Lab 2 - Setup and Introductory Webapp

Summary

The purpose of this assignment is to complete our setup:

- Create a MySQL user, database and corresponding NetBeans connectivity for use in remaining projects
- Integrate source control with GitHub
- Learn the basics of Maven by creating our first project
- Create a simple web application to introduce yourself
- Execute and review a Sonar analysis

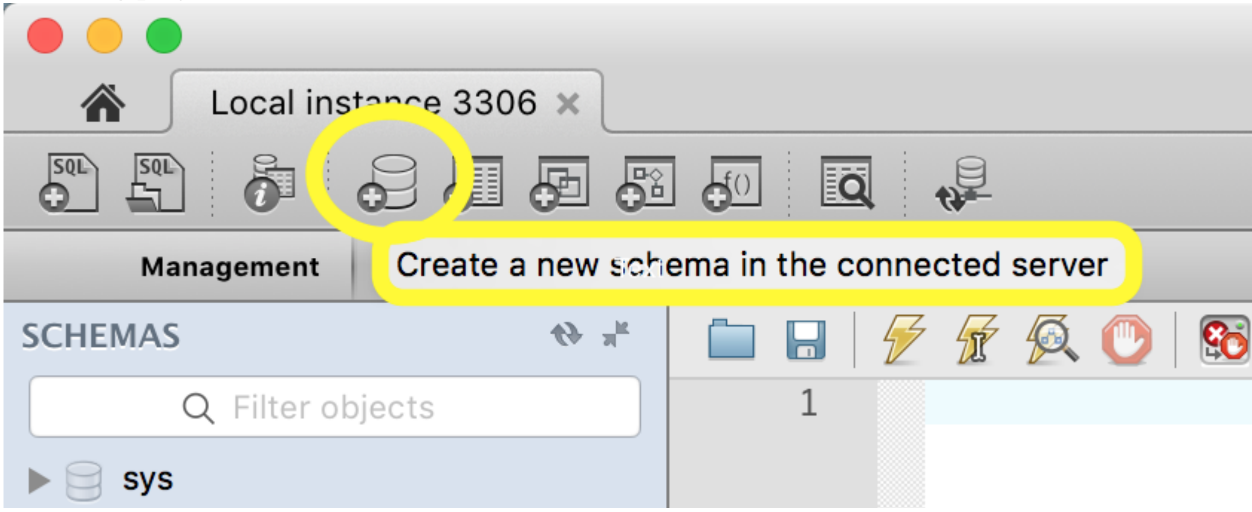
Requirements

Documentation

Create a Word or PDF document for your documentation requirements.

