# How to setup the express server

1. Execute the db\_initialize.sql file in SQL files folder.
2. Run server.js in Frontend
3. Run server.js in Backend

# DB layout

User

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Null constrain** | **Description** |
| userid | Int | Not null | Identifies the row |
| Username | Varchar 100 | Not null | Username |
| Email | Varchar 100 | Not null | Email of user |
| Type | Varchar 100 | Not null | Type of user |
| Password | Varchar 100 | Not null | Password of user |
| Profile\_pic\_url | Varchar 200 | Not null | Profile pic url of user |
| Created\_at | timestamp | Not null | Insert current time |

Game

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Null constrain** | **Description** |
| Gameid | Int | Not null | Identifies the row |
| Title | Varchar 50 | Not null | Title of the game |
| Description | Varchar 200 | Not null | Description of game |
| Year | Int 4 | Not null | Year of game |
| Image\_url | Varchar 200 | Not null | Image of game |
| Created\_at | Timestamp | Not null | Insert current time |

gameprices

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Null constrain** | **Description** |
| Priceid | Int | Not null | Identifies the row |
| Gameid | Int | Not null | Gameid of game |
| Price | Float | Not null | Price of game |
| Platformid | Int | Not null | Platformid of game |

Category

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Null constrain** | **Description** |
| Catid | Int | Not null | Identifies the row |
| Catname | Char 100 | Not null | Name of category |
| Description | Char 200 | Not null | Description of category |

Platform

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Null constrain** | **Description** |
| Platformid | Int | Not null | Identifies the row |
| Platform\_name | Varchar 20 | Not null | Name of platform |
| Description | Varchar 100 | Not null | Description of platform |

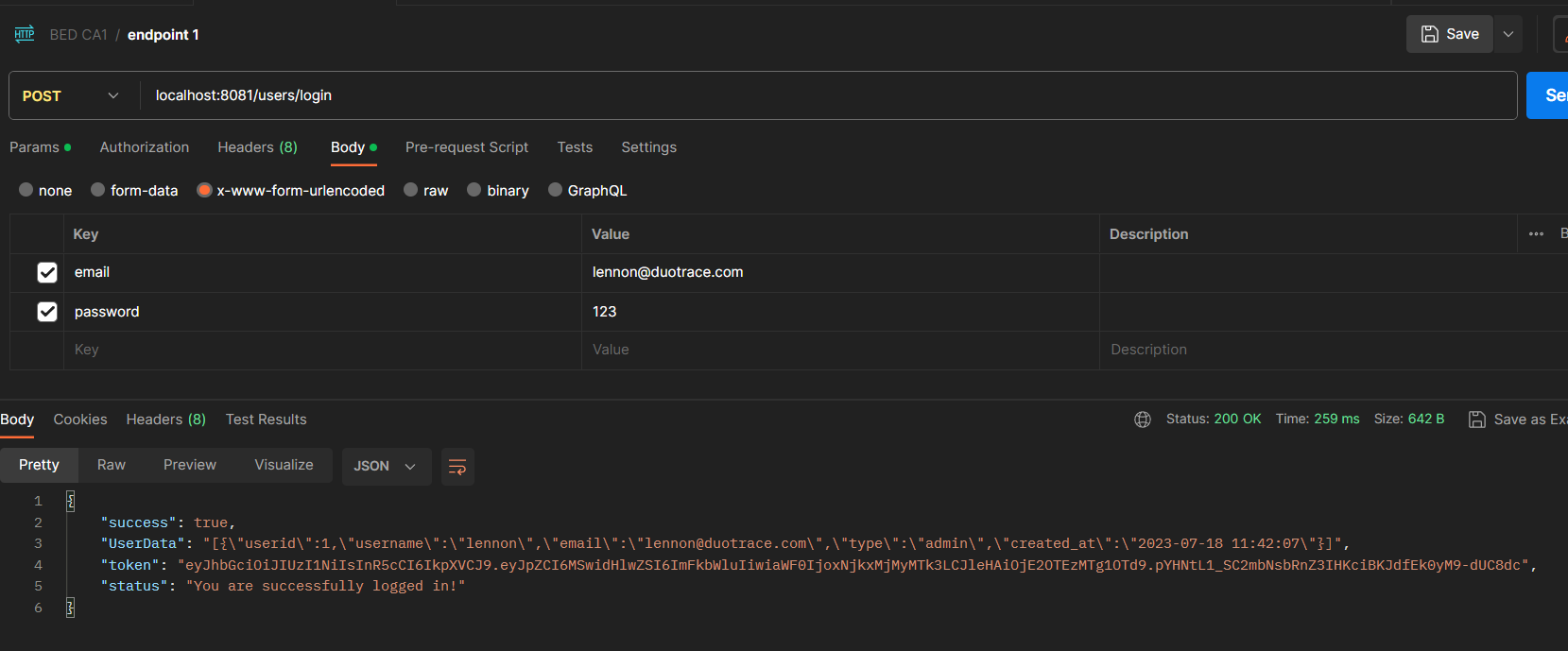
Review

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Null constrain** | **Description** |
| Reviewid | Int | Not null | Identifies the row |
| Gameid | Int | Not null | Gameid of game |
| Userid | Int | Not null | Userid of user |
| Content | Varchar 200 | Not null | Content of review |
| Created\_at | Timestamp | Not null | Insert current time |
| Rating | Int 1 | Not null | Rating value |

