LAB- Understanding GIT and GITHUB

In this lab you will understand about git and github

I assume that you have already installed git and gitbash on your machine.

STEP1 – Working with git

Create a folder with name   c:\ gitrepos\firstrepo

We want to initialize it as your local git repository

Open cmd and cd to c:\ gitrepos\firstrepo

Execute the below command .

```
git init
```

You should observe that a .git folder is created.

Create a file with name **a.txt** under c:\ gitrepos\firstrepo and add following content:

This is first line

Execute **git status** and observe the output.

To stage the files in current directory, execute git add .

After staging we can commit using below command:

git commit –m “This is my first commit”

If you are executing commit for the first time u will get an error message asking you to set your name and email. Use the below commands to do that setting

```
git config –global user.name “Your Name”
git config –global user.email “Your email”
```

Now execute **git log** command to check the commit logs

Now modify a.txt file and add another line “This is second line”

Stage this changes and commit using below commands:

git add .

git commit –m “This is my second commit”

Now execute **git log** command to check the commit logs

Create a branch with name test using **git branch test**

Switch to test branch using **git checkout test**

Now modify a.txt file and add another line “This is Third   line added in test branch”

Use below commands to commit the changes to test branch

git add .

git commit –m “commiting to test branch”

Now, we want to merge changes to master branch.

First switch to master branch using **git checkout master**

Now execute **git merge test**

Execute below command to check the commit logs

git log

Till now, we worked on local git repository using git

We want to create a repository with same name in github and want topush local changes to remote repository.

STEP2 – Working with Github

Login to your github account.

Create a public repository with same name “firstrepo” . Do not Create Readme.md file while creating repository.

Execute the following steps after creating the respository in github:

(make sure that you are in your local repo folder in cmd prompt)

git remote add origin yourrepourl   # eg: https://github.com/sivalwww/firstrepo.git

git branch -M main

git push -u origin main

Now refresh your repository page in the browser and make sure that the local files in current local repository are pushed in to remote repo.

Observe the 3 commits in the remote repository also

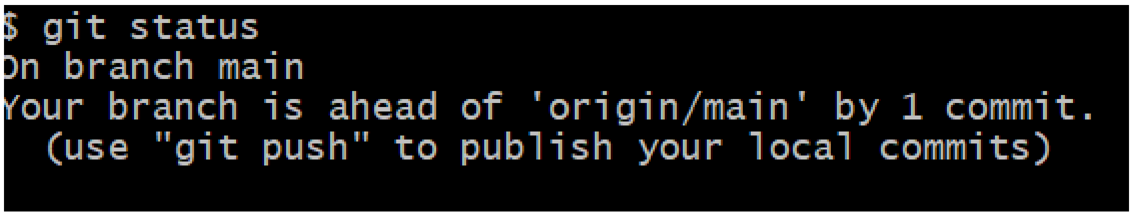
Now modify a.txt and add a line   “Making a change in local repo after pushing code to remote repo”

Commit the changes to local repo using below commands:

git add .

git commit –m “committing to local repo after pushing ”

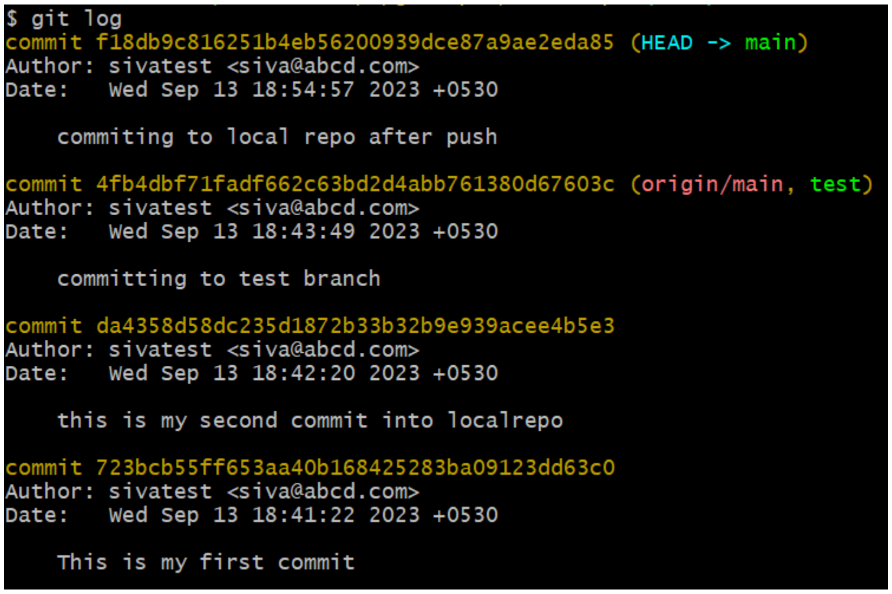
 Now execute **git status** command and observe that it displays following message which says that our local branch is ahead of remote origin by 1 commit



```
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
```