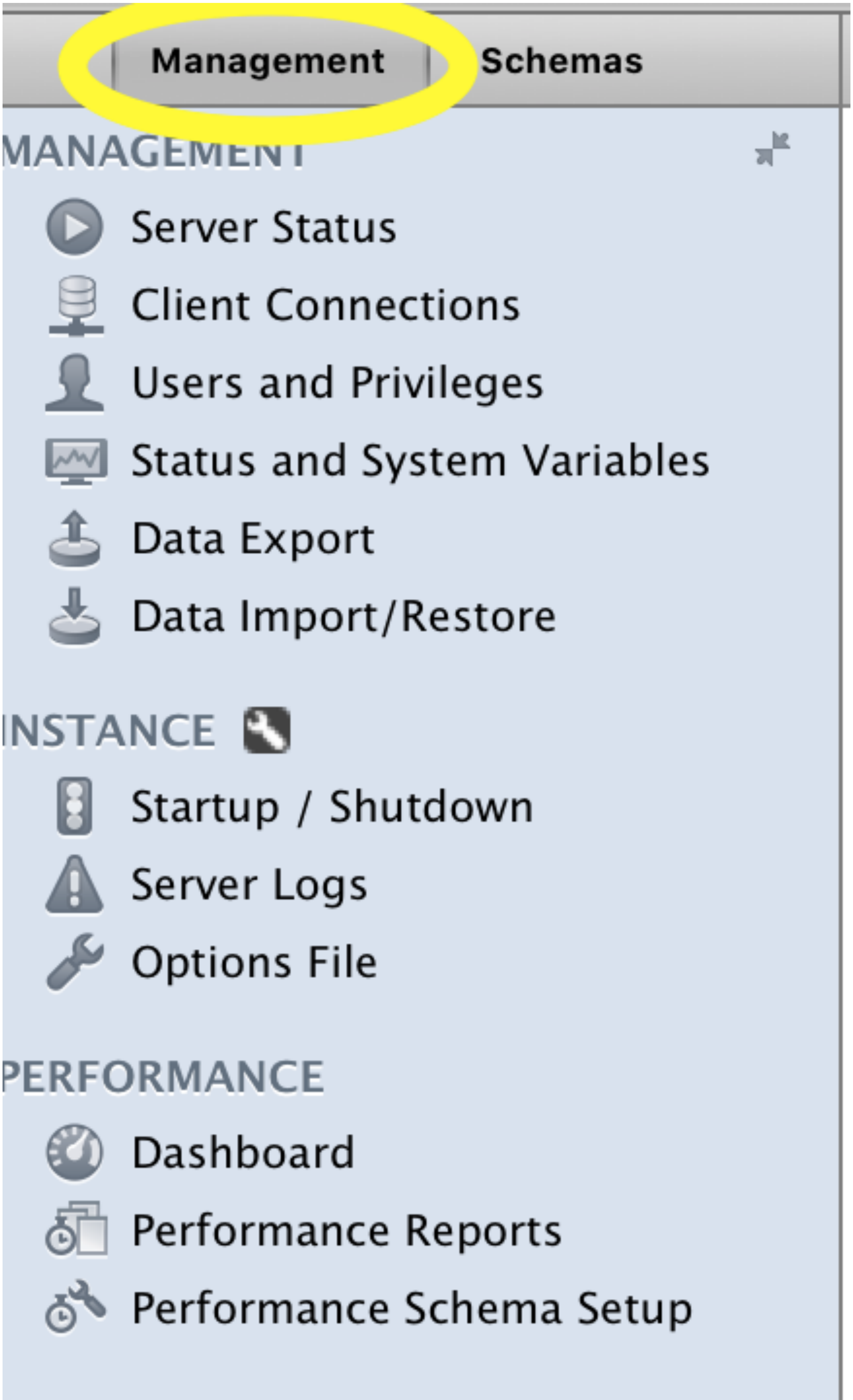
MySQL Database and User

The purpose of this is to simulate real organizations and projects at work. Most of you will not have access to OS or database root passwords. Following the principle of least privilege, you (or your application) will have access to an account that has enough privileges to function, but not system-wide administrative privileges. We will discuss this more in class.

1. Create a new MySQL database schema named **itmd4515**. We will be using this database for the remaining projects this semester.
    1. 