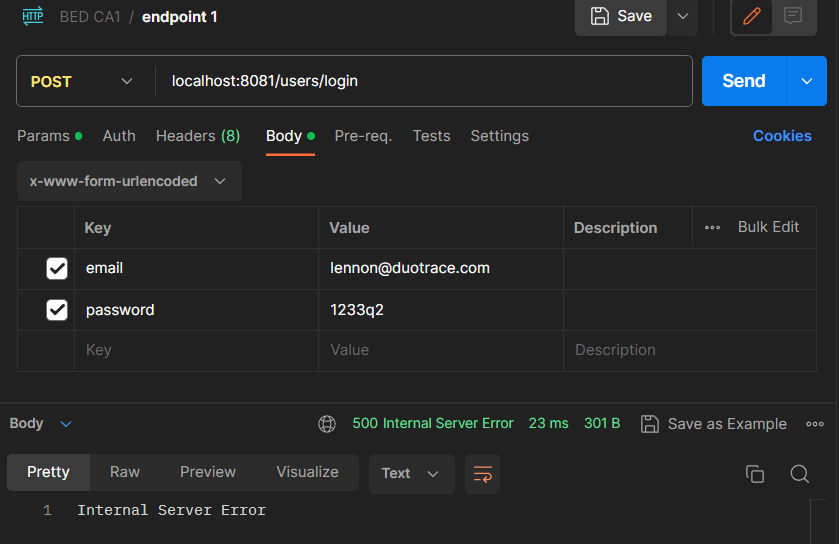
Gamecategory

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Null constrain** | **Description** |
| Id | Int | Not null | Identifies the row |
| Gameid | Int | Not null | Gameid of game |
| Catid | Int | Not null | Catid of category |

