

ThoughtWorks®

LEARNING TO CREATE AUTOMATED TESTS

Darrell Grainger - Quality Analyst
dgrainger@thoughtworks.com

1

HOW TO APPROACH AUTOMATION

- When I learned a new language, e.g. French, I thought in English and converted it to French
- After a while I started thinking in French
- I approached automation the same way
- I knew how to create a manual test case
- I just needed to figure out how to convert it to programming
- After a while, I'd start thinking about the test case as a set of programming steps rather than manual steps
- Until that happens, I recommend writing out a test case in your native tongue then converting it to programming automation

HOW TO GET A PROJECT STARTED

- Test automation is programming.
- So basic programming practices are part of writing automation
- So you should know the following:
 - Having good tools to develop with (IDE, compiler, etc.)
 - Using source control (usually Git but some still use CVS or SVN)
 - How to program in the test automation language
 - Frameworks which help with test automation (unit frameworks)
 - How to use a shell (important for adding to a build pipeline)
- For this presentation I'm going to use Git, Java and Selenium
- For a good shell see the checklist at the end of this presentation

JAVA DEVELOPMENT KIT

- Your computer might have Java already installed
- If it does not go to the Java website and download the latest version of JDK 8.0
- You'll need the full JDK and not just the JRE (Java Runtime Environment)
- Do not download version 9.0 or 10.0
- Selenium currently works with Java 8.0
- Current version of Java, 8.0 build 181:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- See the list at the end of this presentation for the link to Java

4

JAVA DEVELOPMENT TOOLS

- There are a few good, free Java development tools
- Most are called Integrated Development Environments or IDE
- Some tools are completely free and have to paid version
- Others have a community edition for free and something with more features for paid
- I prefer to use the community edition because often companies will opt for the paid version
- So you can build on what you learned from the community edition
- I like IntelliJ IDEA from Jet Brains
- You can download it from <https://www.jetbrains.com/idea/>

WHAT DOES A GOOD IDE DO FOR YOU?

- When you start developing a Java application in say notepad then you need to know the following:
 - The syntax of the language
 - Which *import* statements you need
 - How to include a library and where to download it from
 - Help on using methods and classes
- With something like IntelliJ, it will help you with these things
- For example, to print something I'd use *System.out.println()*
- In IntelliJ I'd just enter: **sout** and it would expand to the full command
- You'll learn more as we go along

SOURCE CONTROL - WHERE TO GET GIT FROM

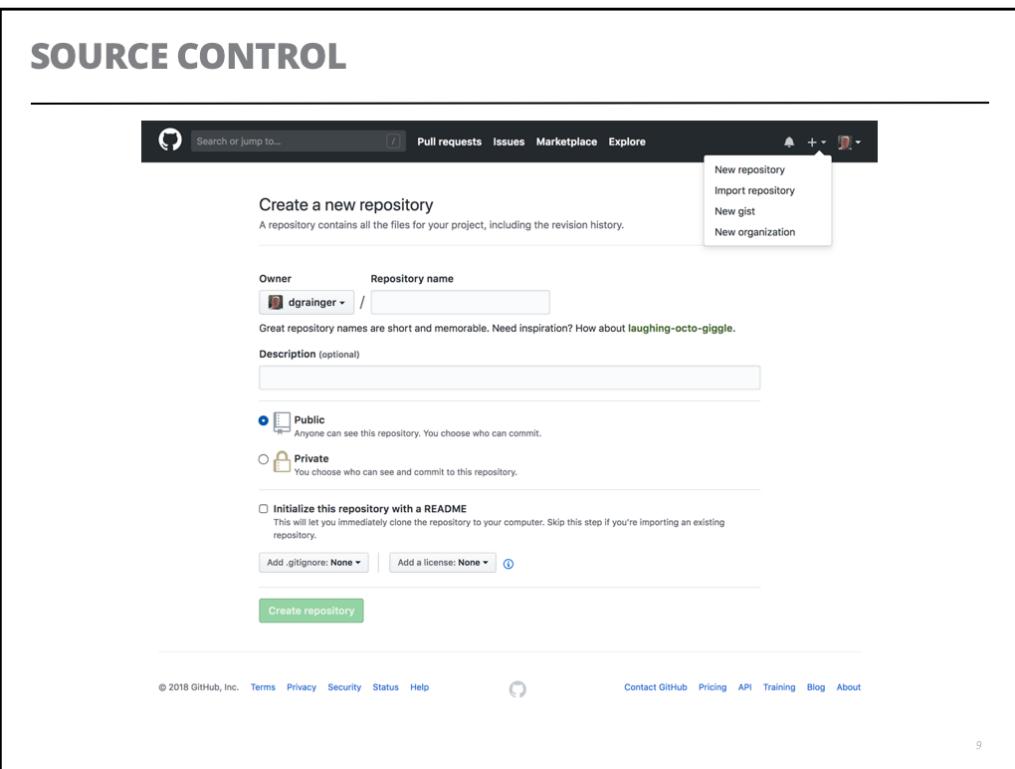
- macOS
 - If you are using a macOS computer then it comes with Git installed
- Windows
 - If you are using Windows you'll need to install Git
 - I'd recommend a version of Git which comes with a Bourne shell
 - If you can use Git for Bourne shell you can use it on Linux and macOS
 - Download it from: <https://git-scm.com/download/win>
- Linux
 - It depends on your Linux distribution
 - It might already be installed
 - If not, you'll have to figure out how to install *packages* and add Git

7

SOURCE CONTROL

- When you create test automation it will take weeks, possibly months to create everything
- How do you save your work each day?
- You can make backups but restoring a backup can be difficult
- This is where source control is helpful
- It is sort of like creating backups in an organized and easy way to restore
- To get started, we'll use GitHub
- Go to <https://www.github.com> and create a free account
- Once you have an account and are logged in we'll create a repository

