**GitHub**

**Overview**

GitHub is a web-based hosting service for version control using git. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features [1].

**Installation**

To install git on your machine follow the instructions from <https://git-scm.com/downloads>. Once it’s installed, run the command *git* in the terminal, and you should receive instructions on how to use the commands.

**Important commands**

*git add file.txt*

*git commit –m “commit message”*

*git push*

*git pull*

**Composer**

**Overview**

Composer is an application-level package manager for the PHP programming language that provides a standard format for managing dependencies of PHP software and required libraries [2].

**Installation**

Composer can be installed following the guide in <https://getcomposer.org/doc/00-intro.md>; make sure it’s a global installation. Once it’s installed, run *composer* in the terminal, and you should receive the composer screen. In case of an error, make sure you follow the steps correctly with admin privileges (e.g. using sudo on a Unix machine).

**Important composer commands**

*composer require package/library*

* Will update composer.json with the new dependency or create a new composer.json and composer.lock if they don’t exist

*composer install*

* Installs the vendor packages according to composer.lock (or creates composer.lock if not present)

*composer update*

* Will regenerate composer.lock with the new composer.json dependencies and versions, no matter if composer.lock exists or not

*composer dump-autoload -o*

* Regenerates the list of all classes that need to be included in the project (vendor/composer/autoload\_classmap.php).

**Example composer.json file**



*Explanation of composer.json:*

**require**: {…} lists the dependencies required for the project (propel, slim, etc...)

**autoload**: {…} lists extras to be included when *composer dump-autoload –o* is ran. classmap allows directories to be included in the autoload process. The psr-4 autoload is used to define the mapping from namespaces to directories. The example filename filename would be app/hello.php containing an App\Hello class.

**Composer notes**

Composer will download the dependencies into a vendor/ folder, which it will create if it doesn’t exist. Every dependency will have a folder, for example, propel will be inside vendor/propel. *composer dump-autoload –o* will **not** work in the previously shown composer.json example if data/models and app/ folders don’t exist. If you want to test it, simple remove everything but require: {…}.

**Works Cited**

[1] Wikipedia. March 23, 2018. Github. <https://en.wikipedia.org/wiki/Composer_(software)>

[2] Wikipedia. March 24, 2018. Composer. <https://en.wikipedia.org/wiki/Composer_(software)>