Execute **git log** command and observe that u see output similar to below:

It says that out current HEAD is pointing to latest commit where as origin/main is pointing to earlier commit.



```
$ git log
commit f18db9c816251b4eb56200939dce87a9ae2eda85 (HEAD -> main)
Author: sivatest <siva@abcd.com>
Date:   Wed Sep 13 18:54:57 2023 +0530

    commiting to local repo after push

commit 4fb4dbf71fadf662c63bd2d4abb761380d67603c (origin/main, test)
Author: sivatest <siva@abcd.com>
Date:   Wed Sep 13 18:43:49 2023 +0530

    committing to test branch

commit da4358d58dc235d1872b33b32b9e939acee4b5e3
Author: sivatest <siva@abcd.com>
Date:   Wed Sep 13 18:42:20 2023 +0530

    this is my second commit into localrepo

commit 723bcb55ff653aa40b168425283ba09123dd63c0
Author: sivatest <siva@abcd.com>
Date:   Wed Sep 13 18:41:22 2023 +0530

    This is my first commit
```

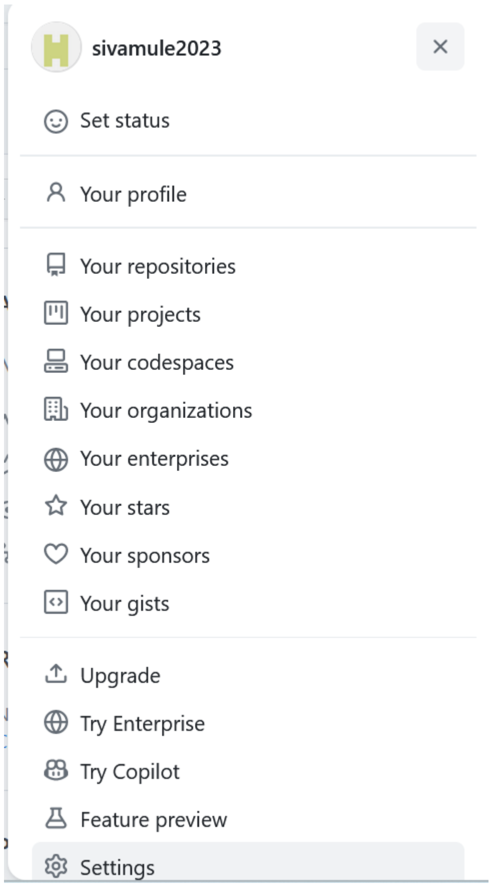
Now push to remote repo using :

git push -u origin main

STEP3

We need to generate a personal access token to integrate anypoint studio with Github.So, let us first generate the personal access token.

Login to your github account .click on your profile icon on your right top corner and select settings.



In the left sidebar, click **Developer settings**

In the left sidebar, under **Personal access tokens**, click **Fine-grained tokens**.

Click **Generate new token**
Under **Token name**, enter a name for the token.
Under **Expiration**, select all repositories
Under **Permissions**, select which permissions to grant the token. For our training purpose, select all permissions.

Under **Repository access**, select which repositories you want the token to access.

Click **Generate token**.