8



Point out how to open the menu on the + symbol

Give the repository a name

Give a description if you want

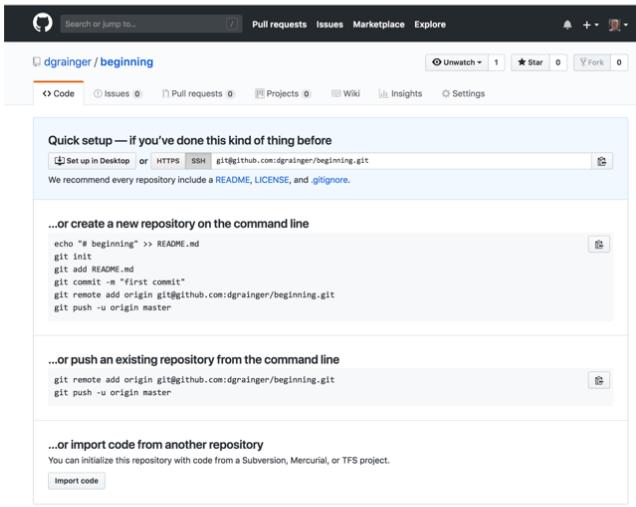
Leave it public (private repositories cost money)

You can add a README later

So will skip that

And we'll add a .gitignore manually as well

SOURCE CONTROL



The screenshot shows the GitHub interface for creating a new repository. At the top, there's a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below the header, the repository name 'dgrainer / beginning' is displayed, along with options to Unwatch, star, fork, and settings. A navigation bar below the repository name includes tabs for Code, Issues, Pull requests, Projects, Wiki, Insights, and Settings. The main content area is titled 'Quick setup — if you've done this kind of thing before' and provides instructions for setting up the repository. It includes three code snippets for command-line users:

- ...if you've done this kind of thing before:

```
Set up in Desktop or HTTPS git@github.com:dgrainer/beginning.git
```

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).
- ...or create a new repository on the command line:

```
echo "# beginning" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:dgrainer/beginning.git
git push -u origin master
```
- ...or push an existing repository from the command line:

```
git remote add origin git@github.com:dgrainer/beginning.git
git push -u origin master
```

Below these snippets, there's a section for importing code from another repository, with a 'Import code' button. A note at the bottom says: 'ProTip! Use the URL for this page when adding GitHub as a remote.' At the very bottom of the page, there are links for Contact GitHub, Pricing, API, Training, Blog, and About.

After you create the repository you should see something like this

On the line which says HTTPS and SSH you want to copy the URL into your clipboard

MAKING A LOCAL COPY TO WORK ON

```
Last login: Fri Aug 31 07:06:12 on ttys001
phoebe2:~ darrellgrainger$ cd workspace
[phoebe2:workspace darrellgrainger]$ git clone git@github.com:dgrainger/beginning.git
Cloning into 'beginning'...
warning: You appear to have cloned an empty repository.
[phoebe2:workspace darrellgrainger]$ cd beginning/
[phoebe2:beginning darrellgrainger]$ ls -l
[phoebe2:beginning darrellgrainger]$ ls -la
total 0
drwxr-x--x  3 darrellgrainger  staff  96 Aug 31 14:31 .
drwxr-x--x  15 darrellgrainger  staff  488 Aug 31 14:31 ..
drwxr-x--x  9 darrellgrainger  staff  288 Aug 31 14:31 .git
phoebe2:beginning darrellgrainger$
```

- To make a local copy you want to open a shell on your computer
- Make a directory to put the work in using ***mkdir workspace***
 - I made the directory already, so I just change directory using ***cd workspace***
 - You will ***mkdir*** once then ***cd*** to it all other times
- Then you want to enter ***git clone*** followed by a paste from your clipboard
- This is assuming the URL from the previous page is still in your clipboard
- If you called the repository ***beginning*** then ***cd beginning*** to switch to it
- Once in the repository you can use ***git*** commands

11

Talk about how the top window here is my shell and some commands I have typed

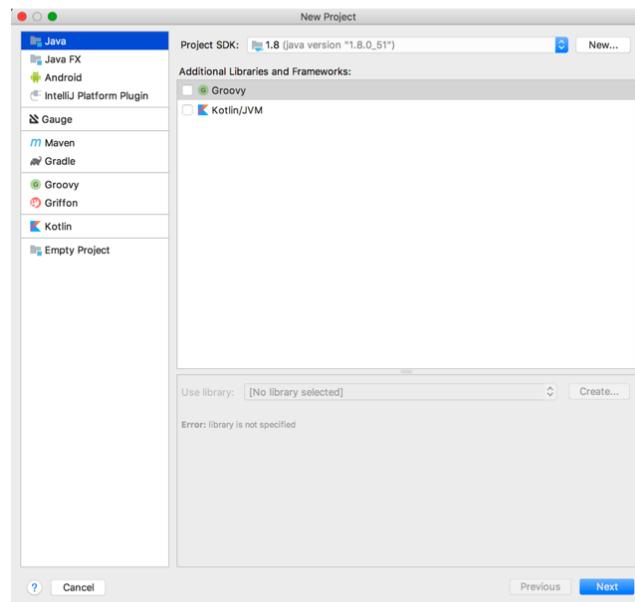
LET'S SEE WHAT IT LOOKS LIKE IN THE IDE



12

Assuming you downloaded and installed IntelliJ, this is the startup screen
We want to select Create New Project