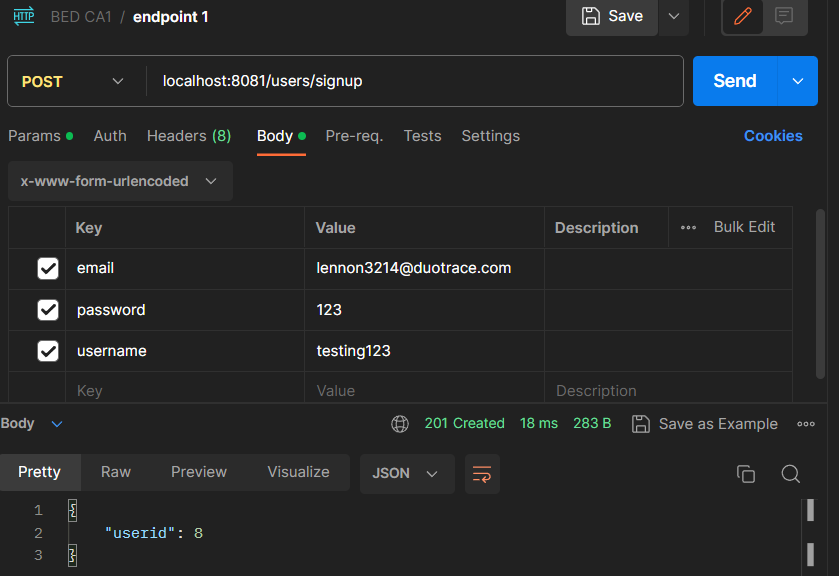
# Login API /users/login PASS



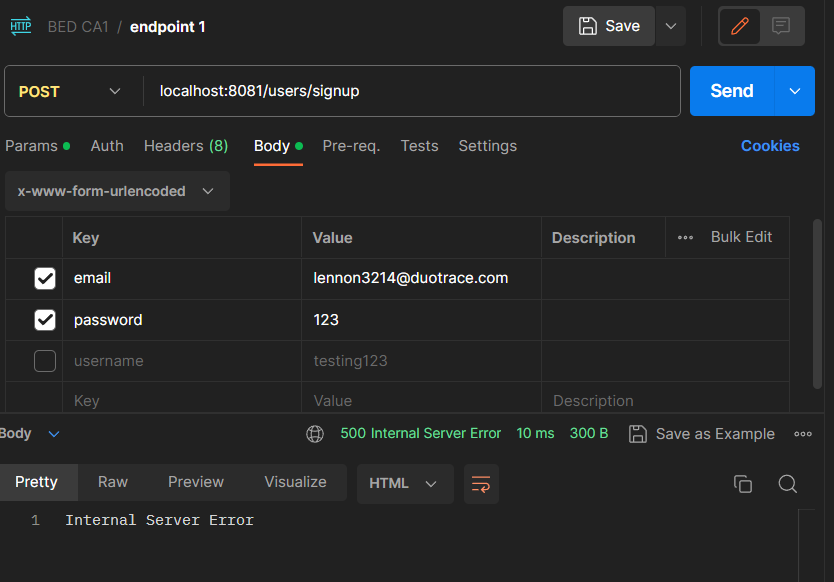
# Login API /users/login FAIL

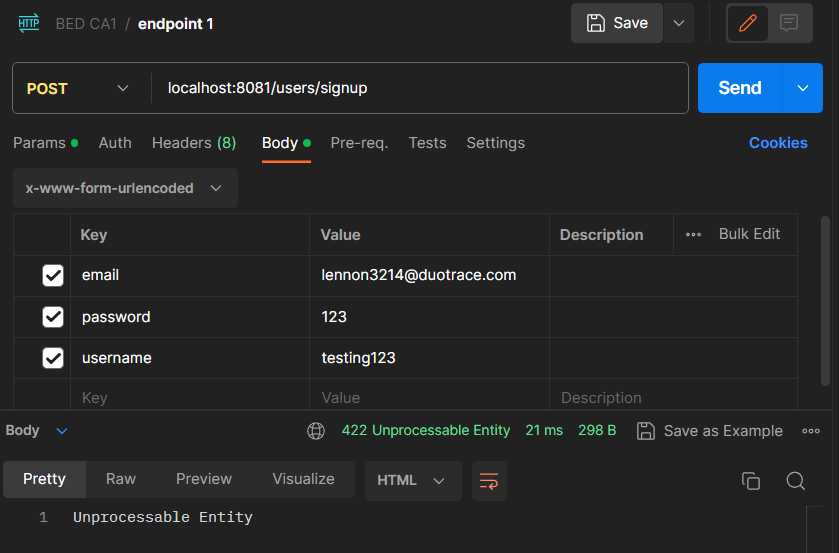


# Registration API /users/signup PASS

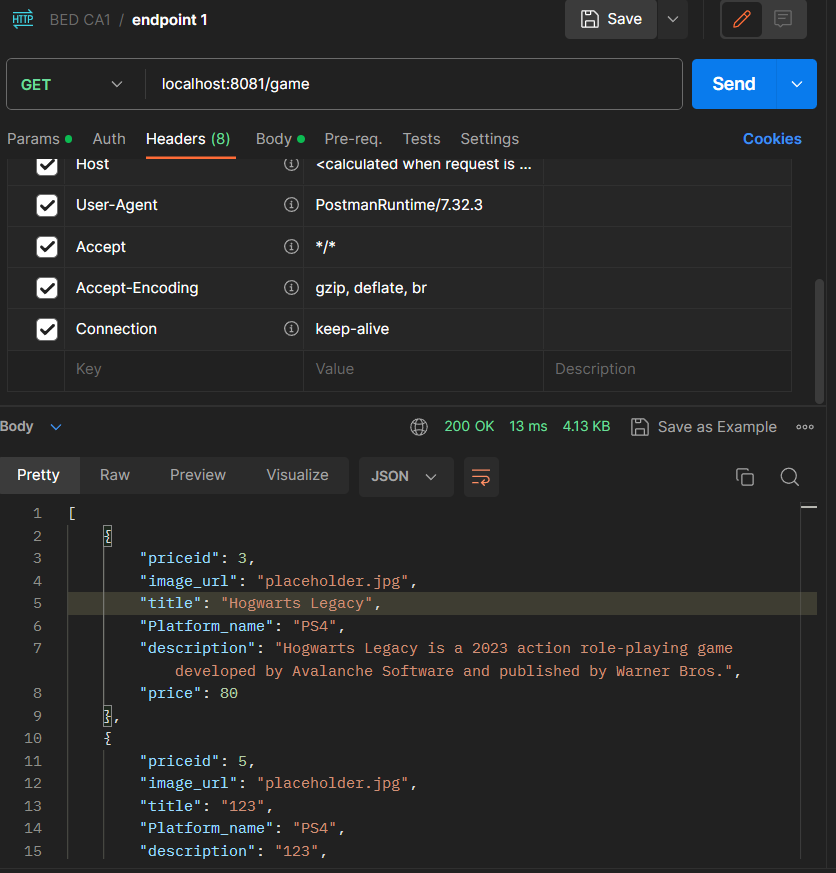


