# Setting up for UTM Web API Development on Windows

## Install Helpful Applications

### Tortoise Git

We will be using Git, so to make it most usable install Tortoise Git from here: <https://tortoisegit.org/>.

### Sublime Text Editor

While this isn’t necessarily a requirement, you can use your own text editor, this development setup document assumes that you will be using Sublime. Install it from here: <http://www.sublimetext.com>.

Once it is installed, go to <https://packagecontrol.io/installation> and copy the whole text inside the “sublime text 3” box. Then run Sublime Edit, click **view** from the menu and select **show console**. Paste what you have copied inside the console in the bottom and press <enter>, then re-start Sublime Edit. Press <ctrl-shift-P> and type:

install package

…and press <enter>. Then type:

gosublime

…and press <enter>.

## Install Required Applications

### Install Go

Install **go** for Windows from <https://golang.org/doc/install>.

Make sure that the following environment variables have been set up:

* **$GOROOT**, point to the go installation directory, something like **c:\Go**.
* **$GOPATH**, pointing to where you will store your code, something like **c:\projects\gocode**. Make sure that this directory exists.

For convenience, you may also wish to add **c:\projects\gocode\bin** to your path.

### Install GCC

**Go** builds under GCC. You need a good copy of GCC installed and it must support 64 bit operations. You could try Cygwin64 but, as well as GCC, the Cygwin installation requires the mingw32 and mingwex libraries and I couldn’t make GCC find them correctly. Since some MinGW bits seem to be required in any case, I next tried installing MinGw64 but **Go** still complained that it didn’t have 32 bit support. The smallest/simplest/neatest solution turned out to be to install <http://tdm-gcc.tdragon.net/> (with the on-demand installer). This is a 64-bit GCC for Windows that plays well out of the box.

### Install Mongo

From the Mongo website.

### Install NPM

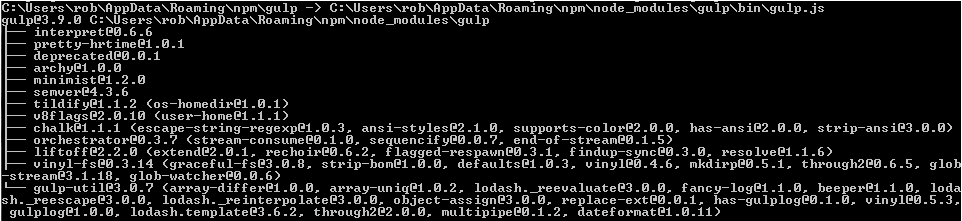
Install **NPM** from <https://nodejs.org>.

### Install Gulp

With NPM installed, from a command prompt, install **Gulp** by typing:

**npm install --global gulp**

…and pressing <enter>. You will see something like:



### Install Gin

At a command prompt, install **Gin** by typing:

**go get github.com/codegangsta/gin**

…and pressing <enter>.

## Clone And Build The Repository

At a command prompt, type the following:

**go get github.com/u-blox/utm**

…and press <enter>. This will clone the code repository, **github.com/u-blox/utm**, onto your computer under the **c:\projects\gocode\src directory**, plus any dependent packages under the **c:\projects\gocode\pkg** directory, and build it, putting the binaries into a **c:\projects\gocode\bin** directory. If it succeeds it will say nothing. If it fails it will tell you stuff. You need to resolve any errors before continuing.

You should now be able to open the project files in Eclipse (Mars or later).

## Install The Built Packages

CD to the **gocode/src/github.com/u-blox/utm/static** folder. Type:

npm install

…and press <enter>. When it has completed you should now have a **node\_modules** folder off this directory. It may say something like:

npm WARN EPACKAGEJSON [UTM-WEB@1.0.0](mailto:UTM-WEB@1.0.0). No description  
npm WARN EPACKAGEJSON [UTM-WEB@1.0.0](mailto:UTM-WEB@1.0.0). No repository field

This is not an issue.

## Run Gulp

Open a **new** command window. CD to the **gocode/src/github.com/u-blox/utm/static** folder. Type:

npm install

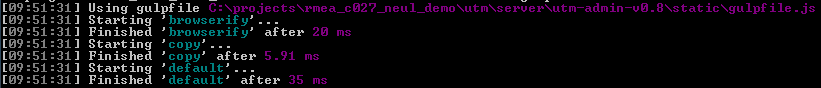
…and press <enter>. When it has completed, you may see something like:

npm WARN EPACKAGEJSON [UTM-WEB@1.0.0](mailto:UTM-WEB@1.0.0). No description  
npm WARN EPACKAGEJSON [UTM-WEB@1.0.0](mailto:UTM-WEB@1.0.0). No repository field

This is not an issue. Type:

gulp

…and press <enter>. You will see something like:



…and the command will not terminate. Gulp is now running. Always have this command window running while you develop, it looks for client-side changes you make in the **static** folder and automagically produces the required new Java Script output in the **dist** folder.

You should now have a **dist** folder off this directory.

## Run Mongo

Open a new command window. CD to the **gocode/src/github.com/u-blox/utm** folder. Type:

mongod

…and press <enter>.

## Run Gin

Open a **new** command window. CD to the **gocode/src/github.com/u-blox/utm** folder. Type:

gin

…and press <enter>. You will see any debug output from the running server in this window.

## Look At The Web Interface

Open a browser and navigate to:

<http://localhost:8080>

You should see a prompt to enter a username and password (which must be present in the mongo users database). If not, something has gone wrong.

Rob