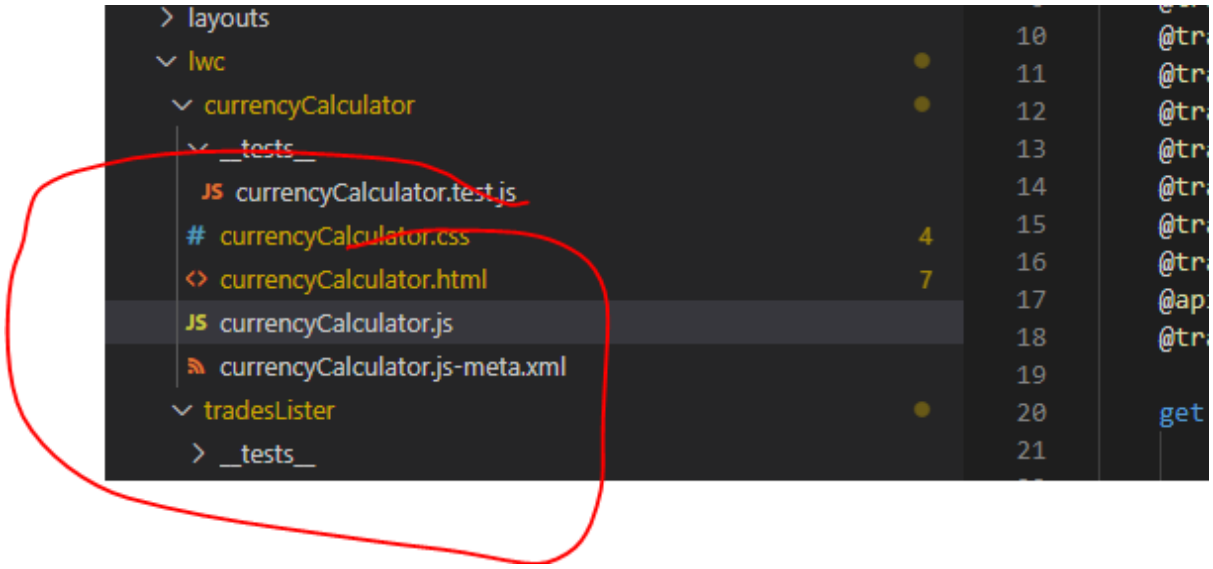
Ebury Test Documentation

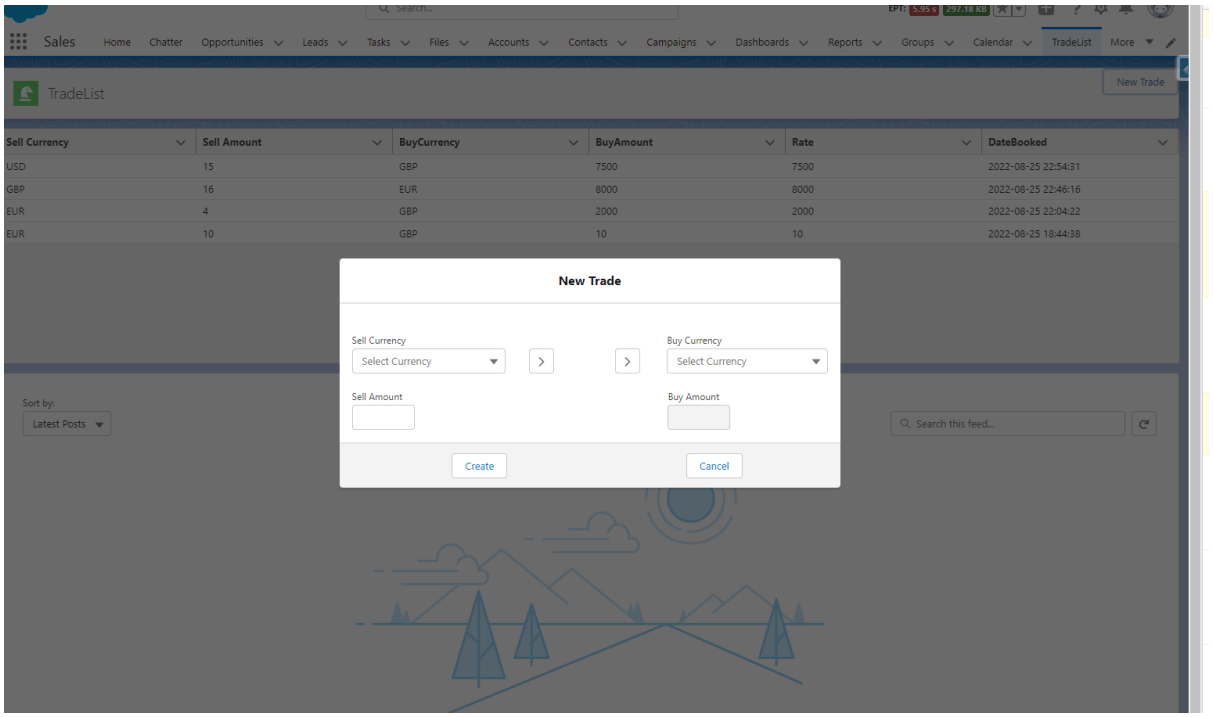
The project actually is designed in two components:

"**CurrencyCalculator**" component: It is responsible for elaborating the logic and it will be opened in a modal.

