**Description Video link**

https://www.youtube.com/watch?v=wgprZL7CHX4&ab\_channel=MichaelTsepelev

**AvailableProducts component**

⁃ Data required to be shown in the table columns are located on different objects fields. So these obects are queried on server side and that data put in a wrapper class to supply the data table.

⁃ Datatable loads data by chunks of records when loadMore method called when user scrolls down the table. Thus it does not have to load all potential thousands of records at startup and freeze the UI.

⁃ When Add button is pressed the component publish certain message type to Lightning Message Channel holding information about selected rows.

⁃ When Second component receive this message it calls apex method to upsert selected records and return calculated Order amount.

⁃ Both components use imperative Apex calls in order to exclude influence of Lightning Data Service caching for certain conditions when cached data is not relevant. So by imperative Apex calls tables show only actual data from database.

⁃ Both components are publishers and subscribers for message channel and process each other events.

**CurrentOrder component**

⁃ Have confirm button which makes HTTP callout to external system which is made by invoking Apex method

⁃ If callout was successfull this method will update Order Status field to Activated and a couple of custom fields that indicate that no more processing should be done for this record.

⁃ If callout was not successful, for example if server returned not 200 code or it was an exception generated for callout the method will update two custom fields that will indicate that this record should be picked up by certain batch job in order to retry the callouts in defined time interval.

**Batch job**

* The job should be run before using the component if we want to retry Confrimation requests to external system in case of network errors
* The job will pick up Order records with Reconfirm = true, resend the request and update certain fields to indicate if retry was succesfull or not