# Registration API /users/signup FAIL

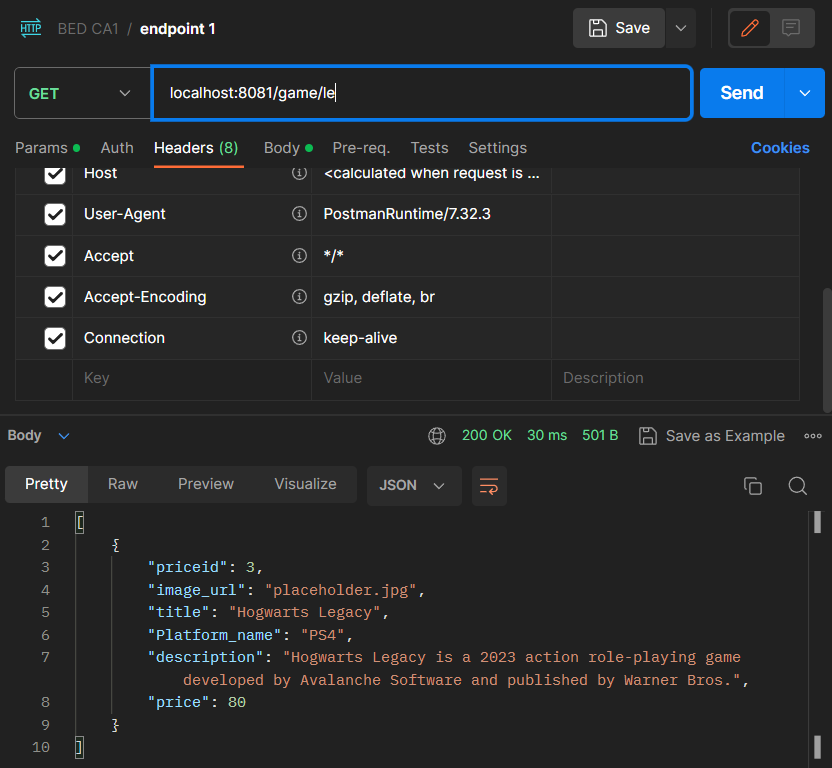


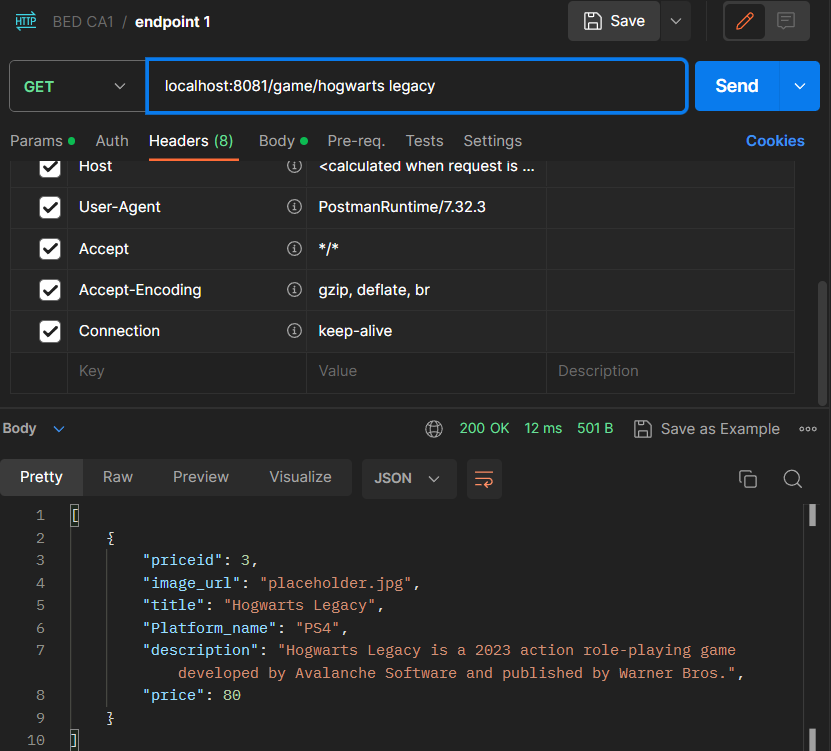


# Loading games API /game

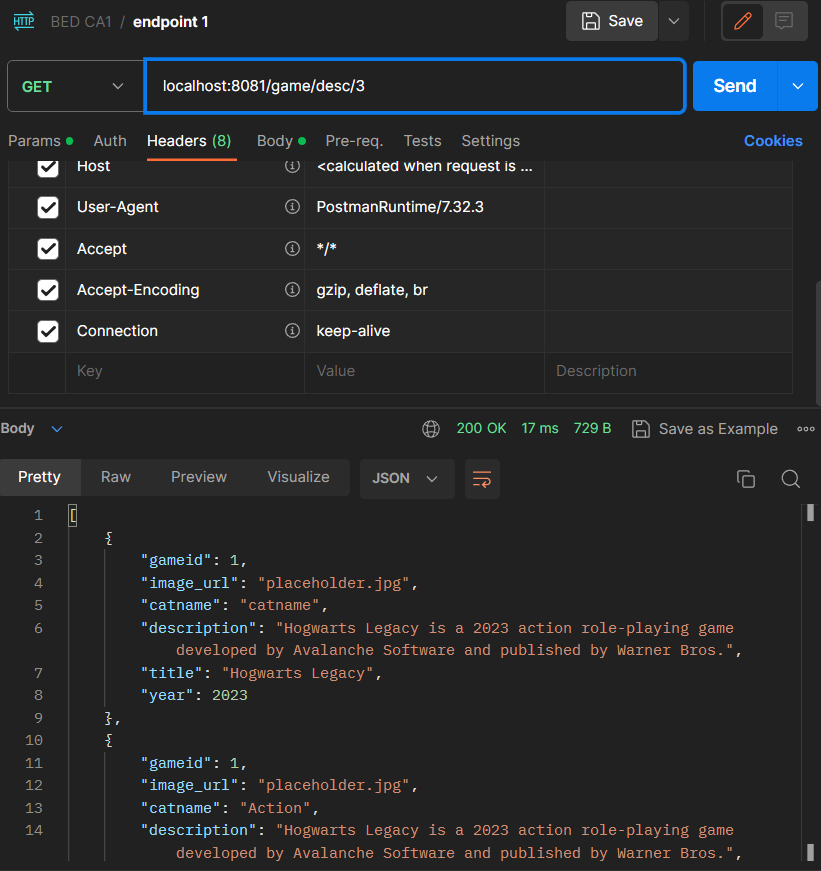


Searching for games API /game/:search PASS