The LWC component in code:

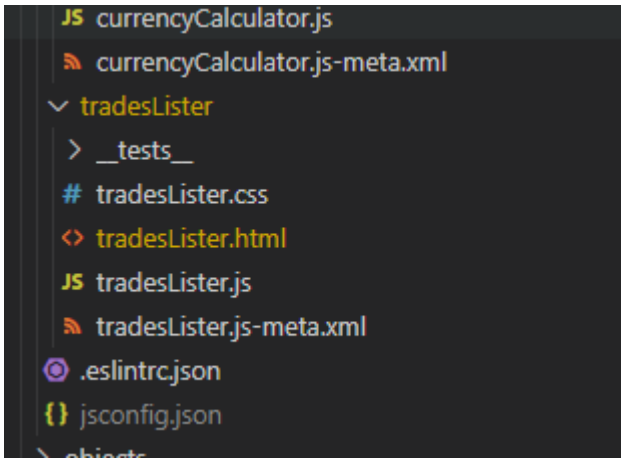


The output of the component:

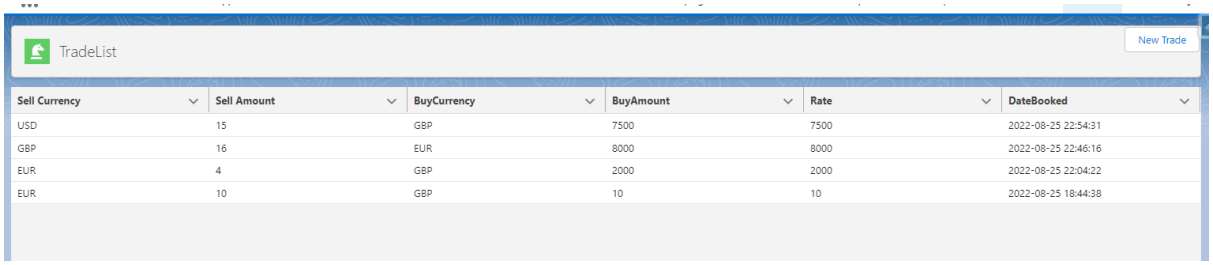


"**TradeLister**" component: It is responsible for lsiting all the trades stored in Salesforce, it is also responsible for opening a modal containing the previous LWC compnonet called "**CurrencyCalculator**" when we click on New trade button.

The LWC component in code:



The output of the component:



<u>Backend Side (Apex</u>

Our backend Architecture is divided in:

Trade Service Apex Class: It is responsible for Creating new trades and also returning Trades stored in Salesforce; Therefore, we have two methods to accomplish the previous concern:

ReturnTrades(): It returns the trades from salesforce.

CreateTrade(): It creates a new trade into salesforce from data coming from LWC component.

FixerIO Caller Apex Class: Respecting the 'Separation of concern' pattern, I choose to make API call from a new different class called FixerIOCaller in order to get data from the API handled by FixerIO platform.

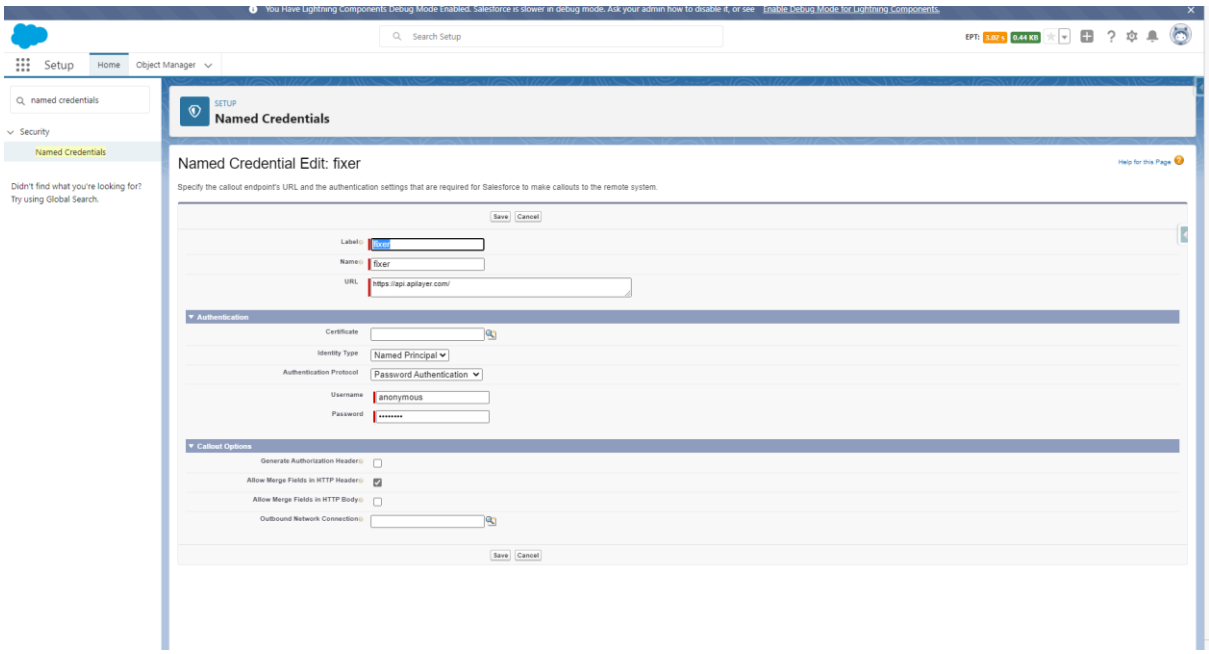
IdFormatter(): it contains an algorithm that creates an Id with the needed format (eg:TR0000055)

