**Lead Squared -Cache Problem**

**Description:**

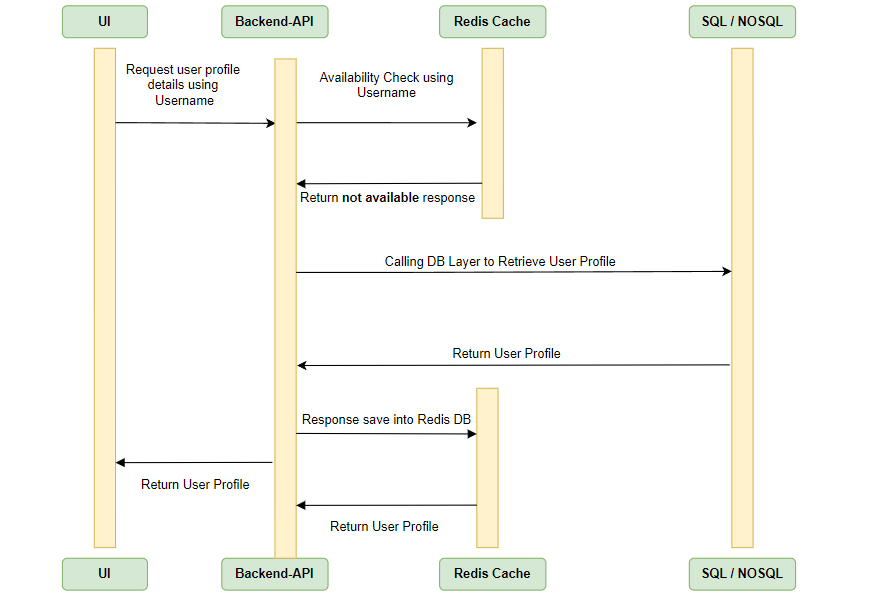
Propose your solution of a cache class (group of classes).

**Solution:**

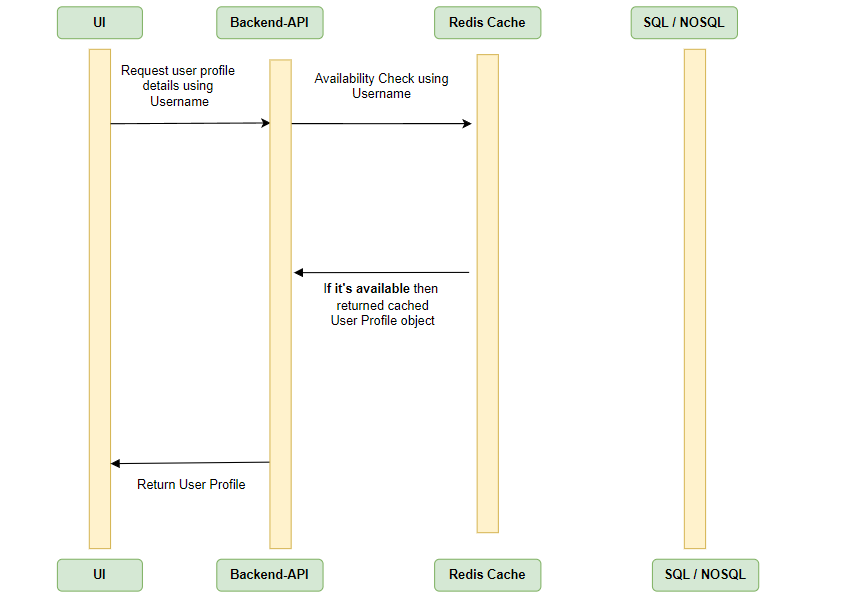
* For Demo Purpose, I have used Microsoft User Identity to all Crud Operation in DB. Haven’t Used any EF core as our main goal is to achieve Cache.
* I have provided option for using either Memory /Distributed Cache in our user profile.
* For Distributed Cache, have used Redis.
* Cache Provider Adapter will decide which cache instance to be used based on the flag (Cache Type) set in the appsetting.json.
* Have Used “NSwag” for Swagger and “JWT Bearer” for Token Generation and “StackExchangeRedis” for Redis Cache.

