Documentation Consumo de Servicios

Prueba Tecnica

1. Consumo de Servicios UserController y FamilyController

Para esto necesitan generar un Bearer JWT

Se consume el servicio (/User/GenerateBearer) La cual solicita su nombre de usuario y contraseña el cual retorna un Bearer que queda registrada en la base de datos y tiene una vigencia de una hora de su fecha de generación

Con la cual este bearer será solicitado para el resto de servicios

1.1 Servicios de UserController:

Encontraremos los siguientes servicios

-User/CreateUser: Servicio para crear los usuarios

-User/GetUser: Servicio para obtener los usuarios

-User/DeleteUser: Servicio para eliminar usuario

-User/UpdateUser: Servicio para Actualizar información del usuario

-User/UpdateUserPass: Servicio para actualizar contraseña del Usuario

* 1. Servicios de FamilyController

-/Family/CreateFamily: Servicio para crear familiar del usuario

-/Family/DeleteFamily: Servicio para eliminar familiar

-/Family/UpdateFamily: Servicio para actualizar información del usuario

-/Family/GetFamilyUser: Servicio para obtener los familiares por usuario

-/Family/GetFamilyByDocument: Servicio para obtener familiar por Numero de Documento

Ademas en cada servicio de estos antes mencionados se implementa método para crear logs en la base de datos en la tabla Services\_Log en la cual se almacena tanto el request y el response de cada petición que solicite por medio de esta y en caso tal motivo de error.

1. Para Consumir Servicios PostsController y CommentsControllers

Para estos servicios se requieren una Autorización **Basic Auth**

User: XYZUser

Password: XYZPass

Estos son las credenciales para poder realizar los consumos de los servicios generados

* 1. Servicios PostsController:

Encontramos los siguientes Servicios

-/Posts/InsertMassivePost: servicio para hacer inserción masiva según la documentación brindada.

-/Posts/CreatePosts: servicio para crear un post.

-/Posts/GetPost: servicio para obtener los posts.

-/Posts/GetPostById: servicio para obtener un post por id.

-/Posts/UpdatePost: servicio para actualizar posts.

1.2 Servicios CommentsController:

-/Comments/InsertMassiveComments: servicio para hacer inserción masiva según la documentación brindada.

-/Comments/CreateComments: servicio para crear un Comments.

-/Comments/GetComments: servicio para obtener los Comments.

-/Comments/GetCommentsById: servicio para obtener un Comments por id.

-/Comments/UpdateComments: servicio para actualizar Comments.

1. Servicios extras incluidos:

[http://localhost:42218 /api/JsonPlaceCommentsholder/Comments](http://localhost:42218%20/api/JsonPlaceCommentsholder/Comments)

http://localhost:42218 /api/JsonPlaceholder/Posts

son 2 servicios que cumplen la misma función o similar de reflejar los posts y comments

cabe recalcar que el puerto que se entrega puede variar(solo es reemplazarlo)

1. Información a considerar

* los servicios fueron construidos en C# .net core
* el modelo de visualización es con swagger V.2
* todos los servicios para la comunicación con las tablas en las bases de datos se hacen por procedimientos almacenados
* en el proyecto contiene diferentes carpetas para tener un orden dentro del propio código los cuales son:

-DataObjects: Carpeta donde contiene las Classes de request y response de cada uno de los servicios mencionados anteriormente.

-Repository: Carpeta donde se contiene los consumos con la base de datos cada repositorio creado es para cada diferente controlador generado.

-Utilities: Carpeta donde se encuentran diferentes utilidades las cuales son:

-Clase útil donde Contiene métodos los cuales son utilizados en diferentes controladores y/o servicios.

-Clase ApiEnumerables la cual contiene los diferentes procedimientos almacenados creados y son llamados para su consumo.

* La arquitectura de respuesta que se maneja en los diferentes controladores y/o servicios consumidos es la siguiente:

ResponseGeneric que contiene los siguientes parámetros

Int CodeError: codigo de error propio que se entrega según sea el caso.

String Message: mensaje relacionado al error o respuesta generada.

Object Data : objeto relacionado o parámetro que identifica el resultado del servicio consumido.