This solution has 2 projects

1. **MeDirect.ExchangeApi**

* I use EFCore 5 Code First as ORM and SQL Server to save data
* Use Redis for cashing user exchanges and rates that users call
* For Authentication users, we can use IdentityServer4. Due to lack of time, I did not use it.
* For logging I used Serilog and save logs in files, we use elastic search to save the logs and Kibana to show the logs
* There is a middleware for logging every request
* I use Swager Library to document the APIS
* I Have written five services, I can use every service used in the separate project as a service

and connect each other with RabbitMQ

1. **User service**: for login and saving user's currency exchange into the database. I have supposed that users before have registered on the site.
2. **Rate API**: as a provider to obtain the latest currency exchange rates.
3. **Caching Rent Service**: for caching the rents which call from Rate API. After 30 minutes service cleans the rate of currency exchange.
4. **Caching User Service**: for caching users' information as used as authentication
5. **Exchange Service**: for Currency exchange, I suppose the transfer of the money has been done in another service, and in this service, we only store the exchange data.

**2. MeDirect.Test**

* For Test, I have used the NUnit library, also I used InMemoryProvider for testing the database so it did not need to mock the database.