API (incl. Models, DTOS/ClientObjects, and Controllers)

Server Side

View(s)

Model file(s)

Component files (except CSS and HTML), services, resources

HTML

CSS

Model(s)

Controller(s)

Database(SQL)

.Net (C#)

Client Side

**Detailed Diagram of Each Stack Layer (and How to Debug)**

Server side:

Database- holds all data according to business logic, is accessed (CRUD) through .Net./.NET Core, debugged using SQL language (I am unsure)

.Net/.NET Core – interacts with both client side (through API) and database (through SQL), hold business logic in model, repository, and helper files, is debugged in Visual Studio (EQR) or VSCode (Directed Payments) (I think)

Client Side:

Angular-

Models files reflect the data structure that the client has requested.

View (HTML and CSS) are what the client displays and are created for each component.

Controllers are component .ts files, including resource and service files. Resource files interact directly with API (in .Net/.NET Core) to send and receive information (CRUD)

This layer can be debugged using Chrome Dev Tools (That I have used thus far) – Network tab, breakpoints in Sources tab, Console tab, and Elements tab.

Re-Data sent to DQ issue

What I knew before Getting your help was that nothing was changing when I removed “Date sent to DQ”, but “should” have changed. I did not understand Subjects, and Will sent me a simple app to try and understand. I knew the logic for the issue was in .Net meaning, Will and I set up the if statement for the DateSentToDQ (to verify if it exists and the StatusTypeId’s value) to change the statusTypeId to change depending on the outcome of that statement. It wasnt changing, and I was unsure how to debug this situation.

After using Chrome Dev Tools to check what information was coming from the API (through the Network tab), then modifying the RoundCreated Subject to receive a parameter, we discovered the underlying issue was that UpdateRound helper was taking in a RoundAssignment object (the last saved info from the db) instead of a RoundAssignmentView object. RoundAssignmentView would use the info sent from the client side, which is what we needed.

The advice I would give another developer if they experienced the same issue would be to use the Chrome Dev tools to ensure the correct information is coming from the api when triggered, and if not, to debug the Repository (and its Helpers) of that api. Also, I have to admit that I didn’t really understand the different between RoundAssignment and RoundAssignmentView, so learning that difference definitely cleared that info up for me.