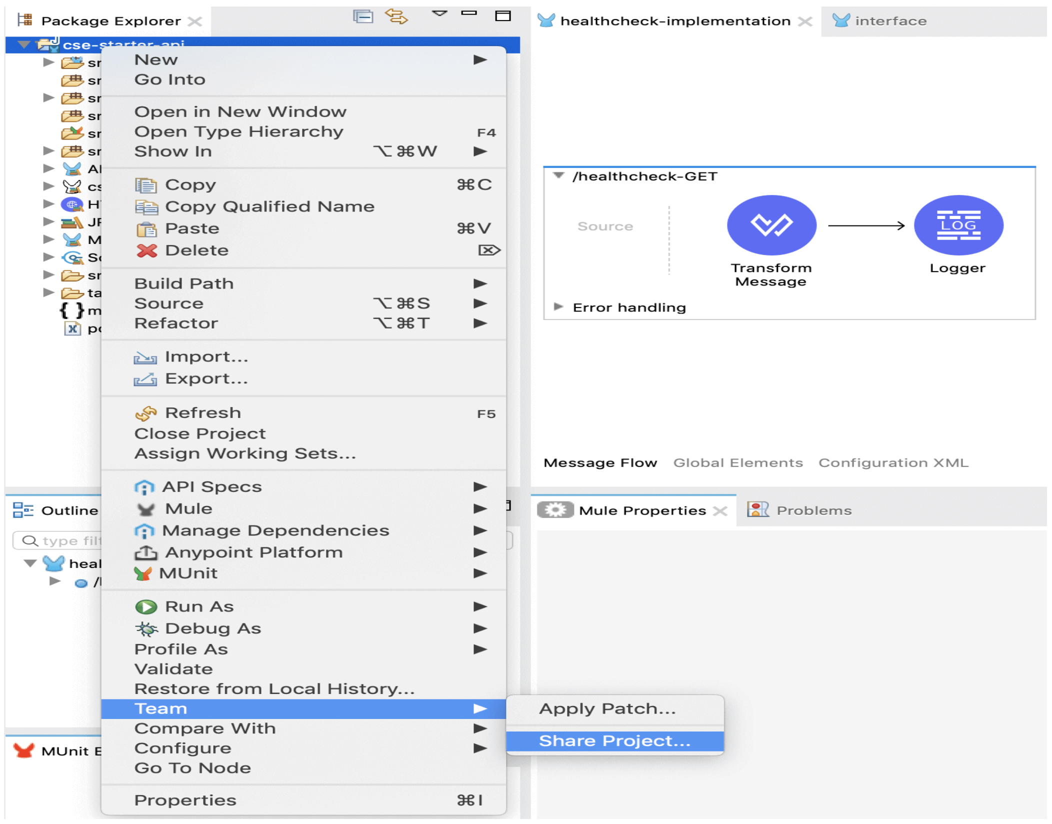
Copy the generated token so that u can use it in future when you push the code from local repository to remote repository

STEP4

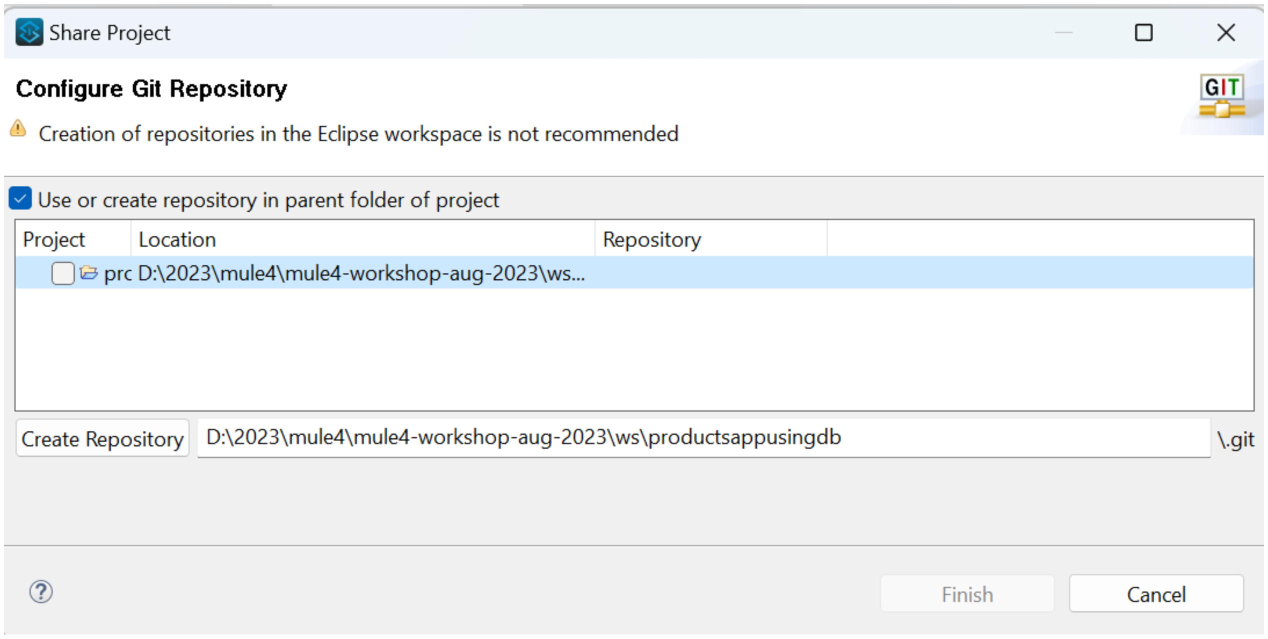
In this step, we will push the existing mule project into github repository.

