Development commits for Organization model to Git repository

Please add your suggestions to this document after you reading this document.

As per Minutes of Meeting yesterday, below is the basic plan for the GIT repository for on going production

**Figure 1 Master Branch**

Create the FEATURE branch from MASTER

Feature branch2

**Figure 2 Create the STAGING branch from MASTER**

**Figure 3 MERGE the STAGING with MASTER**

Feature branch1

**Staging branch**

Feature branch3eature

saasxasa

Do commits for feature branch to

staging branch.

The above diagram represents basic structure to commit in the GIT.

Step 1: First thing is to create a repository with MASTER branch in GITHUB

Step 2: Create the STAGING branch from MASTER branch.

Step 3: Create the FEATURE branch from MASTER (whoever developer worked with specific requirement)

Step 4: After development completed, each and every FEATURE branch code has to commit into STAGING branch (Staging branch is going for production environment).

Step 5: Once FEATURE branch is committed to STAGING, merge the STAGING branch with MASTER branch.

Step 6: After the Step 5, MASTER branch code will stable code.

Above steps are apply to Org model services.

Currently, we have SAAS, Whit Label1, Org Model Services etc. In future we may have another White Label service will be also there. So, whatever the new service is going to develop, we have the above same structure to follow.

For example

SAAS (Ufinity APP) – 7 services are there

W.R.T to above diagram structure for SAAS

Master Branch (SAAS)------🡪Staging branch for SAAS

From Master ----🡪 FEATURE branch (SAAS)-🡪commit to STAGING

From STAGING ----🡪 MASTER (merge the STAGING branch to Master once feature branch is committed to Staging)

Like above example, we will have Master branches for SAAS, White Label and Org Model. i.e., total 3 masters and 3 stages branches.

Before implementing this plan, we have to practice in our local/remote public repository with different context.

Below are general things to understand while doing commits (PULL/PUSH requests)

* How to resolve conflicts
* How to merge the branches
* How to revert the code (if it is the case when wrong commit happens)
* How to update the staging from master when anything went wrong happens in PRODUCTION environment
* Understand the REBASE concept
* Understanding Cherry picking concept (it is better to understand)

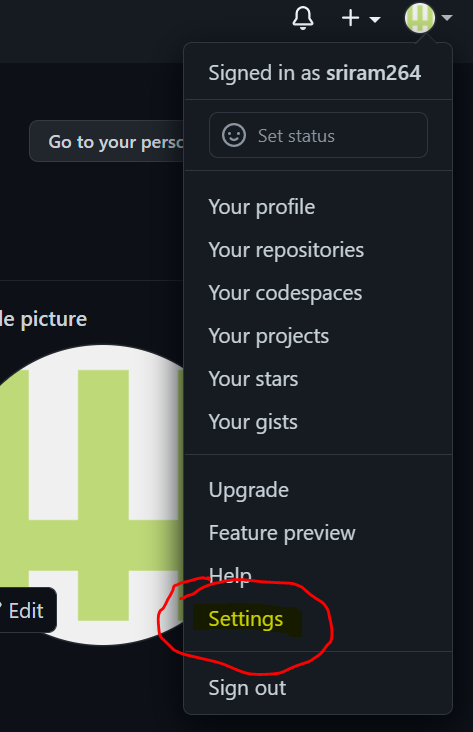
To do the practice I have created the public repository with sample project MASTER branch

<https://github.com/sriram264/springmvc>

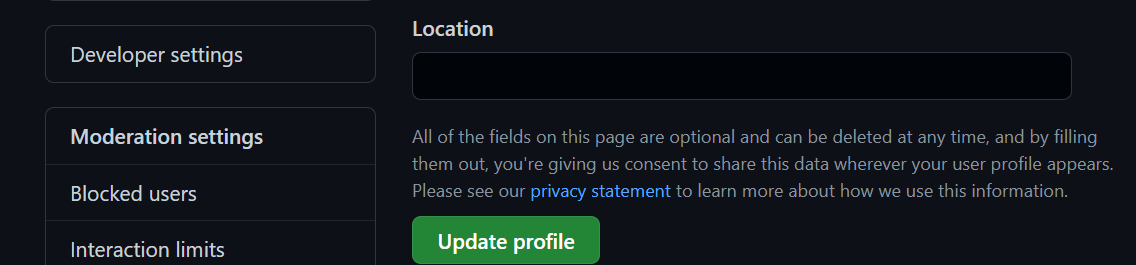
First everyone create an account in GITHUB with your personal email, please don’t use the geekbull account.

