DrillingInfo Launcher

Derek Foundoulis, Jake Meister, Cat Wylie

June 16, 2017

Abstract

This document serves to give an overview of the files contained in this project, their functions and elements.

Contents

1 Source

1.1 main.js

main.js is the main JavaScript for the launcher, it creates the browser window and loads mainscreen.html and login.html into the browser, it also deals with cookies. Cookies can not be handled in any other file due to *Electron*.

1.1.1 Functions

- `createWindow()`: Creates and displays the browser window
- `loadConfig()`: Loads the JSON configuration file
- `loadCookies(nameInfo, urlInfo)`: Loads cookies, used by mainscreen.html
- `loadLogInScreen()`: takes no parameters, returns the login screen
- `loadURL(name)`: Load HTML into browser window

- `runJarFile()`: Runs Jar files from the command line
- `storeCookie(data, nameInfo)`: Stores a cookie, used by mainscreen.html

1.1.2 Global Variables

- `cook`: For cookies
- `password`: Stored from login.html
- `username`: Stored from login.html
- `win`: BrowserWindow

1.2 tokenGrabber.js

tokenGrabber.js is a JavaScript that handles parsing a JSON file that is returned from the server and is used by main.html.

1.2.1 Functions

- `buildAndRunCommand()`: A combination of the above functions for testing purposes
- `curlCommandAssembler()`: A string function that builds the curl command based on username and password
- `execCommand()`: Runs a unix command in a child process
- `getToken()`: Runs a specific hard coded curl command that gets the access token from AWS
- `parser()`: Faster than typing `JSON.parse()`
- `printJSON()`: Prints the access token, for display/debugging purposes

1.2.2 Global Variables

- token: used to store the returned token

1.3 renderer.js

A script file that can be used to run in a renderer process

1.4 login.html

login.html is recycled from the online version of DrillingInfo's login page

1.4.1 Functions

These are functions we added to gain functionality with *Electron*

- check(form): This function is where our usernames and passwords are hard coded in, and the mainscreen.html is asked to load by main.js

1.5 mainscreen.html

mainscreen.html is the html for the main page of the launcher, which is accessed after the login screen, if the user has the correct credentials

1.5.1 Elements

- DrillingInfo Logo
- Version number (displays no version if not yet installed)
- Username
- Entitlement icons
- RSS feed for news
- Tools
- Update button
- Loading bar
- Launch button

1.5.2 Functions

- `checkForPatch()`: Checks for a new patch of Transform
- `deleteAllCookie()`: Removes all cookies using IPC
- `downloadFile()`: Downloads file, takes a file URL and target path
- `execCommand(command, callback)`: Executes a UNIX command
- `frame()`: Advances frame in progress bar, called by moveBar()
- `getCookie()`: Returns cookie from main.js using IPC
- `getRequest(requestURL, setJSON, callback)`: Takes a URL, sets it to a JSON then returns body in callback for debugging
- `getVersionFiles(versionObj, releasesIndex)`: Handles downloading of version files and also cross references with local directory
- `greyOut(id, grayOut)`: Changes the availability of the update and launch buttons
- JQuery function which reads tools from JSON and appends them to HTML
- `launch.onclick()`: Run when launch button is clicked, checks entitlements
- `moveBar()`: Animates the progress bar during an update
- `parser(commandOutput)`: Parses entitlement JSON
- `resetProgressBar()`: Resets the progress bar to 0%
- `setAccessToken()` : Sets the access token from tokenGrabber
- `setApplicationsJSON(obj, callback)`: Sets the applicationsTransform JSON from its URL in the config
- `setConfigJSON(obj)`: Sets the configuration JSON
- `setCookie(data)`: Sets cookie in main.js using IPC
- `setLaunchCommand(releaseIndex)`: Sets command that the launch button uses to execute the transform_X.Y.Z.jar
- `setPlatformAndApplicationURL(callback)`: Sets platform and application URL from the config
- `setPlatformJSON(obj)`: Sets the platform JSON to the one from the server

- `setVersionsJava(obj)`: Sets the java versions JSON to the one from the server
- `setVersionsTransform(obj)`: Sets the current version of Transform
- `showImage(id)`: Takes a element ID and shows the image. Used for entitlements
- `showProgress(received, total)`: Logs download progress
- `tokenHandler()`: Grabs token from server and updates entitlements
- `updateApplication()`: Updates the transform Jar when update is pressed
- `update.onclick()`: Run when update button is clicked
- `updatePlatform()`: Updates java, automatically called when launcher starts
- `updateTransform()`: Updates Transform, called when update button is clicked.
- `updateUsername(name)`: Updates the username.
- `updateVersionNumber(number)`: Used to update the version number
- `userAccess(entitlements)`: Cross references the user's entitlements with the argument[]

1.6 login.css

login.css is recycled from the online version of DrillingInfo's login page. We did not edit this CSS.

1.7 mainscreen.css

This is the style for mainscreen.html.

1.7.1 Styled Elements

- DrillingInfo Logo
- Version number
- Tool Tips
- Username
- Entitlement icons
- Footer
- RSS feed for news
- Tools
- Buttons
- Update
- Loading bar
- Launch
- Body

1.8 main.css

main.css is recycled from the online version of DrillingInfo's login page. We did not edit this CSS.

1.9 patternfly.css

patternfly.css is recycled from the online version of DrillingInfo's login page. We did not edit this CSS.

1.10 zocial.css

zocial.css is recycled from the online version of DrillingInfo's login page. We did not edit this CSS.

2 Icons

The icon directory of this project contains both usable and filler icons and logos, these are outlined below:

- beta_1.png: filler
- di_logo_login.png: usable
- icon.png: usable
- support_1.png: filler
- transform.png: filler

3 Modules

Modules that weren't part of the base node.js:

- `async`
- `mkdirp`