Firstly, let us make the current project as local git repository.

To create a local repository, In the Package Explorer, right-click the name of your project → select Team → select Share Project



In the Configure Git repository window, Select the checkbox for **Use or Create a repository in the parent folder of the project**. Click on the **Create Repository** button and then finish


Share Project

Configure Git Repository

Creation of repositories in the Eclipse workspace is not recommended

☑ Use or create repository in parent folder of project

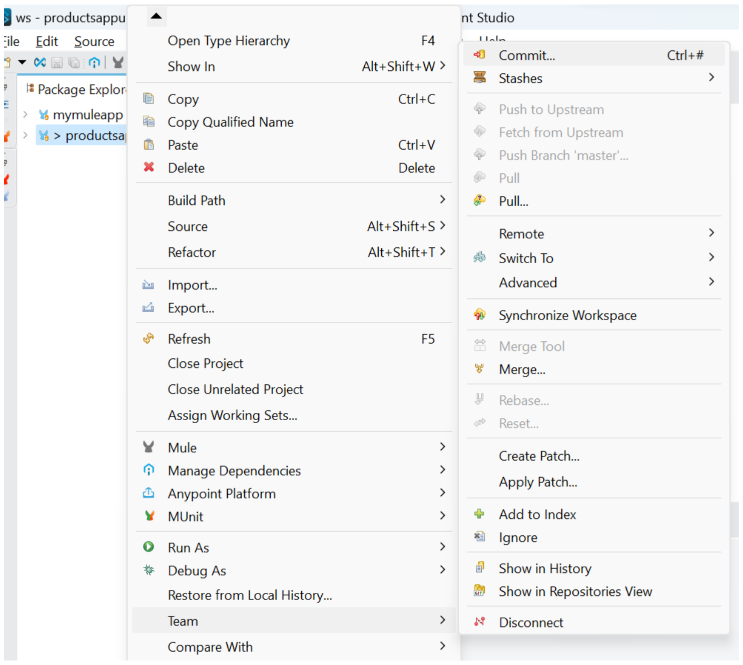
Project	Location	Repository
☐ prc	D:\2023\mule4\mule4-workshop-aug-2023\ws...	

Create Repository D:\2023\mule4\mule4-workshop-aug-2023\ws\productsappusingdb \.git

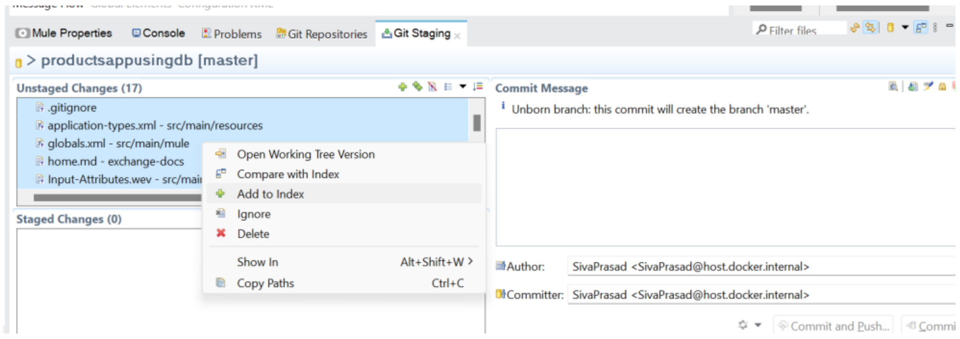
Finish Cancel

In the Anypoint Studio package explorer, you can see the arrow mark and questions mark symbols for your studio files. This icon indicates that you have created a Git repository for your project on your local drive, but you have not yet registered it and are not yet tracking changes to the project.

