FarmClash - API Blueprint for Web-based Idle Game

Camille De Vuyst, Joren Van der Sande, Thomas De Volder, Faisal Ettarrahi, Ferhat Van Herck, Siebe Mees

June 4, 2024

Contents

1 Introduction

This document outlines the API endpoints for a web-based idle game. Each endpoint requires user authentication and certain routes are restricted to admin users only.

1.1 Base URL

The base URL for all API requests is `https://team3.ua-ppdb.me/api/`.

2 Users

- Endpoint: `/users`
- Method: GET
- Description: Get a look at all the users.
- Access: Requires `admin` user privileges.
- Response: A JSON array of user objects, each containing:

```
[
  {
    "created_at": "Date",
    "email": "email@example.com",
    "password": "pbkdf2:sha256:hasedpassword",
    "username": "username"
  },
  {
    "created_at": "Another date",
    "email": "anotheremail@example.com",
    "password": "pbkdf2:sha256:anotherhashedpassword",
    "username": "anotherusername"
  }
  // Additional user objects as needed
]
```

2.1 User statistics

- Endpoint: `/get-user-stats`
- Method: GET
- Description: Get a look at stats of the current user.
- Access: Requires `logged in` privileges.
- Response: A JSON array of user stats like level, attack, defense, coins, each containing:

```
{
    "level": level_of_user_farm, ...//Townhall level
    "attack": attack_rating_of_user_farm,
    "defense": defense_rating_of_user_farm,
    "coins": coins_amount
}
```

3 Maps

- Endpoint: `/maps`
- Method: GET
- Description: Get a look at all the maps.
- Access: Requires `admin` user privileges.
- Response: A JSON array of map objects, each containing:

```
[
  {
    "map_id": 1,
    "username_owner": "ownerUsername",
    "width": 100,
    "height": 100,
    "created_at": "Tue, 26 Mar 2024 13:55:30 GMT"
  },
  {
    "map_id": 2,
    "username_owner": "anotherOwnerUsername",
    "width": 150,
    "height": 150,
    "created_at": "Wed, 27 Mar 2024 14:00:00 GMT"
  }
  // Additional map objects as needed
]
```

4 Resources

4.1 General Resource Information

- Endpoint: `/resources`
- Method: GET
- Description: Retrieves a comprehensive list of resources across all users. Designed for administrative use to monitor and manage resources system-wide.
- Access: Admin privileges required.
- Response: A JSON array of objects, each representing a resource. Each object includes the resource type, amount, owner's username, and a unique resource identifier. Example of a generalized response:

```
[
  {
    "owner": "resource_owner_username",
    "resource_type": "type_of_resource",
    "amount": "quantity_of_resource",
    "last_updated": "date_of_last_idle_update"
  },
  ... // Additional resource objects
]
```

4.2 Specific Resource Information

- Endpoint: `/resources/<string:username>`
- Method: GET
- Description: Retrieves a list of resources owned by a specific user.
- Access: Admin privileges required.
- Response: A JSON array of objects, each representing a resource owned by the specified user. Each object includes the resource type, amount, and an owner username and when it was last updated by idle.js. Example of a generalized response:

```
[
  {
    "owner": "resource_owner_username",
    "resource_type": "type_of_resource",
    "amount": "quantity_of_resource",
    "last_updated": "date_of_last_idle_update"
  },
  ... // Additional resource objects
]
```

4.3 General Resource adding Information

- Endpoint: `/add-resources`
- Method: POST
- Description:list of resources for a user. This endpoint is used to add a quantity of animals owned by a user.
- Access: Requires logged-in user privileges.

- Request: A JSON object with general information about the update. The object include as key the Resources and as value the amount of the resource to be added (subtracted in the case of a negative value). The key 'idle' with value true can be added to update the last$_updated, forthehourlyproductionofresources.Hereis2examples.$

  ```
  {
      "Zucchini": 20,
      "Money": 20,
      "Egg": 20,
      "Wheat": 20,
  }
  {
      "idle": true,
      "Zucchini": 20,
      "Money": 20,
      "Egg": 20,
      "Wheat": 20,
  }
  ```

Response: A JSON object indicating the status of the add operation.

```
{
  "status": "success/error",
  "message": "Description of the outcome of the operation"
}
```

5 Terrain Map

5.1 General Terrain Map Information

- Endpoint: `/terrain-map`
- Method: GET
- Description: Get a look at all the terrain tiles in the map.

- Access: Requires `logged in` user privileges.
- Response: A JSON array of tile objects, each containing:

```
[
  {
    "map_height": 2,
    "map_width": 2,
    "terrain_tiles": [
      [Water1.1, Water1.2],
      [Water2.1, Water2.2]
    ]
  }
]
```

5.2 Specific Terrain Tile Information

- Endpoint: `/terrain-map/<string:friend_username>`
- Method: GET
- Description: Get a look at a user specific terrain tile in the map.
- Access: Requires `logged in` user privileges.
- Response: A JSON object containing the terrain tile information:

```
[
  {
    "map_height": 2,
    "map_width": 2,
    "terrain_tiles": [
      [Water1.1, Water1.2],
      [Water2.1, Water2.2]
    ]
  }
]
```

6 Friendships

- Endpoint: `/friends`
- Method: GET
- Description: Get a list of all friends for the current user.
- Access: Requires `logged in` user privileges.
- Response: A JSON array of relationship objects, each containing:

```
[
  "friend1",
  "friend2",
  "friend3"
]
```