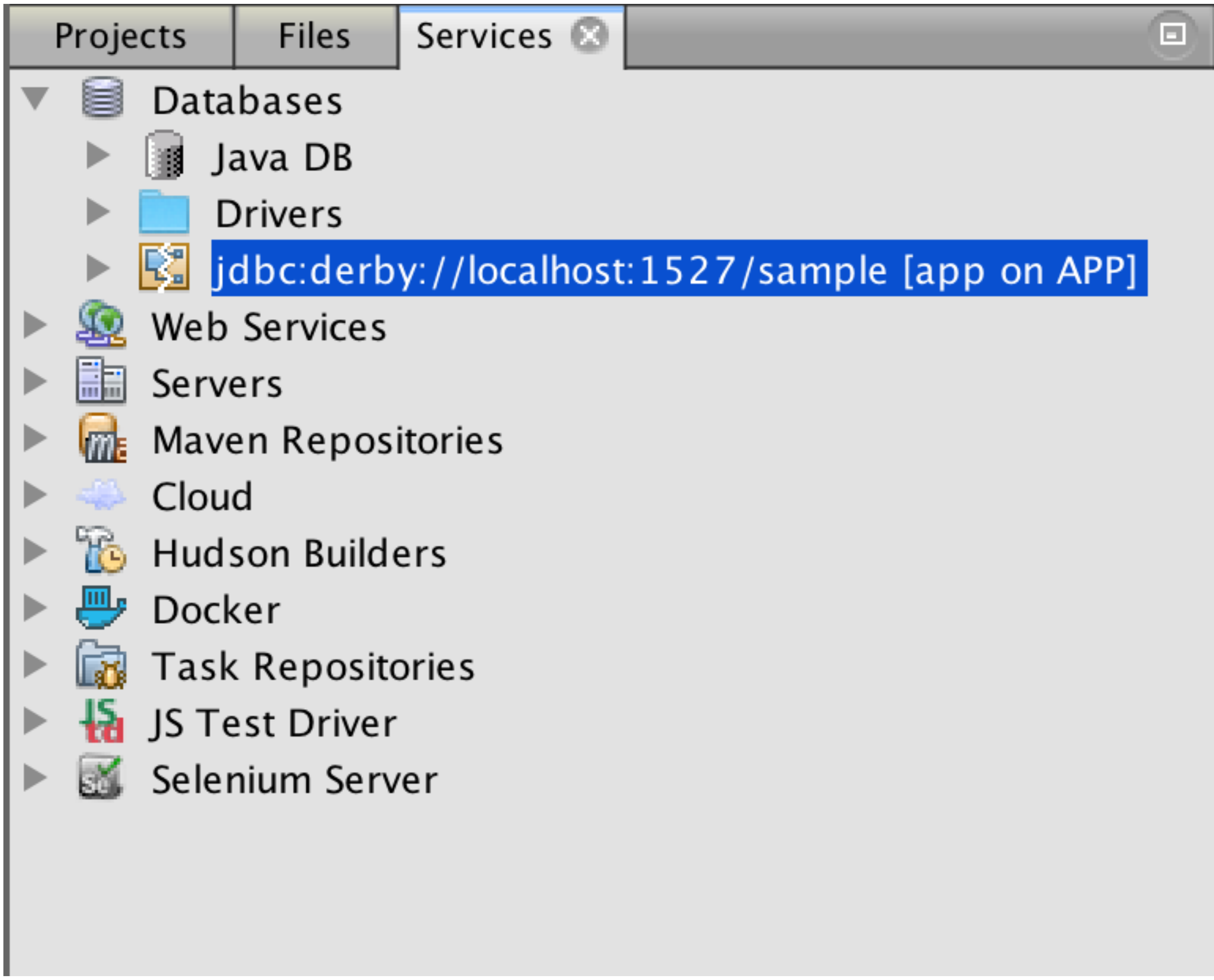
    2. Enter a schema name of itmd4515. Leave the default collation blank. Click apply.

    3. Click apply on the subsequent confirmation window. This window displays the actual SQL commands used to create the database.
2. Create a MySQL user named **itmd4515** with full rights to the database(s). The user must have a password of **itmd4515**. Do not use the root user for connection pools. This user may already exist from your prior home work. If so, you do not need to re-create.
    1. If your Workbench is displaying **Schemas,** then click on **Management** or **Administration** to access the management options


Management
Schemas
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup

3. Click on **Users and Privileges**
4. Click on **Add Account**
   1. Give your new account a Login Name of itmd4515
   2. Limit your new account to Hosts Matching localhost
   3. Give your new account a password of itmd4515
   4. Click on Apply to create the account
   5. Click on **Schema Privileges,** add your new **itmd4515** schema, select "All" privileges, and click apply.
5. Document this with a screenshot showing that you have created the user, and granted full permissions to ***only*** the itmd4515 schema.