Now you need to register the local repository and track changes. To do this go to your package explorer, right-click to the name of your project → select **Team** → select **Commit**



Once you have clicked on the **Commit** button. You will see the **Git Staging** tab opened up. Select all the files that you want to check-in in Git then right-click on the selected files and click on **Add to index**.



> productsappusingdb [master]

Unstaged Changes (17)

.gitignore
application-types.xml - src/main/resources
globals.xml - src/main/mule
home.md - exchange-docs
Input-Attributes.wev - src/mai

Staged Changes (0)

Commit Message

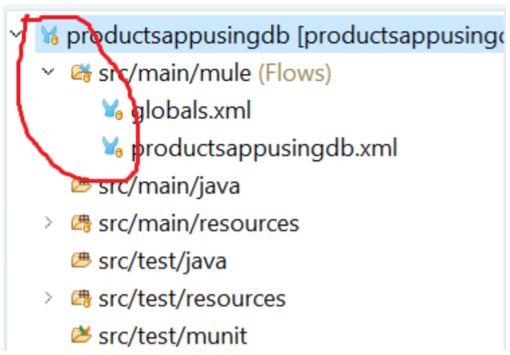
Unborn branch: this commit will create the branch 'master'.

Author: SivaPrasad <SivaPrasad@host.docker.internal>

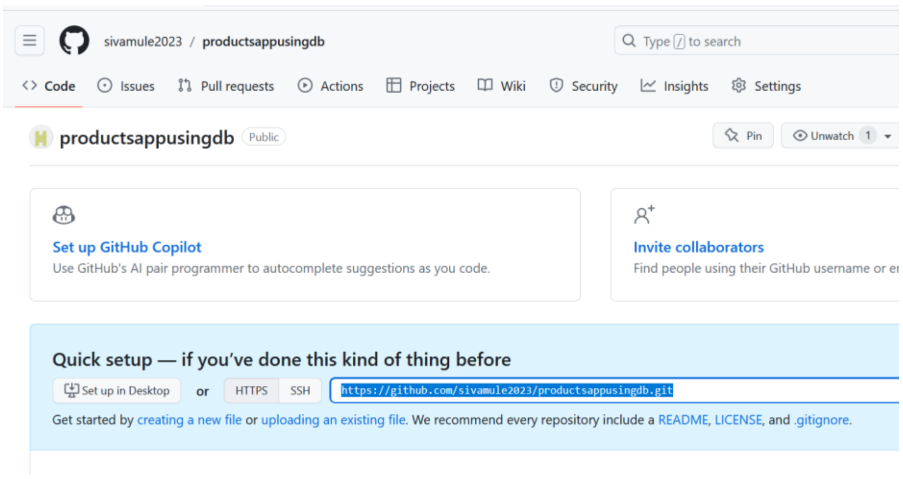
Committer: SivaPrasad <SivaPrasad@host.docker.internal>

Add a commit message to the text box and click on the ‘Commit’ button. Your changes will be stored in the local git repository.

In the package explorer, there is a yellow barrel icon which indicates that the changes have been committed to your local git repo. Also, there is no more arrow icon next to the project name.



Now login to github.com and create a new repository with name same as your mule application. Copy the url of the repository


sivamule2023 / productsappusingdb

productsappusingdb Public

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Invite collaborators
Find people using their GitHub username or e