LET'S SEE WHAT IT LOOKS LIKE IN THE IDE



13

The first page of creating a new project is picking the language

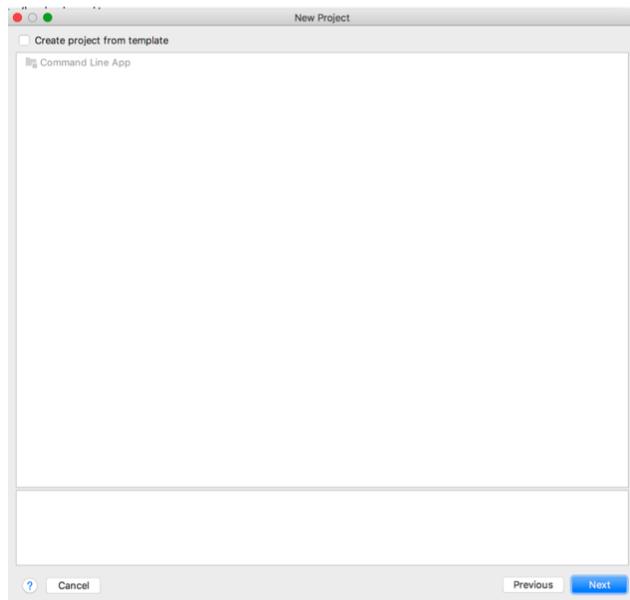
We are going to use Java

Java should already be installed on the machine and added to the IntelliJ configuration

If it is not, we'll have to cancel here, install Java and add it to IntelliJ

Then come back to this screen

LET'S SEE WHAT IT LOOKS LIKE IN THE IDE



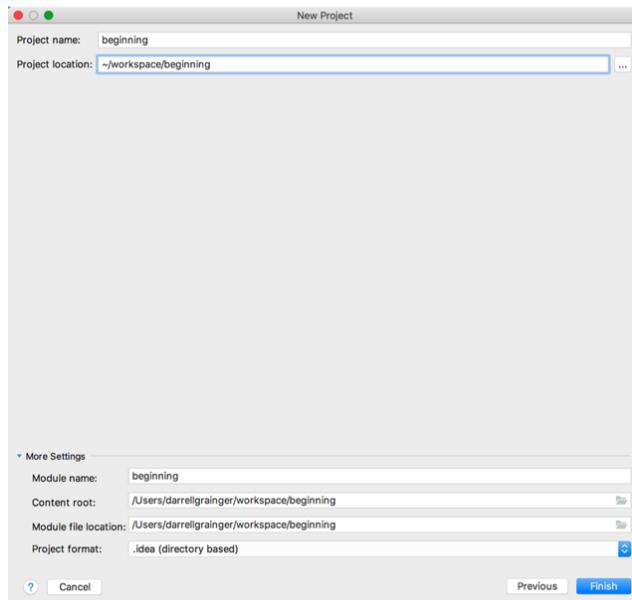
14

After clicking Next button we come to this screen

Nothing to change here

Click Next button again

LET'S SEE WHAT IT LOOKS LIKE IN THE IDE



15

On the final screen we want to edit the location.

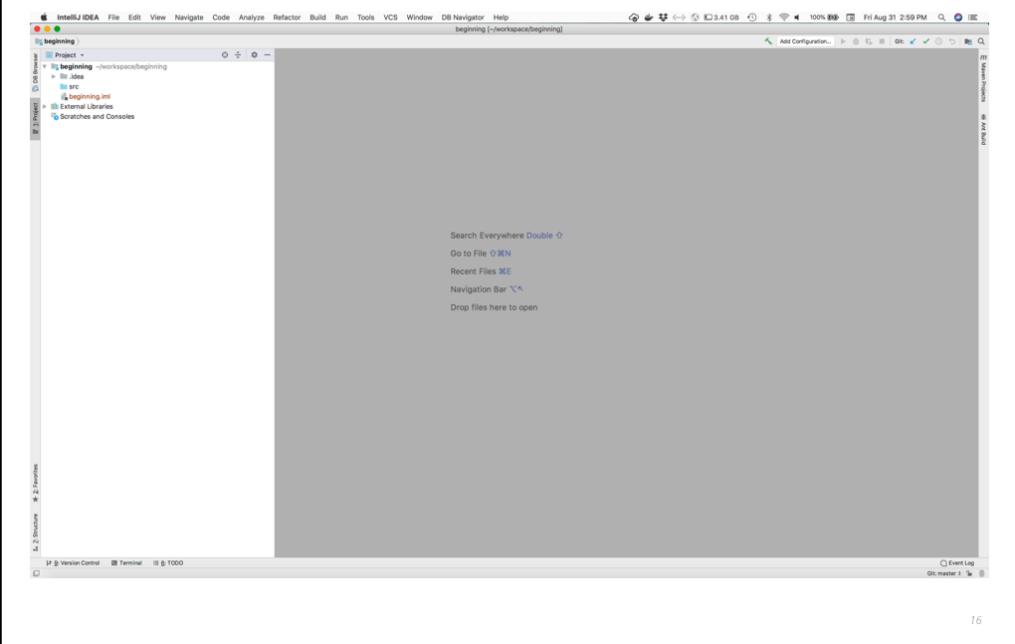
You'll need to know where you cloned the GitHub repository

In my case it was in ~/workspace/beginning

As I type this, it will fill in the Project name, the Module name, etc.

It should look just like mine except the name of the project might be different

LET'S SEE WHAT IT LOOKS LIKE IN THE IDE



After you click Finish button you should see this

On the right is the editor window.

There is nothing there because there is nothing to edit, yet

On the left is the project.

You will notice the beginning.iml file in red

This is the file IntelliJ created for the new project

It is not being backed up by git, so it is red

Around the edges of the window are tools

Like Terminal down at the bottom

ADDING SELENIUM LIBRARIES AND BINARIES

- What if I want to add a library, like Selenium
- First, I need a directory to put all the library files in
- The next picture shows how to do this

17

After you click Finish button you should see this

On the right is the editor window.

There is nothing there because there is nothing to edit, yet

On the left is the project.