3. Create a NetBeans database connection to your new MySQL database. This database connection will be by various tools within NetBeans.
   1. Click on the **Services** tab

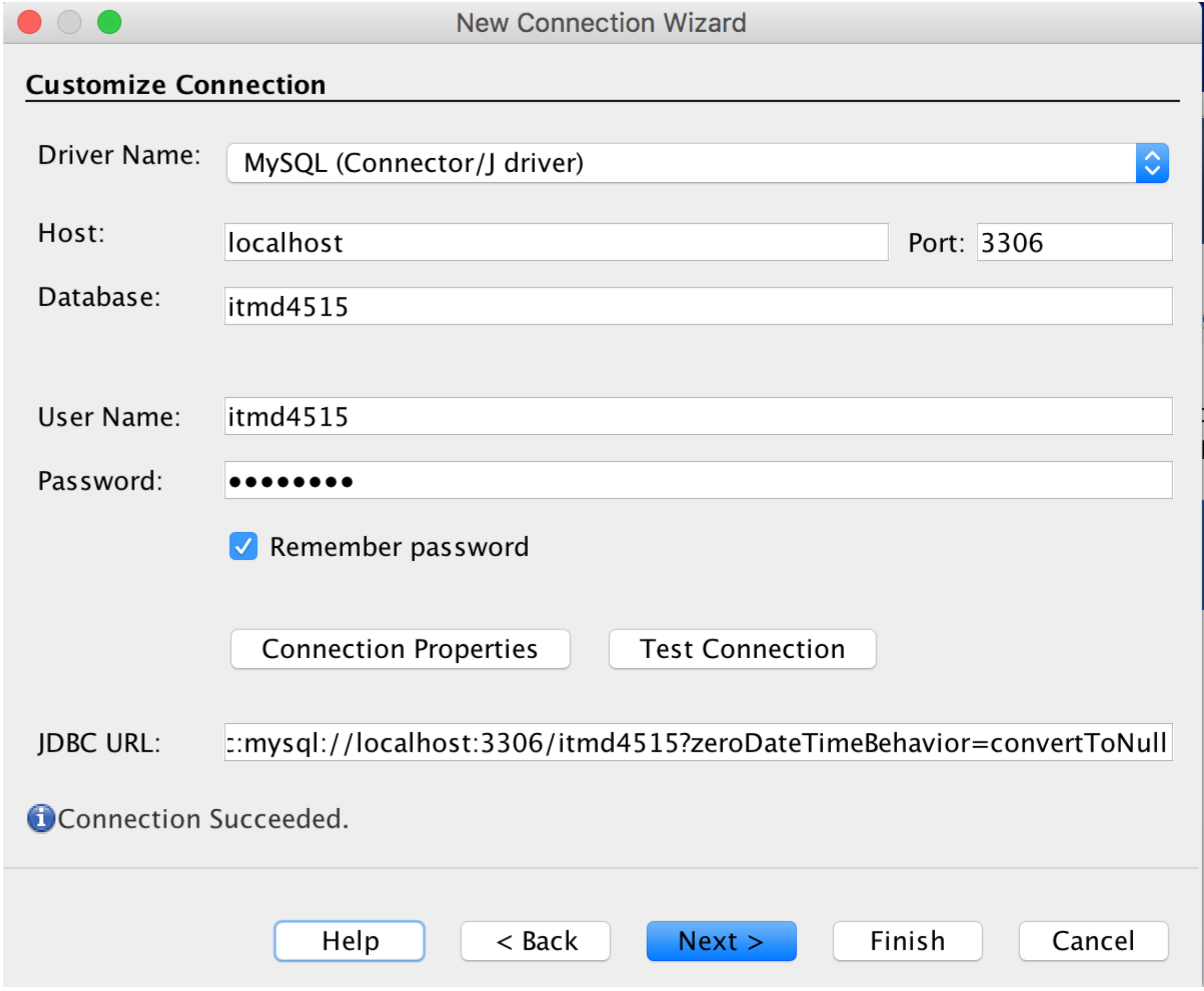
   2.
   3. Right-click on Databases and choose **New Connection**
   4. Select the MySQL JDBC Driver and click Next

      In NetBeans 12, you may need to download MySQL Connector/J (the MySQL JDBC driver) manually from here:

https://dev.mysql.com/downloads/connector/j/. Choose the platform-independent zip file.

