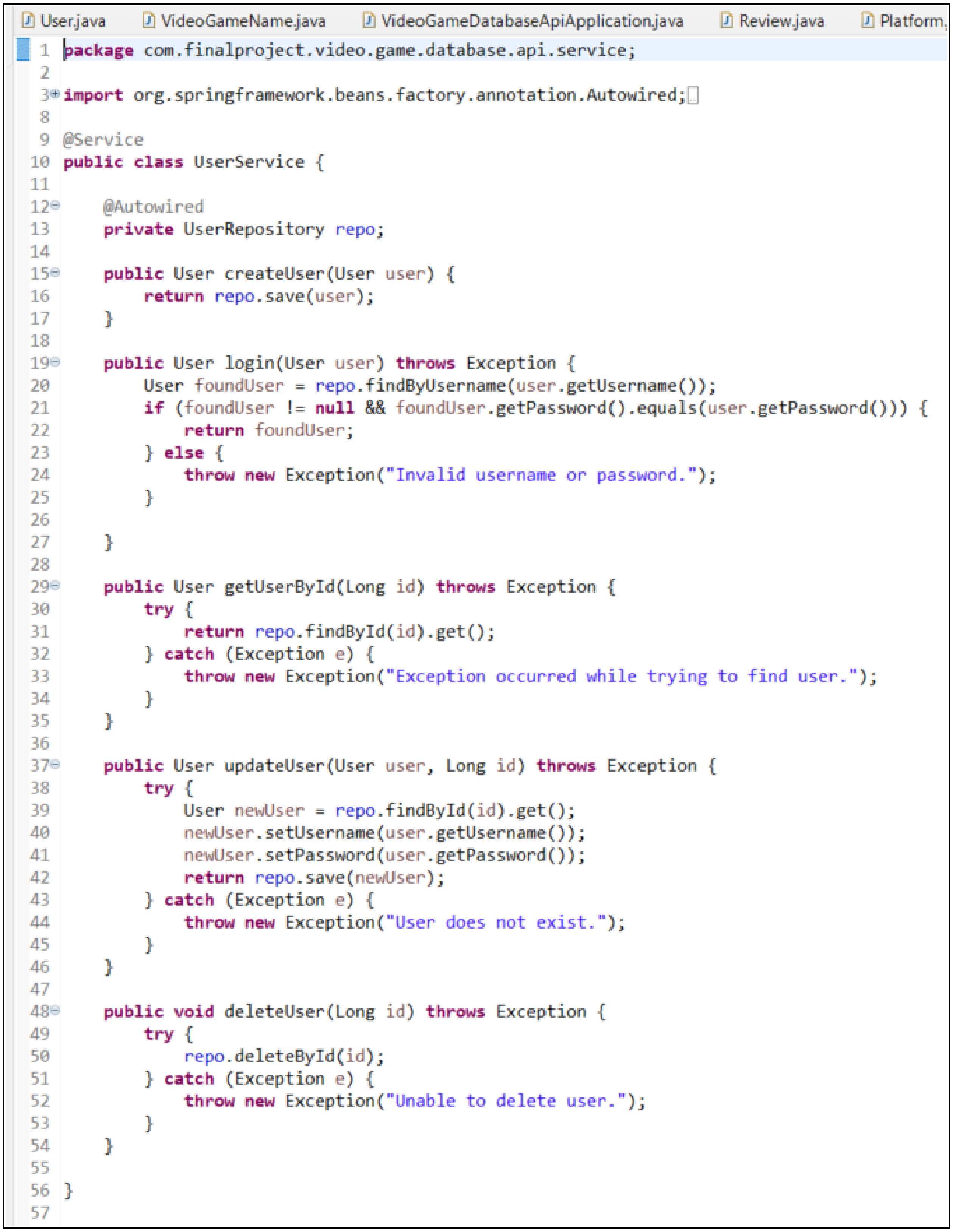
```java
package com.finalproject.video.game.database.api.controller;

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

@RestController
@RequestMapping("/users/{userId}/videGameNames/{gameId}/reviews")
public class ReviewController {

    @Autowired
    private ReviewService service;

    @RequestMapping(method=RequestMethod.GET)
    public ResponseEntity<Object> getAllReviews(){
            return new ResponseEntity<Object>(service.getAllReviews(), HttpStatus.OK);
    }
    @RequestMapping(value="/{reviewId}", method=RequestMethod.GET)
    public ResponseEntity<Object> getReview(@PathVariable Long reviewId){
            return new ResponseEntity<Object>(service.getReview(reviewId), HttpStatus.OK);
    }

    @RequestMapping(value="/{reviewId}", method=RequestMethod.PUT)
    public ResponseEntity<Object> updateReview(@RequestBody Review review, @PathVariable Long reviewId){
        try {
            return new ResponseEntity<Object>(service.updateReview(review, reviewId), HttpStatus.OK);
        } catch (Exception e) {
            return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
        }
    }

    @RequestMapping(method = RequestMethod.POST)
    public ResponseEntity<Object> createReview(@RequestBody Review review, @PathVariable Long userId,
            @PathVariable Long gameId) {
        try {
            return new ResponseEntity<Object>(service.createReview(review, userId, gameId), HttpStatus.OK);
        } catch (Exception e) {
            return new ResponseEntity<Object>(e.getMessage(), HttpStatus.BAD_REQUEST);
        }
    }

    @RequestMapping(value = "/{reviewId}", method = RequestMethod.DELETE)
    public ResponseEntity<Object> deleteReview(@PathVariable Long reviewId) {
        service.deleteReview(reviewId);
        return new ResponseEntity<Object>("Deleted review with id:" + reviewId, HttpStatus.OK);
    }

}
```



