

GamerLand MVP Endpoints

- User Registration: POST /api/users
- User Login: POST /api/users/login
- User Logout: POST /api/users/logout
- Game Search: GET /api/games
- Game Details: GET /api/games/{gameId}
- Review Submission: POST /api/games/{gameId}/reviews
- Review Update: PUT /api/games/{gameId}/reviews/{reviewId}
- Review Deletion: DELETE /api/games/{gameId}/reviews/{reviewId}
- Rating Submission: POST /api/games/{gameId}/ratings
- Rating Update: PUT /api/games/{gameId}/ratings/{ratingId}
- Rating Deletion: DELETE /api/games/{gameId}/ratings/{ratingId}

Personas & Endpoint Usage

Alex:

- Game Search: To allow Alex to filter games by genre and sort by rating, so he can find the best games to play in his preferred genre.
- Game Details: To provide Alex with the necessary information about the game, including its rating, to decide if he wants to play it.
- Review Submission/Update/Deletion: To allow Alex to share his opinion about a game with other users or update his review if necessary.
- Rating Submission/Update/Deletion: To allow Alex to rate a game and update his rating if he changes his mind.

Sarah:

- Game Search: To allow Sarah to filter games by family-friendly or educational categories, so she can find games that she can play with her son.
- Game Details: To provide Sarah with the necessary information about the game, including its rating and reviews from other parents, to decide if it's appropriate for her and her son.
- Review Submission/Update/Deletion: To allow Sarah to share her opinion about a game with other users or update her review if necessary.
- Rating Submission/Update/Deletion: To allow Sarah to rate a game and update her rating if she changes her mind.

Gabriel:

- Game Search: To allow Gabriel to filter games by multiplayer category and sort by popularity and rating, so he can find the best multiplayer games to play with his friends.
- Game Details: To provide Gabriel with the necessary information about the game, including its rating and reviews from other competitive players, to decide if it's suitable for him and his friends.
- Review Submission/Update/Deletion: To allow Gabriel to share his opinion about a game with other users or update his review if necessary.
- Rating Submission/Update/Deletion: To allow Gabriel to rate a game and update his rating if he changes his mind.

Request/Response Examples (Incl. Error Examples)

User Registration: POST /api/users

REQUEST:

POST/api/users HTTP/1.1

Host: gamerland.com

Content-Type: application/json

```
{
  "username": "patrickflood",
  "email": "patrickflood@email.com",
  "password": "password123",
  "confirmPassword": "password123"
}
```

RESPONSE:

HTTP/1.1 201 Created

Content-Type: application/json

```
{
  "id": "abcdefg123456lskjdfasldjfl",
  "username": "patrickflood",
  "email": "patrickflood@email.com"
}
```

ERROR REQUEST:

POST /api/users HTTP/1.1

Host: gamerland.com

Content-Type: application/json

```
{
  "username": "johndoe",
  "password": "password123"
}
```

```
}
```

ERROR RESPONSE:

HTTP/1.1 400 Bad Request
Content-Type: application/json

```
{  
  "error": "Missing required fields"  
}
```

User Logout: POST /api/users/logout

REQUEST:

POST /api/users/logout HTTP/1.1
Host: gamerland.com
Authorization: Bearer abc123abc1223jfia;sljfal;sdjf
Content-Type: application/json

```
{  
  "userId": "12345"  
}
```

RESPONSE:

HTTP/1.1 200 OK
Content-Type: application/json
{
 "message": "User successfully logged out."
}

ERROR REQUEST:

HTTP/1.1 401 Unauthorized
Content-Type: application/json

```
{  
  "error": "Unauthorized",  
  "message": "You are not authorized to perform this action."  
}
```

ERROR RESPONSE:

POST /api/users/logout HTTP/1.1
Host: gamerland.com
Content-Type: application/json

```
{  
  "userId": "12345"
```

```
}  
*missing auth header
```

