How do I - set up developer environment

Make sure to go through all steps of gaining access rights and accounts described in How do I get access to

Instructions for MacOS

Developer applications

1. Install Homebrew: https://brew.sh/
2. Install *git* https://git-scm.com (terminal): `brew install git`
3. Install *gnupg* https://gnupg.org to sign git commits: `brew install gnupg`
4. Currently used Java version is 11 (ask the team if it’s still that, as there are plans for update).
   You can install Java 11 from OpenJDK https://openjdk.java.net via: `brew info openjdk@11`
5. Install *Maven* https://maven.apache.org (terminal): `brew install maven`
6. Install *pre-commit* https://pre-commit.com for shared git hooks: `brew install pre-commit`
7. Install *gh* https://cli.github.com to interact with GitHub from the terminal: `brew install gh`
8. Install *travis* https://github.com/travis-ci/travis.rb to interact with Travis CI from the terminal: `brew install travis`
9. Install Docker Desktop https://www.docker.com/products/docker-desktop via `brew install homebrew/cask/docker`
10. Log in into Quay.io with a previously created RedHat SSO account
11. Go to Account settings Generate encrypted password Choose the docker option
12. Call given `docker login` command in terminal (might be worth to save it in keychain)
13. Install your IDE of choice (or all):
    a. Visual Studio Code https://code.visualstudio.com via `brew install visual-studio-code`
    b. IntelliJ IDEA https://www.jetbrains.com/idea via `brew install intellij-idea` then request a license IntelliJ

Settings and repositories

1. Generate SSH key (remember to use ssh-add after generating it): https://www.siteground.com/kb/how_to_generate_an_ssh_key_pair_in_mac_os/?gclid=EAIaIQobChMI_q7-3PzW9gIViOF3Ch0gpQqREAAYASAAEgKeVPD_BwE
2. Go to GitHub account -> Settings -> SSH and GPG Keys. Add *.pub content as ssh key.
3. Go to ~/.m2
4. Create settings.xml file with template (https://collab.alfresco.com/share/page/site/engineering/document-details?nodeRef=workspace://SpacesStore/49638ec9-5e09-4f7c-8e6b-5df2d59de600). Replace yourusername and yourpassword with appropriate values
5. I recommend creating a new folder in your home directory (~) named “Projects” it should help you keep the repositories in an orderly fashion.
6. You’ll need to clone repositories from (https://github.com/orgs/Alfresco/repositories) using SSH link (to be able to use the SSH authorisation set up in previous steps)

ACS repositories

- acs-community-packaging
- acs-packaging
- alfresco-community-repo
- alfresco-enterprise-repo

Running the environment

Your environment should be working, try to run:

```
mvn clean install -DskipTests
```

in every cloned repo, if any errors occur, contact your team.

Additional ACS commands

Please watch https://collab.alfresco.com/share/page/site/engineering/document-details?nodeRef=workspace://SpacesStore/1716cf80-ec74-4d68-9e6f-e12deb23766c to get an overview of developer commands or actions necessary to run the projects, especially to acquaint yourself with alias commands:

- comR
- entR
- comP
- entP
- envUp/envRm
- envT/envO

Their descriptions along with other aliases are located in acs-packaging/dev/aliases.

> If you want to set up environment (envUp/envRm) for community project, use aliases from:
>
> acs-community-packaging/dev/aliases instead

Technical Overview

Hyland Software Development Life Cycle
https://hyland.atlassian.net/wiki/spaces/OCE/pages/251076867/SDLC+Home

ACS Technical Overview

https://collab.alfresco.com/share/page/site/engineering/documentlibrary?file=ACS%20Technical%20Overview%201%20-%20Welcome%20and%20Agenda.mp4#filter=path%7C%2FACS%2FACS%20internal%2FACS%20Technical%20Overview ACS technical overview videos by @ Stefan Kopf

Engineering Practices Useful read, lower priority than Hyland SDLC above, but contains useful information, e.g. code styling schema loadable into Intellij/Eclipse