**GitHub Repository**

**What is GitHub:**

GitHub is a repository where you can create repos and store your source code, and maintain versions of your code. Optum has a GitHub at - <https://github.optum.com/>

**What is git:**

git is a command line interface (CLI) to communicate with your github. You need to setup your git with your credentials to authenticate yourself and communicate with github.

For our UPM environment you may not be using CLI. But if you wish you could use git.

Most of the time, you will be using Project Creator from DevKit to create or check out your projects from github repo.

**You may familiarize more yourself with the information from the links below:**

1. Access GitHub from here: <https://github.optum.com/>
2. Request Tortoise Client: <http://appstore.uhc.com/AppInfo/AppVersionId/14168?BackToList=/AppList/AppList>
3. Requesting access to Optum Github: <https://hubconnect.uhg.com/docs/DOC-90949>
4. Codehub to Github migration faqs: <https://hubconnect.uhg.com/docs/DOC-107887>
5. Github migration project: <https://github.optum.com/CDE/ghe-migration>
6. Project creator updates for Github: <https://hubconnect.uhg.com/groups/hbs-api-engine/blog/2017/08/18/project-creator-updates>
7. Github training documentation: <https://hubconnect.uhg.com/docs/DOC-113608>
8. UPM4 Github Organization: <https://github.optum.com/api-engine-upm4>
9. API Engine Github Org (you can see some example repos here): <https://github.optum.com/api-engine-lacey>
10. Follow the [CodeHub to GitHub Enterprise Migration](https://hubconnect.uhg.com/groups/codehub/projects/codehub-to-github-enterprise-migration) project page for migration updates and FAQs
11. Follow the [GitHub Hub Connect](https://hubconnect.uhg.com/groups/github) page to stay updated on the tool and collaborate with the existing community
12. Read through our [migration guidelines](https://hubconnect.uhg.com/docs/DOC-113746) before taking action on prepping for migration

**Some understanding on git and GitHub:**

git clone command clones a repository from GitHub to your local directory, called local repository. You make changes to the files in your local repository. If you add new files to your local repository you issue git add command. Then, you run git commit command to commit the changes to your local repository. Notice your git commit command will not commit the changes to the GitHub repository, until you run one more command: git push origin master

master is the place where your code resides, by default, GitHub’s branch is master, also called master branch. At UPM we standardize to use the master branch only where all the checkouts and commits take place. No feature branches be created and worked with.

**Access to GitHub:**

Request access to GitHub by submitting a Secure platform request for your MS id for the group:

**github\_users**

**Request Git 2.7 from AppStore at:**

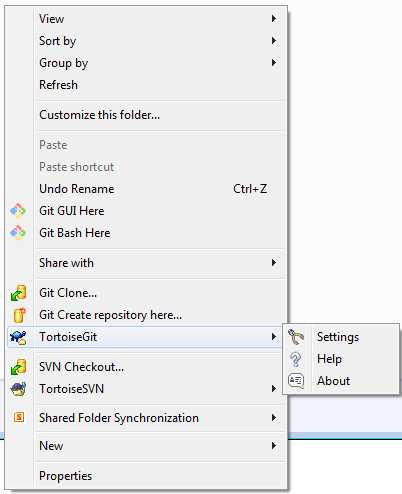
<http://appstore.uhc.com/AppInfo/AppVersionId/12090?BackToList=/AppList/AppList>

**Tortoise SVN style Tortoise Git client:**

If you use the Tortoise SVN client, then you will also probably want the Tortoise Git client:

<http://appstore.uhc.com/AppInfo/AppVersionId/14168?BackToList=/AppList/AppList>

Once the Tortoise Git client is installed you should be able to see the below popup menu on right clik, with Git GUI, Git Bash, Git Clone etc.