Quick setup — if you've done this kind of thing before

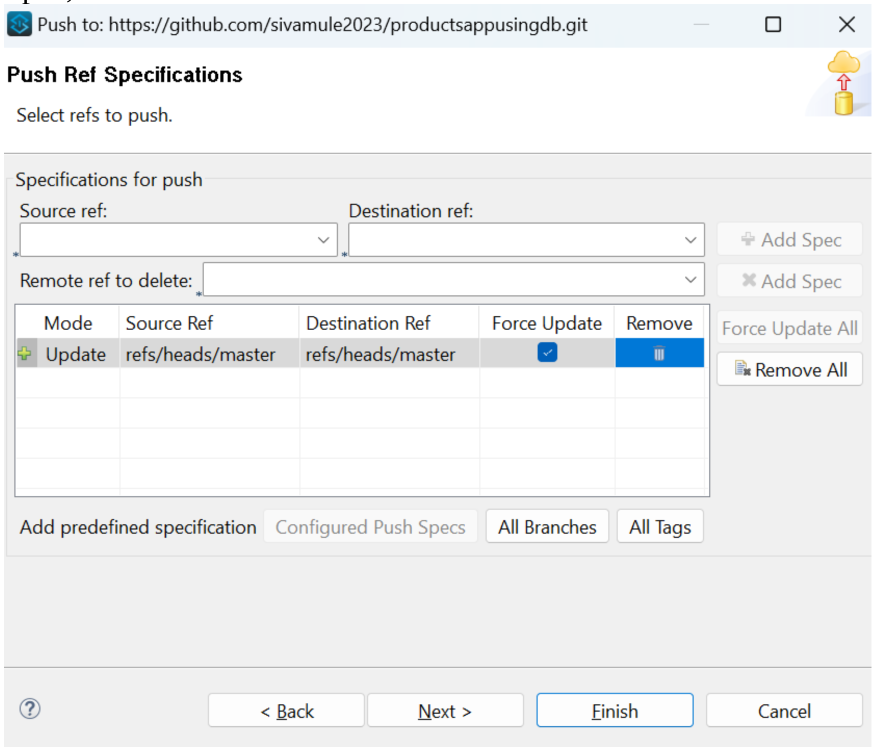
https://github.com/sivamule2023/productsappusingdb.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

Next, commit to your remote git repository. In the Package Explorer, right-click the name of your project navigate to **Team** → select **Remote** → **Push**

Paste the URL of the empty repository created in Remote Git

Select the Specification for Push from the Source ref and Destination ref and then Click on Add Spec, then Click on Finish.


Push to: https://github.com/sivamule2023/productsappusingdb.git

Push Ref Specifications

Select refs to push.

Specifications for push

Source ref:

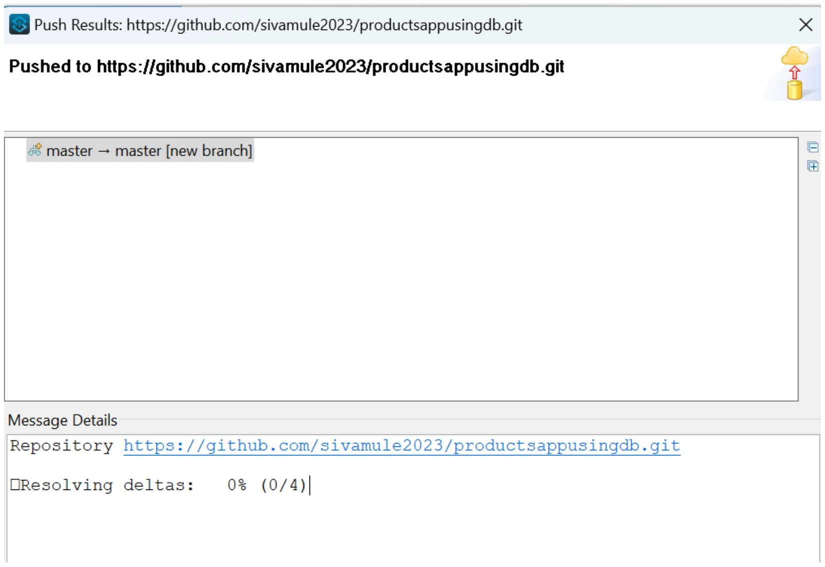
Destination ref:

Remote ref to delete:

Mode	Source Ref	Destination Ref	Force Update	Remove
Update	refs/heads/master	refs/heads/master	✓	

Add predefined specification

After the changes are committed to the remote repository you will see a window which will indicate that the changes are pushed successfully which is similar to the window below. Click on ‘OK’.