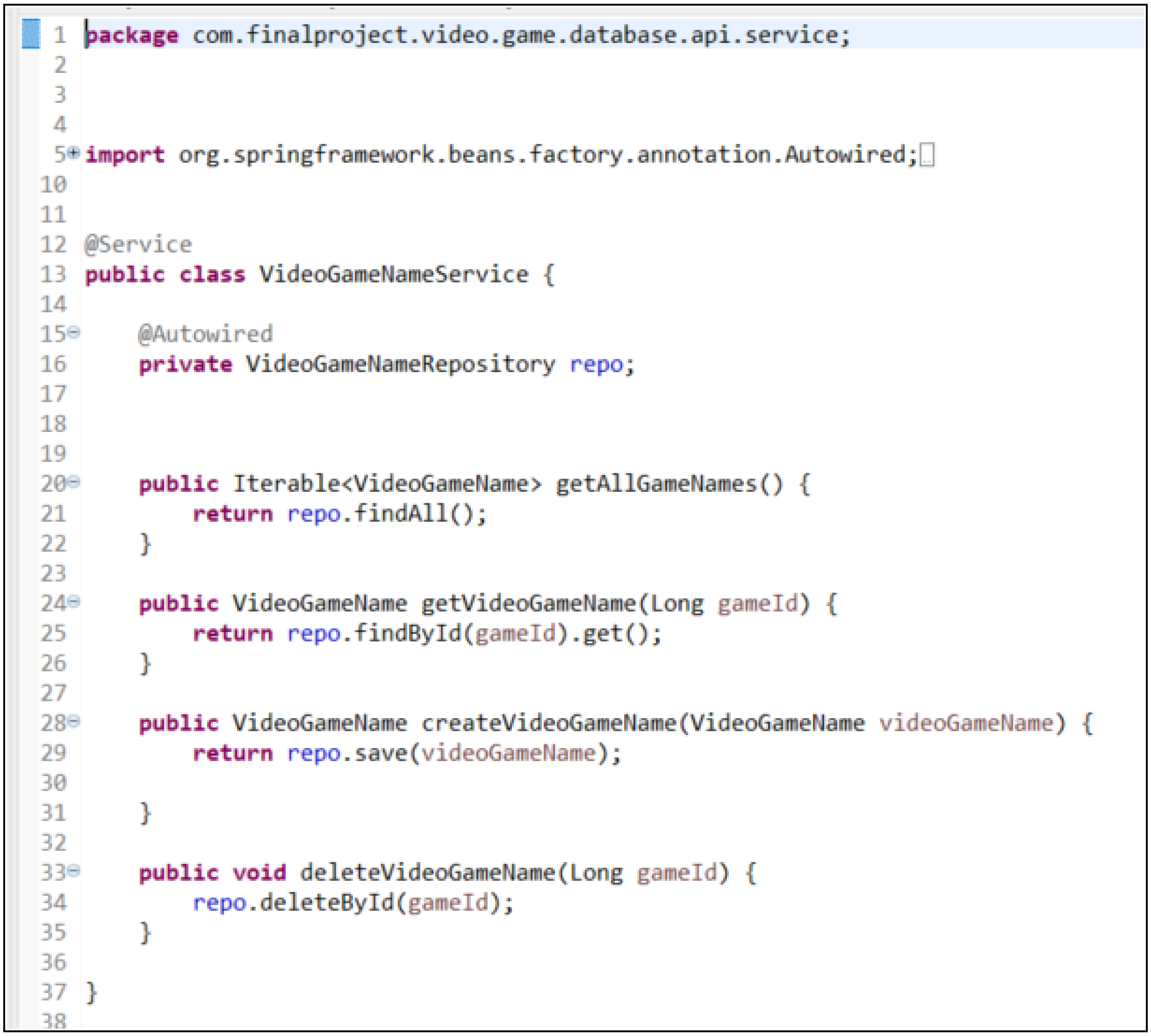
```java
package com.finalproject.video.game.database.api.controller;

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

@RestController
@RequestMapping("/platforms")
public class PlatformController {

    @Autowired
    private PlatformService service;

    @RequestMapping(method=RequestMethod.GET)
    public ResponseEntity<Object> getPlatforms(){
        return new ResponseEntity<Object>(service.getPlatforms(), HttpStatus.OK);
    }

    @RequestMapping(value="/{platformId}", method=RequestMethod.GET)
    public ResponseEntity<Object> getPlatform(@PathVariable Long id){
            return new ResponseEntity<Object>(service.getPlatform(id), HttpStatus.OK);
    }

    @RequestMapping(method = RequestMethod.POST)
    public ResponseEntity<Object> createPlatform(@RequestBody Platform platform) {
            return new ResponseEntity<Object>(service.createPlatform(platform), HttpStatus.OK);
        }

    @RequestMapping(value = "/{platformId}", method = RequestMethod.DELETE)
    public ResponseEntity<Object> deletePlatform(@PathVariable Long id) {
        service.deletePlatform(id);
        return new ResponseEntity<Object>("Deleted platform with id:" + id, HttpStatus.OK);
    }
}

```



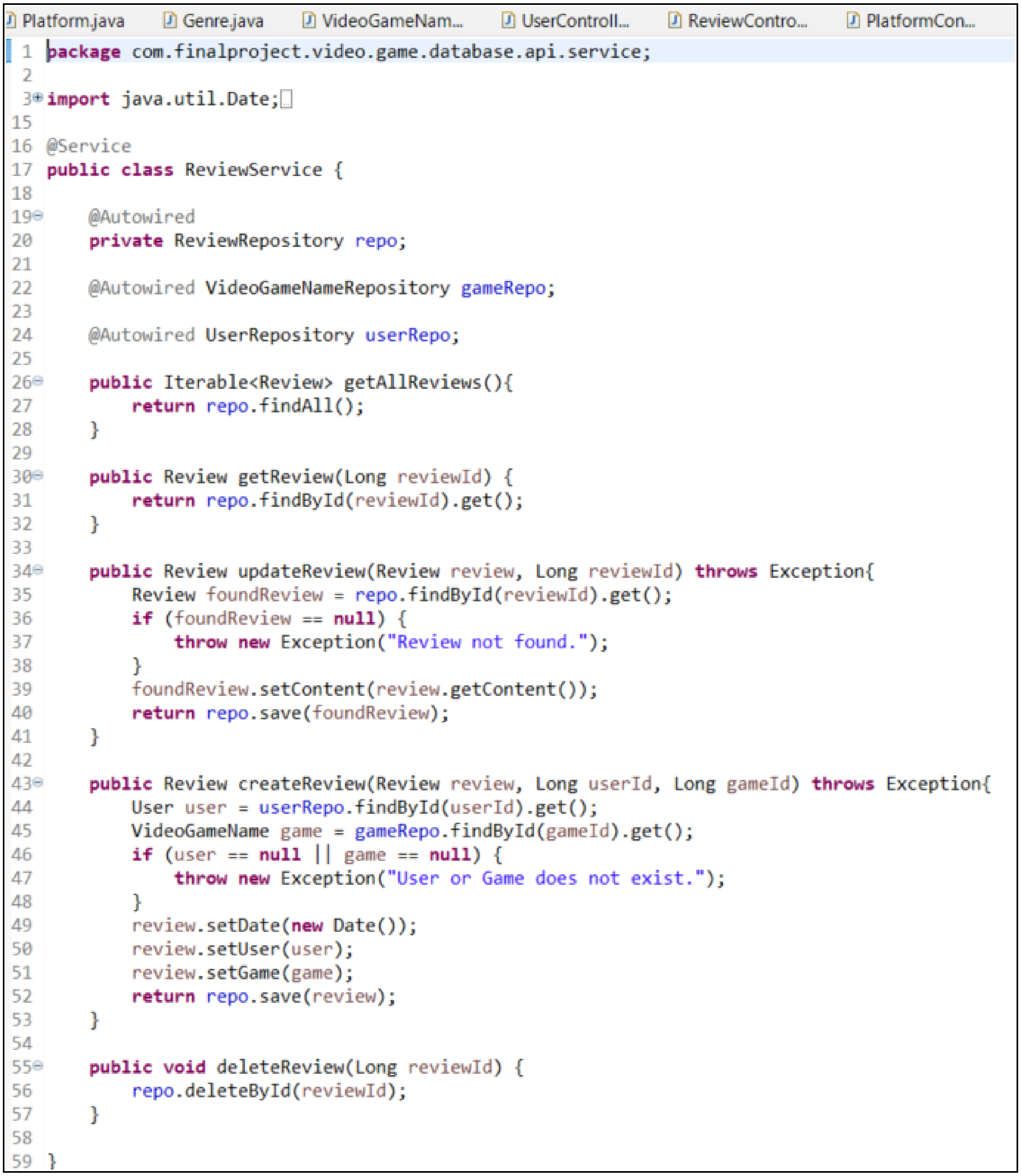
```java
package com.finalproject.video.game.database.api.controller;

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

@RestController

@RequestMapping("/genres")
public class GenreController {

    @Autowired
    private GenreService service;

    @RequestMapping(method=RequestMethod.GET)
    public ResponseEntity<Object> getGenres(){
        return new ResponseEntity<Object>(service.getGenres(), HttpStatus.OK);
    }

    @RequestMapping(value="/{genreId}", method=RequestMethod.GET)
    public ResponseEntity<Object> getGenre(@PathVariable Long id){
            return new ResponseEntity<Object>(service.getGenre(id), HttpStatus.OK);
    }

    @RequestMapping(method = RequestMethod.POST)
    public ResponseEntity<Object> createGenre(@RequestBody Genre genre){
            return new ResponseEntity<Object>(service.createGenre(genre), HttpStatus.CREATED);
    }

    @RequestMapping(value = "/{genreId}", method = RequestMethod.DELETE)
    public ResponseEntity<Object> deleteGenre(@PathVariable Long id) {
        service.deleteGenre(id);
        return new ResponseEntity<Object>("Deleted genre with id:" + id, HttpStatus.OK);
    }

}
```