**UML Flow:**

* Non-Cache Flow:



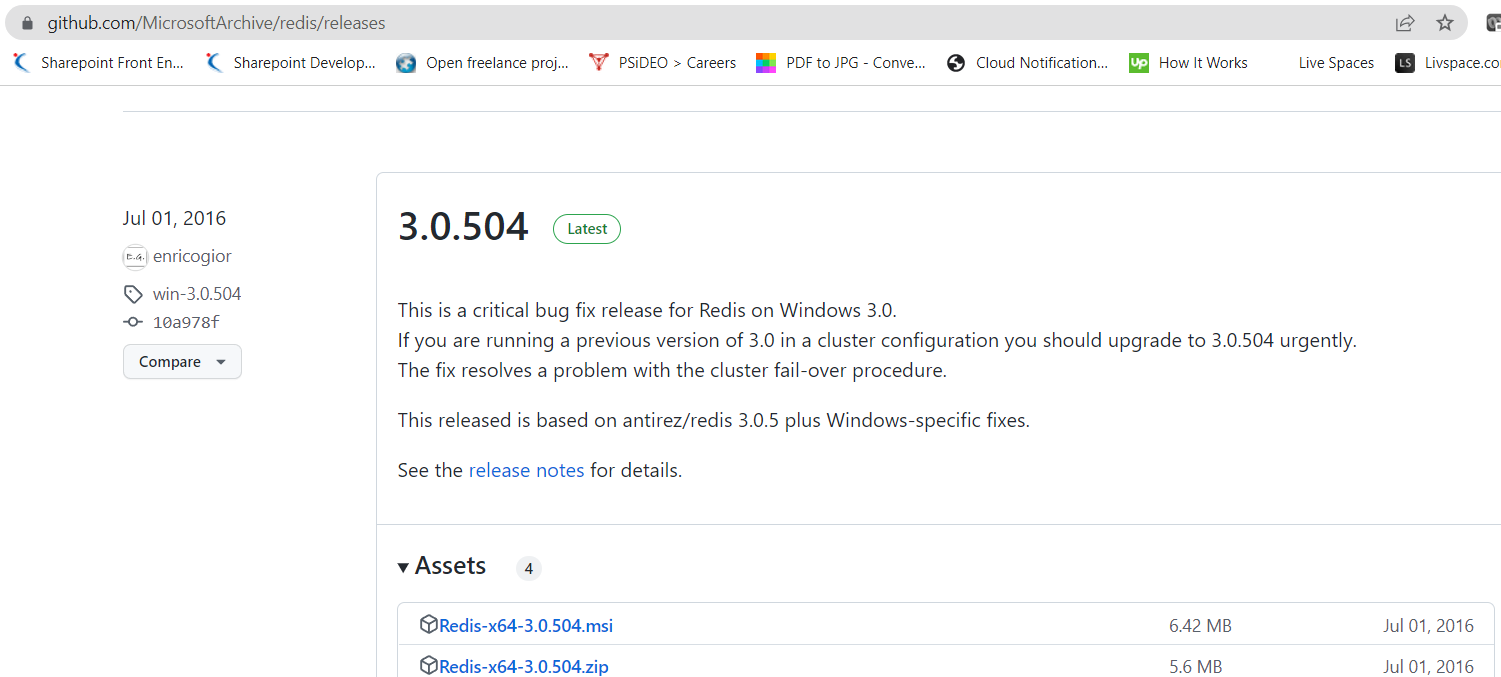
* Cache Flow:



**Software Prerequisite:**

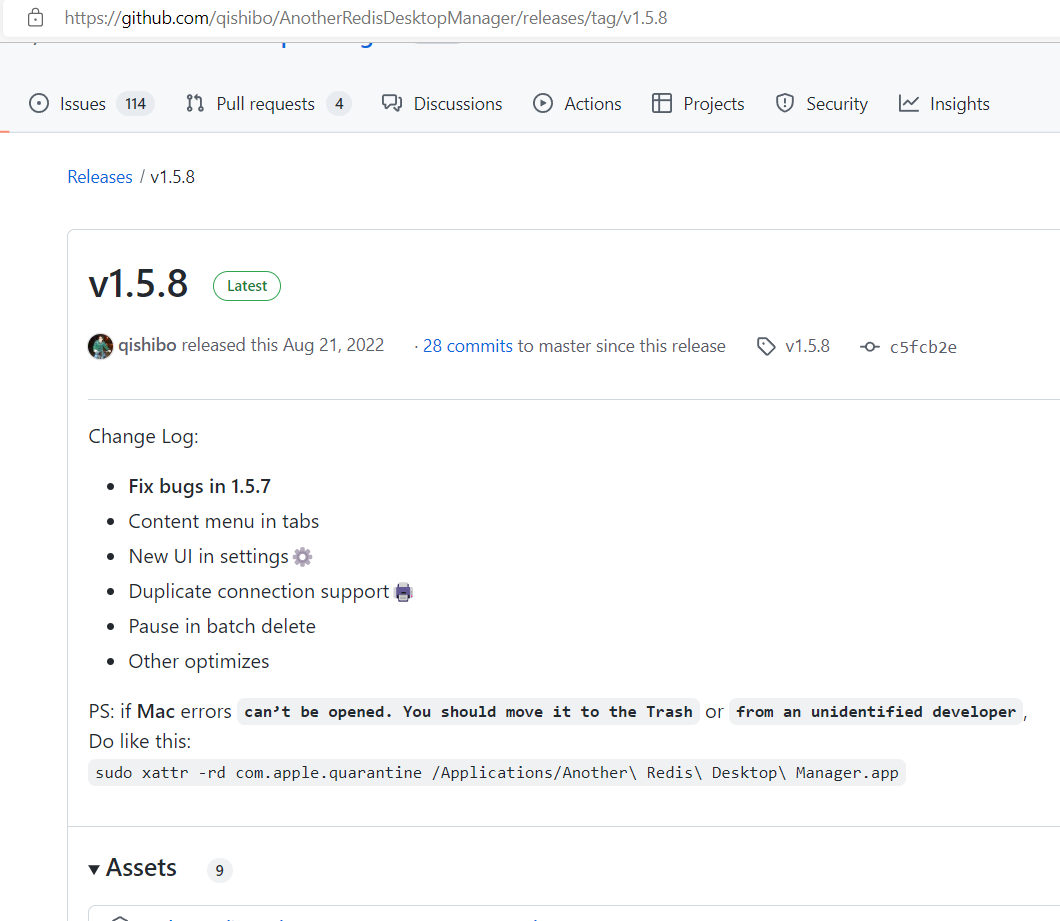
* Download & Install Redis Cache from the below path as per option provided. (Version 3.0.504). Use MSI to install. (Open Source)
* **Use Port (6379). As I have configured in App Settings.**

[**https://github.com/MicrosoftArchive/redis/releases**](https://github.com/MicrosoftArchive/redis/releases)

****

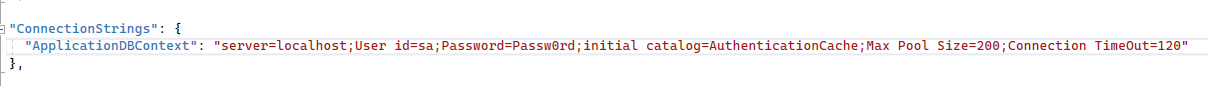
* Download & Install Another Redis Desktop Manager to visualize the data from the below path. (Version 1.5.8). Use exe to install. (Open Source)

[Release v1.5.8 · qishibo/AnotherRedisDesktopManager · GitHub](https://github.com/qishibo/AnotherRedisDesktopManager/releases/tag/v1.5.8)

****

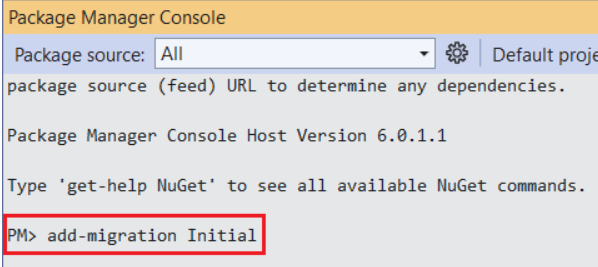
**Project Solution:**

* Unzip the Visual Studio Project in your local workspace. (Require .net Core6 Framework).
* Change the Connection string as per your SQL Connection in “appsettings.json” file.