Save it to a directory of your choosing and register the driver by following the steps in the wizard.

5. Change the database to itmd4515, enter your credentials, and test the connection



New Connection Wizard

Customize Connection

Driver Name: MySQL (Connector/J driver)

Host: localhost Port: 3306

Database: itmd4515

User Name: itmd4515

Password: ●●●●●●●●

☑ Remember password

JDBC URL: ::mysql://localhost:3306/itmd4515?zeroDateTimeBehavior=convertToNull

Connection Succeeded.

6.
7. If you experience an error related to the Timezone when you test the connection, you can adjust the JDBC URL to explicitly include the timezone:

```
jdbc:mysql://localhost:3306/itmd4515?zeroDateTimeBehavior=convert
ToNull&serverTimezone=America/Chicago
```

8. I will explain during class what the difference is between a connection and a server with Services → Databases. If this process registers a MySQL server for you in NetBeans, you can remove it (especially if it is causing timezone errors). I will explain/demo in class.
9. Document this with a screenshot showing the successful results of the test.
10. Click Finish

GitHub and Project Setup

The purpose of this is two-fold. First, to simulate real organizations and projects. Git is widely used, if not industry standard at this point. Second, this will enable me to better assist you with debugging your issues this semester.

For Lab 2, perform the following steps:

1. Create a GitHub Account with your **myIIT username,** and using your @hawk.iit.edu email as your email.
2. Open the Blackboard assignment for Lab 2
3. Click on the assignment invitation URL
4. Login to GitHub using the account your just created with your **myIIT username**
5. Join the classroom roster by selecting *your* **myIIT username** from the list

   Make sure you select your username and not someone else's username. GitHub has made this a one-click deal. If you mistakenly click someone else's username, you will have to email me and ask for help.

6. Accept the assignment
7. Your Lab 2 repository will then be created automatically and shared with me
8. You will receive a confirmation email:
   1. @spyrison has invited you to collaborate on the itmd4515/itmd4515-f20-lab2-sspyriso repository

Next create a **Java with Maven Web Application** project in NetBeans with the following coordinates and configuration:

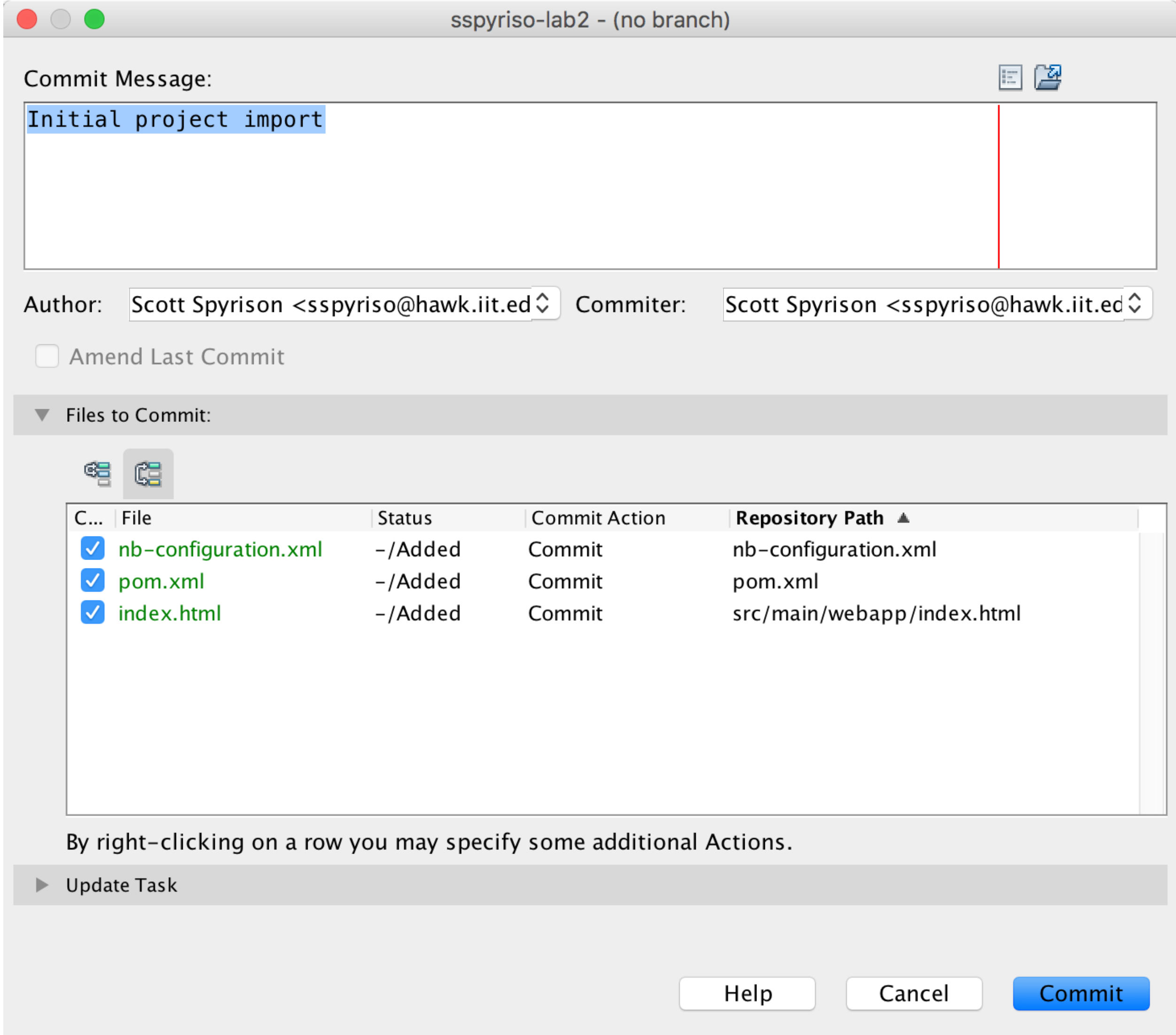
1. Project Name: **uid-lab2** (uid means your myIIT username)
2. Artifact ID: **uid-lab2** (default is fine)
3. Group ID: **edu.iit.sat.itmd4515.uid**
4. Version: **1.0-SNAPSHOT** (default is fine)
5. Use a base package for your Java code of **edu.iit.sat.itmd4515.uid**
6. Under Server, choose Payara Server and Java EE 8 Web

For example, my NetBeans project will be called “**sspyriso-lab2**" with a base package and group id of **edu.iit.sat.itmd4515.sspyriso**.

Deviating from the package convention given above will mean that you can not benefit from Sonar and other automated tools, and I will not be able to fix this. Please follow the specification!

Finally, configure your project for source control:

1. Using NetBeans or the Git command line, initialize your project as a Git repository and add/commit your project

    1. In NetBeans, this can be done in **Team -> Git -> Initialize Repository**
    2. At the root level of your project, choose **Team -> Commit**
    3. Enter a commit message such as "Initial project import" and Commit.
    4. 

sspyriso-lab2 - (no branch)

Commit Message:

```
Initial project import
```

Author: Scott Spyrison <sspyriso@hawk.iit.ed Commiter: Scott Spyrison <sspyriso@hawk.iit.ec

☐ Amend Last Commit

Files to Commit:

C...	File	Status	Commit Action	Repository Path ▲
☑	nb-configuration.xml	-/Added	Commit	nb-configuration.xml
☑	pom.xml	-/Added	Commit	pom.xml
☑	index.html	-/Added	Commit	src/main/webapp/index.html

By right-clicking on a row you may specify some additional Actions.