**Create or Checkout a project from GitHub Repo:**

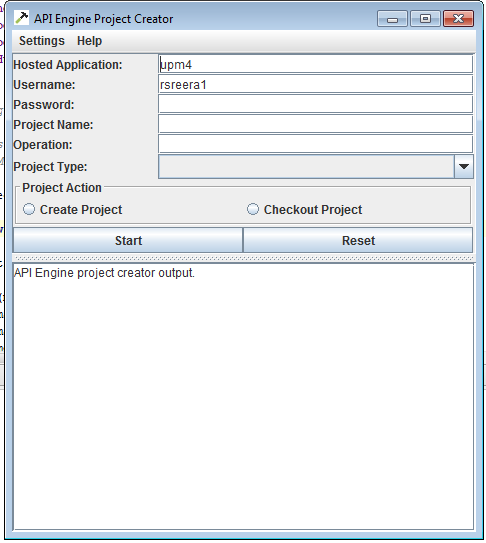
Use the project creator from **DevKit** tool available at: <https://codehub.optum.com/api-engine/apiengine-local-dev-kit/branches/RB-2.0.x>

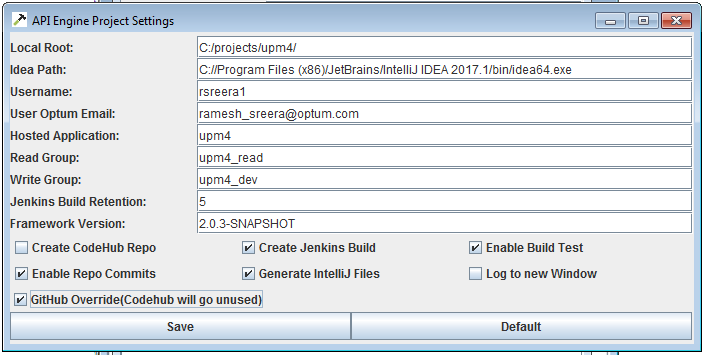
Use this Project Creator jar file to create a project or checkout a project from GitHub. Double click on the project creator, you will be greeted with a window as shown below. Click on settings then on Edit. That will pop up a window to edit project engine settings where you select GitHub Repo to work with.

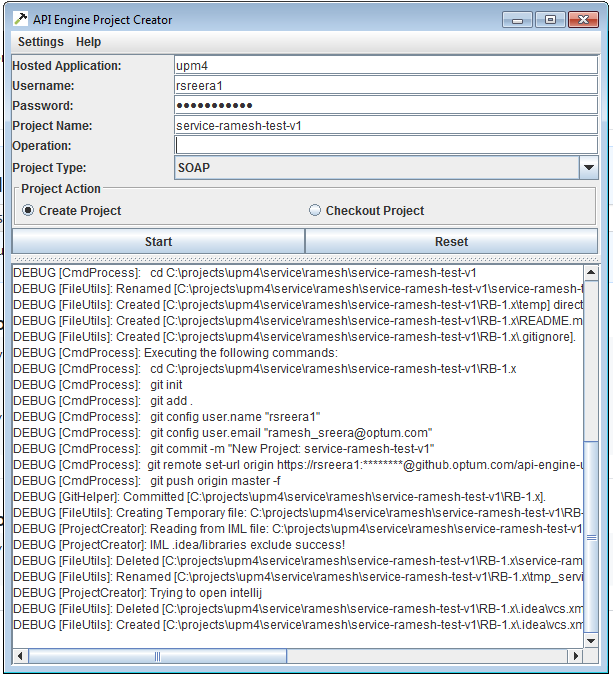
Uncheck the box “Create CodeHub Repo”

Check the box “GitHub Override”

