****

|  |
| --- |
| **Developer Documentation**  **Data Creation Website** |
|  |

\***A** = Added / **M** = Modified / **D** = Deleted

# Revision History

| **Version** | **Date** | **Author** | **A/M/D\*** | **Comments / Title / Brief Description** |
| --- | --- | --- | --- | --- |
| .1 | Aug 25, 2017 | Joshua Sun | A | First version |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\***A** = Added / **M** = Modified / **D** = Deleted

# Contributors

| **Name** |
| --- |
| Joshua Sun |

Contents

[Revision History 2](#_Toc491087582)

[Contributors 2](#_Toc491087583)

[Table of Contents 3](#_Toc491087584)

[1 Purpose 4](#_Toc491087585)

[2 Technologies used 4](#_Toc491087586)

[3 Installation 4](#_Toc491087587)

[3.1 Prerequisites 4](#_Toc491087588)

[3.1.1 Version Check 4](#_Toc491087589)

[3.1.2 ‘Command Not Found’ 4](#_Toc491087590)

[3.2 Cloning From Git 5](#_Toc491087591)

[3.3 Installing dependencies 5](#_Toc491087592)

[3.3.1 Global dependencies 5](#_Toc491087593)

[3.3.2 Project dependencies 5](#_Toc491087594)

[3.3.3 Confirm that dependencies are installed correctly 5](#_Toc491087595)

[3.3.4 Adding more dependencies to this project and installing them on Jenkins 5](#_Toc491087596)

[3.3.5 Essential dependencies explained: 6](#_Toc491087597)

[4 Runing the server 6](#_Toc491087598)

[5 Developing 7](#_Toc491087599)

[5.1 Files and File structure 7](#_Toc491087600)

[5.2 The workflow 7](#_Toc491087601)

[5.3 Backend development 7](#_Toc491087602)

[5.3.1 Server 7](#_Toc491087603)

[5.3.2 Database 7](#_Toc491087604)

[5.4 Frontend development 8](#_Toc491087605)

[5.4.1 Wrting JS - React JS 8](#_Toc491087606)

[5.4.2 Writing HTML 8](#_Toc491087607)

[5.4.3 Writing CSS 8](#_Toc491087608)

[6 Version Control and Deployment to Jenkins 8](#_Toc491087609)

[6.1 Commiting to Git 8](#_Toc491087610)

[6.2 Deploying on Jenkins 9](#_Toc491087611)

[6.2.1 MongoDB on Jenkins 9](#_Toc491087612)

[6.2.2 Dev deploy 9](#_Toc491087613)

[6.2.3 Production deploy 9](#_Toc491087614)

[7 Running Tests 9](#_Toc491087615)

[8 Useful Links 9](#_Toc491087616)

[9 Acknowledgments 9](#_Toc491087617)

[10 Licence 10](#_Toc491087618)

# Purpose

This is the official documentation of the Data Creation web application. The aim of this documentation is to give you a detailed look into the tools and technology used to build this tool, and ways to maintain it.

# Technologies used

Frontend

* JQuery
* Various jQuery plugins, including jsGrid and jQuery dialog
* HTML 5, CSS

Backend

* Node.js And Express
* Microsoft Access Database, driver Node-Adodb

Build tools

* Yarn, Browserify, Browserify-directory NPM-scripts

Version control

* TFS Git

# Installation

## Prerequisites

Make sure you have Node.js, NPM, and Yarn installed. These can be obtained from their official websites. Continue once you have these installed.

### Version Check

On Windows, open Git Bash

On MacOS or Linux, open Terminal

Confirm the following:

$ node -v

7.9.0 (or greater)

$ npm -v

4.2.0 (or greater)

$ yarn --version

0.27.5 (or greater)

### ‘Command Not Found’

If any of the above commands give ‘command not found’ and you have installed the tool, it means that you have not set the PATH variable correctly. To configure you system environment variables, see <https://www.java.com/en/download/help/path.xml>

## Cloning From Git

To clone this project ot your local, run

$ git clone <http://tfs.omers.com/tfs/Production/LEAP/_git/automation_tools>

$ cd DataCreation

You are now at the root folder of the Data Creation project

## Installing dependencies

Confirm that you are at the root folder of the Data Creation project, if not, see 3.2

### Global dependencies

These are command line build tools needed to compile our project, on bash, run:

$ yarn global add nodemon browserify browserify-directory

### Project dependencies

These are project specific dependencies, run:

$ yarn install

the Yarn package manager will install all dependencies automatically. Unlike global dependencies, project dependencies are listed under package.json

### Confirm that dependencies are installed correctly

Run:

$ node app.js

If you get ‘server started on port 3000’, then you are all good to go. Otherwise you have not setup the project correctly.

### Adding more dependencies to this project and installing them on Jenkins

Since our Jenkins server does not have internet access, Yarn install will not work properly. To solve this issue, we must install dependencies in offline mode. This setting is written in .yarnrc file in the project’s root folder. This file tells Yarn that whenever you run $ yarn add to add dependencies, save a compressed version of the dependency in a folder called npm-packages-offline-cache. This folder is uploaded to git along with the project so the dependencies can be installed correctly on Jenkins. For this project, this has already been set up so you don’t need to worry about it.

