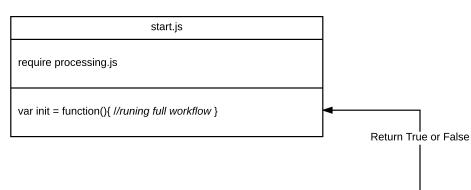
CPA JS Framwork - July 2017 Draft





Helper Part

Helper function for whole framework

constant.js

hostname: qa.emory.edu

githubHost: github.com/emorycpa

helper.js

Error Handing: errorHanding(error) postMessage(format)

Processing Part

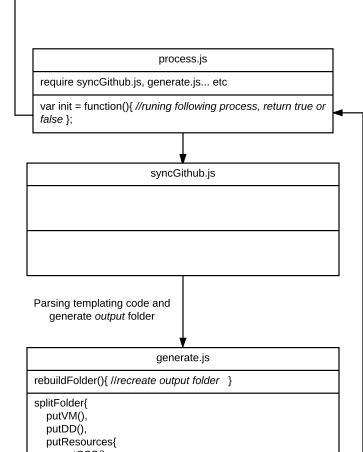
If any callback return False/Error, return false to process.js, then pass error message to start.js

entry point for processing

1. **Fetch/Merge** with files (both front-end and back-end) on Github to sync with other's code

This way garentee Github only has templating code, instead of DD/VM code. DD/VM code is only needed in Cascade Server.

- 1. Running template engine and generate DD/VM code (future) (Option: parse SCSS and TypeScript code to CSS and JavaScript)
- 2. Move generated files to corresponding folder (*output*):
 - 1. velocity -> vm folder
 - 2. data definition -> dd folder
 - 3. resources (css/js..) -> _includes/css, _ includes/js...folder
 - 4 HTML -> hase folder



I/O: readLocalFile(fileName, filePath)

- 3. resources (css/js..) -> _includes/css, _ includes/js...folder 4. HTML -> base folder

2. data definition -> dd folder

- 1. Read local keys: user name/password
- 2. Send request to Cascade 8 to log in If successfully login, allowing following methods runnning, show success message in terminal If fail, showing error message in terminal

- 1. Read corresponding folder in Cascade 8, then get a list of files in this folder.
- 2. Compare list of files in folder with local output folder
- 3. For all files which has been deleted on local machine, run delete method to delete these files in Cascade 8
- 4. For all existing files in *output* folder, run *create* method to add or overwrite these files in Cascade 8

```
putDD(),
   putResources{
      putCSS(),
      putJS(),
      putOthers([fonts,cfm,php...])
  putHTML(),
   Authentication Process
                        authenticate.js
readLocalKey(keyName, pinCode)
init = function(){ //sendRequest to Cascade API}
  For each folder in output
folder, sync with Cascade 8
                        syncCascade.js
processFoloder([vm, dd, resources]){
     //run following methods to process each folder in arrary
parameter
var cascadeFileList= readCascadeList();
var localFileList = readLocalList();
deleteFiles(fileList) {
  foreach file in cascadeFileList:
         if not in localList, delelteFile(file);
  return true;}
deleteFile(file, path): delete single file
if deleteFiles == true
   addFiles(fileList) {
       foreach file in localFileList:
         addFile(file);
  return true;}
add(file, path): add single file
                                     -Return True or False-
```

putvivi(),