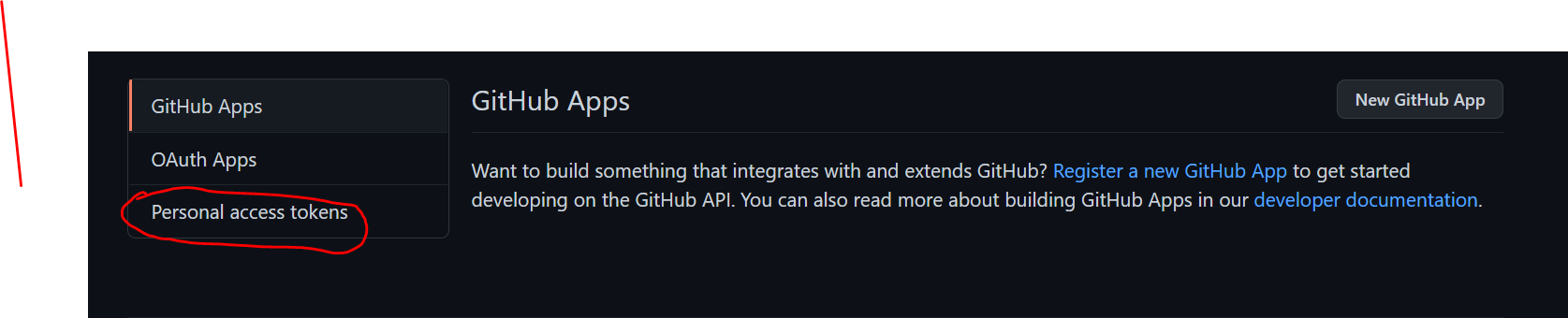
After this create the public access token in your GitHub account. This is why because git hub stops the passwords as credential, instead we have to use public token

