Setting up Haskell IDE for Atom

September 3, 2016

1 Note to Windows/Mac/DICE(Linux) Users

The steps of configuration of Windows is similar to those of Mac/DICE(Linux). All you need is to install some required plugins of Atom and binary dependencies of Haskell. The aim is to set up the atom for a useful Haskell IDE by **ide-haskell** plugin [1].

2 Plugins of Atom

First, following five plugins should be installed, as well as **ide-haskell**. Although you may not need some of them, it is strongly suggested to install them all.

- language-haskell - Required

- ide-haskell-cabal - Use this plugin to build and launch Haskell projects

- haskell-ghc-mod - Use this plugin to type-check your code on save and add type highlights on mouseover

- autocomplete-haskell [2] - Use this plugin to autocomplete code

- ide-haskell-repl [3] - It provides a way to interact with ghci from Atom

The steps of installation are really easy. Once the Atom has been isntalled, you can simply type the following command in **terminal** of Mac/DICE(Linux) or **cmd** of Windows:

```
$ apm install language−haskell haskell−ghc−mod ide−haskell−cabal
ide−haskell autocomplete−haskell ide−haskell−repl
```

Listing 1: Commands of installation

On the other hand, you can also install them through GUI of Atom. Click the menu: Packages − > Settings View − > Install Packages/Themes of Atom. Then you may find the installation interface with a search bar at figure 1. Input the name of the packages as above and install them one by one (the order does not matter).
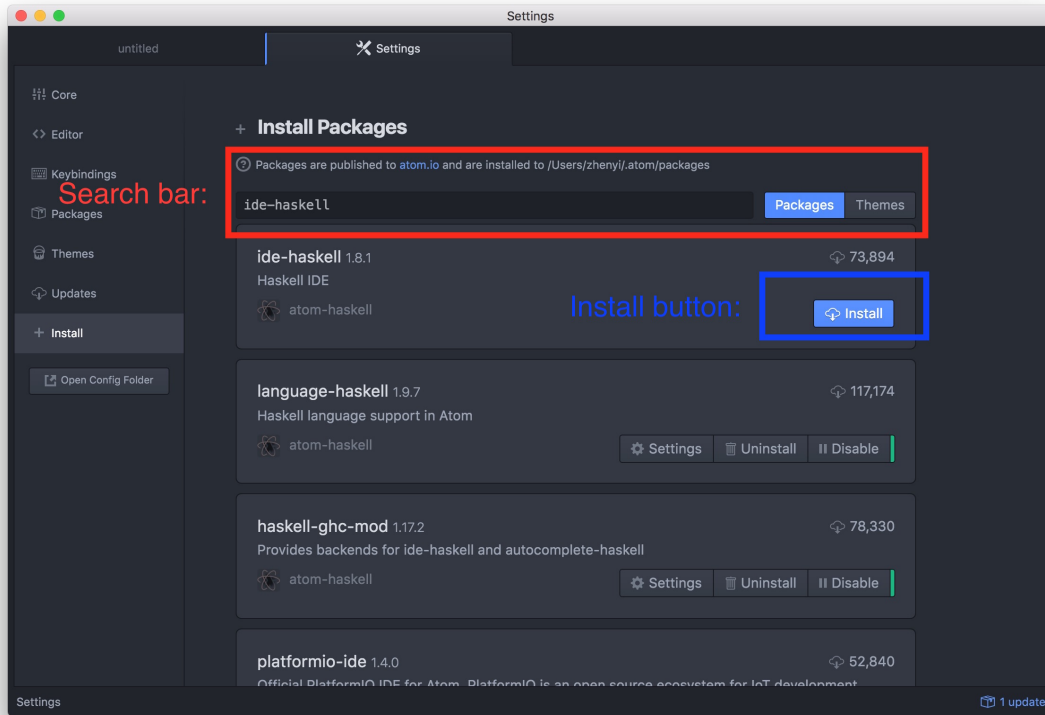
Figure 1: The GUI of installation

3   Dependencies

To make the plugins work, the fundamental abilities and necessary interfaces provided by the following binary dependencies are needed.

- ghc-mod - It could be seen the backend of the **haskell-ide** plugin

- hlint - It could suggest possible improvements to Haskell code

- stylish-haskell - Required by **haskell-ide** plugin

To install them, you should make sure that the haskell-platform has been installed successfully. Then, opening the **terminal** in Mac/DICE(Linux) or **cmd** in Windows and running the following commands.

```
$ cabal update
$ cabal install ghc-mod hlint stylish-haskell
```

Listing 2: Commands of installation

If the version of your Haskell is 8.0.1 or later, you may try the following command:

```
$ cabal update
$ cabal install alex
$ cabal install happy
$ cabal install ghc−mod hlint stylish−haskell
```