You will notice the beginning.iml file in red

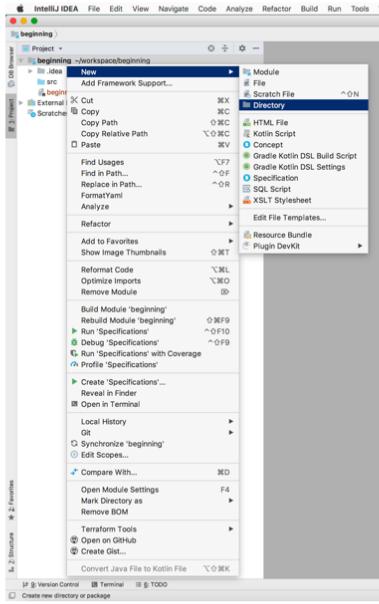
This is the file IntelliJ created for the new project

It is not being backed up by git, so it is red

Around the edges of the window are tools

Like Terminal down at the bottom

LET'S SEE WHAT IT LOOKS LIKE IN THE IDE



18

A lot of the features of a good IDE are available as a right click

Different menus will appear depending on where you right click

I want to add a directory to the project

So I right click on the project name (beginning) and a menu appears

Select New->Directory and add a directory

I'm going to call the directory ***libs***

There is a reason for this but really I could have called it anything

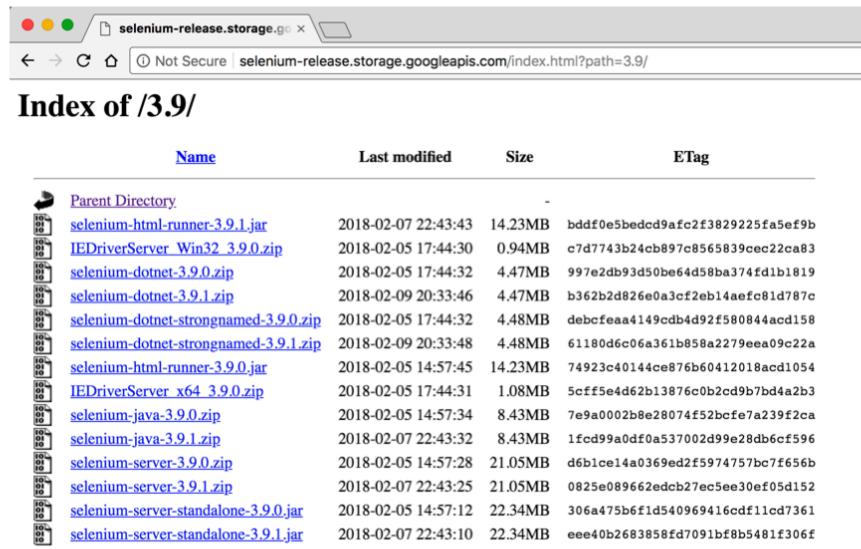
NOW I NEED FILES TO PUT IN THE NEW DIRECTORY

Name	Last modified	Size	ETag
2.39	.	.	.
2.40	.	.	.
2.41	.	.	.
2.42	.	.	.
2.43	.	.	.
2.44	.	.	.
2.45	.	.	.
2.46	.	.	.
2.47	.	.	.
2.48	.	.	.
2.49	.	.	.
2.50	.	.	.
2.51	.	.	.
2.52	.	.	.
2.53	.	.	.
3.0-beta1	.	.	.
3.0-beta2	.	.	.
3.0-beta3	.	.	.
3.0-beta4	.	.	.
3.0	.	.	.
3.1	.	.	.
3.1.1	.	.	.
3.1.2	.	.	.
3.1.3	.	.	.
3.1.4	.	.	.
3.2	.	.	.
3.3	.	.	.
3.4	.	.	.
3.5	.	.	.
3.6	.	.	.
3.7	.	.	.
3.8	.	.	.
3.9.0	.	.	.
3.9	.	.	.
selenium	.	.	.
index.html	2014-01-13 22:12:39	0.01MB	794b0ff41ead1b1428481b7ff3e759e0

19

If you go to the URL noted in the picture, you will find all the Selenium libraries
Let's get the latest set of libraries
This is the 3.9 directory

NOW I NEED FILES TO PUT IN THE NEW DIRECTORY



The screenshot shows a browser window with the address bar containing "selenium-release.storage.googleapis.com/index.html?path=3.9/". The page title is "Index of /3.9/". Below the title is a table listing files and their details:

Name	Last modified	Size	ETag
Parent Directory	-	-	-
selenium-html-runner-3.9.1.jar	2018-02-07 22:43:43	14.23MB	bddf0e5bedcd9afc2f3829225fa5ef9b
IEDriverServer_Win32_3.9.0.zip	2018-02-05 17:44:30	0.94MB	c7d7743b24cb897cb8565839cec22ca83
selenium-dotnet-3.9.0.zip	2018-02-05 17:44:32	4.47MB	997e2db93d50be64d58ba374fd1b1819
selenium-dotnet-3.9.1.zip	2018-02-09 20:33:46	4.47MB	b362b2d826e0a3cf2eb14ae1fc81d787c
selenium-dotnet-strongnamed-3.9.0.zip	2018-02-05 17:44:32	4.48MB	debcfeaa4149cd84d92f580844acd158
selenium-dotnet-strongnamed-3.9.1.zip	2018-02-09 20:33:48	4.48MB	61180d6c06a361b858a2279ea09c22a
selenium-html-runner-3.9.0.jar	2018-02-05 14:57:45	14.23MB	74923c40144ce876b60412018acd1054
IEDriverServer_x64_3.9.0.zip	2018-02-05 17:44:31	1.08MB	5cff5e4d62b13876c0b2cd9b7bd4a2b3
selenium-java-3.9.0.zip	2018-02-05 14:57:34	8.43MB	7e9a0002b8e28074f52bcfe7a239f2ca
selenium-java-3.9.1.zip	2018-02-07 22:43:32	8.43MB	1fcfd99a0df0a537002d99e28db6cf596
selenium-server-3.9.0.zip	2018-02-05 14:57:28	21.05MB	d6b1ce14a0369ed2f5974757bc7f656b
selenium-server-3.9.1.zip	2018-02-07 22:43:25	21.05MB	0825e089662edcb27ec5ee30e0f05d152
selenium-server-standalone-3.9.0.jar	2018-02-05 14:57:12	22.34MB	306a475b6f1d540969416cdf11cd7361
selenium-server-standalone-3.9.1.jar	2018-02-07 22:43:10	22.34MB	eee40b2683858fd7091bf8b5481f306f

20

When I click on 3.9 I see the following.