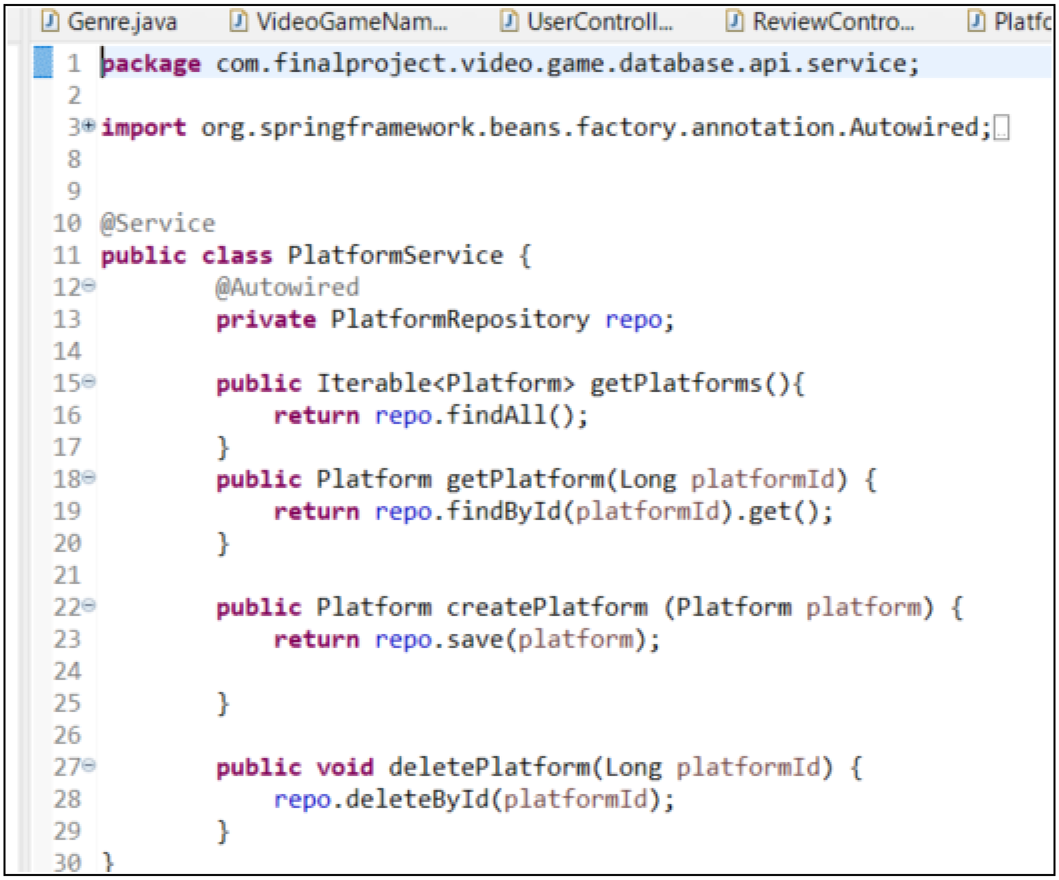
```java
package com.finalproject.video.game.database.api.service;

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

@Service
public class UserService {

    @Autowired
    private UserRepository repo;

    public User createUser(User user) {
        return repo.save(user);
    }

    public User login(User user) throws Exception {
        User foundUser = repo.findByUsername(user.getUsername());
        if (foundUser != null && foundUser.getPassword().equals(user.getPassword())) {
            return foundUser;
        } else {
            throw new Exception("Invalid username or password.");
        }

    }

    public User getUserById(Long id) throws Exception {
        try {
            return repo.findById(id).get();
        } catch (Exception e) {
            throw new Exception("Exception occurred while trying to find user.");
        }
    }

    public User updateUser(User user, Long id) throws Exception {
        try {
            User newUser = repo.findById(id).get();
            newUser.setUsername(user.getUsername());
            newUser.setPassword(user.getPassword());
            return repo.save(newUser);
        } catch (Exception e) {
            throw new Exception("User does not exist.");
        }
    }

    public void deleteUser(Long id) throws Exception {
        try {
            repo.deleteById(id);
        } catch (Exception e) {
            throw new Exception("Unable to delete user.");
        }
    }

}
```



