OAuth Login API

Table of Contents

Project Overview

This project implements an OAuth login API using ASP.Net MVC. The API is designed to authenticate users and provide them with JWT tokens that include their roles and accessible system regions. This allows the game to restrict access to certain features based on the user's role.

Key Features:

- OAuth-based authentication
- JWT token issuance with user roles and scopes
- Static user data for demonstration
- Support for different system regions and roles

Setup and Installation

Prerequisites

- .NET Framework
- Visual Studio or any other C# IDE
- ASP.NET MVC and Web API

Installation Steps

1. **Clone the Repository:**
   git clone <repository-url>
   cd <repository-directory>
2. **Open the Project:**
   Open the solution file (.sln) in Visual Studio.
3. **Install Required Packages:**
   Ensure you have the following NuGet packages installed:
    - Microsoft.Owin.Host.SystemWeb
    - Microsoft.Owin.Security.OAuth
    - Microsoft.Owin.Cors

4. **Build the Project:**

   Build the project to restore any missing dependencies and ensure all configurations are correct.

Usage Guide

Authentication

To authenticate, you need to request a token from the /token endpoint with a username and password.

Request Token

POST /token

Content-Type: application/x-www-form-urlencoded

grant_type=password & username=admin & password=admin

Response

Json

```
{
   "access_token": "<JWT_Token>",
   "token_type": "bearer",
   "expires_in": 86400
}
```

Access API Endpoints

Use the obtained token to access the protected endpoints.

Get Server Time

GET /api/data/forall

Access Authenticated Data

GET /api/data/authenticate

Authorization: Bearer <JWT_Token>

Access Admin Data

GET /api/data/authorize

Authorization: Bearer <JWT_Token>

Code Explanation

RouteConfig.cs

The RouteConfig.cs file configures the routes for your API.

- **Purpose:** Defines URL routing for the API.
- **Key Points:**
    - Routes requests to the UserController by default.
    - Ignores requests to .axd resources, which are used for tracing.

Startup.cs

The Startup.cs file configures the OWIN pipeline and sets up OAuth authentication.

- **Purpose:** Configures the OAuth server and sets up token issuance.
- **Key Points:**
    - Uses CORS to allow requests from any origin.
    - Sets the /token endpoint for token requests.
    - Configures tokens to expire after 1 day.

ApiAuthorizeAttribute.cs

The ApiAuthorizeAttribute.cs file customizes authorization handling.

- **Purpose:** Extends the default authorization behaviour to return a 403 Forbidden status for authenticated but unauthorized requests.
- **Key Points**
    - Checks if the user is authenticated.
    - If authenticated but not authorized, returns 403 Forbidden.

APIAuthorization.cs

The APIAuthorization.cs file defines the OAuth provider for validating users and generating JWT tokens.

- **Purpose:** Manages authentication and generates JWT tokens with roles and scopes.
- **Key Points:**
    - Validates clients with context.Validated().
    - Grants credentials by validating username and password.
    - Adds claims for roles and scopes to the JWT token.

UserController.cs

The UserController.cs file defines the API endpoints for accessing user data and roles.

- **Purpose:** Provides endpoints for getting user data and checking roles.
- **Key Points:**
    - localjost/api/data/forall: Public endpoint to get server time.
    - localjost /api/data/authenticate: Returns user's accessible systems.
    - localjost /api/data/authorize: Restricted to admins, returns user roles and accessible systems.

Configuration

The API is configured to use OAuth for authentication. You can modify the following settings in Startup.cs:

- **Token Endpoint Path:** Change /token to your desired endpoint.
- **Token Expiry:** Adjust AccessTokenExpireTimeSpan to control how long the token is valid.

Testing

Manual Testing

1. **Get Token:** Use tools like Postman to send a POST request to /token with valid credentials.
2. **Access Endpoints:** Use the obtained token to send GET requests to the API endpoints.

Troubleshooting

- **Token Not Issued:** Ensure the username and password are correct.
- **403 Forbidden:** Check if the user has the necessary roles for the endpoint.
- **CORS Issues:** Make sure CORS is correctly configured in Startup.cs.