



## SOFTWARE REQUIRMENT SPECIFICATION

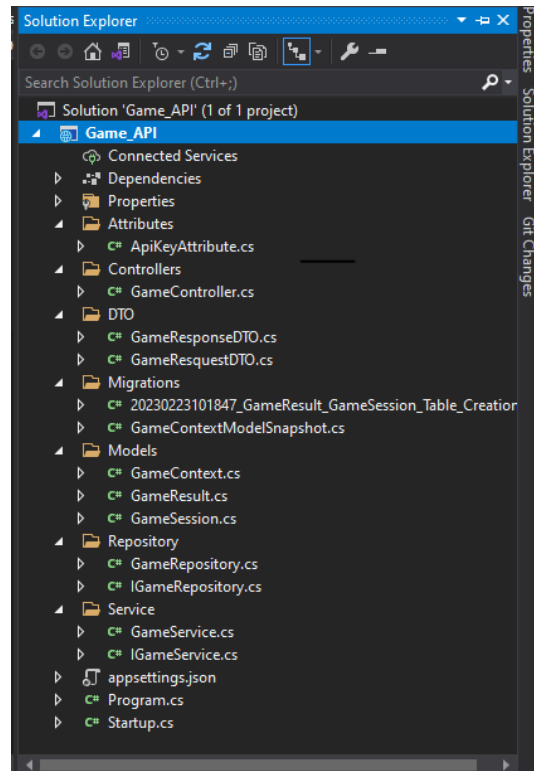
# CONTENTS

Application Architecture.....	2
Task 1 (Post API).....	6
Task 2 (Get API) .....	6
TECHNOLOGY STACK .....	7
THANK YOU .....	8

Technical Guide Concepts and Procedures Release Table

DOCUMENT VERSION	
<b>v1.0</b>	26-Feb-2023

# Application Architecture



- **Attributes Folder**

This folder contains **ApiKeyAttribute** class which will use for Authenticate the user using this class (**OnActionExecutionAsync**) Method. This Method will be executed on every API call first, if user is authenticated then API method body will be executed to perform operations.

This class is inheriting (**Attribute, IAsyncActionFilter**) so, this will be used on class Level.

```
13
14 namespace Game_API.Controllers
15 {
16     [ApiController]
17     [Route("api/[controller]")]
18     [ApiKey]
19     public class GameController : ControllerBase
20     {
21         private readonly IGameService _service;
22
23         public GameController(IGameService service)
24         {
25             _service = service;
26         }
27     }
```

- **Controller Folder**

This folder contains **GameController** which contains API calls of POST and GET

```
[HttpPost]
[Consumes(MediaTypeNames.Application.Json)]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status401Unauthorized)]
public async Task<ActionResult> PostGame([FromBody] GameResult gameResult)
{
    if (gameResult == null)
    {
        return BadRequest();
    }

    try
    {
        // Save the changes to the database
        await _service.Post(gameResult);
    }
    catch (DbUpdateException)
    {
        throw;
    }

    // Return a response with the saved gameSession object
    return CreatedAtAction(nameof(PostGame), new { id = gameResult.UserId }, gameResult);
}
```

If POST/GET call is does not contain Authentication Key than 401 Code will be returned with message  
"Api Key was not provided".

Responses

Curl

```
curl -X GET "https://localhost:44374/api/Game?userId=1&playerId=111&gameName=1" -H "accept: text/plain"
```

Request URL

```
https://localhost:44374/api/Game?userId=1&playerId=111&gameName=1
```

Server response

Code	Details
401 <i>Undocumented</i>	<p>Error:</p> <p>Response body</p> <pre>Api Key was not provided</pre> <p>Response headers</p> <pre>content-length: 24 content-type: text/plain; charset=utf-8 date: Sun26 Feb 2023 12:20:38 GMT server: Microsoft-IIS/10.0 x-powered-by: ASP.NET</pre>

For request to be successfully authenticated you must enter API key. Copy API key from file  
"appsettings.json"