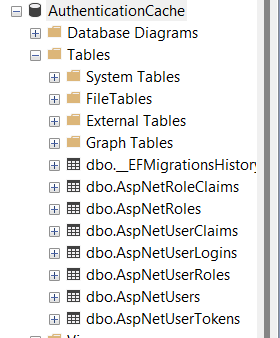
* As I am following Code First Approach, so install the Db using below Migration command in NuGet Package Manager Console.

1)***add-migration Initial***

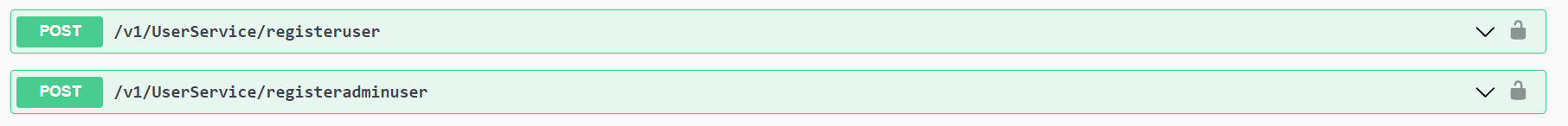


2) ***update-database***

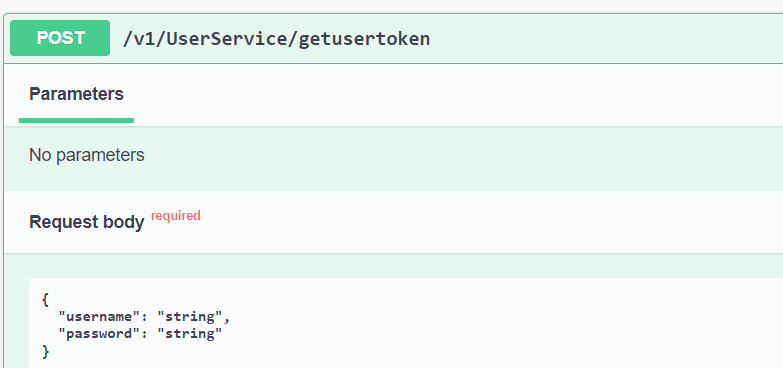
* If you check the database using SQL server object explorer, you can see that the tables below are created inside the database. We are using very few tables for our Demo purpose. Keeping the other Tables for future enhancements.