Listing 3: Commands of installation

4   Setting Up Atom

Finally, the packages are configurable via: Packages − > Settings View − > Manage Packages, then finding the certain package and click the setting button.
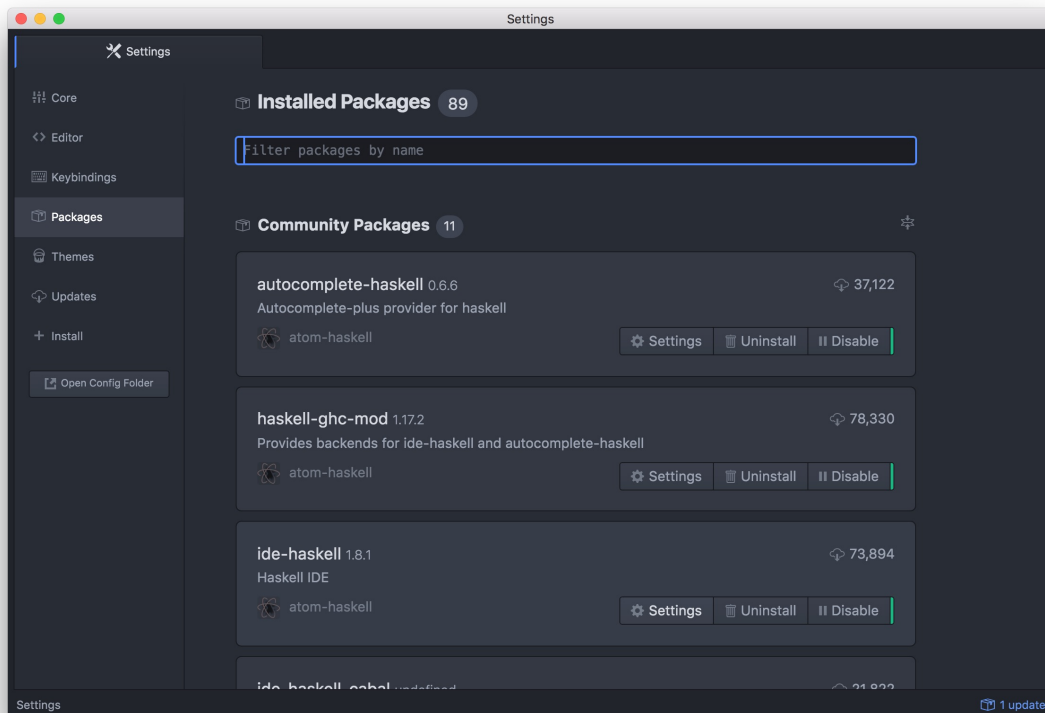

Figure 2: Setting up Atom

The two main plugins should be configured before using. In ide-haskell settings, set 'Stylish Haskell Path' to:

- DICE(Linux): $HOME/.cabal/bin/stylish-haskell

- Mac: $HOME/Library/Haskell/bin/stylish-haskell

- Windows: the path of packages installed

In haskell-ghc-mod settings, set 'Ghc Mod Path' to:

- DICE(Linux): $HOME/.cabal/bin/ghc-mod

- Mac: $HOME/Library/Haskell/bin/ghc-mod

- Windows: the path of packages installed

Make sure that replacing the $HOME environment variable to absolute directory, for example: "/home/username" in Linux. The full path may look like: "home/username/.cabal/bin/stylish-haskell".

Using the command: "echo $HOME" in DICE(Linux)/Mac to obtain the value of $HOME. For Windows, there is no environment variable of $HOME. Hope you could figure it out.

5   How to Use

Like most of the text editor, opeing a haskell file at first. then you may find the "Haskell IDE" appears after "Help" in menu bar. Just editing your haskell file in the text area. For running the haskell program, click "Open REPL" in "Haskell IDE" menu, and Haskell IDE will launch GHCi in a buffer of Atom (figure 3). In general, you should choose 'none' or 'cabal', if you do not use 'stack' to build where 'cabal' is the default choice for Haskell platform(figure 4).

Input any expression you want to evaluate and then click "shift + enter" in Windows/DICE(Linux) or "cmd + enter" in Mac (figure 5).

References

[1] Nikolay Yakimov Alexander Chaika. *IDE-Haskell*. URL: https://atom.io/packages/ide-haskell.

[2] Nikolay Yakimov. *Autocomplete-haskell*. URL: https://atom.io/packages/autocomplete-haskell.