Having read up on Selenium, how it works, etc.

I know what all these files mean

For now just now that you only need some of them

Others are for C# or Internet Explorer

We'll need:

selenium-java-3.9.1.zip

selenium-server-standalone-3.9.1.jar

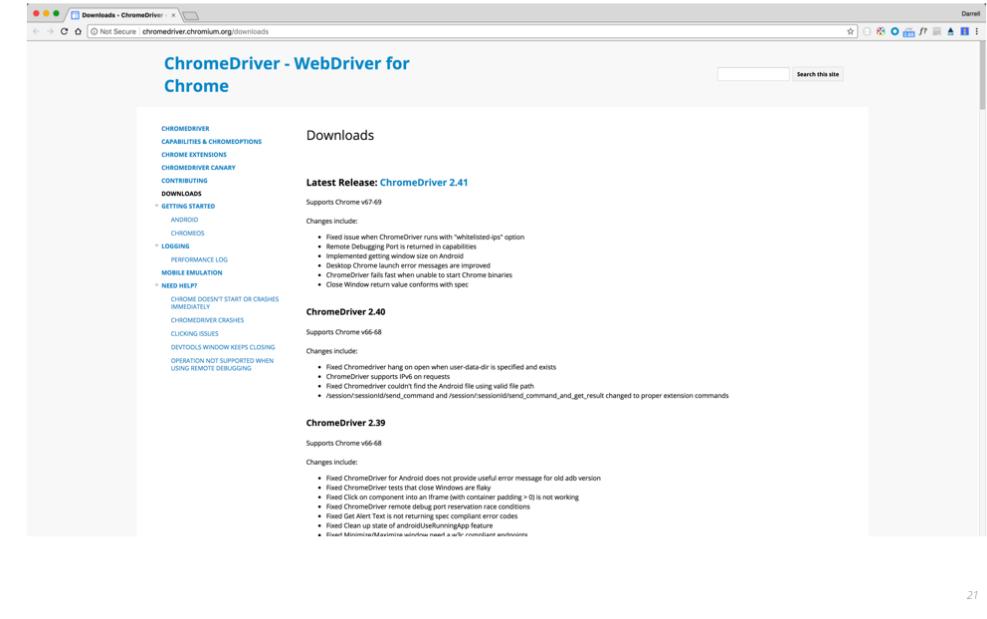
Download the server jar file into the libs directory your created

Download the zip file anywhere but you need to unzip it into the libs folder

You can either download the zip file, unzip it and copy all the files to the libs folder

Or you can use the command line and unzip to unzip the files

NOW I NEED FILES TO PUT IN THE NEW DIRECTORY



Selenium also requires a binary to allow the Selenium client to talk to the browser

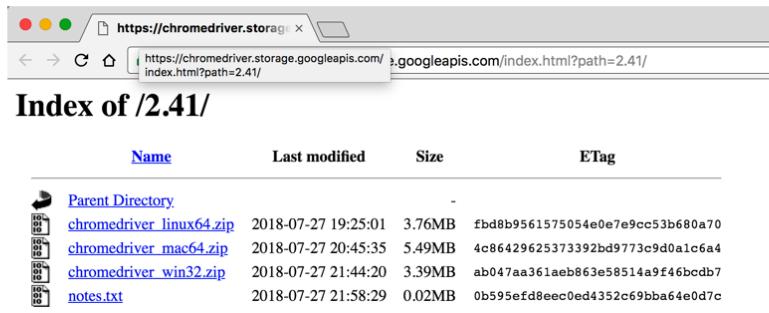
We are going to use Chrome to start

So we need to download the chromedriver

Go to the URL in the picture

If you click the “ChromeDriver 2.41” it will take you to the downloads for that version

NOW I NEED FILES TO PUT IN THE NEW DIRECTORY



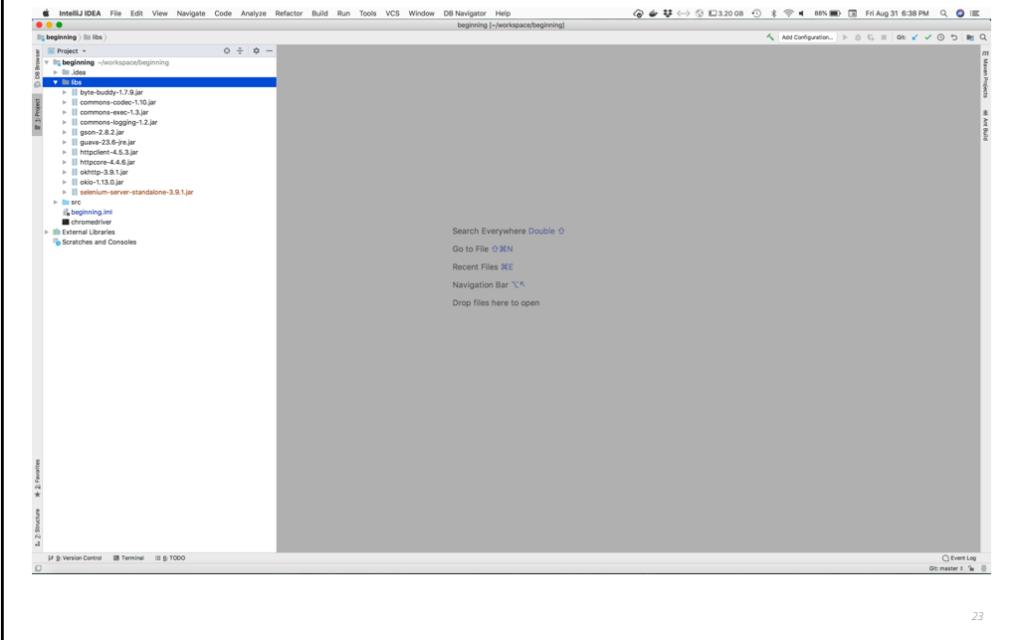
The screenshot shows a web browser window with the URL <https://chromedriver.storage.googleapis.com/>. The page title is "Index of /2.41". Below the title is a table listing files and their details:

