**A picture containing text, sign

Description automatically generated**

**How to use GitHub with Eclipse**

By

Tarun Narasimha Varma Muppalla

A04297061

Graduate Assistant – Research

Department of Computer Science

**Table of Contents**

1. Cloning your GitHub Account
2. Push Changes (Commit)
3. Pull changes from upstream
4. Conclusion
5. References

**1. Cloning your GitHub Account**

1. Make sure you have your GitHub account setup. Create a repository. For information regarding how to create a GitHub account and a repository check Howto (GitHub) document.
2. Access the repository you are currently working on in your GitHub account. Selecting "Code," go to the HTTPS tab and copy the URL. A screenshot of a computer

   Description automatically generated
3. Go to Eclipse, right click anywhere under the Package Explorer tab. Select “Import”.

A screenshot of a computer

Description automatically generated

1. Once you have entered the Import Menu, open the dropdown under “Git”. Select “Projects from Git” and click on Next.

A screenshot of a computer

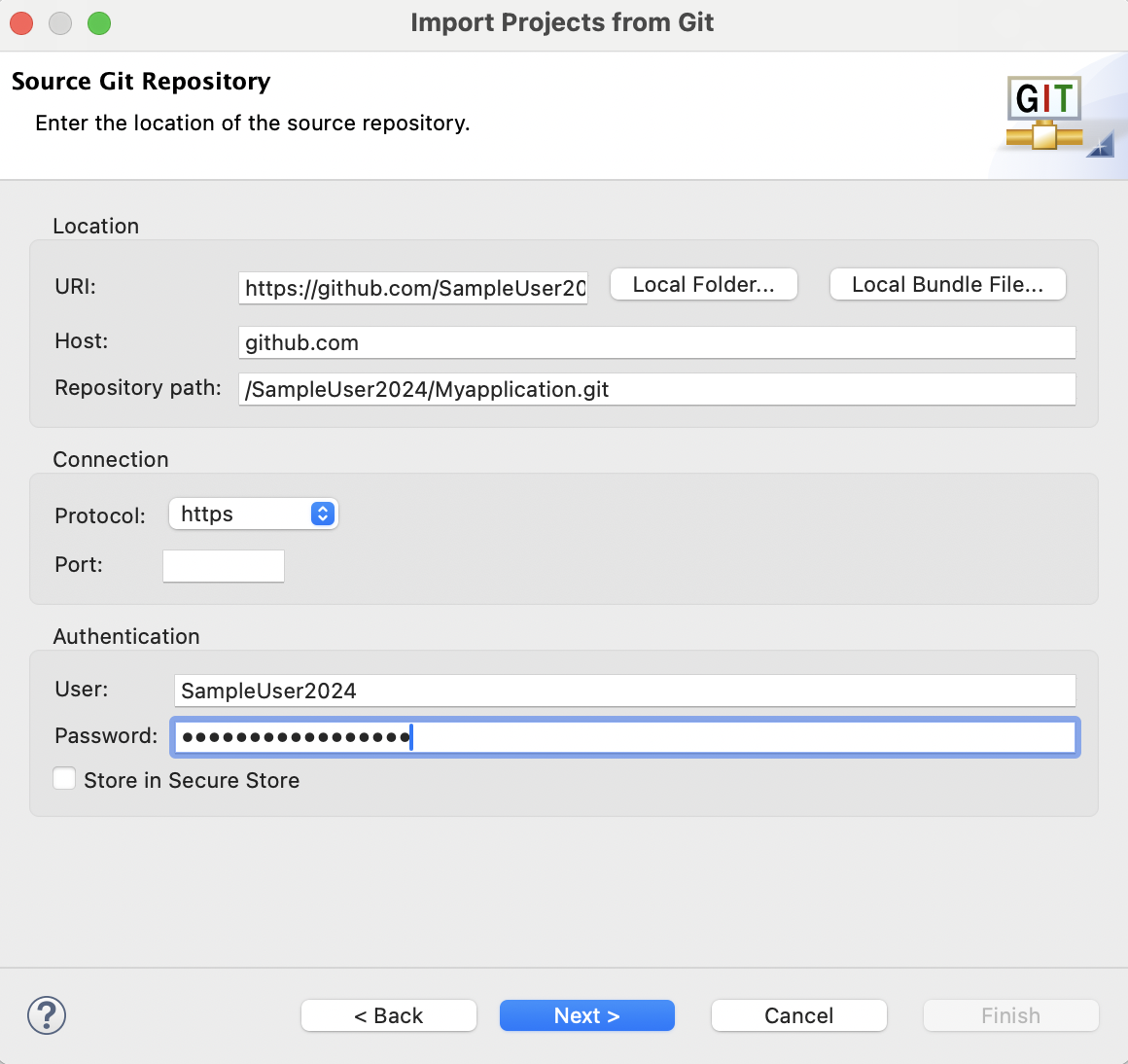
Description automatically generated

1. Select the option “Clone URL” as we are using a repository that’s already existing.

A screenshot of a login box

Description automatically generated

1. Enter the Copied HTTPS URL. Enter your GitHub username and password.



1. Select the Branch you desire to work on.A screenshot of a computer

   Description automatically generated
2. Select the Directory you would like the files to be saved in

A screenshot of a computer

Description automatically generated

1. Now you should be able to see the folder on your Eclipse IDE.

**2. Push changes (Commit)?**

1. Make changes to your file, right click on the java project. Go to Team-> CommitA screenshot of a computer

   Description automatically generated
2. Drag and drop all the changed files into staged changes. Click on “Commit and Push”.

A screenshot of a computer

Description automatically generated

Occasionally, you may encounter difficulties when attempting to commit changes. In such instances, you will need to create a token using your GitHub account and attempt to use the token ID as your password. For creating a token follow the process below.

1. In the upper right corner of your GitHub account. Navigate to Settings, choose developer settings by scrolling down. SELECT “Tokens (classic)” Click on “Generate new token (classic)”

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

A screenshot of a phone

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Add a Note, enable all the options you want and click on “Generate new token”.

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

**Note**: Make sure you copy your personal access token as you might not be able to access it again.

1. Select Team -> Commit now. Drag and drop every modified file to the “stage changes”. After adding a comment, select Push Head or Commit and Push. Enter your GitHub user ID when prompted for credentials, and the generated personal access token as the password. Your edits may have been accepted.A screenshot of a computer

   Description automatically generated

A screenshot of a computer

Description automatically generated

**Note**: you might have to enter the credentials multiple times, to avoid this you can save them by enabling the “Store in Secure Store” option.

A screenshot of a login box

Description automatically generated

**3. Pull updated Files from Upstream:**

1. Perform a right-click in Package Explorer, choose the folder, and then pick Team -> Pull. This will retrieving all the updated and modified files from the repository.

A screenshot of a computer

Description automatically generated

**Conclusion**

Integrating GitHub with Eclipse streamlines the development workflow, fostering collaboration and version control. This seamless integration enhances code quality, facilitates team coordination, and ensures efficient project management. By following the outlined steps, developers can effortlessly synchronize their Eclipse projects with GitHub repositories, leveraging the full potential of both tools for a more productive and organized software development process.

**References**

|  |  |
| --- | --- |
| **URL** | **Description** |
| <https://www.eclipse.org/downloads/> | Eclipse IDE download link. |
| <https://www.youtube.com/watch?v=yA9t5Dz68GQ> | Video Tutorial of how to use GitHub in eclipse. |
|  |  |