*Project Structure / Directory Outline*

IN MAIN JAVA FOLDER

- com.revature.assignforce

\* *AssignForceV2Application.java* - Application entry point: com.revature.assignforce.domain.dao is JpaRepository and com.revature.assignforce.domain scanned for components

\* *PersistenceContext.java* - Context:

DataSource defined, entityManagerFactory defined, transactionManager defined

- com.revature.assignforce.domain - POJOS

\**Activatable.java* - Interface, meaning an item can be active or inactive

*\* Batch.java* - Single batch info

*\* BatchStatusLookup.java* - NOTE: not currently being used

*\* Building.java -* Physical building / campus. Location has many buildings

*\* Certification.java -* Trainer’s certification information. Contains a URL and name, as well as trainer id.

*\* Curriculum.java* - Set of skills a batch will learn, such as Core Java or SDET, or Capital One focus

*\* Location.java* - Batch main location

*\* Room.java* - Batch room. Building has many rooms

\* *Setting.java* - Settings saved in the db. Each setting has an id, name, and value.

*\* Skill.java* - represents the skill table in the database. Holds an id (skill Id), name, and active.

*\* Trainer.java* - trainers and co-trainers, Holds a trainerId, firstName, lastName, resume, and active. Contains a list of unavailability, skills, and certifications.

*\* Unavailable.java* - This one is special - a list of two dates: start of range and end of range.

- com.revature.assignforce.domain.dao

\* *ActivatableObjectRepository.java* - Fancy file by August - no actual content, but anything activatable should extend this interface, such as a room or a trainer.

*\* BaseRepository.java* - Anything not activatable should extend this interface.

*\* BatchRepository.java* - extends BaseRepository

\* *CertificationRepository.java* - extends BaseRepository

*\* CurriculumRepository.java* - extends ActivatableObjectRepository

*\* LocationRepository.java* - extends ActivatableObjectRepository

*\* RoomRepository.java* - extends ActivatableObjectRepository

\* *SettingRepository.java* - extends BaseRepository

*\* SkillRepository.java* - extends ActivatableObjectRepository

*\* TrainerRepository.java* - extends ActivatableObjectRepository. There was a note that this is necessary for Spring Data. Also contains:

Trainer findByFirstName(String name);

List<Trainer> findByActiveIsTrue();

*\* UnavailableRepository.java* - extends BaseRepository

\* text file descriptor -“*dao stuff goes here.txt*”

- com.revature.assignforce.domain.dto

\*\*DTO’s reflect the POJO’s. They are the in-between. In our REST controllers, we accept respective DTO formats coming from our javascript files. The major differences between the POJO’s and the DTO’s is that DTO’s do not contain defined constructors, and they are not mapped to database tables.

\* *BatchDTO.java*

*\* BuildingDTO.java*

*\* CertificationDTO.java*

*\* CurriculumDTO.java*

*\* LocationDTO.java*

*\* ResponseErrorDTO.java -* This one is special. It has no associated mapped POJO, and so it has a defined constructor. It’s only item, however, is a String object, errorMessage.

*\* RoomDTO.java*

*\* SettingDTO.java*

*\* SkillDTO.java*

*\* TrainerDTO.java*

*\* UnavailableDTO.java*

- com.revature.assignforce.service

\* *ActivatableObjectDaoService.java* - intention - to inactivate activatable objects instead of deleting them.

*\* AuthService.java* - intention - intention - to handle authorization via token passed by Parasol.

*\* BatchDaoService.java* - Uses DaoService

\* *CertificationDaoService.java* - Uses DaoService

*\* CurriculumDaoService.java* - Uses ActivatableObjectDaoService

*\* DaoService.java* - handles DAO operations on BaseRepository POJO’s

*\* LocationDaoService.java* - Uses ActivatableObjectDaoService

*\* RoomDaoService.java* - Uses ActivatableObjectDaoService

\* *SettingDaoService.java* - Uses DaoService

\* *SkillDaoService.java* - Uses ActivatableObjectDaoService

*\* TrainerDaoService.java* - Uses ActivatableObjectDaoService, adds method:

Trainer findByFirstName(String name

\* *UnavailableDaoService.java* - Uses DaoService

- com.revature.assignforce.web

*\* AuthController.java* - intention - use for authorization via token passed by Parasol

\* All other REST Ctrl’s are for translating javascript objects via DTO to their corresponding POJO’s (or vice versa), and make calls to the db to update, delete, create new, or retrieve records.

*\* BatchCtrl.java - \**

*\* BuildingCtrl.java - \**

*\* CurriculumCtrl.java - \**

*\* DefaultRoutingController.java* - used for redirecting to the index page when certain other URL’s are attempted. Also looks like there is intended use for Parasol via healthCheck method. \*\*\*\* Not a REST controller

\* *gCalCredentialsCtrl.java* - \* Special case, does not require DTO. It is used to get credentials for the shared Google Calendar.