Name	Last modified	Size	ETag
Parent Directory		-	
chromedriver_linux64.zip	2018-07-27 19:25:01	3.76MB	fbdb8b9561575054e0e7e9cc53b680a70
chromedriver_mac64.zip	2018-07-27 20:45:35	5.49MB	4cb6429625373392bd9773c9d0a1c6a4
chromedriver_win32.zip	2018-07-27 21:44:20	3.39MB	ab047aa361aeb863e58514a9f46bcd7c
notes.txt	2018-07-27 21:58:29	0.02MB	0b595efd8eec0ed4352c69bba64e0d7c

22

There are different versions of chromedriver depending on your operating system
I am using a macOS computer, so I download the chromedriver_mac64.zip file
I then want to unzip this file into the root of my project (not the libs folder)

WHEN EVERYTHING IS SETUP



Once everything is installed it should look like this:

There is a libs folder

In the libs folder are the Selenium libraries (jar files)

In the root of the project is the chromedriver binary

ADDING THE NEW FILES TO GIT

```
Terminal
+ inflating: libs/httpclient-4.5.3.jar
+ inflating: libs/httpcore-4.4.6.jar
+ inflating: libs(okhttp-3.9.1.jar
+ inflating: libs(okio-1.13.0.jar
phoebe2:beginning darrelgrainger$ rm selenium-java-3.9.1.zip
phoebe2:beginning darrelgrainger$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    beginning.jml
    chromedriver
    libs/

nothing added to commit but untracked files present (use "git add" to track)
```

24

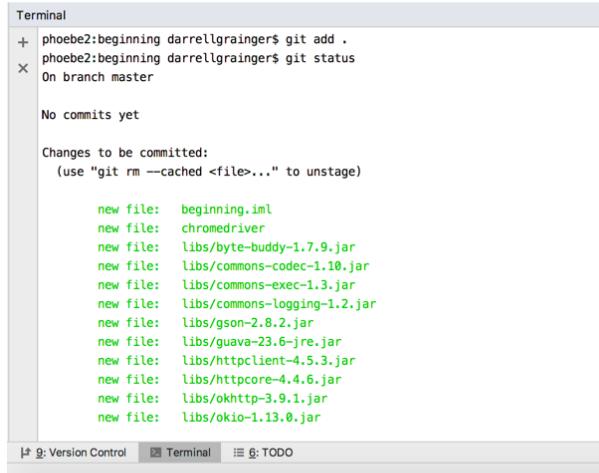
If we open the Terminal at the bottom of the IDE we can type shell commands here
To unzip the files the command is: (write this somewhere)

```
cd ~/workspace/beginning
unzip ~/Downloads/selenium-java-3.9.1.zip 'libs/*'
```

Next I can enter **git status** to see what is happening with my local git repository

I can see that there are changes to the project file, the chromedriver is new and the libs/ directory is new

ADDING THE NEW FILES TO GIT



```
Terminal
+ phoebe2:beginning darrellgrainger$ git add .
phoebe2:beginning darrellgrainger$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file: beginning.iml
    new file: chromedriver
    new file: libs/byte-buddy-1.7.9.jar
    new file: libs/commons-codec-1.10.jar
    new file: libs/commons-exec-1.3.jar
    new file: libs/commons-logging-1.2.jar
    new file: libs/gson-2.8.2.jar
    new file: libs/guava-23.6-jre.jar
    new file: libs/httpclient-4.5.3.jar
    new file: libs/httpcore-4.4.6.jar
    new file: libs/okhttp-3.9.1.jar
    new file: libs/okio-1.13.0.jar

25
```

If I type **git add .** it will stage all the untracked files to be added

This doesn't actually add any of the files to the repository

I can now see all the files which are going to be added to the repository, if I use **git commit**

ADDING THE NEW FILES TO GIT

```
Terminal
+ phoebe2:beginning darrellgrainger$ git commit -m "Add the Selenium 3 libraries to the project"
[master (root-commit) cc0bae2] Add the Selenium 3 libraries to the project
 12 files changed, 11 insertions(+)
create mode 100644 beginning.iml
create mode 100755 chromedriver
create mode 100644 libs/byte-buddy-1.7.9.jar
create mode 100644 libs/commons-codec-1.10.jar
create mode 100644 libs/commons-exec-1.3.jar
create mode 100644 libs/commons-logging-1.2.jar
create mode 100644 libs/gson-2.8.2.jar
create mode 100644 libs/guava-23.6-jre.jar
create mode 100644 libs/httpclient-4.5.3.jar
create mode 100644 libs/httpcore-4.4.6.jar
create mode 100644 libs/okhttp-3.9.1.jar
create mode 100644 libs/okio-1.13.0.jar
phoebe2:beginning darrellgrainger$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
phoebe2:beginning darrellgrainger$
```