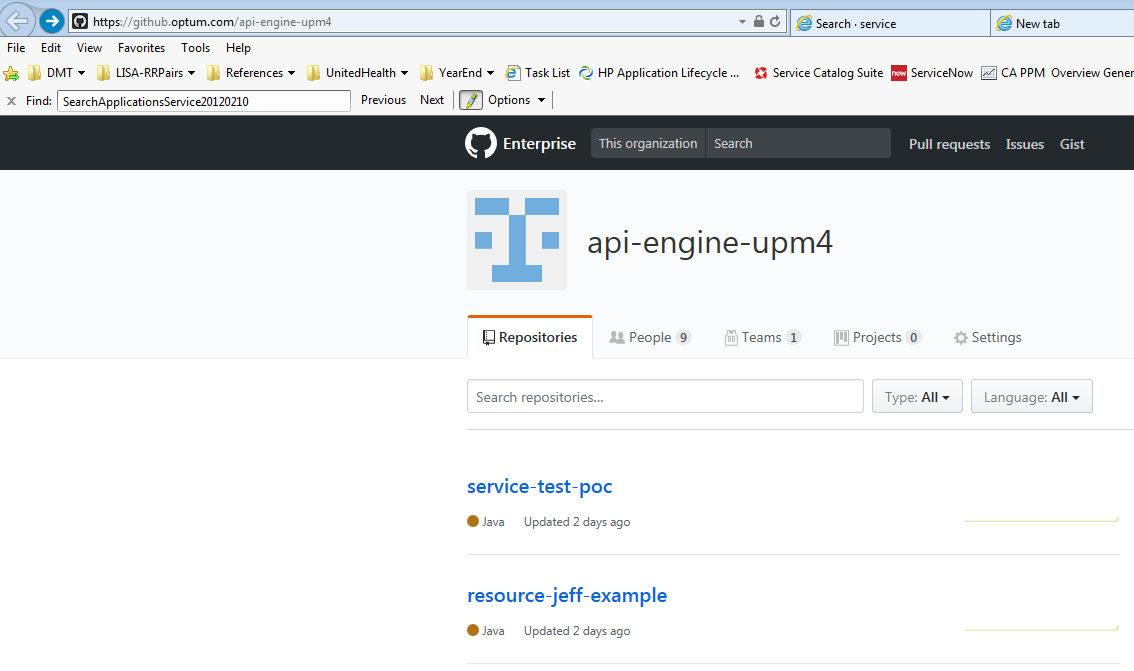
Save, and close the dialog boxes, to run the MainDriver with saved settings.





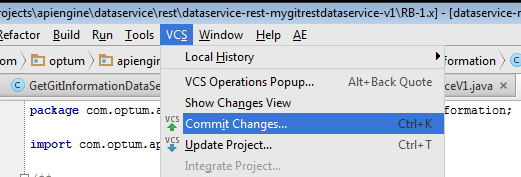


This should create your project in GitHub and do a checkout into your local folder.

