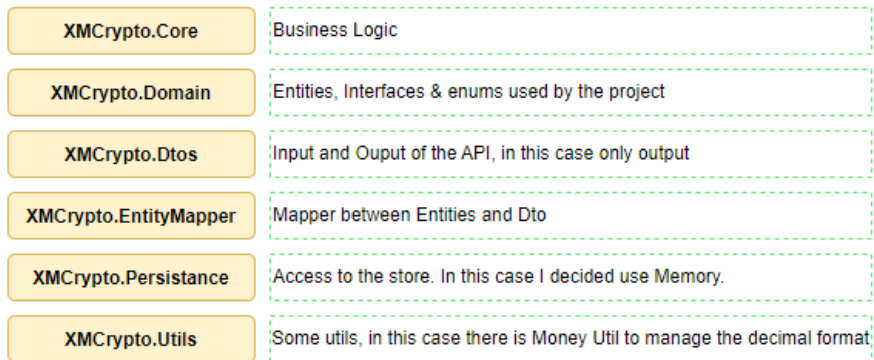


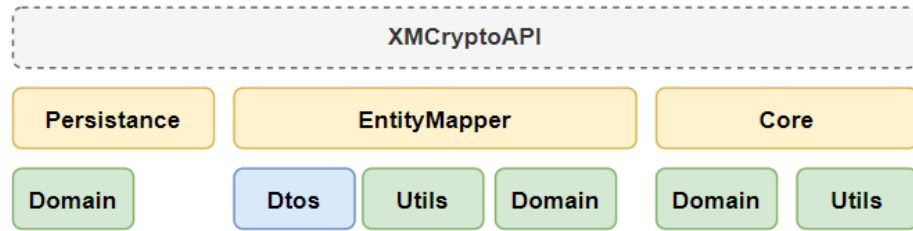
Project Structure

I decided implemented a clean architecture, split the code in different modules

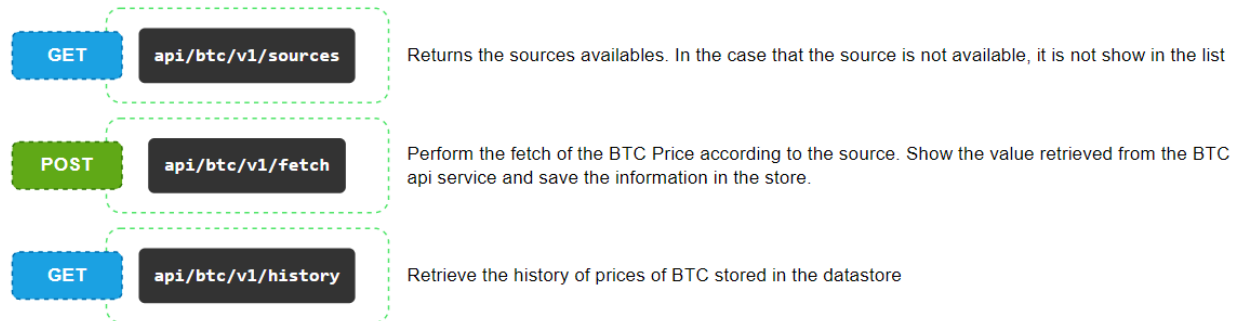


Dependencies

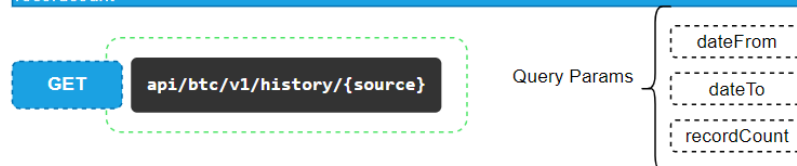
Below is the dependencies between the assemblies



Controllers

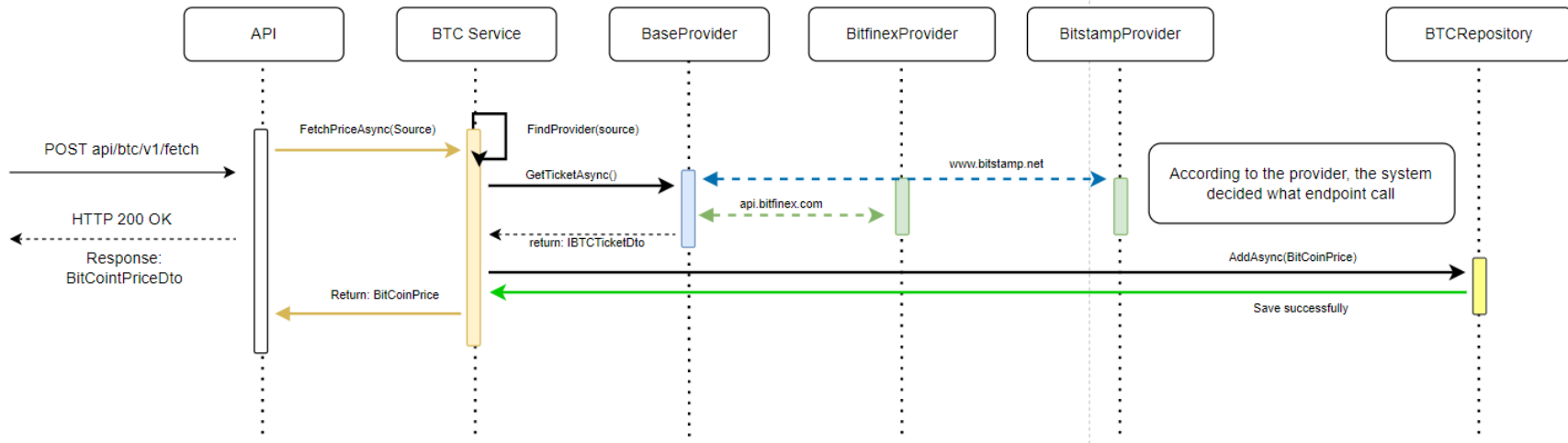


I added an additional endpoint to query the history. This endpoint allow get the information from the history by source, date range filter and recordcount



FETCH API

I would like to make some comments about the more complex API Endpoint: **FETCH**



I created the **BaseProvider** class that allows me to manage the connections with different services. I decided to perform this approach because allows me to keep simple the code of the different providers in the case that the definition of the API matches the definition in my base class. Anyway, if there is a big difference in the new provider, you could override all the methods necessary. On the other hand, we have the code of the **BTCService** class, that, in the case of adding a new provider, there is no necessity to modify this class, because this class is generic and it adapts for new providers.