7 Chat

- Endpoint: `/messages/<string:friend_name>`
- Method: GET
- Description: Get a list of all chat messages for the current user.
- Access: Requires `logged in` user privileges.
- Response: A JSON array of chat message objects, each containing:

```
[
  {
    message_id: 1,
    "sender": "senderUsername",
    "receiver": "receiverUsername",
    "message": "Hello, how are you?",
    "created_at": "Tue, 26 Mar 2024 13:55:30 GMT"
  },
  {
    message_id: 2,
    "sender": "receiverUsername",
    "receiver": "senderUsername",
    "message": "I’m good, thanks!",
    "created_at": "Tue, 26 Mar 2024 13:56:30 GMT"
  }
  // Additional chat message objects as needed
]
```

8 Leaderboard

- Endpoint: `/leaderboard`
- Method: GET
- Description: Get a list of top 3 users, 2 friends and yourself sorted by their score.
- Access: Requires `logged in` user privileges.
- Response: A JSON array of user objects, each containing:

```
[
  {
    "place": 1,
    "username": "username",
    "score": 100
  },
  {
    "place": 2,
    "username": "anotherUsername",
    "score": 200
  }
  // Additional user objects as needed
]
```

9 Market

- endpoint: @game_blueprint.route('/update-building-map', methods=['POST'])
- Method: POST
- Description: Handles POST requests to insert JSON data into the database. Expects JSON data in the request body.
- Access: Requires `logged in` user privileges.
- Response: A JSON object with status and message indicating success or failure.

10 Building

- endpoint @game_blueprint.route('/fetch-building-information', methods=['GET'])
- Method: GET
- Description: Handles GET requests to fetch building information for the current user.
- Access: Requires `logged in` user privileges.
- Response: A JSON object containing building information.

10.1 Building Information by type

- Endpoint: `/fetch-building-information-by-type/<string:building`$_type >$ $Method : GET$
- Description: Handles GET requests to fetch building information for the current user with the given type.
- Access: Requires `logged in` user privileges.
- Response: A JSON object containing building information.

10.2 Exploration Information

- Endpoint: `/exploration`
- Method: GET
- Description: Handles GET requests to fetch exploration information for the current user if one is ongoing.
- Access: Requires `logged in` user privileges.
- Response: A JSON object containing exploration information. examples:

```
{
    "ongoing": true,
    "owner": owner of exploration,
    "chickens": chickens sent on exploration,
    "goats": goats sent on exploration,
    "pigs": pigs sent on exploration,
    "cows": cows sent on exploration,
    "exploration_level": level of bay building once exploration was starte
    "augment_level": augment level of bay building once exploration was st
    "started_at": time exploration was started,
    "duration": duration of exploration in minutes,
    "surviving_goats": surviving goats on exploration,
    "rewards_of_goats": amount of rewards the goats brought with them,
    "surviving_pigs": survining pigs on exploration,
    "surviving_cows": surviving cows on exploration,
    "rewards_of_cows": amount of rewards the cows brought with them,
    "surviving_chickens": surviving chickens on exploration,
    "base_rewards": base rewards of the exploration,
}
{
    "ongoing": false
}
```

10.3 Starting an exploration

- Endpoint: `/start-exploration`
- Method: POST
- Description: Handles POST requests to start an exploration with given information for the current user.
- Access: Requires `logged in` user privileges.
- Request: A JSON object exploration information. example:

  ```
  {
    "chickens": amount of chickens sent on exploration,
    "goats": amount of goats sent on exploration,
    "pigs": amount of pigs sent on exploration,
    "cows": amount of cows sent on exploration,
    "exploration_level": level of bay building,
    "augment_level": augment level of bay building,
    "remaining_time": duration of exploration in minutes
  }
  ```

- Response: status of the exploration:

  ```
  {
      'status': 'success',
      'message': 'Exploration started successfully.'
  }
  {
      'status': 'error',
      'message': 'Failed to start exploration.'
  }
  ```

10.4 Stopping an exploration

- Endpoint: `/stop-exploration`

- Method: POST
- Description: Handles POST requests to stop an exploration for the current user.
- Access: Requires `logged in` user privileges.
- Response: status of stopping the exploration:

```
{   'status': 'success',
    'message': 'Exploration stopped successfully.'
}
{
    'status': 'error',
    'message': 'Failed to stop exploration.'
}
```

10.5 General Animal Update Information

- Endpoint: `/add-animals`
- Method: POST
- Description:updated list of animals for a user. This endpoint is used to update the quantity of animals owned by a user.
- Access: Requires logged-in user privileges.
- Request: A JSON object with general information about the update. The object includes as key the animal and as value the amount that needs to be added (subtracted in the case of a negative value). The key 'idle' with value true can be added to update the last$_{u}pdated, forthehourlyproductionofan$

```
{
    "Pig": -7,
"Cow": 5,
"Chicken": 7,
"Goat": -5
}
```

```
 ...// 'idle' is used when last_updated needs to be reset aswell this is t
{
    "idle": true,
    "Pig": 2,
"Cow": 2,
"Chicken": 3,
"Goat": 1
}
```

Response: A JSON object indicating the status of the update operation.

```
{
  "status": "success/error",
  "message": "Description of the outcome of the operation"
}
```