Go to the settings page

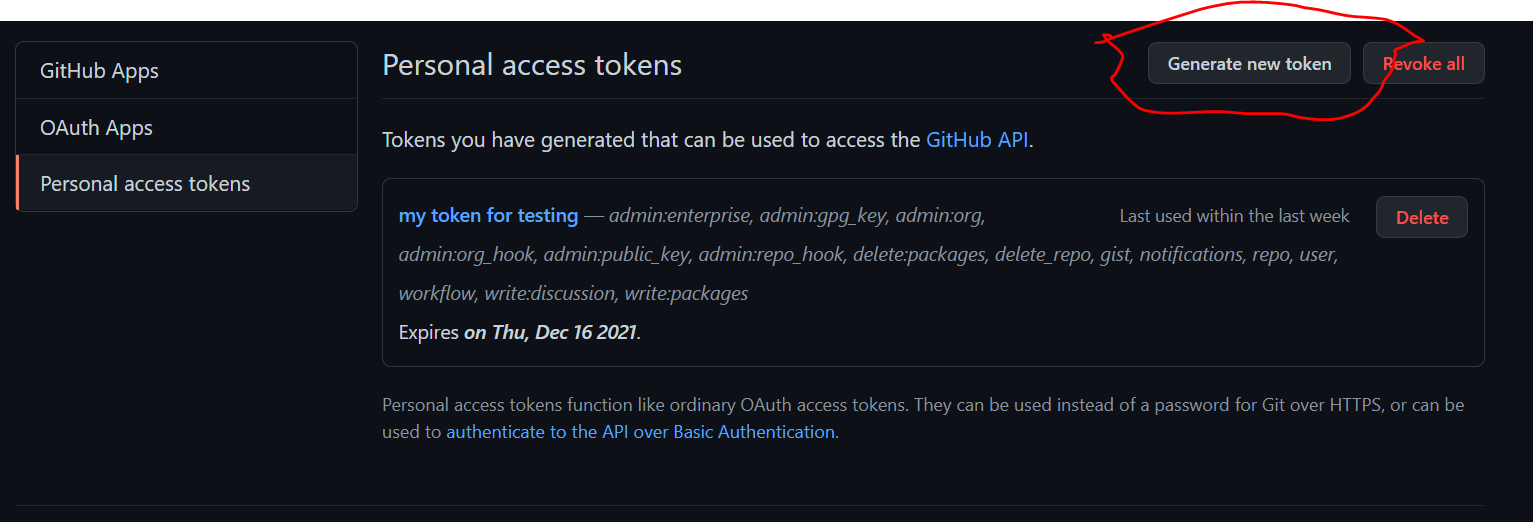


Next go to developer’s settings



After click on the developer settings goto personal access tokens

After click on the personal access tokens generate new token



After this step, clone the project the from remote GIT into your eclipse.

At repository level I have create feature branch with my name sriram\_branch.

Let’s first take the feature branch, I have created sample spring \_mvc\_example project. Within that we have HomeController.java file.

First task is resolving conflict

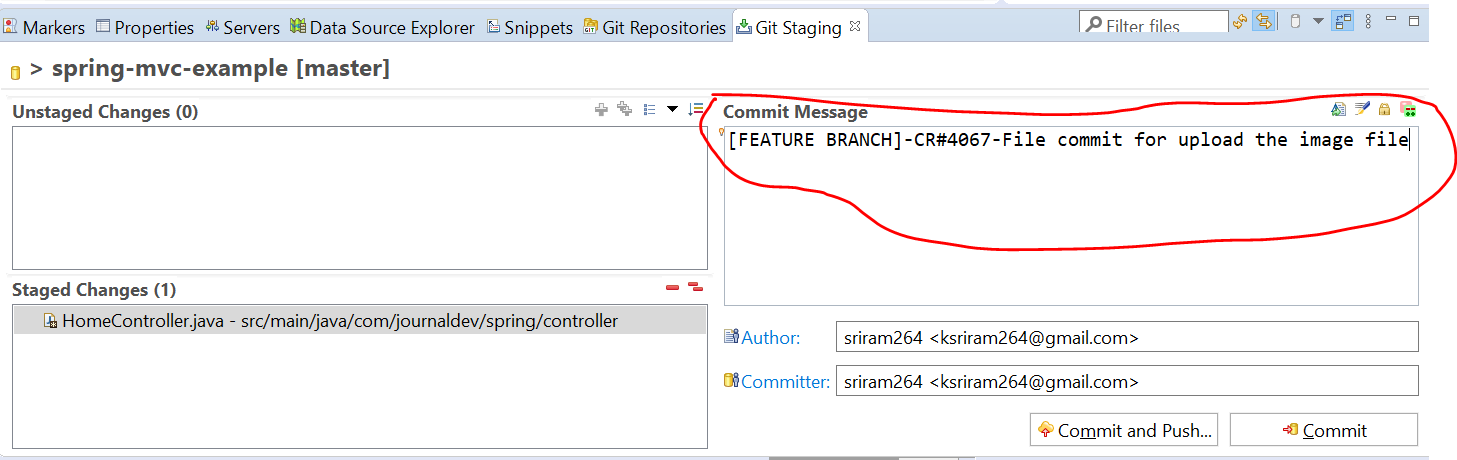
Second task is merging the feature branch to master.

Third task is to revert the code it means that undo changes in the remote.

