Table of content:

1. App configurations
2. Guidelines about Git flow
3. Sprint Tasks
4. Submit app tasks
5. Localizations
6. Notifications and Deep links
7. Modelers system
8. Accounts & credentials

App configurations:

iOS works with external config that is managed from API and Database. Please reference to server documentation “p\_listfromDBandAPIdesign.docx” contained in “Development Documentation” folder.

db tables involved:

1. **AppConfig** – Contains actual values of all the configurations. Its structure represents hierarchical tree.
2. **ConfigValueTypes**- Contains value type that can be used in AppConfig (like string, int, bool, dictionary, float, array)
3. **AppVersion** – Contains the mapping between application version and its assigned config version. Changing **AppVersionConfigVer** to higher number will let clients know that configuration has changed

Client guidelines are as follows:

1. If new property needed, add it to latest active configuration, no need to create a copy and assign new config for version.
2. If existing property must change and it will break older versions. Copy the latest config (**AppConfig**) and assign new **AppVersion** . Here is example of SQL script to run on each DB:

/\*Creates new mapping between app version and config version\*/

INSERT INTO `AppVersion` (`AppType`,`AppVersionBuildID`,`AppVersionNeedUpdate`,`AppVersionMustUpdate`,`AppVersionConfigVer`,`AppConfigVersionID`)

VALUES(1,{{new version number string like '1.7.0'}},0,0,0,{{new config version number like 4}});

/\*Copy existing configuration into new configuration\*/

insert into AppConfig select {{new config version number like 4}},AppConfigID,ConfigValueTypeID,AppConfigDesc,AppConfigValue,AppConfigParentID from AppConfig where AppConfigVersionID = {{Config version you are coping from like 1}};

3. When creating new config, create modelers config as well by following same steps and changing config version number to appropriatly.

Guidelines about GIT:

1. No direct commits to Master branch.
2. Any new feature should be developed under Feature branch.
3. Finished features are merged back to Development branch.
4. Once all features in current Development branch are finished create Release\_{Sprint Name}\_vX.x.x branch and finish all last minute fixes on it.

Sprint tasks:

* Hockeyapp crash reports uploading ipa + dsym (store locally production submit file)
* Localizations
* Merging feature branches after app submit to development and master
* Create tag for each release in Master branch
* Create feature branches for each feature.

Submit tasks:

1. Verify scheme for “Archive” as “production\_release”
2. Verify configmanager’s config api urls are correct.
3. Update app version to next one (like 1.7.0, 1.8.0 etc…)
   1. Go to Homestyler project from Navigation
   2. Choose Homestyler under “Targets”
   3. Choose General tab and update Version number

**Note: use must increase version number each time you submit to apple**

1. Update build version when sending ET build (using Terminal command “agvtool next-version –all” from project’s root folder)

**Note: Don’t forget to commit these changes back to GIT**

1. Run archive build option
2. After archiving is done, open Organizer, choose last archive and click “Distribute” option
3. In distribute wizard choose “Save for Enterprise or Ad Hoc Deployment”
4. Finally upload the ipa file to ET and send it to hagai.galai@autodesk.com

Localizations:

1. Go to folder {Your git root}/CmyCasa/en.lproj/Base.lproj/
2. Run the following “Terminal” commands to generate updated “.strings” files for the storyboards:

#iPad storyboards

ibtool --generate-strings-file LoginStoryboard\_iPad.strings LoginStoryboard\_iPad.storyboard

ibtool --generate-strings-file ProfsStoryboard\_iPad.strings ProfsStoryboard\_iPad.storyboard

ibtool --generate-strings-file GalleryStoryBoard\_iPad.strings GalleryStoryBoard\_iPad.storyboard

ibtool --generate-strings-file MainStoryboard\_iPad.strings MainStoryboard\_iPad.storyboard

ibtool --generate-strings-file RedesignStoryboard\_iPad.strings RedesignStoryboard\_iPad.storyboard

ibtool --generate-strings-file NewProfileStoryboard\_iPad.strings NewProfileStoryboard\_iPad.storyboard

#iPhone storyboards

ibtool --generate-strings-file LoginStoryboard\_iPhone.strings LoginStoryboard\_iPhone.storyboard

ibtool --generate-strings-file ProfsStoryboard\_iPhone.strings ProfsStoryboard\_iPhone.storyboard

ibtool --generate-strings-file GalleryStoryBoard\_iPhone.strings GalleryStoryBoard\_iPhone.storyboard

ibtool --generate-strings-file MainStoryboard\_iPhone.strings MainStoryboard\_iPhone.storyboard

ibtool --generate-strings-file RedesignStoryboard\_iPhone.strings RedesignStoryboard\_iPhone.storyboard

ibtool --generate-strings-file ProfileStoryBoard\_iPhone.strings ProfileStoryBoard\_iPhone.storyboard

1. Cut and copy the “.strings” files into CmyCasa/en.lproj/en.lproj (override the existing files)
2. Go to CmyCasa/en.lproj and create zip file that will include en.lproj + localizable.strings file and send them to Selina.Xu@autodesk.com
3. Selina returns the localized files as folders for each language. Each folder will contain all .strings files for storyboards and localizeable.string file.
4. Replace the content of each language folder under CmyCasa/en.lproj/{specific language folder) with .strings files from Selina.
5. Compile the project again and fix the double quotes issues in .strings files by escaping them such as \”

Notifications and Deep Linking:

Modelers system:

Credentials:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Account | Contact | Site | username | password |
| appsflyer | elad@appsflyer.com | http://www.appsflyer.com | Sergei.berenson@autodesk.com | developer#1 |
| Mixpanel | ishan.verma@autodesk.com | http://www.mixpanel.com |  |  |
| Hockeyapp |  | http://hockeyapp.net/features/ | Sergei.berenson@autodesk.com | developer#1 |
| Apple account | Andy.Chang@autodesk.com |  |  |  |
| bitly | http://bit.ly/pro/dashboard |  | autodeskinc | autodeskrocks! |
|  |  |  |  |  |