SP2013Framework Documentation

Integrating Modern JavaScript tools into SharePoint 2013

# Installation Process

### Step 1: Install Global Dependencies

Make sure the following open source programs are installed on your local machine globally before starting the installation process. Please refer to their documentation for instructions on how to install them. Note that you must install NodeJS before anything else.

|  |  |
| --- | --- |
| **Program** | **URL** |
| NodeJs | <https://nodejs.org/en/download/> |
| Gulp | <https://github.com/gulpjs/gulp/blob/master/docs/getting-started.md> |
| Yeoman | <http://yeoman.io/learning/index.html> |
| Typescript | <https://www.typescriptlang.org/index.html#download-links> |
| Bower | <https://bower.io/> |

### Step 2: Register Yeoman Generator

Since this project is not in the npm public registry, you must manually install the generator’s dependencies and link the project to npm.

1. Navigate to the generator’s root directory and run “npm install”



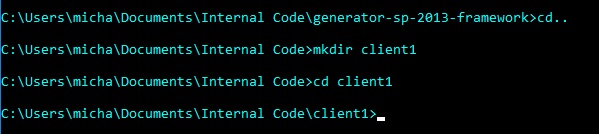
1. Once finished, run “npm link” to register the project’s name to npm’s global symlink.



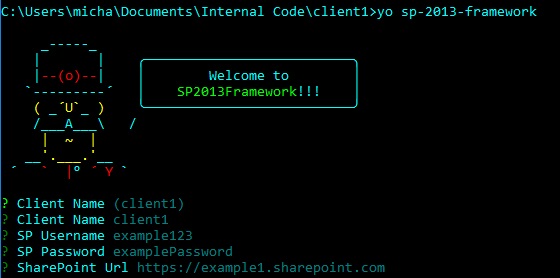
### Step 3: Using the Yeoman Generator

Now that the generator is linked to npm, you will be able to create a new project using Yeoman from the command line in any directory.

1. Create a new directory using the client’s name.

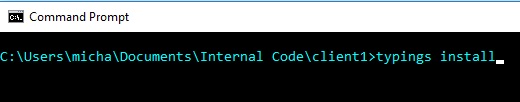


1. You will now be able to run the Yeoman generator which will ask you for your account credentials. These credentials must be correct for the framework to interact with the SharePoint site otherwise you will get a nasty error message in Step 4.



* 1. If you are installing this setup on a sub site or site collection, the full url will be required in the SharePoint Url parameter.
  2. This current version will not encrypt passwords (see the bug section).
  3. Once you are done, Yeoman will automatically install all node and bower dependencies.

1. Since Yeoman won’t automatically install typescript’s “definitely typed” files, you will have to install them yourself by running “typings install”



### Step 4: Install the framework on the SharePoint site

To make sure everything was installed correctly, it is time to test the link from the framework to your SharePoint site. Run the command “gulp push:sharepoint” to add all theme files to the SharePoint site.C:\Users\micha\AppData\Local\Microsoft\Windows\INetCacheContent.Word\push_sharepoint.jpg

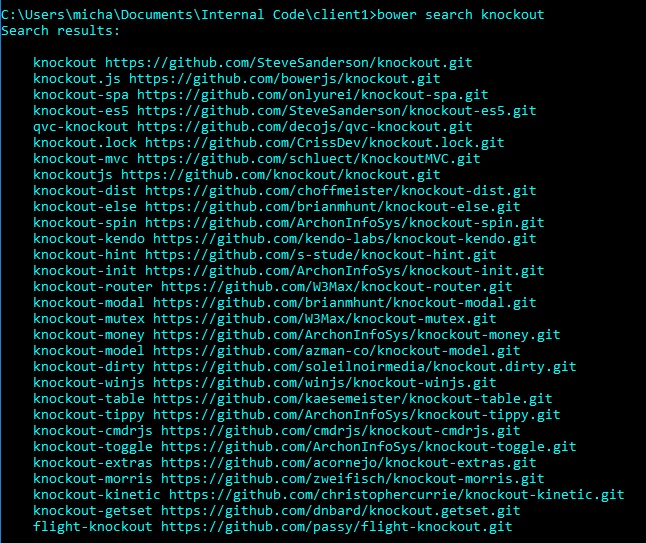
All files will now be pushed to the SharePoint site if everything was installed correctly. In this case, you will find a new masterpage named “custom.html” inside the “client1” folder in the sharepoint masterpage directory. To view a full list of gulp commands, see the “gulpfile.js” on the root of your project.

# Package Manager

SP2013Framework uses bower for managing its dependencies. Each library will also require its own definitely type file so typescript can register their properties in the function calls. To keep track of the dependencies, it is recommended to first use bower to install the library and then use typescript to get the library’s definitely typed files. To demonstrate this, we are going to install the knockout.js framework to our example client1’s sharepoint site.

### Step 1: Install Knockout using bower

To check if bower has a library in its registry, you can run something like “bower search knockout” to pull up a list of projects bower can install.



For now, we will only need to install the first one. Run “bower install knockout –save” to install knockout and save its url to your bower.json file. It is important to run this command so our framework can keep track of dependencies and will make it easier if you need to move the project to another machine.