### Essential dependencies explained:

Build dependencies

These are listed under ‘dev-dependencies’ in package.json and will not be installed on our production environment on Jenkins

* Nodemon: watches for changes in our project and restarts the server automatically
* Eslint: a linter that checks for JavaScript syntax and style errors
* Browserify: bundles your JavaScript files into a single bundle.js and enables CommonJS modules on the frontend
* Forever: restarts server if it crashes
* Mocha: runs unit tests
* Browserify-directory: allows us to use browserify and output multiple files

Server and Frontend dependencies:

These are dependencies needed for the server and frontend to run. These must be installed on Jenkins.

* Express: powerful framework for Node that abstracts away many low level server operation
* EJS: HTML templating engine, compiles .ejs files into HTML files
* Body-parser: parses the request body into a JS object in req.body
* Moment: Date & time formatting library
* Node-adodb: database driver that works with Microsoft access
* JQuery: library for manipulating the DOM, attaching event listeners and making AJAX requests

# Runing the server

## Vanilla Server

There are a number of ways to start the server, at the most basically level, you can run:

$ node app.js

In the project’s root folder.

## Compiling Javascript files

This project uses CommonsJS modues in the frontend. This means we must compile the JS files using Browserify before the browser can read these files. This is written as a NPM script:

Run:

$ npm run build-js

This replaces all the ‘requires’ in ./src/js/\*.js with the actual code that they are importing and outputs to ./public/js

NOTE: this step is essential if you change any files in ./src since without it, files in ./public will not be updated!

## Passing in Environment Variables

It is easy to pass in environment variables. For example, to change port:

$ PORT=1234 node app.js

# Developing

## Files and File structure

* Node\_modules – where dependencies are saved locally, not uploaded to git
* Doc – project planning and documentation files
* Public – static assets such as Javascript, images and CSS
* Routes – route handler functions
* Src – precompiled JS file, need browserify to be outputted to ./public folder
* Views – EJS files (precompiled HTML files)
* .eslintrc – linter configurations
* App.js – entry point of our server
* LICENSE – this project uses GPL 3.0 freeware license
* Package.json – lists all denpendencies and project meta information
* README.md – markup files of the documentation
* Yarn.lock – auto generated by yarn

## The workflow

This section will walk you through how this project is compiled when you type

$ npm run build-js

1. Goes to ./src/js and reads all the JS file.
2. Whenever a ‘require’ keyword is seen, Browserify travels down the dependency tree and import all the code to replace the ‘require’.
3. Finally, output js files are saved in ‘./public/js’ and they can be included in our html files

## Backend development

### Server

The Node.js server is responsible for routing and database queries. The server setup file is ./app.js, the route handlers and database queries are written in files in ./routes, and the database Schema are in ./model

For documentation on the Express framework, see <https://expressjs.com/>

### Database

This app uses Microsoft Access database. This is not a fully supported backend in NodeJS, but the Node-Adodb library provides some access for this backend. In the future we will migrate to MongoDB

## Frontend development

### Wrting JS

This project takes the traditional approach of using jQuery and Ajax calls on the frontend. All js source files are in ./src and MUST BE COMPILED BY BROWSERIFY by running $ npm run build-js.

### Writing HTML

You can write HTML in ./views. The ejs files are used by the ejs template engine to compile to HTML.

To see documentation on EJS, go to: <http://ejs.co/>

### Writing CSS

This project uses css, which are located in public/css

# Version Control and Deployment to Jenkins

## Commiting to Git

Make sure all new features are commited on the dev branch and thoroughly tested before merging to the master branch. Once you are finished making changes, issue the following commands

$ git add .

$ git commit -m “commit message”

$ git fetch

$ git rebase

$ git push

After $ git rebase, you should get fast-forward. Merge conflicts are rare and if they do come up, you are probably not commiting your changes right.

## Deploying on Jenkins

### Dev deploy

After pushing your changes to the dev branch go to

<http://10.115.17.17:8080/view/Automation%20Tools/job/data_creation_dev/>

And run the job, this starts the dev server on port 3001

### Production deploy

After pushing your changes to the dev branch go to

<http://10.115.17.17:8080/view/Automation%20Tools/job/data_creation_master/>

And run the job, this starts the dev server on port 3000

# Running Tests

Units tests for this project are currently unavailable. If you would like to write unit tests, I recommend using the Mocha library:

<https://mochajs.org/>

# Useful Links

* Git repository URL: <http://tfs.omers.com/tfs/Production/LEAP/_git/EnrollmentDataCreation>
* Node.js documentation: <https://nodejs.org/en/docs/>
* Express.js documentation: <https://expressjs.com/>
* React js documentation: <https://facebook.github.io/react/>
* SASS documentation: <http://sass-lang.com/>
* MongoDB documentation: <https://docs.mongodb.com/?_ga=2.191918509.667046423.1503338162-1605050878.1503338162>
* Mongoose documentation: <http://mongoosejs.com/>
* Yarn documentation: <https://yarnpkg.com/lang/en/>
* EJS documentation: <http://ejs.co/>
* Javascript ES6: <http://dev.venntro.com/2013/09/es6-part-1/>

# Acknowledgments

* Parasar Saha for coming up with the idea
* Jannie Zheng for writing the Selenium job to dump data

# Licence

This project is licensed under the GNU Public General License Version 3, see LICENSE.md for details