Trade Service Wrapper Class: It stands for a wrapper calls in order to perform json parsing.

ChatterPoster Apex: The trigger is triggered after each after insert in the object Trade__c and it handles information for a new created FeedItem object in order to pass it to the Chat feed.

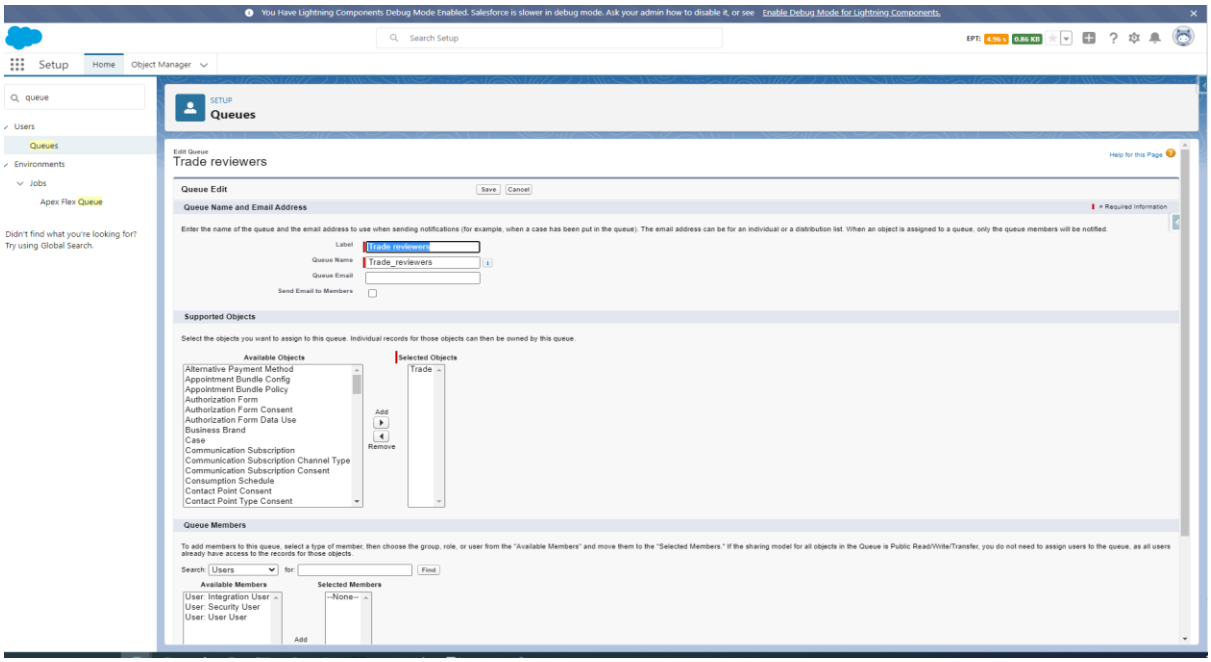
Api Configuration Steps:

We have to configure the named credentials in our environment (Our Scratch org) in order to be able to get data from FixerIO API.



Queue Creation and adding user to it

We have to create a queue named Trade reviewers and assign Trade__c object for the supported objects also we need to add users to the queue.



NB:

Since I was pressed with time (huge work in current work before the end of project this month) I didn't cover some requirements in the test, for instance:

- Implement the exact design in the requirement file.
- I didn't implement test cases for my classes (I can do them I am used to them in my current work)
- I didn't assign the FeedItem to the queue of specific users.

What I can do more?

- Dependency Injection
- More refactoring
- Jest test (LWC unit tests)
- Improve design using SLDS.
- Implement Apex unit tests. (boost test coverage)

NB: I will be happy to join your team. Thanks