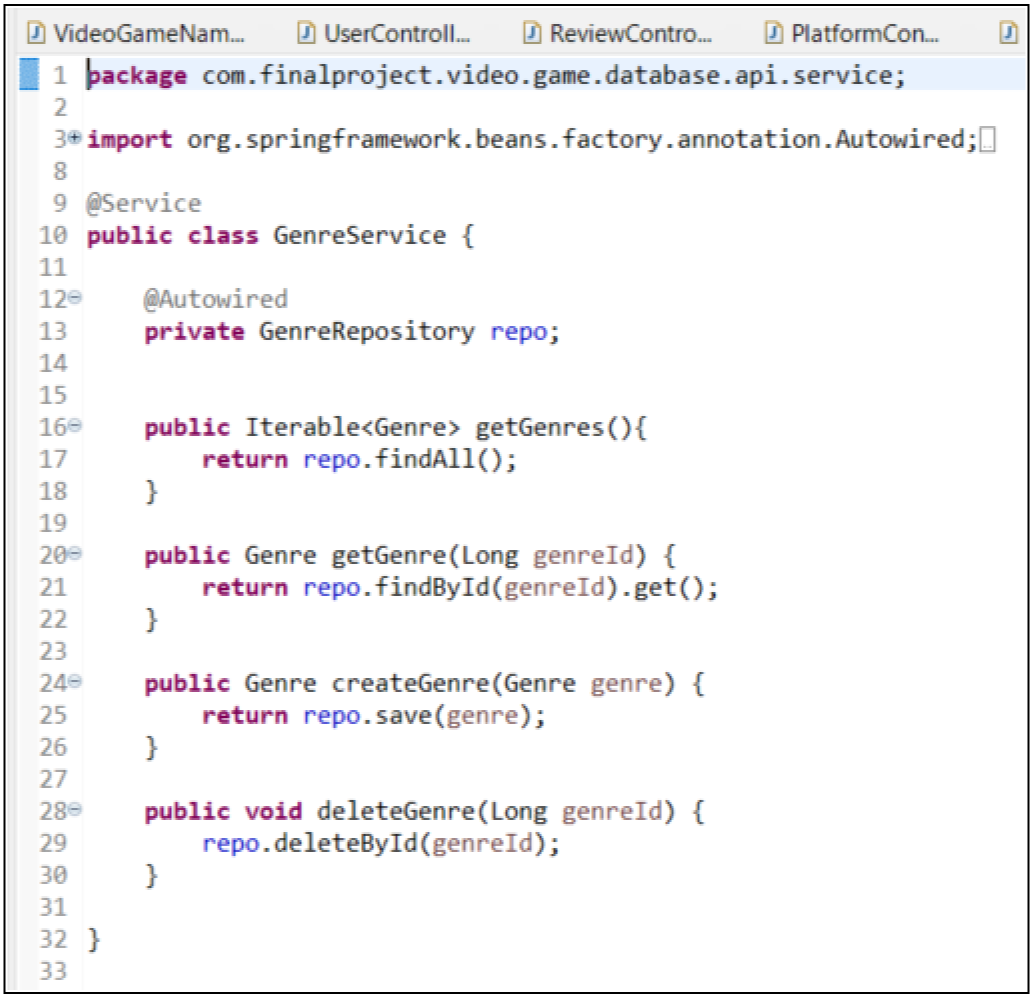
```
package com.finalproject.video.game.database.api.service;



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


@Service
public class VideoGameNameService {

    @Autowired
    private VideoGameNameRepository repo;



    public Iterable<VideoGameName> getAllGameNames() {
        return repo.findAll();
    }

    public VideoGameName getVideoGameName(Long gameId) {
        return repo.findById(gameId).get();
    }

    public VideoGameName createVideoGameName(VideoGameName videoGameName) {
        return repo.save(videoGameName);

    }

    public void deleteVideoGameName(Long gameId) {
        repo.deleteById(gameId);
    }

}
```



```java
package com.finalproject.video.game.database.api.service;

import java.util.Date;

@Service
public class ReviewService {

    @Autowired
    private ReviewRepository repo;

    @Autowired VideoGameNameRepository gameRepo;

    @Autowired UserRepository userRepo;

    public Iterable<Review> getAllReviews(){
        return repo.findAll();
    }

    public Review getReview(Long reviewId) {
        return repo.findById(reviewId).get();
    }

    public Review updateReview(Review review, Long reviewId) throws Exception{
        Review foundReview = repo.findById(reviewId).get();
        if (foundReview == null) {
            throw new Exception("Review not found.");
        }
        foundReview.setContent(review.getContent());
        return repo.save(foundReview);
    }

    public Review createReview(Review review, Long userId, Long gameId) throws Exception{
        User user = userRepo.findById(userId).get();
        VideoGameName game = gameRepo.findById(gameId).get();
        if (user == null || game == null) {
            throw new Exception("User or Game does not exist.");
        }
        review.setDate(new Date());
        review.setUser(user);
        review.setGame(game);
        return repo.save(review);
    }

    public void deleteReview(Long reviewId) {
        repo.deleteById(reviewId);
    }

}
```

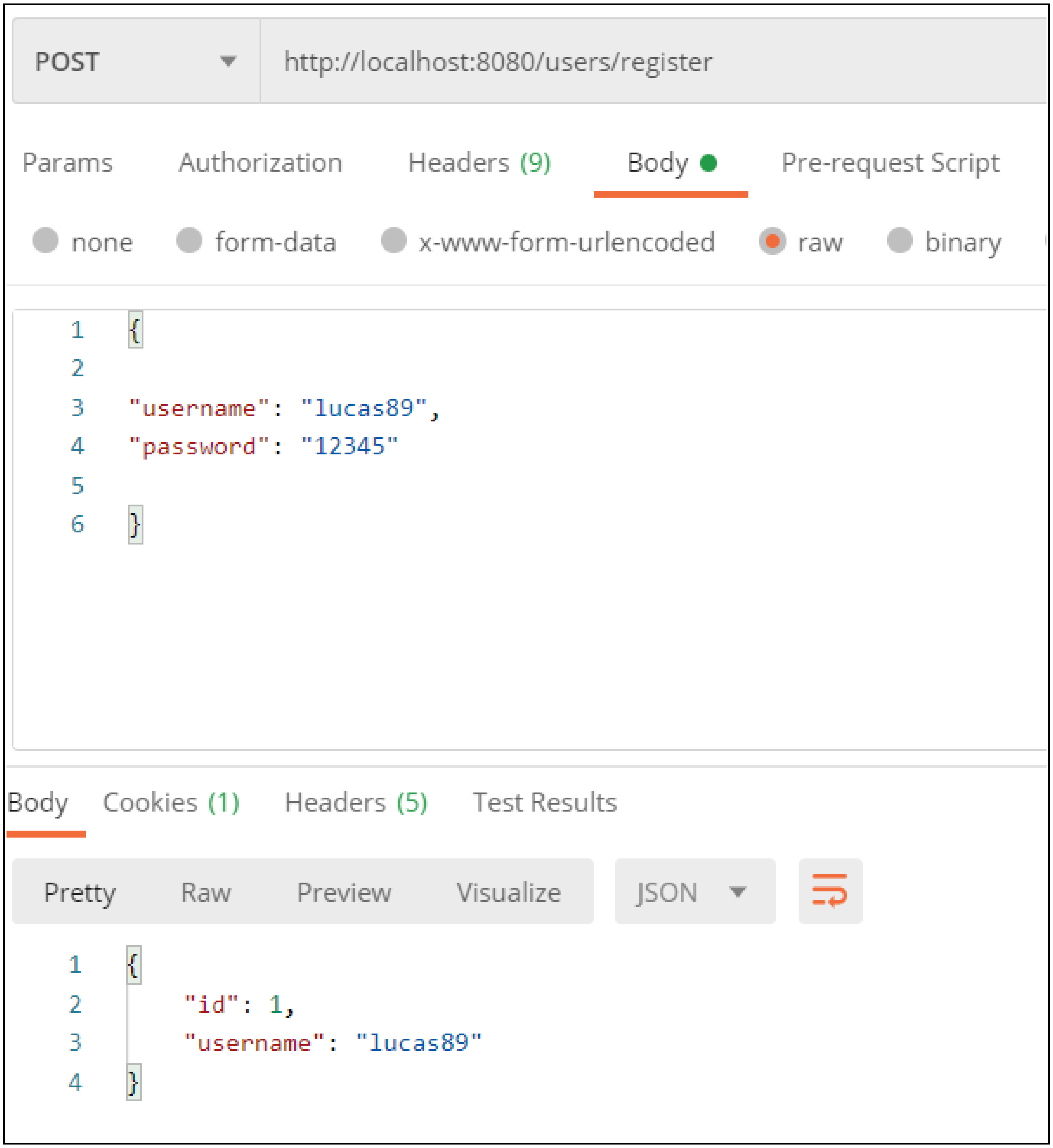


```java
package com.finalproject.video.game.database.api.service;

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


@Service
public class PlatformService {
        @Autowired
        private PlatformRepository repo;

        public Iterable<Platform> getPlatforms(){
            return repo.findAll();
        }
        public Platform getPlatform(Long platformId) {
            return repo.findById(platformId).get();
        }

        public Platform createPlatform (Platform platform) {
            return repo.save(platform);

        }

        public void deletePlatform(Long platformId) {
            repo.deleteById(platformId);
        }
}
```



```java
package com.finalproject.video.game.database.api.service;

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

@Service
public class GenreService {

    @Autowired
    private GenreRepository repo;


    public Iterable<Genre> getGenres(){
        return repo.findAll();
    }

    public Genre getGenre(Long genreId) {
        return repo.findById(genreId).get();
    }

    public Genre createGenre(Genre genre) {
        return repo.save(genre);
    }

    public void deleteGenre(Long genreId) {
        repo.deleteById(genreId);
    }

}

