Guide to Front-End Development (new-gui)

Introduction:

If you are new to Texera Front-End Development team or have little front-end experience using angular frame work (version 6), this read intends to provide you with a simple guide of how to get started.

Preparation phase:

In a nut shell, angular provides modularity, scalability, and robustness to traditional front-end code design. It separates a website into different individual components that can each perform a certain level of independent tasks. It then connects different components with services so they can work collaboratively. It also provides unit testing at the component level as well as application level.

Other than these, angular largely inherits the traditional way of creating a web page. Each component contains four foundational files (.ts | .html | .css | spec.ts), corresponding to typescript (which is basically JavaScript with better scalability), HTML, CSS, and unit testing respectively. Just like how web pages were traditionally written, you will be coding in

1. html: the structure of the component
2. css: the style of the component
3. typescript: the content of the component

and additionally:

1. unit tests: so that the component can be debugged in the future if it breaks

Don’t be overwhelmed. You don't have to be a master in all these four fields to start working on texera Front-End. If you have basic web development experience, you can jump to next section to get started with learning angular. If you have no such experience, you should at least spend a few hours getting familiar with HTML, CSS and JavaScript. The following links might be helpful.

An overview of HTML: <https://www.youtube.com/watch?v=LcS5IgnAeUs>

An overview of CSS: <https://www.youtube.com/watch?v=Eogk9jWYeMk>

Simple JavaScript example: <https://www.youtube.com/watch?v=LFa9fnQGb3g>

The following links are documentations and examples, don't try to master all the knowledge from these websites at once, use them as dictionaries. They will be helpful when you start coding so don't waste too much time on them now.

HTML: <https://www.w3schools.com/html/>

Typescript: <https://www.tutorialspoint.com/typescript/typescript_overview.htm>

CSS: <https://www.w3schools.com/Css/>

Angular Tutorial Phase:

At this point, you should at least be able to interpret a HTML/CSS/Typescript file with your own knowledge and the information you can find online. For the next few weeks,

1. go through the tutorial provided on Angular official website, <https://angular.io/guide/quickstart>
2. watch tutorial videos, (ask front-end group leader to share the videos with you on google drive)
3. especially pay attention to the rxjs videos, you will need them a lot.

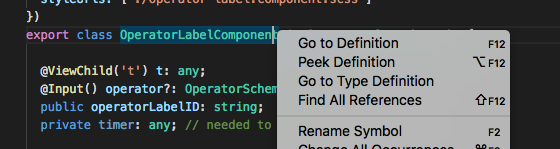
Although these tutorial videos are helpful, it can take a long time to finish watching them. Meanwhile, it is easy to forget what you have learned if you do not practice coding it. Therefore, I recommend you begin the next phase once you finish step 1.

Front-end Code Base:

At this point, you should know how to approach a simple angular application and interpret it using your own knowledge and the information you can find online. Download Visual Studio Code and relevant extensions, get access to Texera front end code base (instructions can be found [here](https://github.com/Texera/texera/wiki/Setup-Development-Environment)). You should:

1. have a general understanding of the structure of the new-gui, what components are there? What do they do? What services are connecting them.
2. You should have a feature in mind that you want to implement. Locate the component and services that are relevant to the feature you want to implement. Carefully read through the code in those sections, make sure you understand what is going on behind the scene.
3. Start coding, then debug, and repeat. ☺
4. Look for solutions in the tutorial videos I mentioned in the previous phase step 2&3 when you have questions.
5. Make good use of google, stack overflow, etc. However, be aware that a lot of code examples online can be outdated since we are using the most recent version of angular with rxjs.

2 useful tricks that you should know how to use.

1. "What does this OperatorLabelComponent do? How is it defined? Where is it used?" Right-click the thing you don't understand, you should find it helpful. 
2. Right-click web page and inspect elements
3. You can Console.log(ThingsYouWantToInspect) in the code base; the logged information will appear in the console window after you do step 2.

Unit testing:

Don’t worry about unit testing at the beginning. Finish the feature first and then write unit tests for it.