*\* LocationCtrl.java - \**

*\* RoomCtrl.java - \**

*\* S3CredentialsCtrl.java - \* Special case, does not require DTO. It is used to get credentials for the S3 bucket.*

*\* SettingCtrl.java - \**

*\* SkillCtrl.java - \**

*\* TrainerCtrl.java - \**

*\* UnavailableCtrl.java - \**

IN RESOURCES FOLDER

- public/

-*index.html* - SOURCE REFERENCE FOR ALL JS SCRIPTS GO HERE

- css/ - contents not listed for brevity

- html/

- templates/

- *batchGenTemplate.html* - For use with reports tab

- *gradTableTemplate.html* - For use with reports tab

- *incomingTableTemplate.html* - For use with reports tab

- dialogs/

*- buildingDialog.html* - Pop up box for adding a building

- calendarDialog.html - Modal for embedded google calendar

- *deleteDialog.html* - Pop up box to confirm deletion of a location

*- locationTemplate.html* - Pop up box for adding a location

- ptoRequestDialog.html - Modal for submitting a pto request

*- roomDialog.html* - Pop up box for adding a room

- *skillDialog.html* - For adding a skill

- *trainerDialog.html* - For adding a trainer

- toolbars/

- *batchGenToolbar.html* - For use with reports tab

- *gradTableToolbar.html* - For use with reports tab

- *traineeTableToolbar.html* - For use with reports tab

- views/ \*These represent the different tabs of the application

- *batches.html* - Batches tab - This is the batch creation/graph page

- *curricula.html* - Curricula tab - For viewing curricula, creating foci, creating skills

*- home.html* - Overview tab - Landing page

*- locations.html* - Locations tab - Lists locations, buildings, and rooms

*- menu.html* - This is the top button-menu (tabs set) on every page

- *profile.html* - Profile tab - intention - to view the currently signed-in trainer’s page

*- reports.html* - Reports tab - Highcharts/predictions reporting

- *settings.html* - Settings tab - For saving global settings

*- trainers.html* - Trainers tab - Lists trainers and their skills

- img/ - revature logo, etc. contents not listed for brevity

- js/

-*assignforce.js* - listed at the bottom -

-controllers/

- dialogs/

- *bldgDialogController.js* - Data and functions for buildingDialog.html

*- deleteDialogController.js* - Data and functions for deleteDialog.html

*- locationDialogController.js* - Data and functions for locationDialog.html

*- roomDialogController.js* - Data and functions for roomDialog.html

- *skillDialogController.js* - Data and functions for skillDialog.html

- *trainerDialogController.js* - Data and functions for trainerDialog.html

- *authController.js* - intention - control user access and facilitate user authorization

*- batchController.js* - D&F regarding the batch creation section and the batch list of the Batches tab.

- *curriculaController.js* - D&F regarding the Curricula tab.

*- homeController.js* - D&F regarding the Overview tab, such as filtering and csv export.

*- locationController.js* - Handles CRUD for locations via Locations tab.

*- menuController.js* - Has one function which: “sets navbar to current page…”

- *profileController.js* - D&F regarding Profile tab, such as uploading certs.

- *ptoController.js* - controls google calendar for pto’s.

*- reportController.js* - D&F for reports page/generation, for Reports tab.

- *settingsController.js* - D&F for the Settings tab.

*- timelineController.js* - D&F for graphical timeline on Batches tab.

*- trainerController.js* - D&F for the Trainers tab.

- directives/

- *fileUploadDirective.js* - not written or used by our team

- *mdDataTableDirective.js* - “ “

*- reportsDirectives.js* - “ “

- filters/

- *activeFilter.js* - filters items based on their active attribute.

*- availableModeFilter.js* - not written or used by our team

- *availableSelectionFilter.js* - filters items (currently rooms or trainers) based on a batch’s selected dates and the items’ sets of unavailabilities.

*- timeModeFilter.js* - not written or used by our team

- services/ \*Most services are used to access REST methods for each POJO

*- batchService.js* - \*

- *buildingService.js* - \*

*- calendarService.js* - calculator service - provides number of weeks / days /

milliseconds between two given dates and returns a new date when passed in

milliseconds.

*- curriculumService.js* - \*

- *gCalCredentialsService.js* - fetches credentials for Google Calendar

*- locationService.js* - \*

- ptoService.js - handles google api calls

*- roomService.js* - \*

- *S3CredentialsService.js* - fetches credentials for S3 Bucket

- *settingService.js* - \*

*- skillService.js* - \*

*- trainerService.js* - \*

*- unavailableService.js* - \*

*- userService.js* - not written or used by our team. Regards user authorization.

-misc/

- *d3-tip.js* - Used for the tooltip when you mousover batches on the timeline on the Batches tab.

- *highCharts.js* - For wherever highcharts are used (for now, the Reports tab)

- *typeahead.js* - placeholder for typeahead functionality for the future.

- *application.properties* - server port to run from, db configuration, hibernate configuration. Note: you can run the project from straight localhost if your port is set to 80.

IN TEST JAVA FOLDER

- com.revature.assignforce

- AssignForceV2ApplicationTests.java - placeholder for testing class

*Shared link* in Google Docs, for real-time editing viewable / editable by all: <https://docs.google.com/document/d/1EYpdUfsXGqMF4PSzXNuD9DyT7VjOYAgnUgmfuJEVJ40/edit?usp=sharing>

Make sure to update the documentation within the project folder as well after significant changes have been made, preferably by one person.