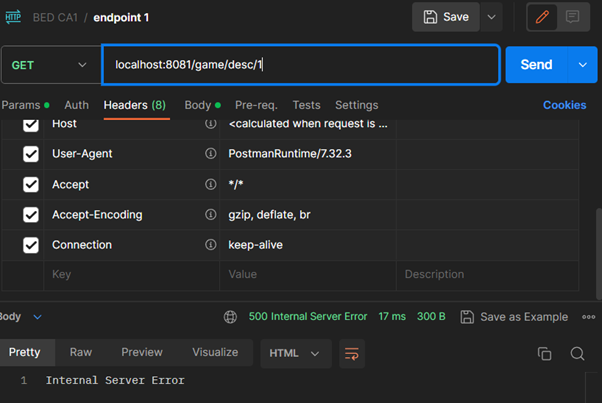




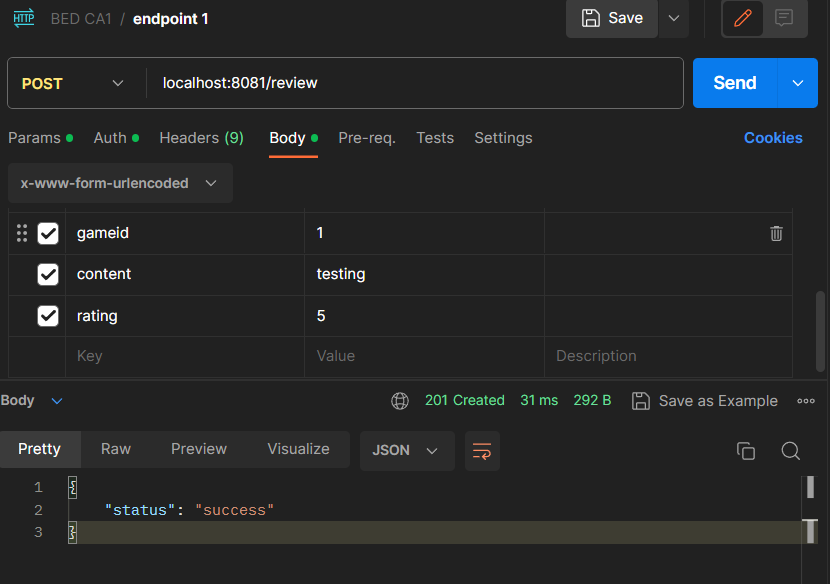
# Getting game details from server API /game/desc/:priceid PASS



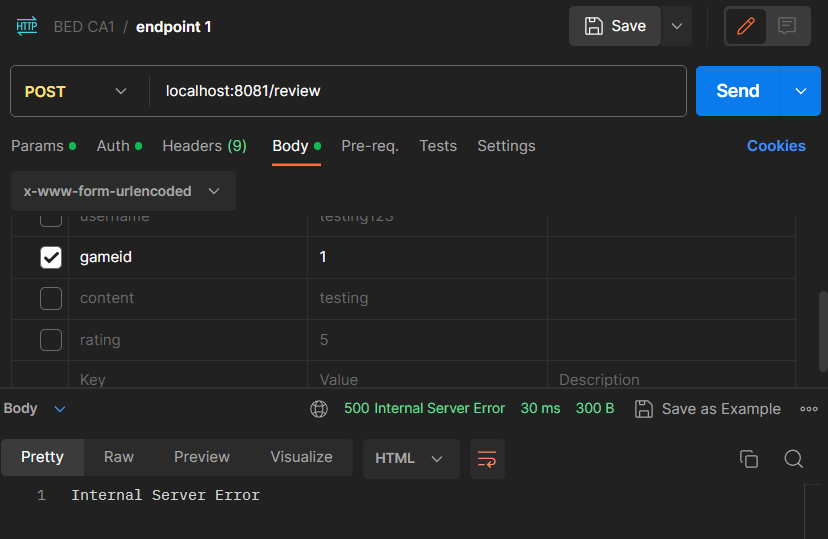
# Getting game details from server API /game/desc/:priceid FAIL



