DISTRIBUTED SYSTEMS

Distributed Systems Report on Web API

Scott Kikumu

Contents

Introduction.

The main aim of the report is to document creation of a WebApi. The WebApi general functionality in this case is to:

- Manage a user database using Entity Framework Package from Visual Studio using SQL.
- Creation of a stateless server which manages data sent from a client to the server.

This report will also cover:

- Defining and explaining where and how HTTP verbs used.
- RSA/AES algorithm used within the WebApi
- Entity Framework

1 -Definition of A WebApi and Request Management.

A WebAPI is an extensible framework for building HTTP based services that can be accessed in different applications. It manages incoming HTTP requests by issuing controllers and actions to the request to carry out different tasks within the server.

This API used however is regarded as stateless. This is because the server, does not store client information between requests. All client information is stored in a database attached to the server and as soon as the server returns a resource to the client, it forgets the client. A stateless server works by first creating, storing(to the Database) and issuing an ID to a client. After this is done, the server clears its memory of the client. It is the responsibility of the client to retain its ID for the duration of resource exchange between client and server.

A stateful server works the same way. However, it remembers the client between requests. The client does not have to constantly send over its ID after every request. The server then allocates part of its memory to save data about who the client is and what the client is doing. The connection between the client and the server is also constantly kept open. It is however difficult to determine when to get rid of client data from the server.

This therefore concludes that the requests are managed differently depending on whether the API is being used on a stateless or stateful server.

2 -Route Mapping and Actions Definition.

Is a set of templates which determine what controller and action method to execute in an API. The ID parameter is part of the route mapping in which data is sent over from the client to the server. Depending on the type of parameter sent over(simple or complex type), the API will attempt to read the parameter from the URI of the request or the body of the request. Actions are contained in the controller in which our methods are mapped to CRUD(Create, Retrieve, Update, Delete) operations.

3 -Requests Definition.

The HTTP request protocols are mapped in the following manner:

3.1 -HTTPGet.

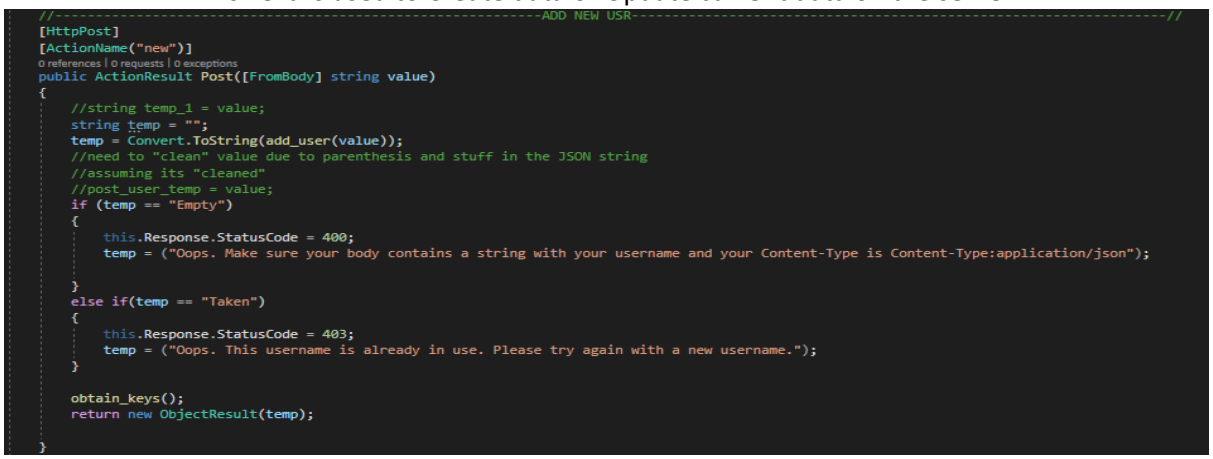
This verb is used to retrieve and request specific information from the server.

```
[HttpGet]
[ActionName("new")]
0 references | 0 requests | 0 exceptions
public string New([FromQuery]string name) //gets name value from get request and checks database if user exists
{
    name = Convert.ToString(search_username(name));
    obtain_keys();
    //convert to JSON string so that client can accept
    return name;
}
```

Figure 3.1: Get Request Implementation

3.2 -HTTP Post

This verb is used to Create data or Update current data on the server.