```

Screenshots of Running Application:



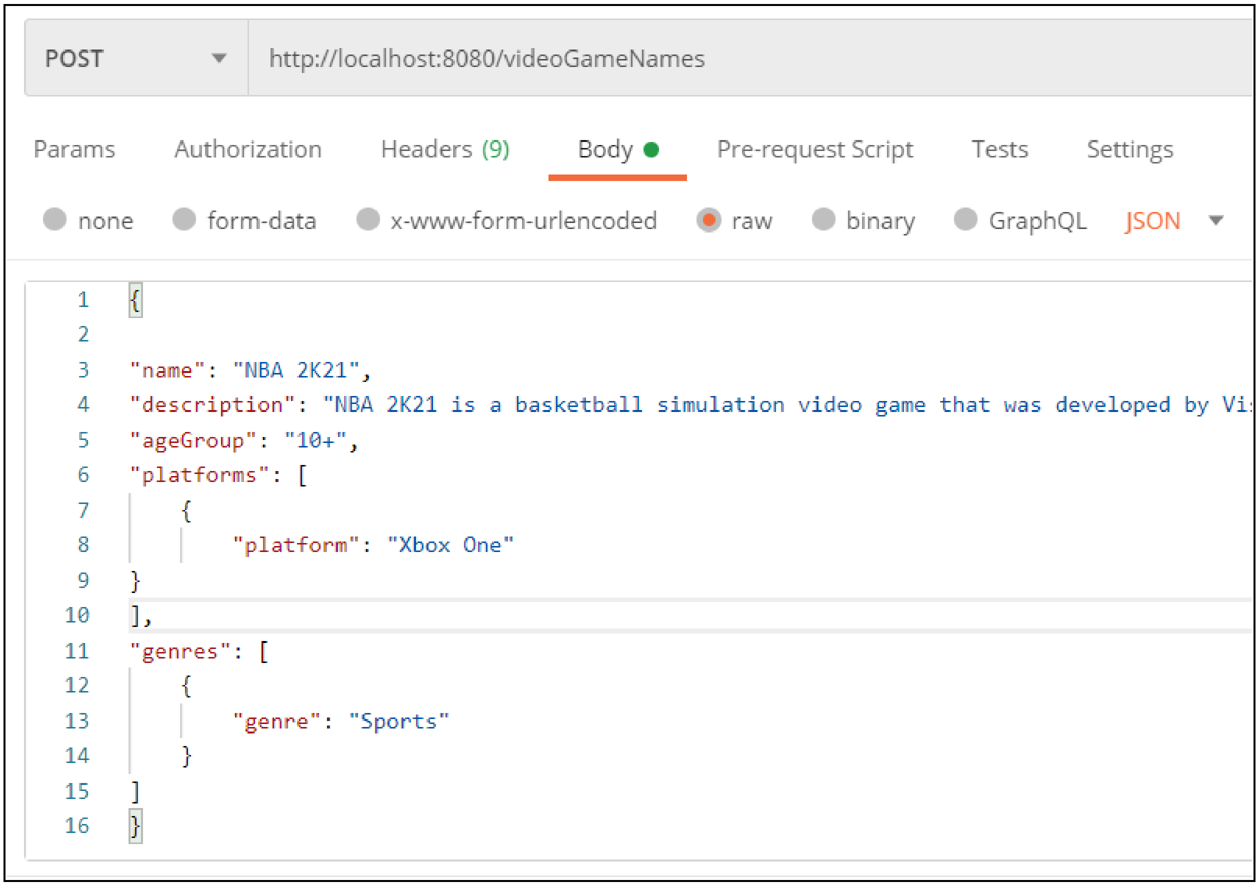
POST http://localhost:8080/users/register

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary

```
{

"username": "lucas89",
"password": "12345"

}
```