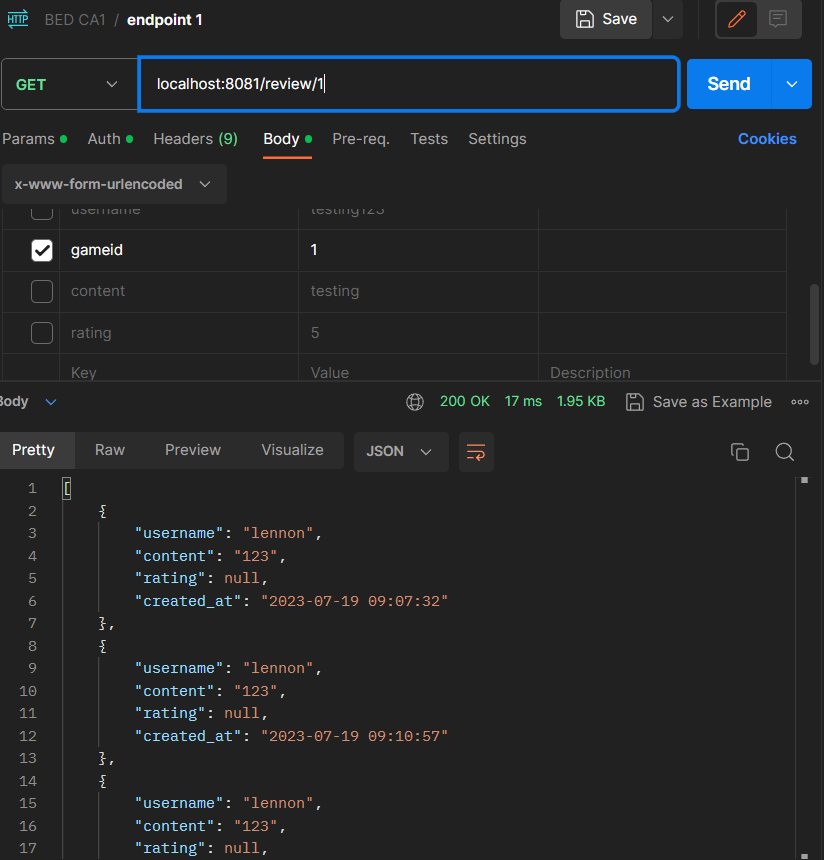
# Sending review API /review PASS



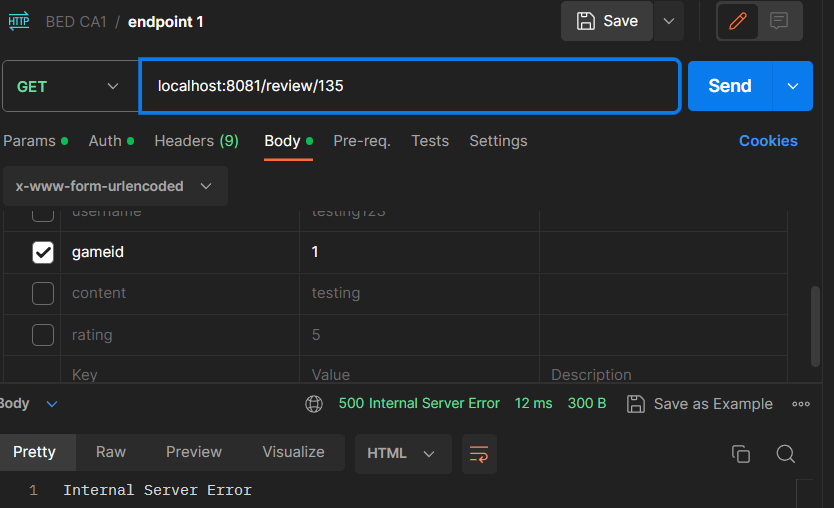
# Sending review API /review FAIL



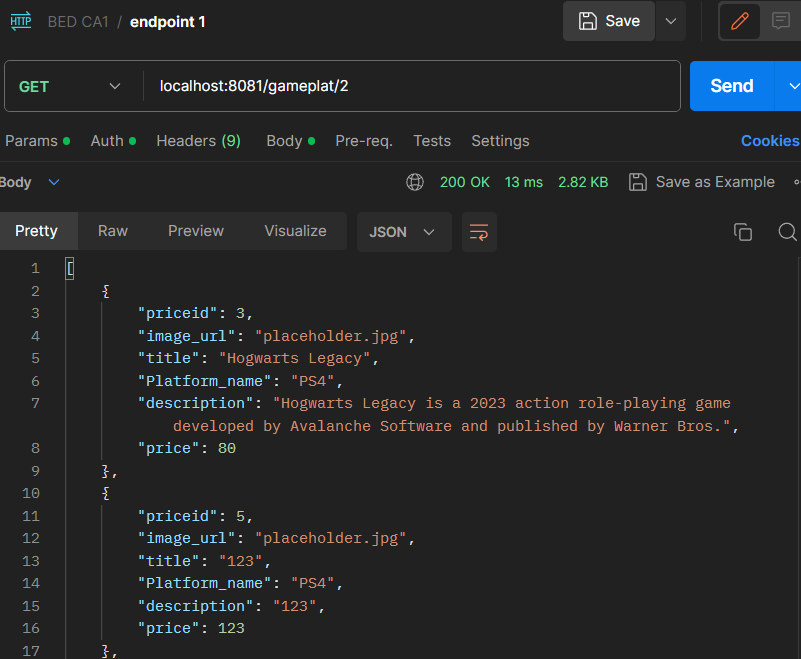
# Getting review based on gameid API /review/:gid PASS



