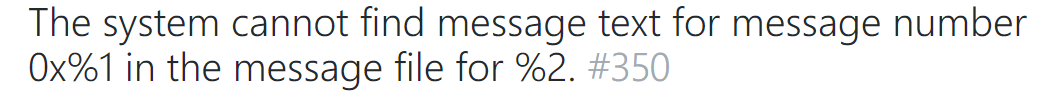
Data Analyzer Electron-Python Desktop App

* In the folder create the app folder where our all css, html, js and python files will be present
* Main folder will contain folders and files:
* App folder
* Main.js
* Package.json
* Package.json can be generated through this command:
* Npm init
* Define all the parameters
* Under dev dependencies mention the electron version
* There will be problem in installing zerorpc package for which you will add this line in your package.json under dependencies
* "zerorpc": "git+https://github.com/exb333/zerorpc-node.git"
* To install all the packages of electron and zerorpc use this command:
* Npm install

**Problems we will face while setting up the system:**

Zeromq error solve by:

Problem:

Solution:

* Before npm.start
* npm rebuild zeromq --runtime=electron --target=1.8.6
* Now npm.start

Zerorpc error solve by:

Add this field below license in your package.json file:

"dependencies": {

"zerorpc": "git+https://github.com/exb333/zerorpc-node.git"

}

Normal installation for zerorpc-node was not working that’s why we forked the repository and using from github

JQuery Problem Solved:

If you want to include JQuery then do this:

Run this command:

npm install jquery –save

Then insert this in your html

<script>window.$ = window.jQuery = require('jquery');</script>

Or

<script> let $ = require("jquery") </script>

Oracle Instant client Problem:

* Pip install cx\_Oracle
* Then download cx\_Oracle instant client64 or 32 bit based on your python version
* Put instant client in C:\\Oracle\\instant\_client\_12\_2
* Set Path in environmental system and user variables.

There will be problem in sending data from One .js file to another .js file or One electron window to another window.:

To solve this problem, we will install another package electron-window using this command:

npm i --save electron-window

The alternative and more easy way I found to send data in form of messages from one js to another:

* Simply use window objects web contents method:

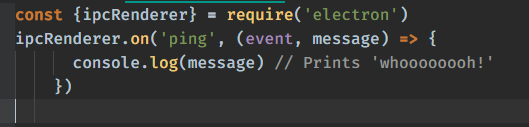


We are using event did-finish-load and result is a variable defined above and is used to collect result value from python in it. Like shown below:



The above code will go from where you want to send data in form of message.

You will retrieve data in this form:



Use the above code in the window where you want to call the data.

<https://github.com/electron/electron/blob/master/docs/api/web-contents.md#contentssendchannel-arg1-arg2->

Another alternative of it is Using IPC main and render:

Sender side:



Receiver Side:



<https://github.com/electron/electron/blob/master/docs/api/ipc-main.md>