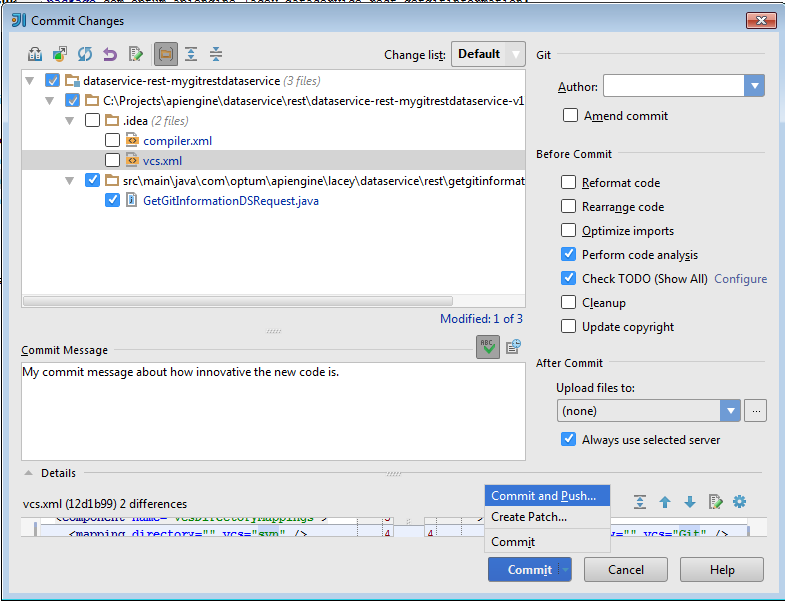


**Working with GitHub from within Intellij:**

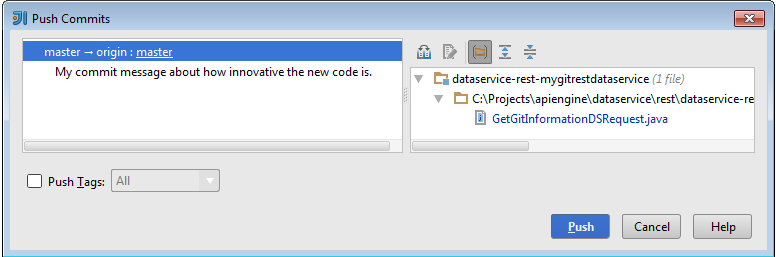
Step 1: Pick Commit Changes

[](https://hubconnect.uhg.com/servlet/JiveServlet/showImage/38-43634-364000/pastedImage_2.png)

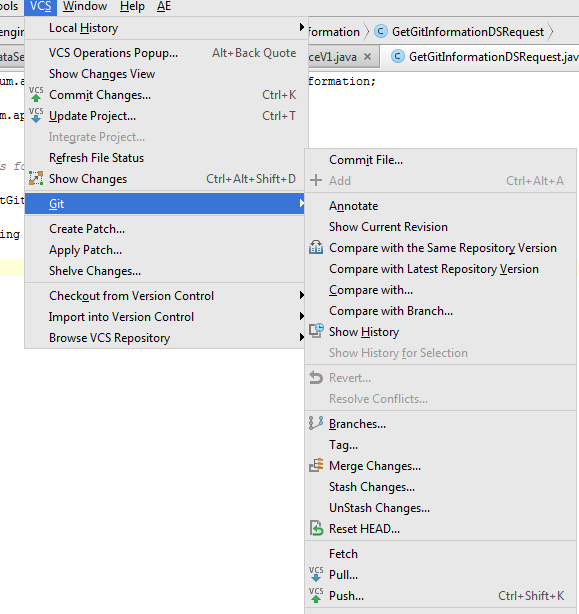
Step 2: Make sure to do the Commit and Push option, as a commit just establishes the change revision locally, whereas the action of the PUSH actually sends the commit to the remote master branch

[](https://hubconnect.uhg.com/servlet/JiveServlet/showImage/38-43634-364001/pastedImage_3.png)

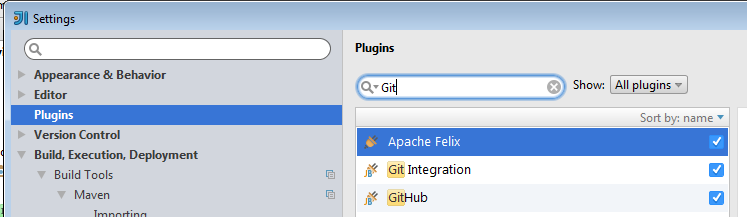
Step 3: You should see a dialog pop up detailing where your push is going (origin -> master) , if so click push and send your source code along its way!

[](https://hubconnect.uhg.com/servlet/JiveServlet/showImage/38-43634-364002/pastedImage_4.png)

On a side note the Git menu has some great options as well, such as comparing your local against the latest repository options and many other choices you are familiar with in SVN:

[](https://hubconnect.uhg.com/servlet/JiveServlet/showImage/38-43634-364003/pastedImage_5.png)

**I should note as well that to enable fully integrated Github + Git in Intellij you need these two plugins installed (Git Integration, GitHub):**

[](https://hubconnect.uhg.com/servlet/JiveServlet/showImage/38-43634-364004/pastedImage_6.png)