RHINO DELOYMENT

1. Connection
2. Deploy

*(\*) Must run rebuild languages from website Metadata if there is any related change*

1. Database: work on ‘master’ branch

* Pull ‘master’ branch from ‘Rhino\_docs’ repository
* Check and verify the database that scripts should be run.
* Run scripts in ‘Latest’ folder.
* Move all scripts from ‘Latest’ folder to a new folder named as deployed date

*Ex: [24May2019]*

1. Backend: work on ‘client\_test’ branch with ‘client test’, on ‘master’ branch with ‘live server’

* From ‘Rhino\_Backend’ repository
  + Client Test: merge ‘develop’ branch to ‘client\_test’ branch
  + Live server: merge ‘client\_test’ branch to ‘master’ branch
  + Commit & push merged branches
* Rebuild the project with ‘release’ mode (or publish with Local profile)
* Backup latest version from deployed server
* Overwrite bellow folders from new build to server.
  + MetadataApi
    - Bin
    - Content
    - Htmls
    - Scripts
  + RhinoApi
    - Bin
    - EmailTemplate
    - Fonts
    - Images
    - **ScheduledScript: if have anychange**

1. Frontend: work on ‘client\_test’ branch with ‘client test’, on ‘master’ branch with ‘live server’

* From ‘Rhino\_Frontend’ repository
  + Client Test: merge ‘develop’ branch to ‘client\_test’ branch
  + Live server: merge ‘client\_test’ branch to ‘master’ branch
  + Commit & push merged branches
* Backup Websites (MetadataApp & RhinoApp) from deloyed server
* UIBuilder/MetadataApp
  + Run ‘grunt releaseClient’ for ‘Client Test’ or ‘grunt releaseLive’ for ‘Live server’. The files for deploying will be build and copied to output folder
  + Overwrite files from output folder to correct web folder
* Sba/RhinoApp
  + Run ‘grunt releaseClient’ for ‘Client Test’ or ‘grunt releaseLive’ for ‘Live server’. The files for deploying will be build and copied to output folder
  + Overwrite files from output folder to correct web folder
  + Sign CSS theme: do this if there is any change in the css files
    - There are 4 urls must be signed for 4 css files (gray\_theme.min.css, green\_theme.min.css, blue\_theme.min.css, red\_theme.min.css)
    - Copy the full urls of css files

Ex:

<https://testsite.rhinoapps.com/rhino/css/green_theme.min.css>

<https://testsite.rhinoapps.com/rhino/css/gray_theme.min.css>

<https://testsite.rhinoapps.com/rhino/css/blue_theme.min.css>

<https://testsite.rhinoapps.com/rhino/css/red_theme.min.css>

* + - With each url, go to this website

<https://www.srihash.org/>

to generate a hashcode and update to /commonData/config.json in cssHash block.

1. Mobile: work on ‘client\_test’ branch with ‘client test’, on ‘master’ branch with ‘live server’

* From correct branch in the project ‘sba’ of ‘Rhino\_Frontend’ repository
* Run ‘grunt releaseDeviceClient’ for ‘Client Test’, or ‘grunt releaseDeviceLive’ for ‘Live Server’. This will copy source code to mobile project in ‘www’ folder
* From ‘mobile’ project
  + Android:
    - run ‘cordova build android’
    - *… [Some step to sign app with keystore* rhinoAndroidKeystore.jks*]*
    - Deploy to Google play the .apk file
  + iOS
    - run ‘cordova build ios’
    - Work with xCode to build .ipa and deploy to Apple store