# Linking Git with Spring Tool Suite (STS)

## Introduction

Spring Tool Suite (STS) is an Eclipse-based IDE tailored for Spring development. Integrating Git with STS allows developers to manage version control directly from the IDE, streamlining the development workflow.

## 1. Install Git Integration in STS

### Checking for EGit Plugin

Most versions of STS come with built-in support for Git through the EGit plugin. To check if it's installed:

1. Open STS.

2. Navigate to Help > About Spring Tool Suite.

3. Click Installation Details and look for "EGit" in the list of installed software.

### Installing EGit via Eclipse Marketplace

If EGit is not installed:

1. Go to Help > Eclipse Marketplace.

2. Search for "EGit".

3. Click Install and follow the prompts to complete the installation.

## 2. Configure Git in STS

Before using Git in STS, you need to configure it with your user information:

1. Open STS.

2. Navigate to Window > Preferences.

3. Expand Team > Git > Configuration.

4. Click New Entry and add your user name and email:

- Key: `user.name`

- Value: `Your Name`

- Key: `user.email`

- Value: `your.email@example.com`

5. Click Apply and Close.

## 3. Cloning a Repository

To clone an existing Git repository into STS:

1. Go to File > Import.

2. Select Git > Projects from Git and click Next.

3. Choose Clone URI and click Next.

4. Enter the repository URI and authentication details (if required), then click Next.

5. Select the branches you want to clone and click Next.

6. Choose the local directory for the cloned repository and click Finish.

Alternatively, using Git command line:

```sh

git clone <repository\_URI>

```

## 4. Creating a New Git Repository

To create a new Git repository from an existing project in STS:

1. Right-click the project in the Project Explorer.

2. Choose Team > Share Project.

3. Select Git and click Next.

4. If no repository exists, click Create to initialize a new repository.

5. Click Finish.

Alternatively, using Git command line:

```sh

cd /path/to/your/project

git init

```

## 5. Basic Git Operations in STS

### Adding and Committing Files

- Add Files to Staging Area:

1. Right-click the file(s) in the Project Explorer.

2. Go to Team > Add to Index.

Alternatively, using Git command line:

```sh

git add filename

git add .

```

- Commit Changes:

1. Right-click the project or file.

2. Go to Team > Commit.

3. Enter a commit message and click Commit.

Alternatively, using Git command line:

```sh

git commit -m "Your commit message"

```

### Pushing Changes

- Push to Remote Repository:

1. Right-click the project.

2. Go to Team > Push to Upstream.

Alternatively, using Git command line:

```sh

git push origin branch\_name

```

### Pulling Changes

- Pull from Remote Repository:

1. Right-click the project.

2. Go to Team > Pull.

Alternatively, using Git command line:

```sh

git pull origin branch\_name

```

### Branching and Merging

- Create a New Branch:

1. Right-click the project.

2. Go to Team > Switch To > New Branch.

Alternatively, using Git command line:

```sh

git branch branch\_name

```

- Switch Branches:

1. Right-click the project.

2. Go to Team > Switch To and select the desired branch.

Alternatively, using Git command line:

```sh

git checkout branch\_name

git checkout -b branch\_name

```

- Merge Branches:

1. Right-click the project.

2. Go to Team > Merge and select the branch to merge from.

Alternatively, using Git command line:

```sh

git merge branch\_name

```

### Viewing Git History

- Show Commit History:

1. Right-click the project.

2. Go to Team > Show in History.

Alternatively, using Git command line:

```sh

git log

```

## 6. Additional Tips

### Staging View

Use the Staging View for a visual representation of your staging area:

1. Navigate to Window > Show View > Other.

2. Expand Git and select Git Staging.

### Synchronization View

To see differences between your local and remote repositories:

1. Navigate to Window > Show View > Other.

2. Expand Team and select Git Synchronize.

## Conclusion

By integrating Git with STS, you can effectively manage your source code and collaborate with team members directly from your IDE. This setup enhances productivity and maintains a smooth workflow in Spring-based projects.