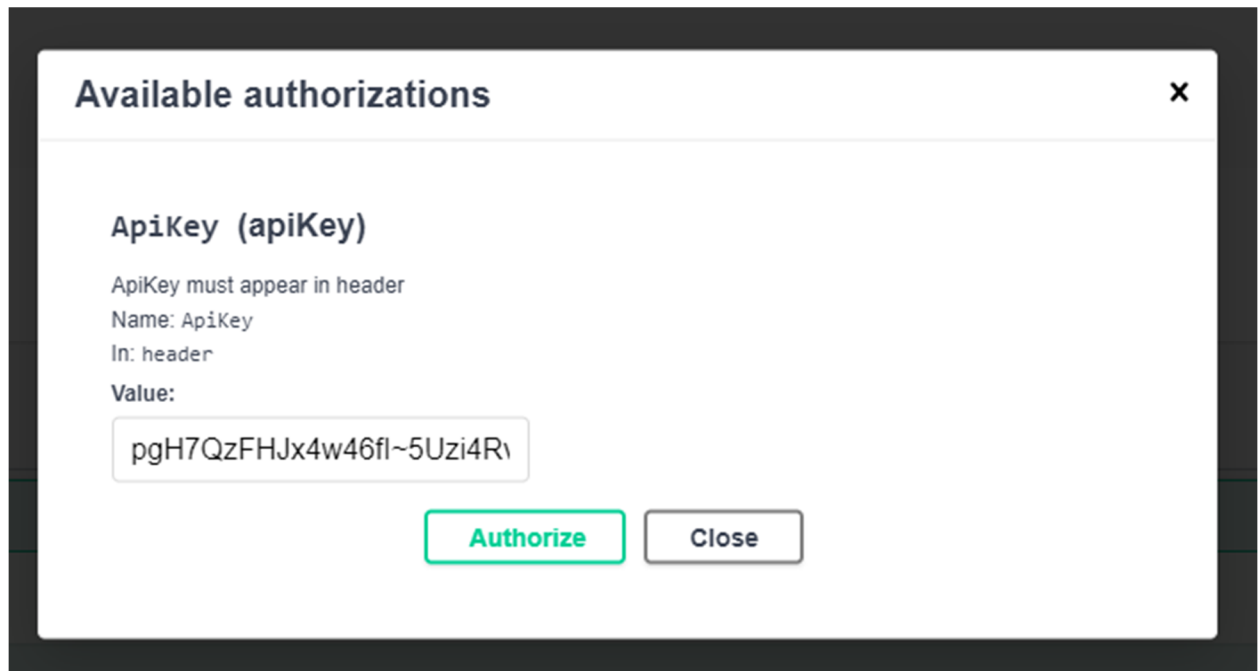
Schema: <https://json.schemastore.org/appsettings.json>

```
1  {
2    "Logging": {
3      "LogLevel": {
4        "Default": "Information",
5        "Microsoft": "Warning",
6        "Microsoft.Hosting.Lifetime": "Information"
7      }
8    },
9    "AllowedHosts": "*",
10   "ApiKey": "pgH7QzFHJx4w46fI~5Uzi4RvtTw1EXp",
11   "ConnectionStrings": {
12     "Game": "Data Source=DESKTOP-BL30DIO;Initial Catalog=Game;Integrated Security=True"
13   }
14 }
15
```

Click on Authorize button as highlighted in below image.

The image shows the Swagger UI interface for an API. At the top, there's a header with the Swagger logo and a dropdown menu labeled 'Select a definition' with 'Game\_API v1' selected. Below this, the title 'Api Key Auth' is displayed with a 'v1' tag and a 'OAS3' badge. A red arrow points to the 'Authorize' button, which is located to the right of the title. Below the title, there's a section for the 'Game' endpoint, which is a POST request to '/api/Game'. This section includes a 'Parameters' tab (currently selected) showing 'No parameters', a 'Try it out' button, and a 'Request body' section with a dropdown menu set to 'application/json'.

Below Pop will appear



Paste the key in text box and click on Authorize.

- **DTO Folder**

GameRequestDTO is used for GET API call parameters.  
GameResponseDTO is used to send response to swagger/UI.

- **Models**


**GameResult** and **GameSession** has one to many relations.  
GameResult class is used in POST API call parameters, because json object will be sent in Request body, JSON object will map all the values to GameResult class.


**GameContext** is responsible for creating database connection, this class is injected in Startup.cs class.


- **Service & Repository**

Service is used to send the API calls to Repository & service is also responsible for returning. GameResponseDTO to controller which is used in GET API call for response object.

## Task 1 (Post API)

**Game** 

POST /api/Game 

GET /api/Game/NextDifficultyLevel 

**Schemas** 


GameSession >


GameResult >

ProblemDetails >

GameResponseDTO >

## Task 2 (Get API)

GET /api/Game/NextDifficultyLevel 

Parameters 

Name	Description
<b>userId</b> * required string (query)	<input type="text" value="userId"/>
<b>playerId</b> * required Integer(\$int32) (query)	<input type="text" value="playerId"/>
<b>gameName</b> * required string (query)	<input type="text" value="gameName"/>

## TECHNOLOGY STACK

1. Visual Studio 2022
2. .NET Framework: 7.0
3. Microsoft SQL Server (I used SQL Server and database, schema automatically created through entity framework migration, mongo database not installed on my laptop)



THANK YOU



Thanks & Best Regards,  
Mudassar Shahbaz