Recitation 1: Version Control using BitBucket

CSE 219

Notify your recitation TA upon completion of this recitation so that they can verify your work.

In this recitation you will be introduced to the version control system (VCS) Git via BitBucket, a website that provides an online repository for your code. You will use some form of version control throughout a career in software development, and though there are many different VCS, git is among the most popular. Understanding git will help you to understand the core principles of these systems.

Introduction

Computer Science is arguably the field that is evolving the fastest. In the industry, potential employers will expect you to be up to date with the current technology and tools. So, in order to not become obsolete, you have to commit to keeping up!

In this course, you will be required to use several development tools. They may seem painful and even futile for just a semester-long project, but these (and other similar) tools are essential to professional software development.

One of the first things you will need to know are organizational procedures, including the build process. The build refers to the required steps the company uses for building, testing, and ultimately deploying completed work. Not all companies use the same tools, but learning on one set of such tools helps you learn how to learn.

The crucial tool categories are:

• Integrated Development Environment (IDE), which includes a debugger and profiler.
  − You can use either NetBeans or IntelliJ IDEA. CSE 219 “official” support and instructions will only be provided for NetBeans, however.

• Version Control System
  − We will use Git.

• Build Process Management (e.g., Maven)

• Modeling Tool (e.g., VioletUML)

• Unit Testing (e.g., JUnit, Arquillian)

Both NetBeans documentation and IntelliJ IDEA documentation are extensive. Throughout the semester, you should visit the IDE documentation pages as and when you need to, and understand how to perform basic tasks that aid in software development using these tools.

Task

To set up a private, online, git repository to commit your code and to clone it locally using your IDE.

• Create a new BitBucket Account: Use your SBU Gmail for the email address to create this account. For your username, please follow the convention FirstnameLastnameCSE219. For example, Ritwik Banerjee’s account name would be RitwikBanerjeeCSE219. This is important because the TAs need your name to identify you, so you must fill that in properly. The camel-case notation must be strictly followed for easy readability. Throughout the semester, there may be points associated with proper and regular use of this account for version control of your code. Also keep in mind that you may want to publicize your work in future through this account, so use your real and correct name in the account setup.

• Create a private repository called “recitation-1”. Be sure to choose the repository type as git (ignore the other options like ‘Mercurial’).

• Save the repository URL for later use.

It will be of the form https://bitbucket.org/USERNAME/REPOSITORYNAME.

• Add the recitation TAs to this repository. For this, go to Settings, and then User and group access, where you can add them by using their email addresses with “Read” permission. This will be important for your homework assignments, where you will have to add the TA grading your assignment.

• “Clone” the repository locally by using your IDE. For NetBeans, go to Team → Git → Clone. You should then be presented with a dialog where you should enter the full BitBucket URL of your repository as well as your username and password for your BitBucket account. Finally, you should specify the directory into which you wish to clone the repository. Make a directory on your local machine named

---

1You will need to do this with each assignment this semester
“recitation-1” (make sure it is in a location where you have permission to read/write), and then browse to that directory. Click next through the next screen and make sure the clone name is “recitation-1”. Note that one could change this to do a little sandboxing (i.e. branch the project to work on some experimental changes). Since there is no code in the repo yet, you may leave the Checkout Branch as No Branch and the Remote Name as origin. Once you press Finish you’ll get a little dialog box stating Clone Completed that will prompt you Do you want to create an IDE project from the cloned sources?, at which time you should press Create Project.

- **Make a new JavaFX Project:** Clicking Create Project will trigger the new project wizard. Once open, select JavaFX Application and name the project Recitation1 making sure to leave Create Application Class checked so that it makes an app named Recitation1. Once you have made the project, try running it. It should pop open a simple little window with a button which when you click it, should result in printing Hello World! to the console window.

- **Commit the project:** Right-click on your project and select Git → Commit. When the Commit window opens enter a Commit Message saying Initial Commit: this contains the initial Recitation1.java JAVAFX application. Under Files to Commit check Recitation1.java and uncheck all the other files.

- **Modify code:** The application currently only has one button. Let’s add another one. Add to the source code such that the application window has 2 buttons, one that says Hello World!, and one that says Goodbye Cruel World!. For this second button, clicking on it should print that text to the console window.

- **Commit your changes** and make sure you provide a Commit Message stating the changes you made to the project.

- **Push your changes:** Commits are made to your local repository. None of your changes are on Bitbucket yet! Now let’s push these changes to BitBucket. To do this, right-click on your project and select Git → Remote → Push and for the repository choose Select Configured Git Repository Location which should have your previously cloned BitBucket repository. Once selected, click Next. You should then be prompted to select a Local Branch, for which you should select master. The same then goes for the Remote Branch. You should select these options and press Finish and when prompted to track remote branches, say yes.

- **Now you can view your commits on Bitbucket.** In your Web browser, go to your BitBucket repository and go to the Source tab. There you should see a directory for your code and if you click on it you will be able to navigate to the .java file you committed. In addition, if you click on the Commit tab you’ll find the two Commits you’ve made, which should each include their own Commit Message. You must always provide good commit messages with your commits. In addition, you should commit periodically, after each significant, tested, change. This is important for your own sanity, especially to prevent any catastrophic computer failure from causing you to lose all of your work. **You will never lose your code again!** Try clicking on the Commits now and you will find color coded code that will help you see what actually changed in the source file as a result of each commit.