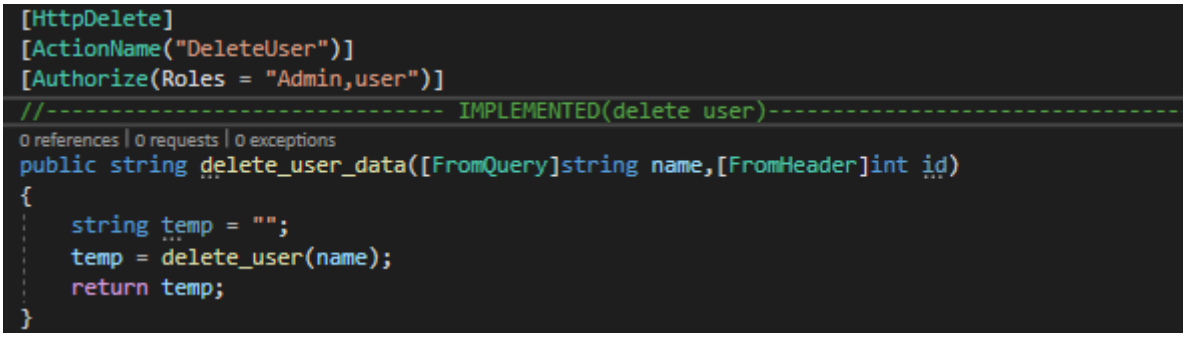


```
//-------------------------------------------------------------ADD NEW USR-------------------------------------------------------------//
[HttpPost]
[ActionName("new")]
0 references | 0 requests | 0 exceptions
public ActionResult Post([FromBody] string value)
{
    //string temp_1 = value;
    string temp = "";
    temp = Convert.ToString(add_user(value));
    //need to "clean" value due to parenthesis and stuff in the JSON string
    //assuming its "cleaned"
    //post_user_temp = value;
    if (temp == "Empty")
    {
        this.Response.StatusCode = 400;
        temp = ("Oops. Make sure your body contains a string with your username and your Content-Type is Content-Type:application/json");

    }
    else if(temp == "Taken")
    {
        this.Response.StatusCode = 403;
        temp = ("Oops. This username is already in use. Please try again with a new username.");
    }

    obtain_keys();
    return new ObjectResult(temp);

}
```

Figure 3.2: Post Request Implementation.

3.3 -HTTP Delete.

This verb is used to Delete data from the server.



```
[HttpDelete]
[ActionName("DeleteUser")]
[Authorize(Roles = "Admin,user")]
//------------------------------- IMPLEMENTED(delete user)--------------------------------
0 references | 0 requests | 0 exceptions
public string delete_user_data([FromQuery]string name,[FromHeader]int id)
{
    string temp = "";
    temp = delete_user(name);
    return temp;
}
```

Figure 3.3: Delete Request Implementation.

4 -Entity Framework.

This framework is a Microsoft relational database tool designed to work with SQL. It can create a database for a user based on:

- Programmer defined objects in code.
- User-defined database schema.
- Using a pre-existing database.

There are 3 approaches to create database structures using entity framework which are:

- Model – First: Creating using this method involves creating a visual representation of the database using design tools. The visuals drawn will then be used to autogenerate a SQL-database when fed into Entity Framework.
- Database- First: Involves manually creating the SQL script to make the Database and then utilizing Entity framework to update the database accordingly.

- Code – First: Allows the user to define database objects using standard classes without using a design utility.

Code first uses migrations to change Database attributes. Migrations is a feature in entity framework which enables the user to make and propagate changes to a Database model.

5 -RSA and AES algorithm.

RSA algorithm is a form of asymmetric encryption which provides two different keys for encryption and decryption. AES is a form of symmetric encryption(Block cipher) where both endpoints use the same key to encrypt and decrypt data.

5.1 -Steps taken in RSA algorithm.

Plain text is first converted to bytes then the following steps occur:

- The public key is sent over which contains 2 parts, **n** and **e**. and a private key contains 3 parts, **p**, **q** and **d**. Such that, p and q contain 2 large prime numbers. When p and q are multiplied together, it gives n. e is public encryption exponent whereas d is private decryption exponent such that:
  $d = e\text{-}1 \bmod \phi(n)$
- The message is turned into a number **m** smaller than **n.** The cypher text, **c** , is then computed corresponding to **$c = m^e \bmod n.$**
- The message can be decrypted using private key **d** such that: **$m = c^d \bmod n.$** Given **m** the original distinct prime numbers can be discovered by applying **Chinese remainder theorem** which yields **$m^{ed} = m \bmod pq$**.

5.2 -Steps taken in AES algorithm.

Plain text is first converted to bytes then converted into 'blocks' of data. The following steps occur:

- The bytes in the block are each mathematically calculated and substituted using a substitution table. Inverse substitution can be applied on each bit.
- Rows of data are then shifted in relation to one another to the right n number of times depending on the round.
- Columns of data are then shifted the same way using XOR operations and dot products.
- Part of the key is then added to the data using XOR operations to give an output for that round.
- The same process is then repeated for a certain number of rounds.

6 -How API key is utilized in the server.

The API key is stored in the header of the requests and is used in identifying and authorizing user resources to a client. The API key is a good option to identify users because each API key generated is unique and no API key can be the same. The API key is however not safe in this project because it is not encrypted when it is sent back and forth from client to server. This can be combatted in the real world by encrypting the API key during data transmission.

7 -Conclusion.

Each task was successfully implemented and completed. Encryption is very important in maintaining user privacy in a server/client. I encountered problems while using keys to encrypt and decrypt data. The Microsoft forums and student labs helped greatly in tackling the issues.