

CoMPAS OpenSCD (next) migration notes

Work in progress: This is not a definitive or comprehensive list

Identified actions:

- ~~Create a component(s) repository within the CoMPAS org~~
- ~~Move Compas-open and Compas-save to the same repository~~
- Compas-loading should maybe be in another repo (util repo?) (define util repo)
- ~~Remove CompasUploadVersion? // No~~
- Refactor save/saveAs? (code duplication) (define solution)
- ~~Should the compas foundation files and services exist in the compas-open-scd(next) repository or should they:~~
 - ~~a) be split between each repo where their dependent on (example: save and open depend on compas-sel-data-service so it's put under that repo)~~
 - ~~b) be group into a single repo "compas-oscd-data-core" where all foundation files and services exist~~
- Cleaner API for accessing the database (services) that allows for separating this logic from the compas components (define interfaces)
- Move existing tests properly
- New repos to fulfill this migration need to have admins and the admins should be updated before a team member leaves
- Once all of the dependencies are fully extracted from compas-oscd-docops, a review of unused package.json dependencies should be done
- Check if we can extract a file from /compas/foundation.ts that only has the functions needed by compas-oscd-docops components/plugins.

Proposed questions by [Jakob Vogelsang](#) :

Question to be answered:

3. are all tests complete, do we need screenshot test?

*4. is the API complete, do we need to reduce/add **properties**, methods **slots**?*

Definition of **compas-oscd-docops**:

This a web components' repo where the following elemental UI components from compas-open-scd exist alongside a set of interfaces that define the desired inputs/outputs when used within a plugin:

- CompasChangeSetRadiogroup
- CompasCommentElement
- CompasLabelsField
- CompasOpen
- CompasSave
- ~~CompasSelList~~
- ~~CompasSelTypeList~~
- ~~CompasSelTypeSelect~~

The UI components should be "dumb". No functionality should be defined in them and plugins should provide functionality for these UI components via props/slots and event handlers.