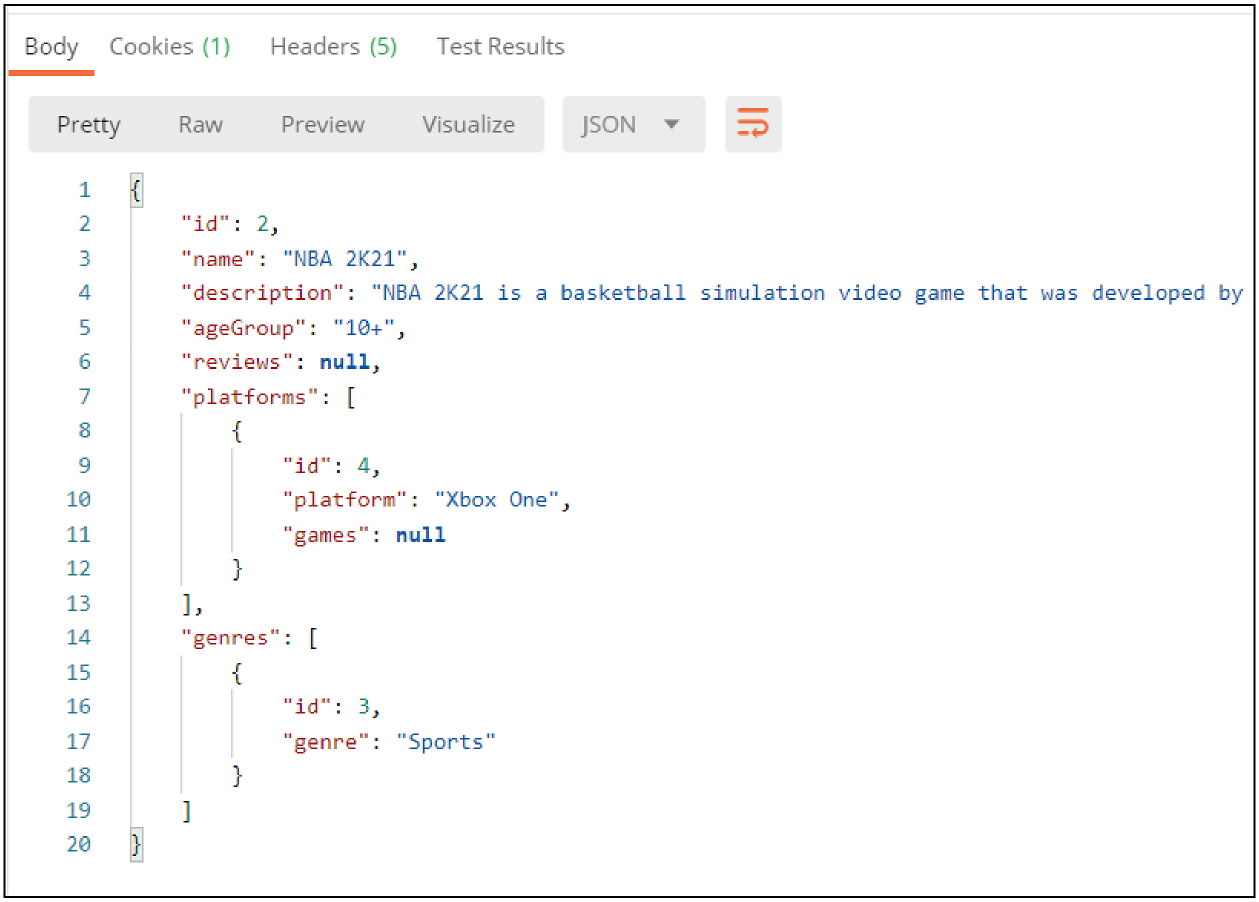
JSON

```
{
    "id": 1,
    "username": "lucas89"
}
```

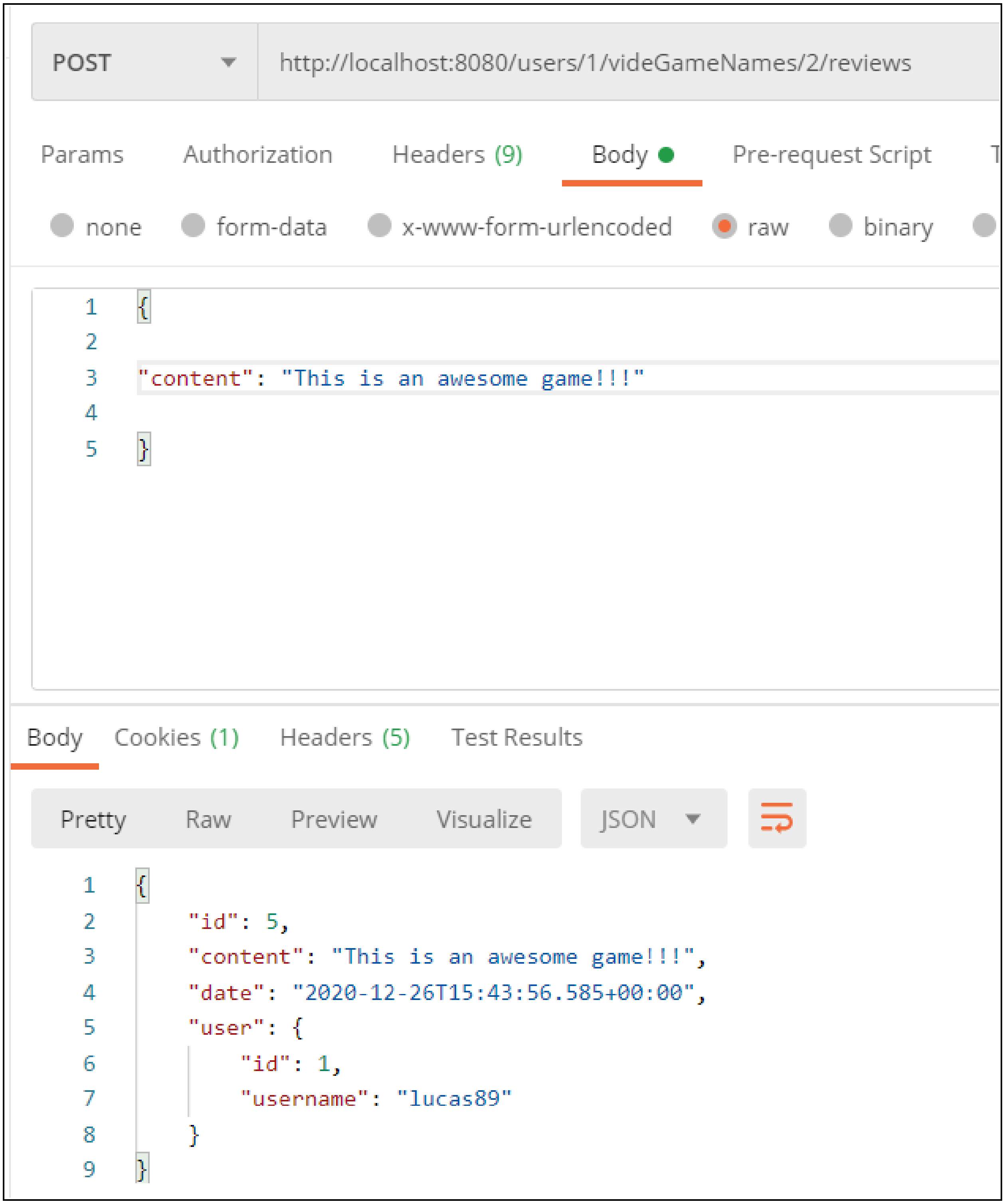


POST http://localhost:8080/videoGameNames

```
{

"name": "NBA 2K21",
"description": "NBA 2K21 is a basketball simulation video game that was developed by Vi
"ageGroup": "10+",
"platforms": [
    {
        "platform": "Xbox One"
}
],
"genres": [
    {
        "genre": "Sports"
    }
]
}
```