Game Search: GET /api/games

REQUEST:

GET /api/games?genre=action&platform=PS4&sort=rating

RESPONSE: HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "games": [  
    {  
      "id": "123",  
      "title": "Example Game 1",  
      "genre": "Action",  
      "platforms": ["PS4"],  
      "rating": 4.5  
    },  
    {  
      "id": "456",  
      "title": "Example Game 2",  
      "genre": "Action",  
      "platforms": ["PS4"],  
      "rating": 4.0  
    }  
  ]  
}
```

ERROR REQUEST:

GET /api/games?genre=invalid_genre&platform=PS4&sort=rating

ERROR RESPONSE:

HTTP/1.1 400 Bad Request

Content-Type: application/json

```
{  
  "error": "Invalid genre provided."  
}
```

Game Details: GET /api/games/{gameId}**REQUEST:**

GET /api/games/1234 HTTP/1.1
Host: gamerland.com
Content-Type: application/json
Authorization: Bearer <access_token>

RESPONSE:

```
{
  "gameId": 1234,
  "title": "The Legend of Zelda: Breath of the Wild",
  "description": "The story follows Link, who awakens from a hundred-year slumber to a mysterious voice that guides him to defeat Calamity Ganon.",
  "releaseDate": "2017-03-03",
  "genres": ["Action-adventure", "Open world"],
  "platforms": ["Nintendo Switch", "Wii U"],
  "publisher": "Nintendo",
  "developer": "Nintendo EPD",
  "rating": 4.9,
  "reviews": [
    {
      "userId": 5678,
      "username": "gamer123",
      "rating": 5,
      "comment": "This game is amazing! Highly recommend it to any Zelda fan.",
      "date": "2022-01-01"
    },
    {
      "userId": 9012,
      "username": "linkfanboi3000",
      "rating": 4,
      "comment": "Great game, but some of the puzzles were a bit too challenging for my taste.",
      "date": "2022-01-03"
    }
  ]
}
```

ERROR RESPONSE:

```
{
```

```
"error": "Game not found"
}
```

Review Submission: POST /api/games/{gameId}/reviews

REQUEST:

POST /api/games/123/reviews HTTP/1.1

Host: gamerland.com

Content-Type: application/json

Authorization: Bearer <access_token>

```
{
  "title": "Amazing game!",
  "body": "I loved everything about this game. The graphics were stunning, the
gameplay was smooth, and the story was captivating.",
  "rating": 5
}
```

RESPONSE:

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "id": 456,
  "title": "Amazing game!",
  "body": "I loved everything about this game. The graphics were stunning, the
gameplay was smooth, and the story was captivating.",
  "rating": 5,
  "user": {
    "id": 789,
    "name": "John Doe"
  },
  "created_at": "2023-03-23T10:30:00Z"
}
```

ERROR RESPONSE:

HTTP/1.1 400 Bad Request

Content-Type: application/json

```
{
```

```
"message": "Invalid review data",
"errors": [
  {"field": "title", "message": "Title is required"},
  {"field": "rating", "message": "Rating must be between 1 and 5"}
]
```

Review Update: PUT /api/games/{gameId}/reviews/{reviewId}

REQUEST:

PUT /api/games/123/reviews/456

Content-Type: application/json

```
{
  "title": "Updated Review Title",
  "content": "Updated review content"
}
```

RESPONSE:

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "id": 456,
  "title": "Updated Review Title",
  "content": "Updated review content",
  "rating": 4,
  "user": {
    "id": 789,
    "username": "example_user",
    "avatar": "https://example.com/avatar.jpg"
  },
  "createdAt": "2023-04-01T15:30:00Z",
  "updatedAt": "2023-04-02T09:45:00Z"
}
```

ERROR RESPONSE:

HTTP/1.1 400 Bad Request

Content-Type: application/json