Fourth task is REBASE

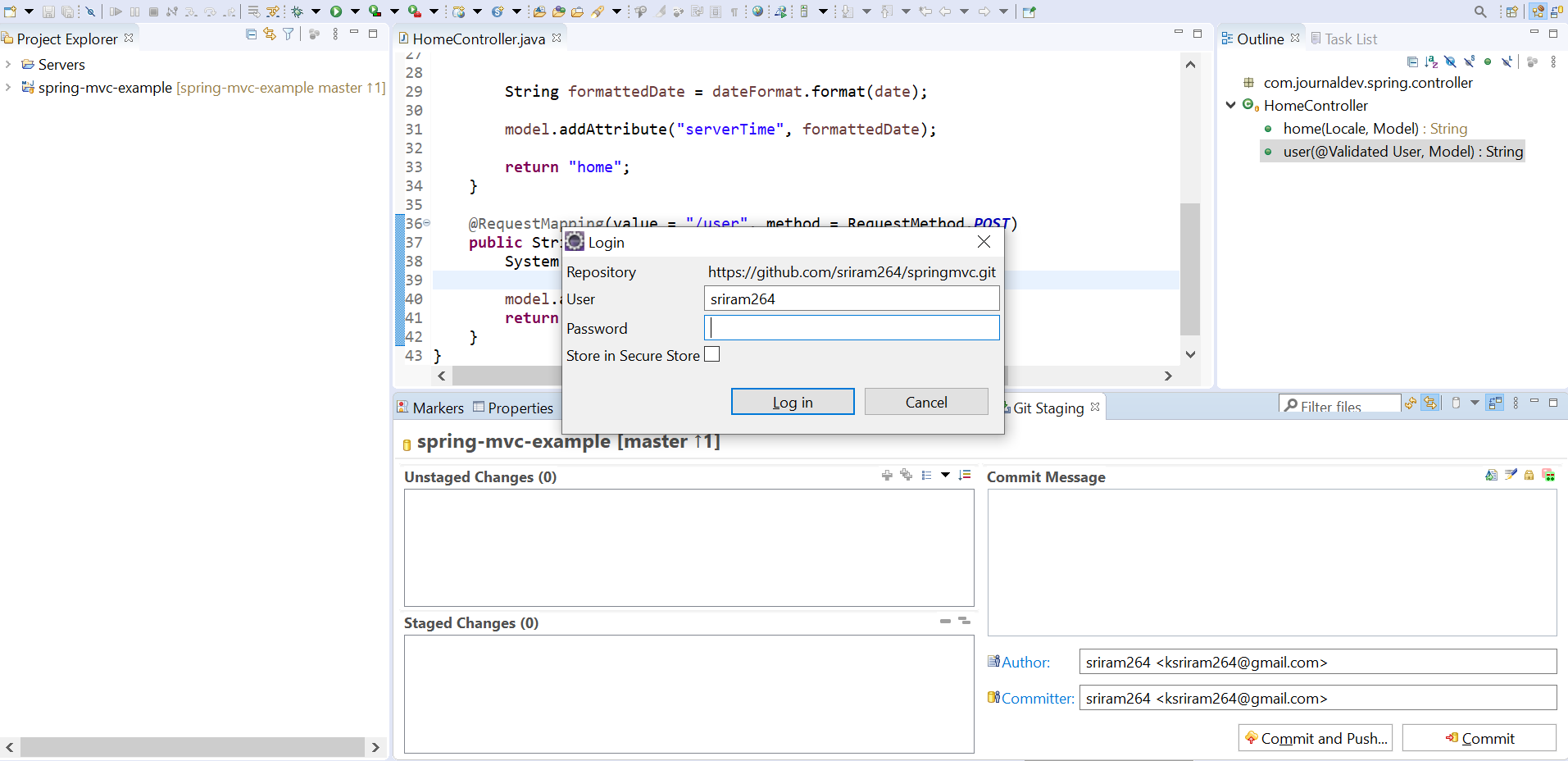
Task 1:

First everyone takes same feature branch with same file (say HomeController.java) do changes and commit the file in eclipse. everyone please remember, while committing the file, In commit message we should have proper message with proper format.