26

Now I issue a **git commit** command

This actually adds all the files to a local repository

This is a repository only on my machine

The files have not be copied to GitHub

ADDING THE NEW FILES TO GIT

```
Terminal
+ phoebe2:beginning darrellgrainger$ git status
On branch master
  Your branch is based on 'origin/master', but the upstream is gone.
    (use "git branch --unset-upstream" to fixup)

    nothing to commit, working tree clean
phoebe2:beginning darrellgrainger$ git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (15/15), 12.14 MiB | 665.00 KiB/s, done.
Total 15 (delta 0), reused 0 (delta 0)
To github.com:dgrainger/beginning.git
 * [new branch]      master -> master
phoebe2:beginning darrellgrainger$
```

27

To copy the files from my local machine to GitHub I issue **git push**

There are parameters I can use with push but if I use all the defaults, git will figure things out

ADDING THE NEW FILES TO GIT

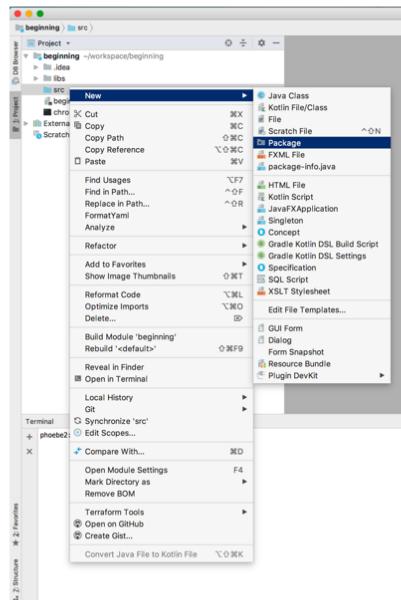
The screenshot shows a GitHub repository page for 'dgrainger / beginning'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The latest commit was made by 'dgrainger' 5 minutes ago. The commits are:

- libs: Add the Selenium 3 libraries to the project
- beginning.iml: Add the Selenium 3 libraries to the project
- chromedriver: Add the Selenium 3 libraries to the project

A message at the bottom encourages adding a README, with a 'Add a README' button.

If I go to my empty repository on <https://www.github.com> I see the following
All the changes I made locally are now copied to GitHub as well
Now I have two backups as well as the original on my machine

JAVA CONVENTIONS



29

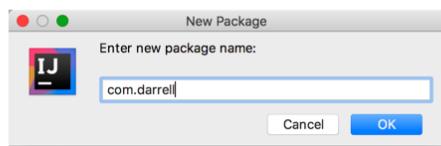
Let's create our first test automation.

First, a Java convention is to organize classes by package

For example, if I want the file ~/workspace/beginning/com/darrell/Example1.java
then I want the package "com.darrell" and the class "Example1"

So this picture shows how to create a package

JAVA CONVENTIONS



30

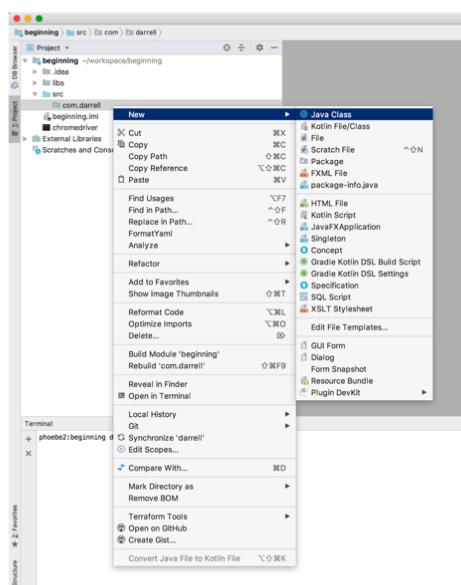
If you look at the Selenium libraries they are often in org.openqa or com.thoughtworks

I like to use something with my name in it

So com.darrell, org.darrell, com.dgrainger, etc.

This is just a convention and won't actually make any difference for running the test code

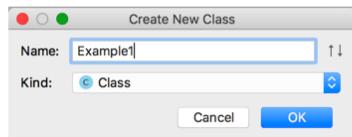
JAVA CONVENTIONS



31

Next I want to create a class to hold my automation code
So I right click on the package name
then select New->Java Class

JAVA CONVENTIONS



32

I'm going to give my class the name Example1

If I was creating automation for say Google I might use the package com.google
and the class might be BasicSearch for testing basic search features

INTELLIJ HELPS



33

After you create the class IntelliJ will ask you if you want to add Example1.java to Git
This is an example of how IntelliJ helps you.

I selected Yes so I didn't have to do this later

You could also click Remember, don't ask again but there might be some files you
don't want added to Git

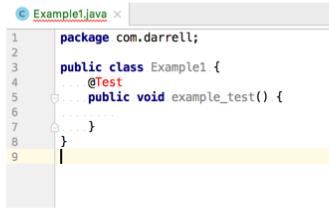
INTELLIJ HELPS



```
Example1.java
package com.darrell;
public class Example1 {
```

- If we look at the file Example1.java we see the above text
- IntelliJ filled in things like the package and class information
- We don't want to change any of this
- We just want to add to it

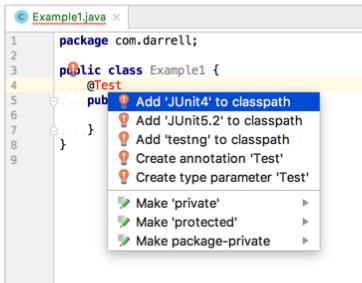
MORE INTELLIJ HELP



```
Example1.java
1 package com.darrell;
2
3 public class Example1 {
4     ...
5     @test
6     public void example_test() {
7         ...
8     }
9 }
```

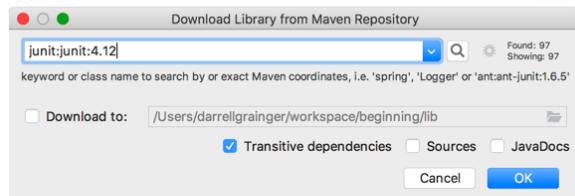
- I want to use the Junit test framework
- I know it has things like `@Test` so I type that
- This is called an annotation because it starts with an @ symbol
- But IntelliJ shows this as red
- I can move the cursor to the word `@Test`
- Then press *alt-enter*