2. Add your GitHub repository as a remote repository. In NetBeans, this can be done via **Team -> Remote -> Push**. The first time you run this command, it will prompt you for the following to setup a remote:

   If you clone from a remote repository, it will automatically be named "origin." In this case, we are going to add our remote repository with the name origin to satisfy that naming convention. Refer to the "Quick Setup" help section on your GitHub repository homepage.

    1. In the subsequent Remote Repository dialog box, enter (copy/paste) the https address of your GitHub repository.
        1. For example, mine would be https://github.com/itmd4515/itmd4515-f20-lab2-sspyriso.git

    2. Enter your GitHub Username and Password
    3. Select master -> master as the local branch, and master -> origin/master as the local reference
    4. Note, this set of steps has also executed a git push for you, to the remote repository you have just configured
3. Browse your GitHub repository - especially if you are new to source control or Git.
4. Document this with a screenshot of the populated repository (Source view, after you push)

Continue to perform git push operations to your remote throughout your development process. I want to see multiple commits in your repository, with commit messages. You can do this via **Team -> Remote -> Push**, only use the repository you configured above, as opposed to setting up a new one each time.

Introductory Maven Web Application

Create an introductory web page (or set of web pages) to introduce yourself to me. This can be as simple as an electronic version of your resume, or as complex as you choose to make it. There are no technical requirements associated with the web page(s). My goal is the following:

1. Ensure your setup is fully functional, including iterative pushes to GitHub
2. Verify that everyone has basic HTML and CSS skills
3. I want to get to know you a little better

You are under no obligation to include personal information on this web page. As I said above, please consider resume content as you decide what to write. You can add additional HTML (or other) files in NetBeans. As you update HTML content, just "Run" your project. It should automatically deploy within the Glassfish environment and start your default browser to display the HTML content.

Maven project setup will be reviewed during class.

1. Document this with a screenshot. The screenshot should display your browser after the application is running inside NetBeans.
2. Submit to Blackboard
    1. Right your uid-lab2 project and select "Clean"
    2. Go to your NetBeans Projects directory. Create a zip file of the uid-lab2 folder and submit it to the Blackboard assignment.