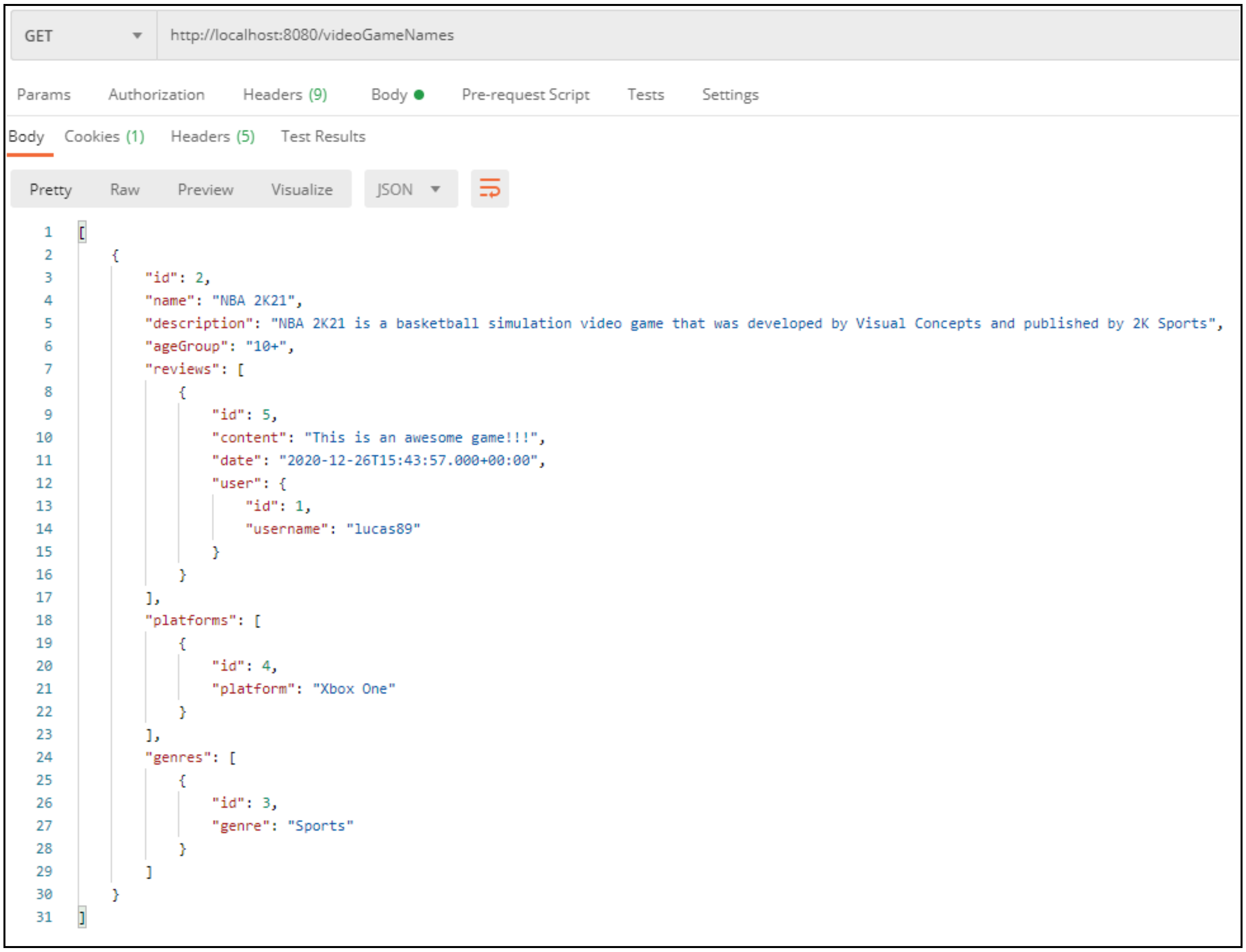
```
{
    "id": 2,
    "name": "NBA 2K21",
    "description": "NBA 2K21 is a basketball simulation video game that was developed by
    "ageGroup": "10+",
    "reviews": null,
    "platforms": [
        {
            "id": 4,
            "platform": "Xbox One",
            "games": null
        }
    ],
    "genres": [
        {
            "id": 3,
            "genre": "Sports"
        }
    ]
}
```



POST http://localhost:8080/users/1/videGameNames/2/reviews

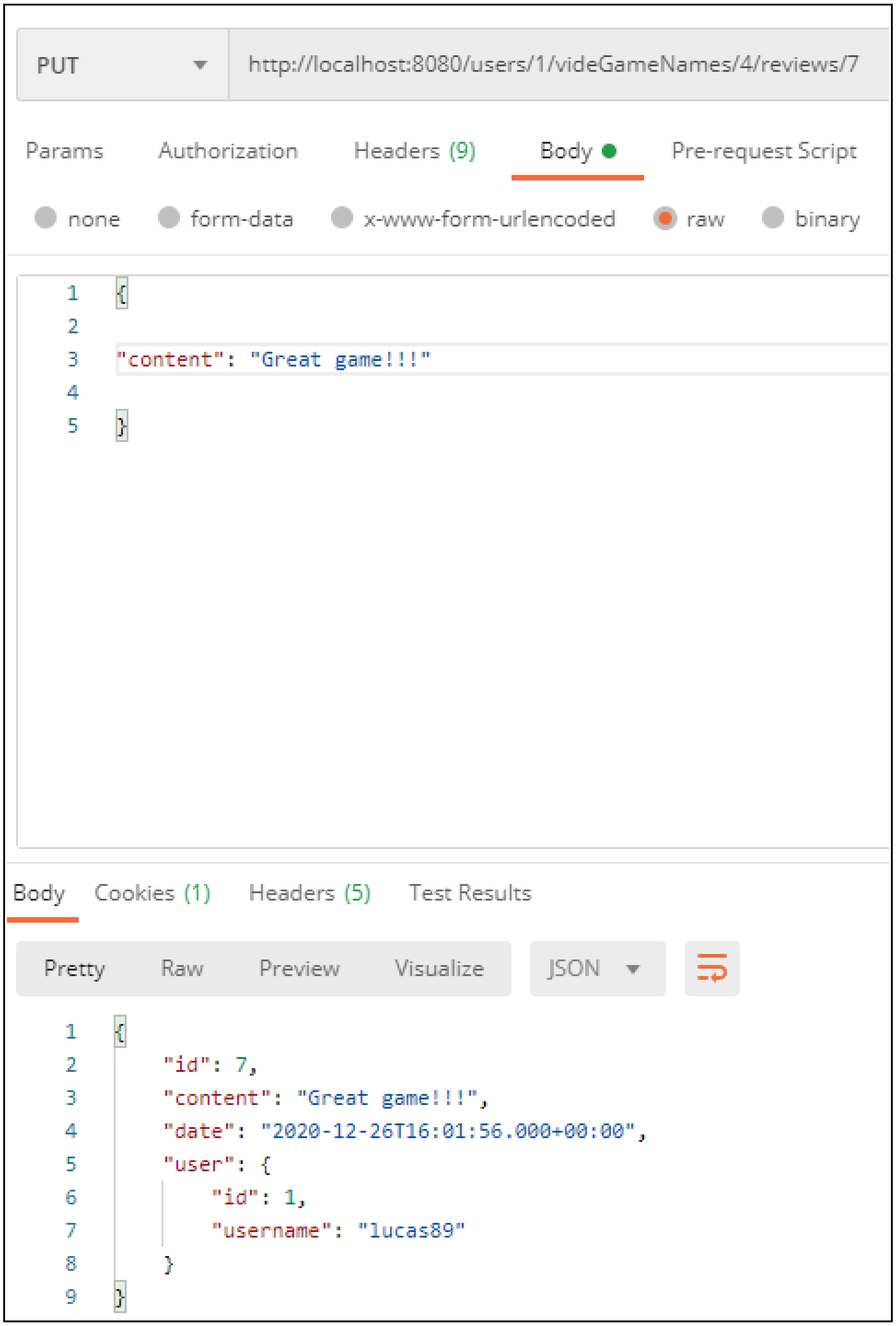
```
{

"content": "This is an awesome game!!!"

}
```

```
{
    "id": 5,
    "content": "This is an awesome game!!!",
    "date": "2020-12-26T15:43:56.585+00:00",
    "user": {
        "id": 1,
        "username": "lucas89"
    }
}
```