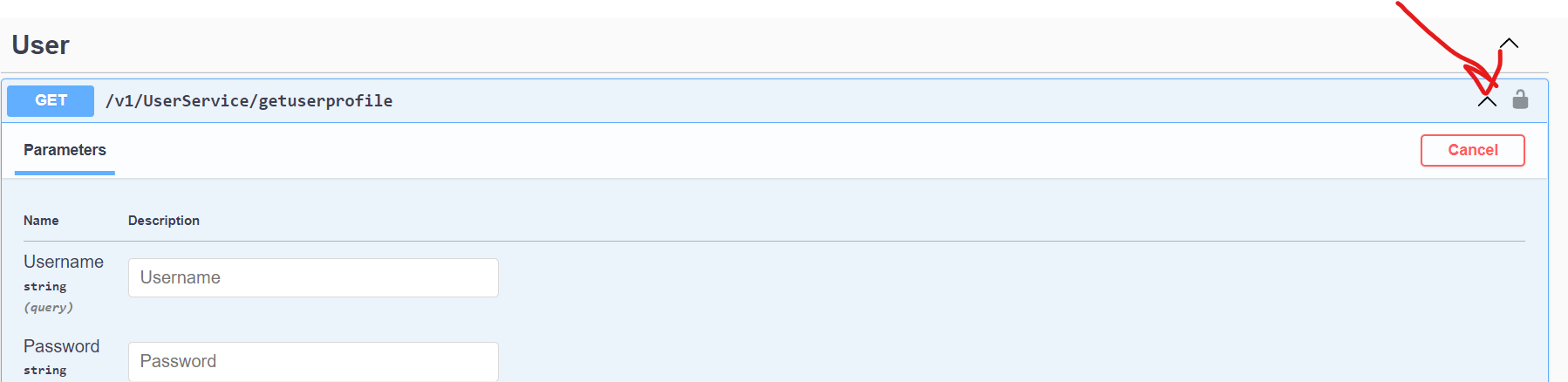


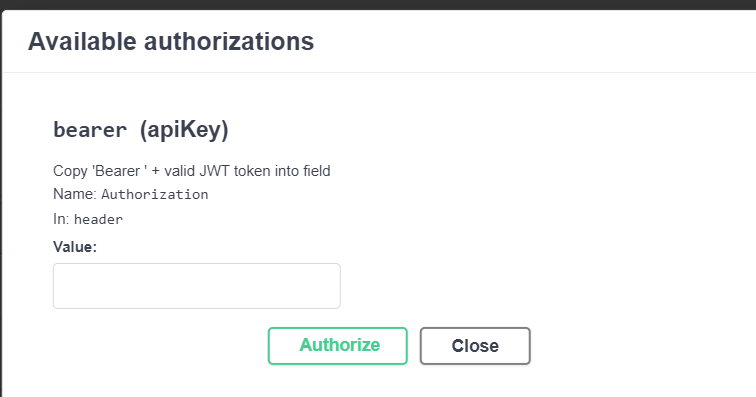
* Run the Application. Swagger Page will Open.
* First Run Few API to Register the normal User and Admin.



* Once the User is Registered then the run the below API to generate the JWT Token. Cache is also in Action to Get/Set the Cache Key.



* To get the Profile from cache, Run the below API which Require Authorization Token. Please use the Generated JWT Token from the above and use them along with bearer.
* 
* Use (**Bearer + Validn JWTGenerated Token**) in the below section and click on Authorize.



* Then run the User Profile from Cache provided by the requested Param.

//Cache Set Object

userProfile.Users = await \_cacheProvider.GetOrSetValueAsync

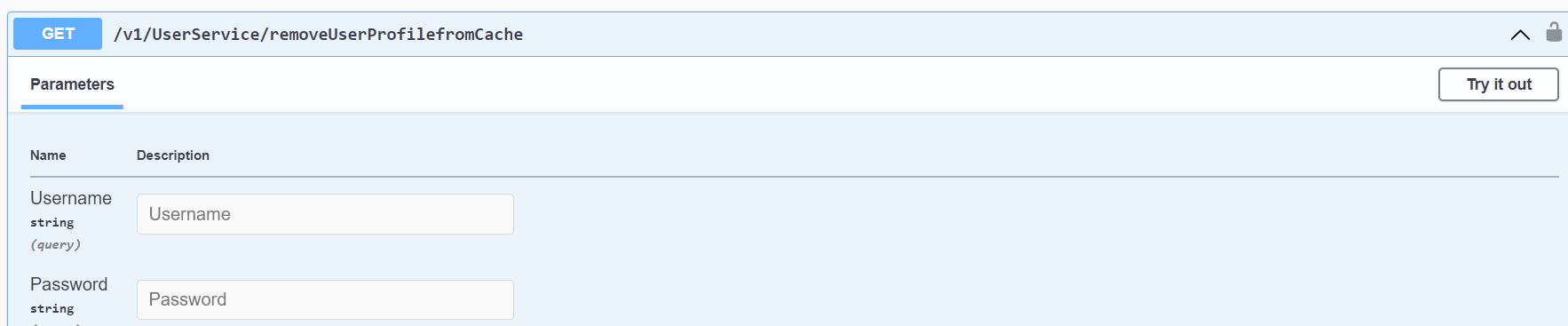
(

$"{nameof(IdentityUser)}:{request.Username}",

async () => { return await \_userManager.FindByNameAsync(request.Username).ConfigureAwait(false); }

).ConfigureAwait(false);

* For Removing the cache for particular User. Please run the Below API, which requires Authorization bearer Token too.



* To Verify/Visualize the token. Please Check in Another Redis Desktop Manager. You cannot view the data as we use Base64 Encoding.