Sonar

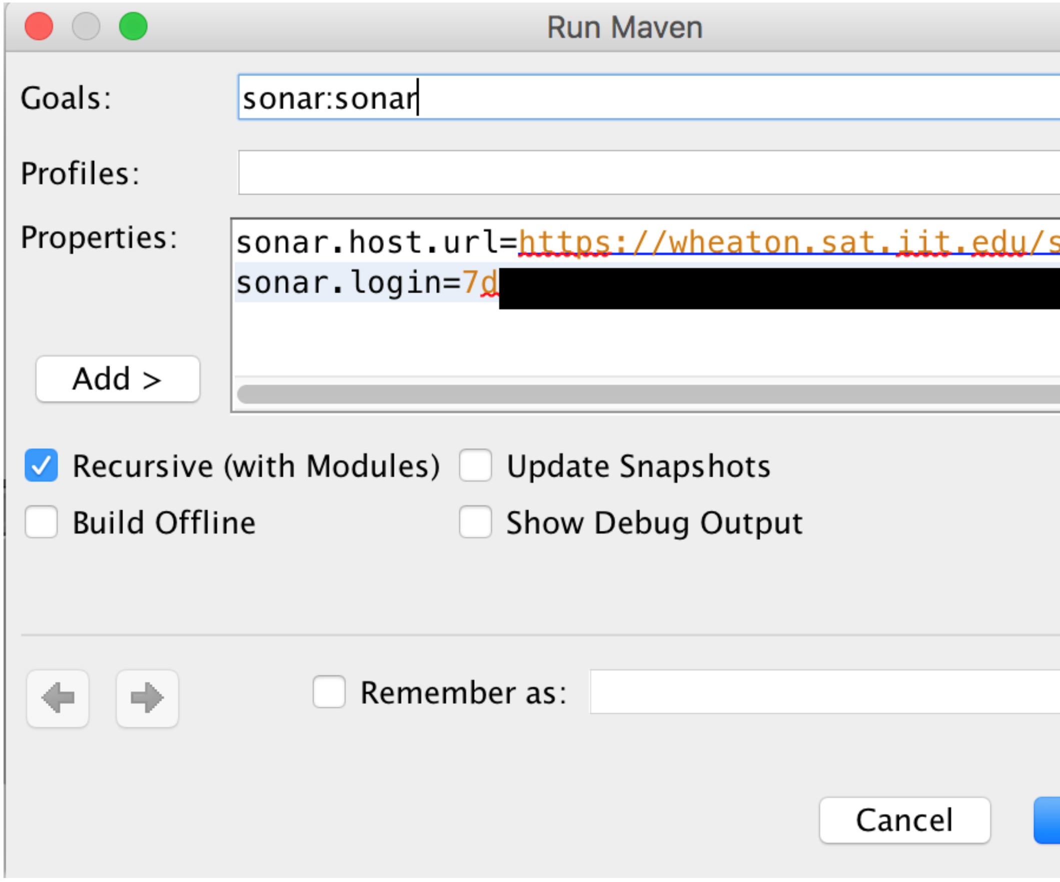
Please note if you cannot connect to the IIT Sonar server, you do not need to do this part. This is optional for Beacon students.

You will be able to make use of static code analysis via SonarQube throughout the semester. This will enable you to check your code before submitting it to me. Do the following:

1. Login and verify your account at https://wheaton.rice.iit.edu/sonar/ (use myIIT username, and your A# as the initial password. You can change your password after logging in)
2. If you click on "All Projects" you should see only your own.
3. Click on your username, and access "My Account"
4. Click on "Security" and then generate a new token. Name your token (you guessed it) **UID ITMD4515 21S**
5. Copy your token to the clipboard, and save it somewhere (secure) you will remember! I recommend KeePass for a local password safe, or use whatever you are accustomed to.
6. In NetBeans, run a Sonar analysis as follows:
    1. First, you will need to build your project again. Right-click your project and choose Clean and Build. If there is no compiled Java code the Sonar analysis will fail (and you just did a clean in the step above before submitting to Blackboard!)
    2. Right click your project and choose **Run Maven → Goals**
    3. Enter the following information:
        1. Goals: `sonar:sonar`
        2. Properties (each on its own line)

        3.
           ```
           sonar.host.url=https://wheaton.rice.iit.edu/sonar/
           sonar.login=PASTE_YOUR_API_KEY_FROM_SONAR
           ```

        4. For example:



Goals: sonar:sonar

Profiles:

Properties:

```
sonar.host.url=https://wheaton.sat.iit.edu/s
sonar.login=7d[redacted]
```

☑ Recursive (with Modules) ☐ Update Snapshots

☐ Build Offline ☐ Show Debug Output

☐ Remember as:

1.

    5. Press OK to run the analysis. Watch the output carefully in your Outlook window.

7. Review the analysis in Sonar by clicking on your project and reviewing the reported issues and "code smells"
8. Document this with a screenshot. The screenshot should display your project after a successful code analysis.

I am only providing NetBeans instructions right now - if you wish to run maven on the command line, please follow the documentation:
https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner+for+Maven

Post a question to Confluence if you need assistance.

I ***will not*** be using Sonar as a grading tool for Lab 2. This is to validate your setup, show the capabilities, and provide you with a tool you can use throughout the semester to improve your Java code.