Push Results: https://github.com/sivamule2023/productsappusingdb.git

Pushed to https://github.com/sivamule2023/productsappusingdb.git

master → master [new branch]

Message Details

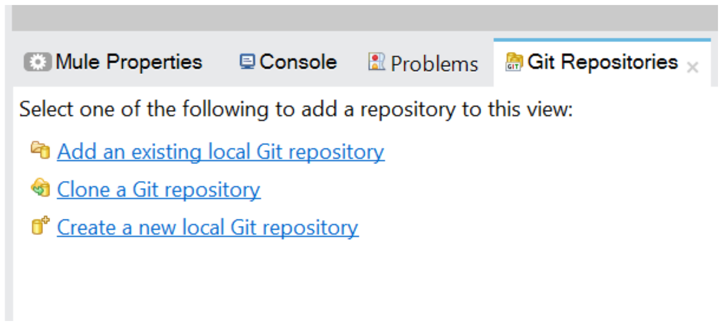
```
Repository https://github.com/sivamule2023/productsappusingdb.git

Resolving deltas:   0% (0/4)
```

Now go to github.com and refresh your repository page. You should see the commited files

Cloning a remote repository in Anypoint Studio

Now as you have committed tha project to github, delete the project from anypoint studio. We will clone the application from github

To clone a remote repository, Go to Menu bar → Window → Show view → Other → Select Git Repositories

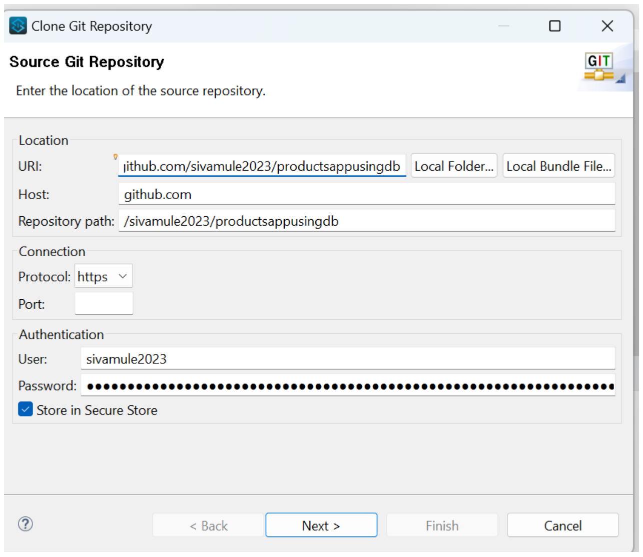


Select one of the following to add a repository to this view:

- Add an existing local Git repository
- Clone a Git repository
- Create a new local Git repository

Click on Clone a git repository

In the next window, give the URL of your repository in github.com , take defaults and finish



Source Git Repository

Enter the location of the source repository.

Location

URI: jithub.com/sivamule2023/productsappusingdb

Host: github.com

Repository path: /sivamule2023/productsappusingdb

Connection

Protocol: https

Port:

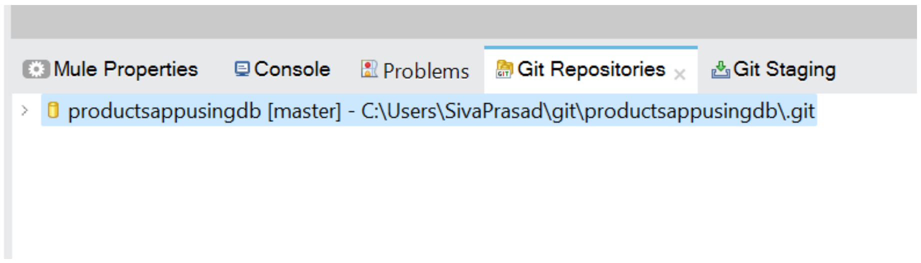
Authentication

User: sivamule2023

Password: ••

Store in Secure Store

You should see the cloned repository as below:



productsappusingdb [master] - C:\Users\SivaPrasad\git\productsappusingdb\.git

Now right click on the repository and select import projects and import the project.

Congratulations!! You understood how to integrate anypoint studio with GIT and manage code

<u>STEP5 – Handling conflicts (optional…don't do .it is just explanation)</u>

If 2 developers start working on same branch at a time and developer1 commited his code after making changes and pushed to github, developer 2 also made changes and trying to commit and push,there will be a conflict.

So, developer2 has to pull the changes first, then resolve conflicts in the files.

Then can commit and push.

This is the end of the Exercise