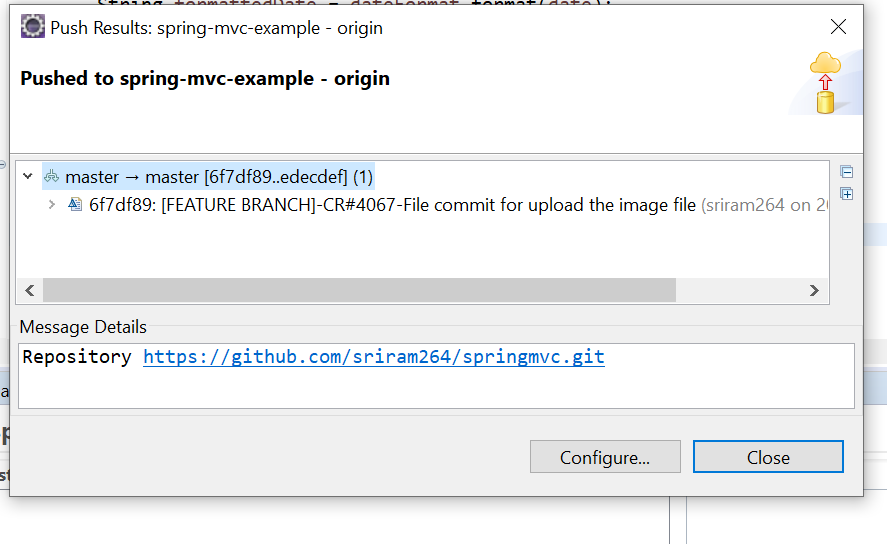


After this step say ‘Commit and Push..’ button , it asks for user and password

You have to give Git user name and password as public token which is generated git repository level

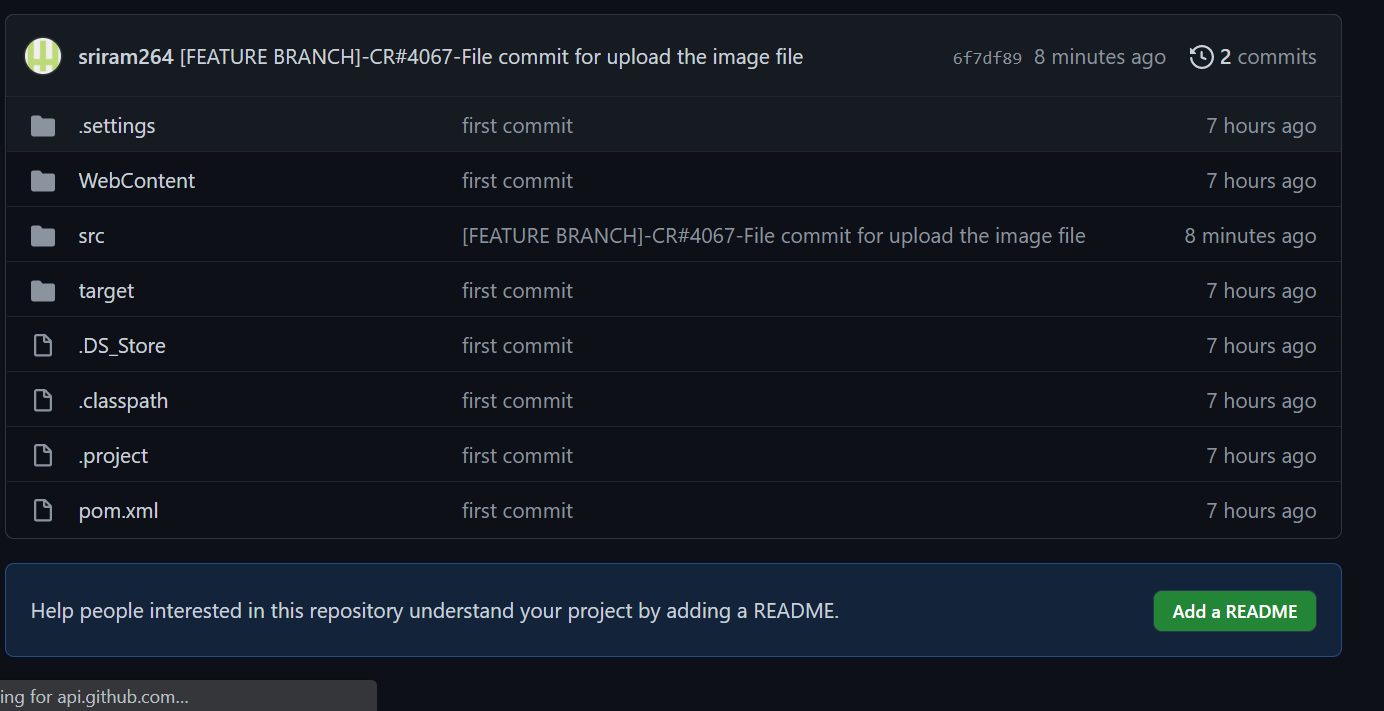


Once you enter the credentials it will commit the files and shows message below



You can check the commit in GIT hub repository.

While push the code , please do PULL request from remote its a Thumb rule .



First task how to commit to GIT is completed. Pick the next task code revert.

Please add your suggestions to this document.