

MySite10- Design Specification

This document defines the design of MySite10.

1. Introduction

This Design Document defines the design of MySite10. It consists of the logical flow relative to modules. Modules

2. Node.JS / Express Modules

This section defines the folder/file structure.

2.1. Express Application Generator

Use the application generator tool, express-generator, to quickly create an application skeleton.

The express-generator package installs the express command-line tool. Use the following command to do so:

```
$ npm install express-generator -g
```

2.2. Folder/File Structure

- settings
 - launch.json – Visual Code (vc) configuration file
- bin
 - www – This is the START of everything. bin/www is what is executed to start this Node.js/Express Server.
- public
 - data – These are files that are accessed
 - cloudService.txt – Where the Download Cloud Service Photos and Videos Selection is saved
 - client_secret_8141.....apps.googleusercontent.com
 - configure.json – User Configuration in this file
 - favicon.ico - Icon

mysite.htpasswd – Username/Password for Photos and Videos pages
(if configured)

resultsOfLastDownload.json – Results of last download that is
displayed when menu selected

users.htpasswd - Username/Password for Download pages

usersGuide.pdf - User's Guide displayed from Contact Page

- images

- cs – Everything in this folder is downloaded from the Cloud
Service using the Download Page

- Photos

- Videos

- banner.jpg

- logo.jpg

- Photos01-900x300.jpg

- Photos02-900x300.jpg

- Photos03-900x300.jpg

- Photos04-900x300.jpg

- Photos05-900x300.jpg

- Photos06-900x300.jpg

- rightsidenav.pug

- txt_about_contents.pug

- txt_contact_contents.pug

- txt_index_contents.pug

- txt_subbanner.pug

- js - This folder contains all the JavaScript modules

- skin – images and css for jPlayer

- stylesheets – Style Sheets for Magnific-popup and MySite10

- routes

- about.js

- contact.js

- download.js

- index.js

- photos.js

- pleasewait.js

- videos.js

- views
 - includes
 - altnav.pug
 - footer.pug
 - include.pug
 - init_rightside_nav.pug
 - initialize_cloud_service.pg
 - navjs.pg
 - rightsidenav.pug
 - slider.pug
 - topnav.pug

about.pug
contact.pug
error.pug
index.pug
initialize.pug
layout_photos.pug
layout_videos.pug
layouy.pug
layoutInitialize.pug
messages.pug
photos.pug
pleasewait.pug
videos.pug

app.js

package.json

3. Flow Charts

bin/www

Create the HTTP/HTTPS Server
Create io.sockets – Used to display progress of Download

app.js

Global variables
Initialize all the /view Pages and /routes

configure.js

Configure mysite10 based on public/data/configure.json

router/index.js
router/photos.js
myJsRoutines.js
router/videos.js
router/contact
router/download.js
download_cs.js
auth.js
writeDownloadreslts.js
router/pleaseWait.js
www Listening on Port

4. Download

The Download Function, creates the same folder/file structure from the Cloud Service selected, to the file system on the server in the public/cs folder. Only the following file extensions are downloaded:

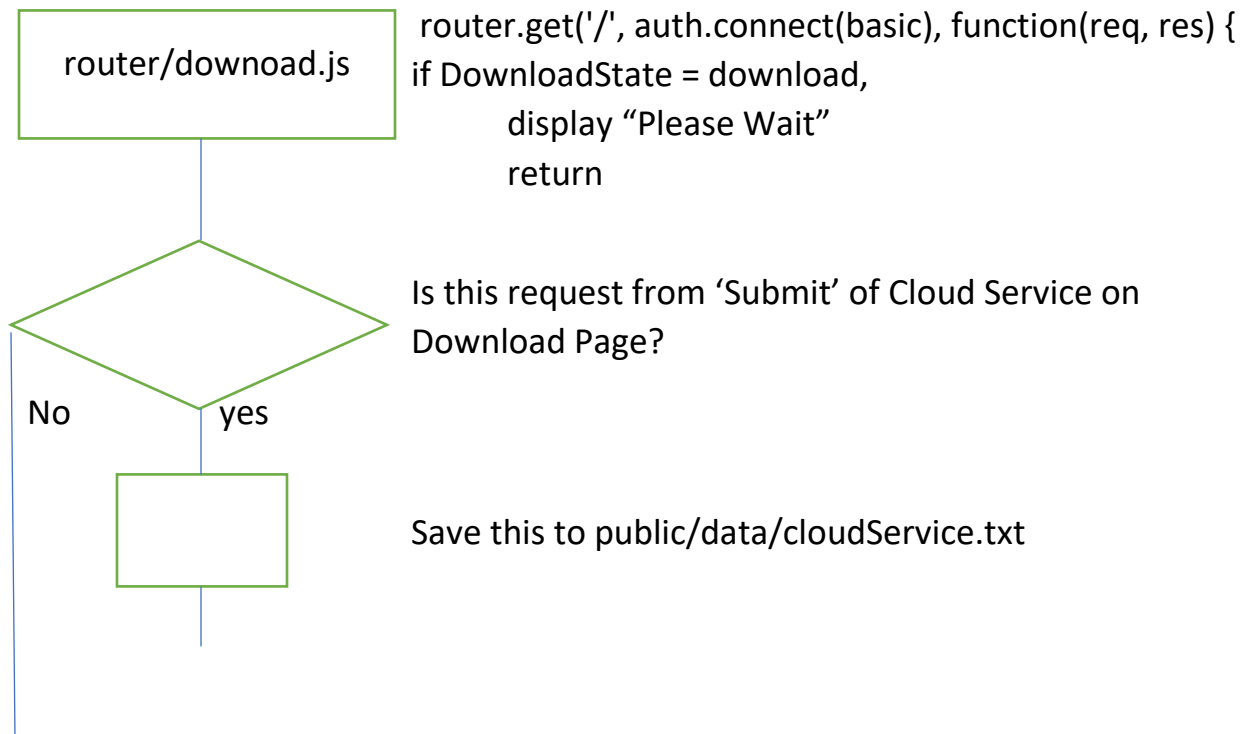
- pug – This is for these files (txt_about_contents.pug, txt_contact_contents.pug, txt_index_contents.pug, txt_subbanner.pug) that are located in the myWebsite folder on the Cloud Service

