FH University of Applied Sciences TECHNIKUM WIEN

Protocol

Software Engineering 1 Labor
BIF3-A1/A2 (Holzer Raoul)

MonsterTradingCardsGame

Armin Dervisefendic if23b040

Project Overview

The Monster Trading Cards Game (MCTG) is designed as a REST-based server application that enables users to manage their trading cards, register, and log in. The application is built using C# and does not rely on any HTTP helper frameworks.

Implementation Steps

1. Project Initialization

- Created a new C# console application named MCTGServer.
- Set up the necessary directory structure and files:
  - **Program.cs**: Main application logic
  - **User.cs**: User model class
  - **Card.cs**: Card model class

2. Server Setup

- Implemented a TCP listener in Program.cs to accept incoming client connections on port **10001**.
- The server listens for HTTP requests and handles basic request parsing.

3. User Registration

- Implemented an endpoint for user registration (/register):
  - Users can send a POST request with a JSON body containing their username and password.
  - Validated input and stored user details in an in-memory list.

4. User Login

- Implemented an endpoint for user login (/login):
  - Users can send a POST request with their credentials.
  - Validated the user and generated a unique token for successful logins.
  - Returned the token in the HTTP response.

5. Card Management

- Added functionality to manage trading cards:
  - Implemented an endpoint to add new cards (/cards, POST):
    - Users can send card details (name, type, power) in a JSON format.
  - Implemented an endpoint to retrieve all cards (/cards, GET):
    - Returned a list of all cards in JSON format.

6. Testing and Validation

- Conducted tests to ensure that the registration, login, and card management functionalities work correctly.
- Logged requests and responses to the console for easier debugging.

How to Run the Monster Trading Cards Game Server

Prerequisites

Before running the server, ensure you have the following installed on your system:

1. **.NET SDK**: Make sure you have the .NET SDK installed.
2. **IDE**: An Integrated Development Environment (IDE) like Visual Studio or Visual Studio Code is recommended for easy code editing and debugging.

Steps to Run the Server

1. Clone the Repository

Clone the project repository to your local machine using the following command:

```
git clone https://github.com/mide553/MonsterTradingCardsGame.git
```

2. Navigate to the Project Directory

Open a terminal or command prompt and navigate to the project directory:

```
cd MonsterTradingCardsGame/MCTGServer
```

3. Server can be run in VS by building the project and running or by a CMD commands:

- dotnet build
- dotnet run

4. Upon successful execution, you should see a message indicating that the server has started and is listening on port 10001:

Output: Server started on port 10001

You can test the server with following commands:

Register:

curl -X POST http://localhost:10001/register --header "Content-Type: application/json" -d "{\"Username\":\"*username*\", \"Password\":\"*password*\"}"

Output: User registered!

Login:

curl -X POST http://localhost:10001/login --header "Content-Type: application/json" -d "{\"Username\":\"*username*\", \"Password\":\"*password*\"}"

Output: {"token": "*generated-token*"}

New card:

curl -X POST http://localhost:10001/cards --header "Content-Type: application/json" -d "{\"Name\":\"*Fire Dragon*\", \"Type\":\"*Monster*\", \"Power\":*50*}"

Output: Card added!

curl -X POST http://localhost:10001/cards --header "Content-Type: application/json" -d "{\"Name\":\"*Water Spell*\", \"Type\":\"*Spell*\", \"Power\":*40*}"

Output: Card added!

Get cards:

curl -X GET http://localhost:10001/cards

Output: [{"Name":"Fire Dragon","Type":"Monster","Power":50},{"Name":"Water Spell","Type":"Spell","Power":40}]