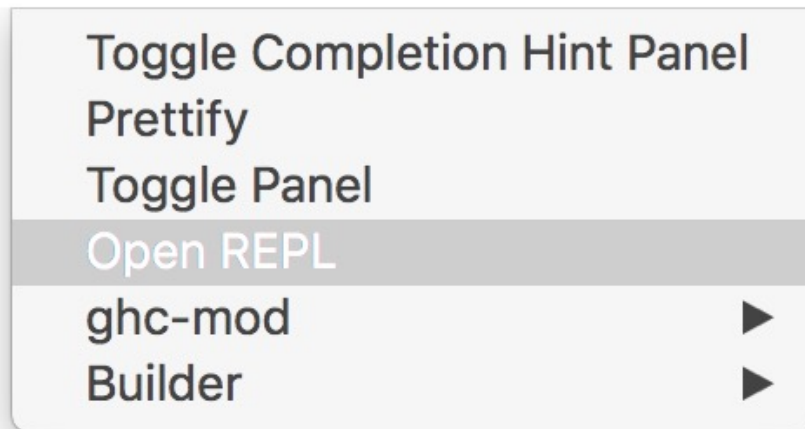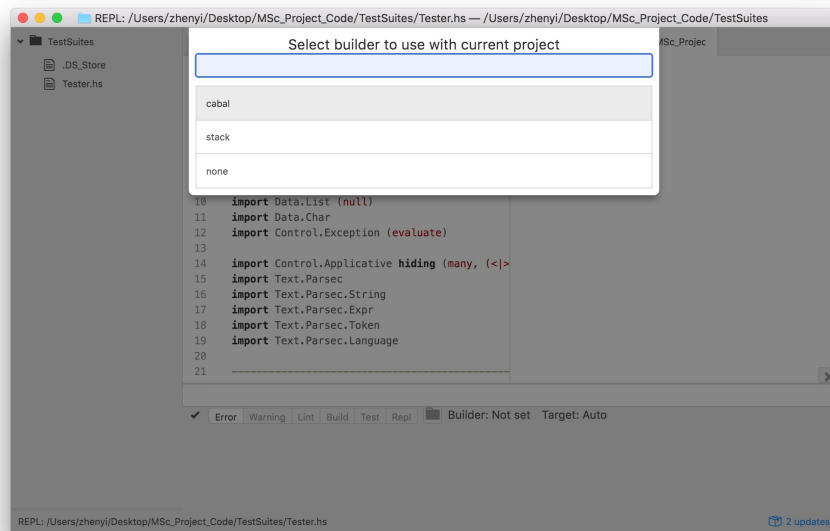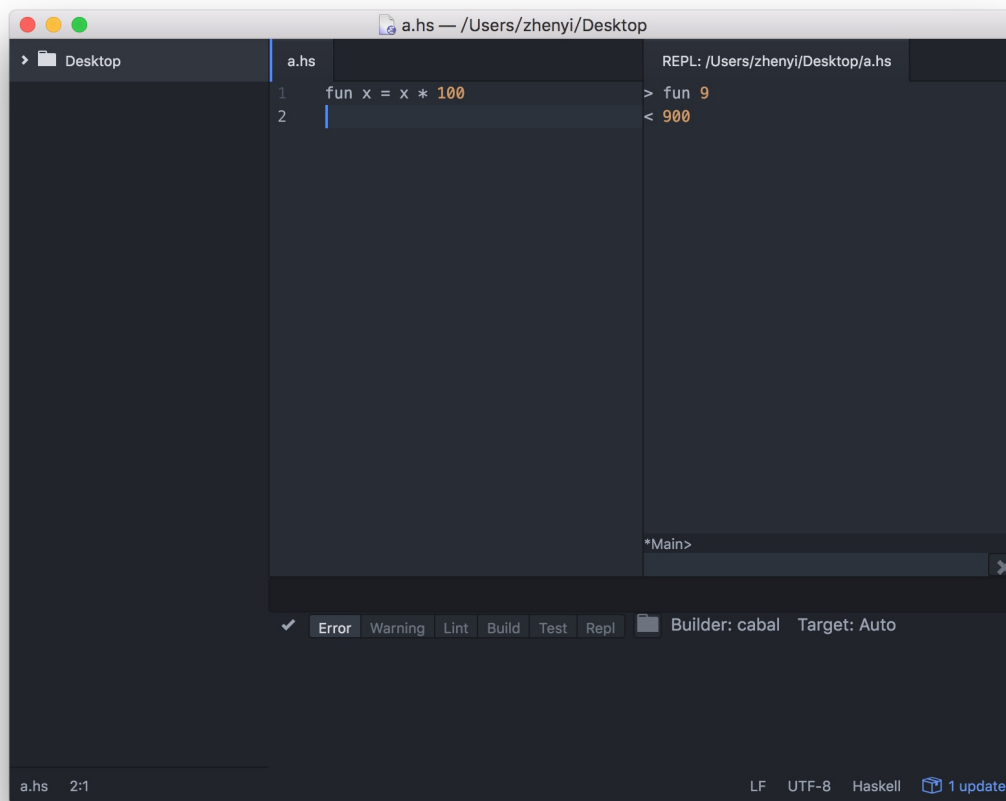[3] Nikolay Yakimov. *IDE-Haskell-repl*. URL: https://atom.io/packages/ide-haskell-repl.

Figure 3: Menu of Haskell IDE

Figure 4: Dialog of repl

Figure 5: Running commands