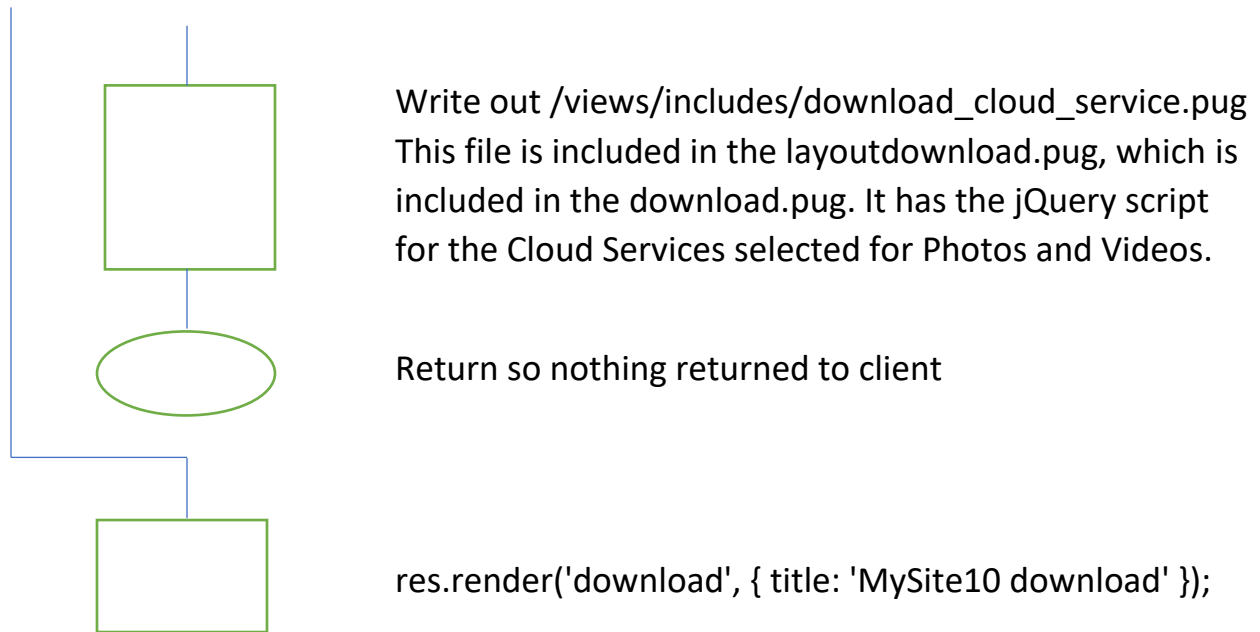
- txt – These are the .txt files that support Photo captions and Video files Playlist
- pdf – Document files that are displayed
- jpg – Photo image files
- jpeg – Photo image files
- png – Photo image files
- gif – Photo image files
- tiff – Photo image files
- bmp – Photo image files
- m4v – Video files
- ogv – Video files
- web – Video files

Apostrophes (') are removed from folder and filenames

Once the download has completed, the pugUpdate function will create the altNav.pug, leftSideNav.pug, and include.pug files in every folder.

4.1. download Flow





4.2. pugUpdate Flow