***Steps for creating automation tests in testing repo.***

Overview: There are currently three different repos that we use. One is qa-platform-ui (all automation tests that has to do with Platform UI) , the second is qa-integration-automation (all automation tests that validate both the Platform UI and backend is working together as expected), and finally the qa-avails-tool (all automation tests that validate the avails tool).

***CLONING A REPO***

Overview: When an automation engineer wants to create or update an automation test they must first clone the correct repo down to their local machine. Currently we are using Bitbucket to store our repo (which is using GIT functionality). Here are the steps below to perform the cloning operation.

***VIA GIT BASH*** (<https://git-scm.com/downloads>)

1. Once you have installed GIT, open a directory you want to clone the directory in.
2. Right click and click “Git Bash Here” (Windows). If you are a MAC user, using terminal change directory to the directory you want and move on to step 3.
3. Go to the bitbucket link for the repo. There you will find on the “Overview” page the URL for the repo. Copy that URL.
4. Type the following in the GIT Bash window/terminal “git clone <URL>”.
5. At this time the repo should be downloaded to your computer.

***IMPORTING MAVEN PROJECT INTO IDE***

Overview: Currently all QA automation tests are using Maven to automatically install the software dependencies, rather than adding the jars to the repo. Maven also allows us to call specific test suites via TestNG. We also use Maven Modules to separate the different function into groups.

***VIA ECLIPSE:***

REQUIREMENT: The given repo we will be import should already be cloned to your local system.

1. Open Eclipse and click on File > Import
2. Type Maven in the search box under “Select an import source:”
3. Select “Existing Maven Projects”
4. Click “Next”
5. Click “Browse” and select the folder that is the root of the Maven project (containing the pom.xml)
6. Click “Next”
7. Click “Finish”

***CREATING LOCAL BRANCH***

Overview: When an automation engineer is going to create an automation test it is very important that they pull the latest master of the repo (to ensure they are working on top of the latest) and create a branch (named the same as the ticket number). This allows for this branch of code to be tested and reviewed before merging into the master code.

***VIA ECLIPSE:***

REQUIREMENT: The user must make sure they have pulled the latest master before creating a new branch.

1. Right click on the project.
2. Go to “Team” > “Switch to” > “New Branch”
3. Popup will appear asking for the name of the branch. Enter the ticket number you are working on. Example: TD-1232
4. Click on “Finish”

***COMMITTING TO A LOCAL BRANCH***

Overview: In order to store your changes into your computer you must commit to your local branch before pushing it to the repo.

***VIA ECLIPSE:***

1. Right click on the project.
2. Go to “Team” > “Commit”
3. Popup will appear where you can review the files you are going to commit, plus it will allow you to add a message.
4. Click “Commit”

***PUSHING TO REPO***

Overview: In order for your local changes to be present in the repo you must push your local branch to the origin.

VIA ECLIPSE:

1. Right click on the project.
2. Go to “Team” > “Push”

***CREATING A PULL REQUEST***

Overview: A pull request allows you and other developers to review the code to make sure all standards are met, and everything is looking as expected. This is a chance to conduct a peer review session.

***VIA BITBUCKET:***

1. Go to the repo on bitbucket. Example: <https://bitbucket.org/adtheorent/qa-integration-automation/overview>
2. Click on “Pull Requests” on the left side of the page.
3. Click on “Create pull request” button on the top right of the page.
4. In the “Create a pull request” page you want to select your branch name in the first box and master in the second box. This is saying you want to merge your changes into master.
5. Add the other developers in the “Reviewers” section of the pull request.
6. Click on “Close <Branch name> after the pull request is merged (this will remove the branch you pushed to the repo from the server.
7. Click on “Create pull request”

Once pull request is create all reviewers will get emails telling them that a pull request is available for them to review. Each reviewer will review the code and either place a comment, approve, or decline the pull request. Once all comments have been addressed and all reviews have approved the pull request you will be able to click on “Merge”. After this stage is completed your code is now placed into the master branch.