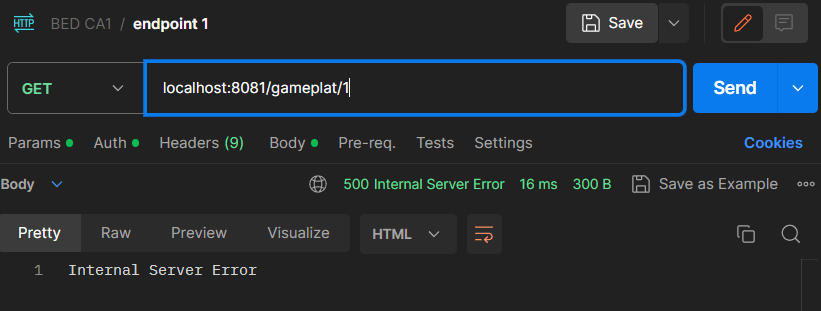
# Getting review based on gameid API /review/:gid FAIL



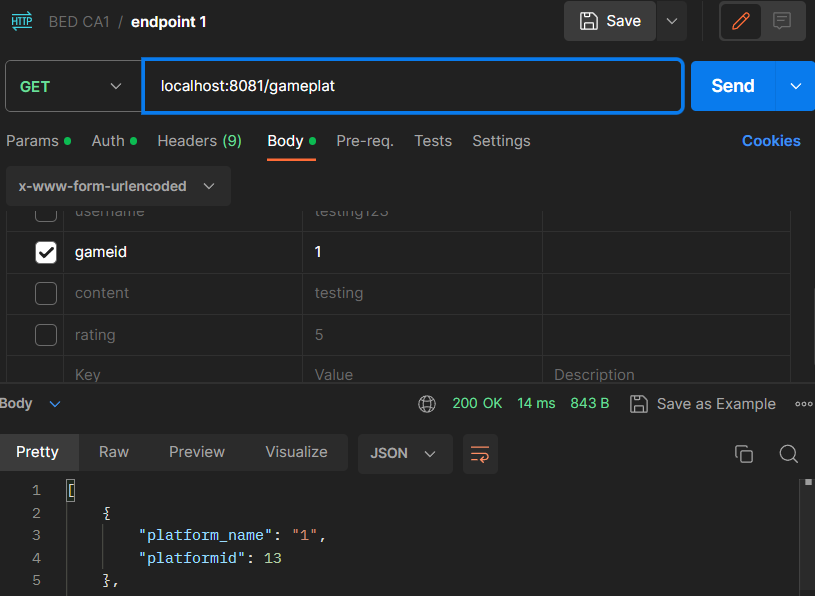
# Getting game based on platform id API /gameplat/:pid PASS



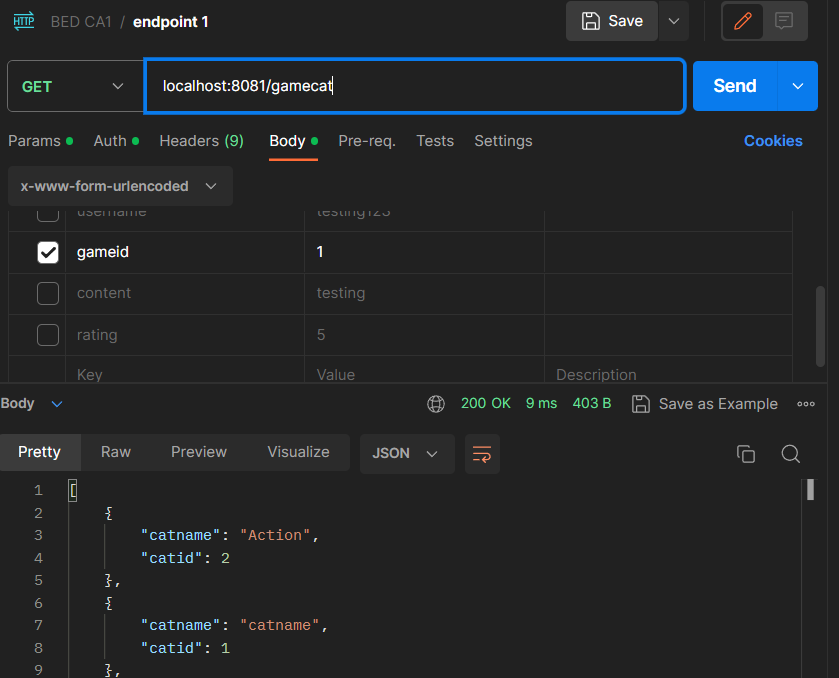
# Getting game based on platform id API /gameplat/:pid FAIL



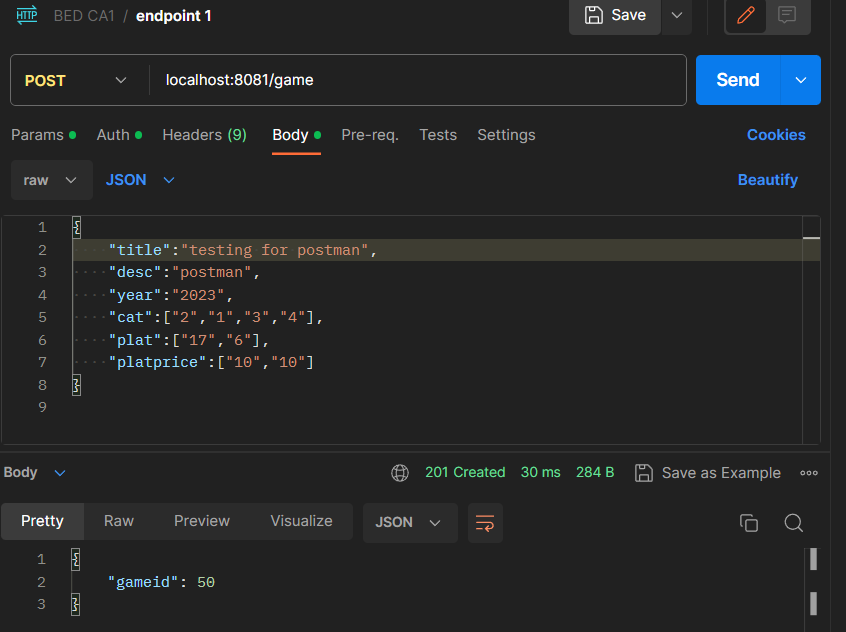
# Getting platform name and platformid API /gameplat PASS



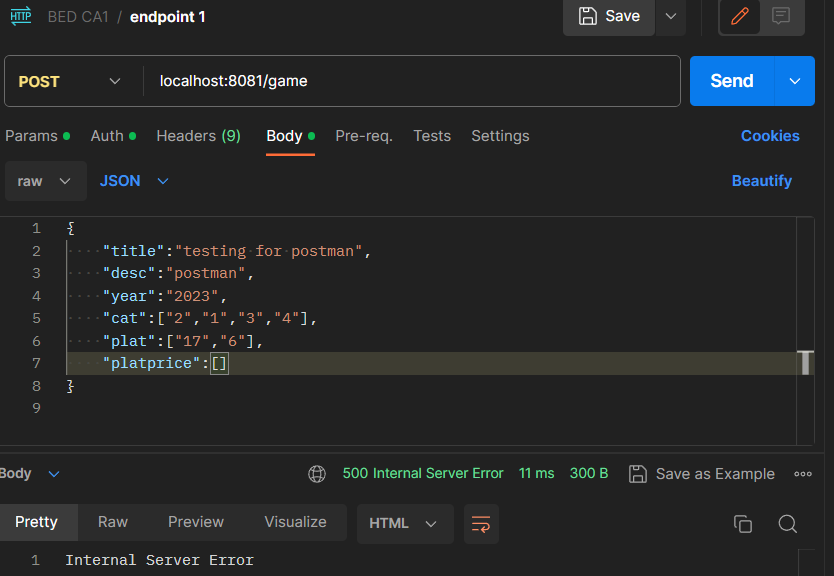
# Getting game category API /gamecat PASS



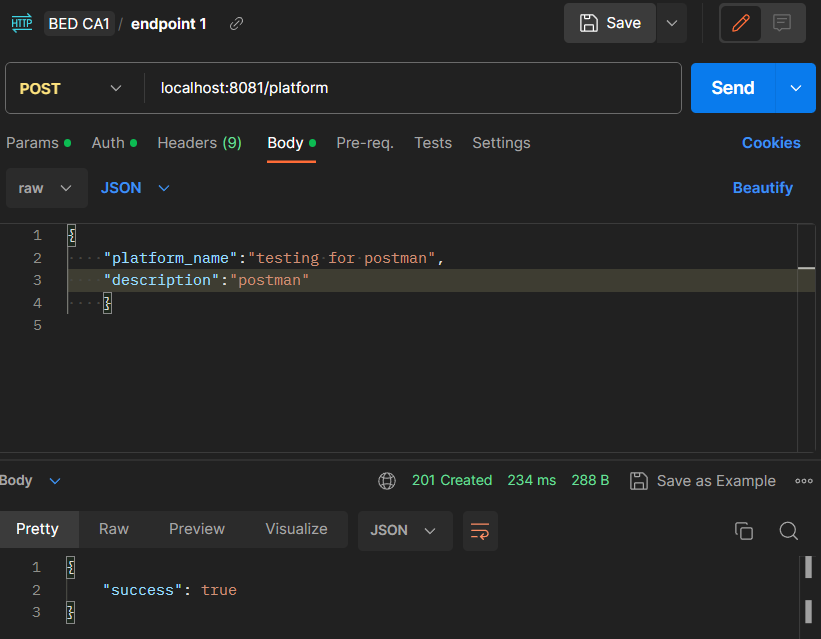
# Uploading game API /game PASS



# Uploading game API /game FAIL



# Uploading new platform API /platform PASS



# Uploading game API /platform FAIL