```
{
  "error": "Missing required field: title"
}
```

Review Deletion: DELETE /api/games/{gameId}/reviews/{reviewId}

REQUEST:

DELETE /api/games/123/reviews/456 HTTP/1.1

Host: gamerland.com

Authorization: Bearer <access_token>

RESPONSE:

HTTP/1.1 204 No Content

ERROR REQUEST:

DELETE /api/games/123/reviews/invalid_review_id HTTP/1.1

Host: gamerland.com

Authorization: Bearer <access_token>

ERROR RESPONSE:

HTTP/1.1 404 Not Found

Content-Type: application/json

```
{  
  "error": "Review not found"  
}
```

Rating Submission: POST /api/games/{gameId}/ratings

REQUEST:

POST /api/games/1234/ratings HTTP/1.1

Host: gamerland.com

Content-Type: application/json

Authorization: Bearer {token}

```
{  
  "value": 8.5,  
  "comment": "This game is amazing!"  
}
```

RESPONSE:

HTTP/1.1 201 Created

Content-Type: application/json

```
{  
  "id": 5678,  
  "value": 8.5,
```



```
"comment": "This game is amazing!",  
"userId": 123,  
"gameId": 1234,  
"createdAt": "2023-04-01T12:00:00Z"  
}
```

ERROR RESPONSE:

HTTP/1.1 400 Bad Request
Content-Type: application/json

```
{  
  "error": "Invalid request",  
  "message": "The 'value' field is required and must be a number between 1 and  
  10."  
}
```

Rating Update: PUT /api/games/{gameId}/ratings/{ratingId}

REQUEST:

PUT /api/games/1234/ratings/5678 HTTP/1.1
Host: gamerland.com
Content-Type: application/json
Authorization: Bearer <access_token>

```
{  
  "rating": 9  
}
```

RESPONSE:

HTTP/1.1 200 OK
Content-Type: application/json

```
{  
  "id": 5678,  
  "gameId": 1234,  
  "userId": 9876,  
  "rating": 9,  
  "createdAt": "2022-04-01T10:00:00Z",  
  "updatedAt": "2022-04-01T11:00:00Z"  
}
```

ERROR REQUEST:

PUT /api/games/1234/ratings/5678 HTTP/1.1
Host: gamerland.com

Content-Type: application/json
Authorization: Bearer <access_token>

```
{  
  "rating": "invalid"  
}
```

ERROR RESPONSE:

HTTP/1.1 400 Bad Request
Content-Type: application/json

```
{  
  "error": "Invalid rating value"  
}
```

Rating Deletion: DELETE /api/games/{gameId}/ratings/{ratingId}

REQUEST:

DELETE /api/games/123/ratings/456 HTTP/1.1
Host: gamerland.com
Authorization: Bearer <access_token>

RESPONSE:

DELETE /api/games/123/ratings/456 HTTP/1.1
Host: gamerland.com
Authorization: Bearer <access_token>

ERROR REQUEST:

DELETE /api/games/123/ratings/789 HTTP/1.1
Host: gamerland.com
Authorization: Bearer <access_token>

ERROR RESPONSE:

HTTP/1.1 404 Not Found
Content-Type: application/json

```
{  
  "error": "Rating with ID 789 not found for game 123"  
}
```

Diagram

The user interface pages will make requests to the GamerLand APIs to communicate with RAWG.io APIs, and OpenVGDB APIs depending on the user's actions.

The GamerLand server will receive requests from the user interface pages and handle them, either by processing them internally or forwarding them to the appropriate external APIs (RAWG.io or OpenVGDB).

The RAWG.io server and OpenVGDB server will receive requests from the GamerLand server and process them, returning results to the GamerLand server.

The GamerLand server will receive the results from the RAWG.io server and OpenVGDB server and use them to respond to the user interface pages.

The GamerLand server will also interact with the database to store and retrieve data related to and RELEVANT to the user, games, reviews, ratings, etc.