INTELLIJ MAKES FIX SUGGESTIONS



- A context menu will pop up and suggest how to fix the error
- I want to use JUnit 4, so I take the first suggestion
- When Java is compiling the code, it will search the classpath for all the libraries
- By adding JUnit 4 to the classpath it helps the Java compiler

INTELLIJ KNOWS TO DOWNLOAD LIBRARIES FOR US



- When you tell IntelliJ to add Junit to the classpath, it is smart enough to realize you need the actual library too
- So it asks you to download it
- You can download it to the standard, global location
- Or we can click the **Download to:** option and download it to our **libs** directory (not the lib directory IntelliJ suggests)

37

GIT STATUS

```
Terminal
+ phoebe2:beginning darrellgrainger$ git status
X On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   src/com/darrell/Example1.java

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:  beginning.iml
    modified:  src/com/darrell/Example1.java

phoebe2:beginning darrellgrainger$
```

38

Let's go to the Terminal and check the git status

Here we see that git recognizes we changed the Example1.java file

And we made changes to the project (classpath, added junit library)

COMMIT THE CHANGES TO GIT

```
phoebe2:beginning darrellgraingers$ git add .
phoebe2:beginning darrellgraingers$ git commit -m "Start first test case"
[master 2f34e58] Start first test case
 2 files changed, 20 insertions(+)
 create mode 100644 src/com/darrell/Example1.java
phoebe2:beginning darrellgraingers$
```

- Again, we want to stage the changes to be committed
- So we *git add .* to add all the changes
 - It should be noted that sometimes we might only commit specific changes
 - I'll talk about that shortly
- Then we want to commit the changes locally
- Notice the *-m* option
- We usually want to give a message about what we are committing

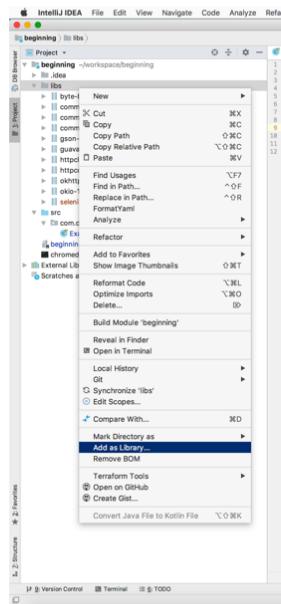
PUSH THE CHANGES TO GITHUB

```
Terminal
+ phoebe2:beginning darrellgrainger$ git status
On branch master
X Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
phoebe2:beginning darrellgrainger$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 983 bytes | 983.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To github.com:dgranger/beginning.git
  cc0bae2..2f3de58  master -> master
phoebe2:beginning darrellgrainger$
```

- Finally, we want to push the changes to GitHub
- Notice that **git status** tells us our local copy is ahead of GitHub
- So there is nothing to *commit* but there is something to *push*

ADDING SELENIUM LIBRARIES TO THE CLASSPATH



41

Just because the Selenium libraries are in the projects libs/ folder, the Java compiler won't be able to find them

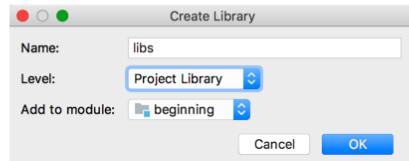
The compiler will only look in the folders you tell it to

So we need to tell the compiler to look in the libs/ folder

Right click on the libs/ folder

Select "Add as Library..."

ADDING SELENIUM LIBRARIES TO THE CLASSPATH



- The name *libs* will automatically be filed in because we right clicked on the *libs* folder
- We want the entire project to be able to use Selenium
- So we add it to the project level and we only have one module
- So it automatically goes to the *beginning* module

COMMIT AND PUSHING TO LOCAL AND GITHUB

```
Terminal
+ phoebe2:beginning darrellgrainger$ git status
X On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   beginning.iml

no changes added to commit (use "git add" and/or "git commit -a")
phoebe2:beginning darrellgrainger$ git add .
phoebe2:beginning darrellgrainger$ git commit -m "Adding changes to project"
[master b4c641a] Adding changes to project
 1 file changed, 1 insertion(+)
phoebe2:beginning darrellgrainger$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 353 bytes | 353.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:dgrainger/beginning.git
    7e663f6..b4c641a  master -> master
phoebe2:beginning darrellgrainger$
```

43

Again, as we done a couple of times,

We go to a terminal and see the git status has changed

The sequence is;

```
git status
git add .
git commit -m "some helpful message"
git push
```

TODO

- Code completion - *WebDriver* (picture 35)
- Automatically adding *import* statements
- More code completion - *ChromeDriver*
- Simple example
- Running the simple example
- Adding a .gitignore
- Using the JUnit framework @Before and @After annotations
- Using *assertEquals* statement for a pass/fail result
- Seeing it fail then fixing the test
- Adding a second test

44

CHECKLIST

- A good book on Git: <https://git-scm.com/book/en/v2>
- A cheatsheet and docs for Git: <https://git-scm.com/docs>
- A good Java IDE: <https://www.jetbrains.com/idea/>
- Java documentation:
 - <http://www.oracle.com/technetwork/java/javase/documentation/index.html>
- Java Development Kit (JDK):
 - Go to documentation then select Downloads tab
 - Selenium currently works with JDK SE 8.0

45

CHECKLIST

- sh (Bourne shell) or bash (Bourne Again Shell)
- macOS
 - This is the built in shell or Terminal.app
- Linux
 - This is the build in shell
- Windows
 - Git-scm: <https://git-scm.com/download/win>
 - Cygwin: <http://www.cygwin.com/install.html>