GET http://localhost:8080/videoGameNames

```json
[
    {
        "id": 2,
        "name": "NBA 2K21",
        "description": "NBA 2K21 is a basketball simulation video game that was developed by Visual Concepts and published by 2K Sports",
        "ageGroup": "10+",
        "reviews": [
            {
                "id": 5,
                "content": "This is an awesome game!!!",
                "date": "2020-12-26T15:43:57.000+00:00",
                "user": {
                    "id": 1,
                    "username": "lucas89"
                }
            }
        ],
        "platforms": [
            {
                "id": 4,
                "platform": "Xbox One"
            }
        ],
        "genres": [
            {
                "id": 3,
                "genre": "Sports"
            }
        ]
    }
]
```

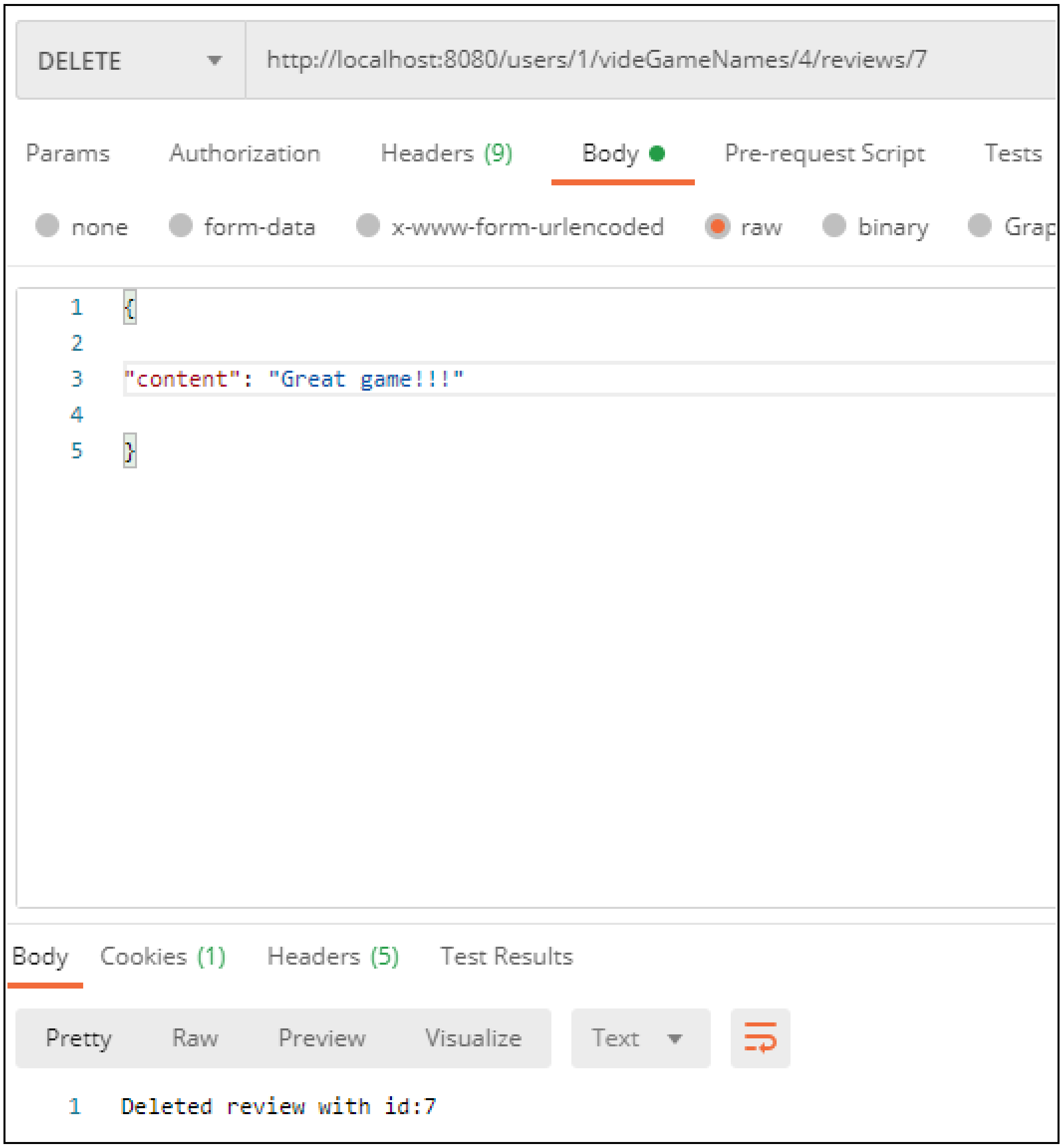


PUT http://localhost:8080/users/1/videGameNames/4/reviews/7

○ raw

```
{

"content": "Great game!!!"

}
```

```
{
    "id": 7,
    "content": "Great game!!!",
    "date": "2020-12-26T16:01:56.000+00:00",
    "user": {
        "id": 1,
        "username": "lucas89"
    }
}
```



DELETE http://localhost:8080/users/1/videGameNames/4/reviews/7

```
{

"content": "Great game!!!"

}
```

```
Deleted review with id:7
```

URL to GitHub Repository: