<https://github.com/nsf/gocode>

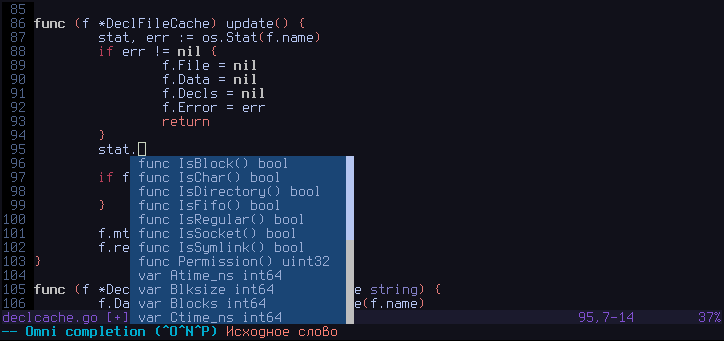
1. **An autocompletion daemon for the Go programming language**

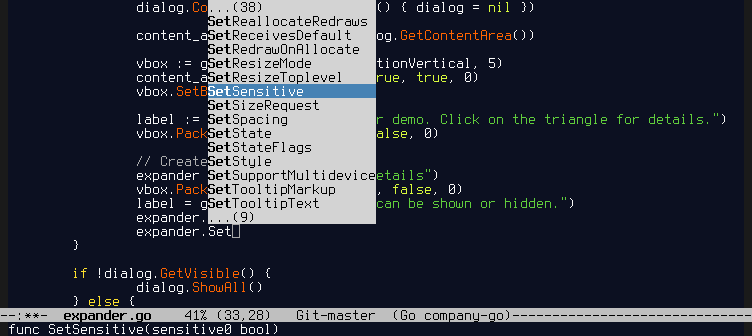
Gocode is a helper tool which is intended to be integrated with your source code editor, like vim and emacs. It provides several advanced capabilities, which currently includes:

* Context-sensitive autocompletion

It is called *daemon*, because it uses client/server architecture for caching purposes. In particular, it makes autocompletions very fast. Typical autocompletion time with warm cache is 30ms, which is barely noticeable.

Also watch the [demo screencast](http://nosmileface.ru/images/gocode-demo.swf).

[](https://github-camo.global.ssl.fastly.net/3b2eb4c4dc72b975032f536a93260dfe7353c23f/687474703a2f2f6e6f736d696c65666163652e72752f696d616765732f676f636f64652d73637265656e73686f742e706e67)

[](https://github-camo.global.ssl.fastly.net/dd80f0004b50ac03d07dba087ed4e20e54b34e3b/687474703a2f2f6e6f736d696c65666163652e72752f696d616765732f656d6163732d676f636f64652e706e67)

1. **Setup**
2. You should have a correctly installed Go compiler environment and your personal workspace ($GOPATH). If you have no idea what **$GOPATH** is, take a look [here](http://golang.org/doc/code.html). Please make sure that your **$GOPATH/bin** is available in your **$PATH**. This is important, because most editors assume that **gocode** binary is available in one of the directories, specified by your **$PATH** environment variable. Otherwise manually copy the**gocode** binary from **$GOPATH/bin** to a location which is part of your **$PATH** after getting it in step 2.

Do these steps only if you understand why you need to do them:

export GOPATH=$HOME/goprojects

export PATH=$PATH:$GOPATH/bin

1. Then you need to get the appropriate version of the gocode, for 6g/8g/5g compiler you can do this:

go get -u github.com/nsf/gocode (-u flag for "update")

Windows users should consider doing this instead:

go get -u -ldflags -H=windowsgui github.com/nsf/gocode

That way on the Windows OS gocode will be built as a GUI application and doing so solves hanging window issues with some of the editors.

1. Next steps are editor specific. See below.
2. **Vim setup**

In order to install vim scripts, you need to fulfill the following steps:

1. Install official Go vim scripts from **$GOROOT/misc/vim**. If you did that already, proceed to the step 2.
2. Install gocode vim scripts. Usually it's enough to do the following:

2.1. vim/update.sh

**update.sh** script does the following:

#!/bin/sh

mkdir -p "$HOME/.vim/autoload"

mkdir -p "$HOME/.vim/ftplugin/go"

cp "${0%/\*}/autoload/gocomplete.vim" "$HOME/.vim/autoload"

cp "${0%/\*}/ftplugin/go/gocomplete.vim" "$HOME/.vim/ftplugin/go"

2.2. Alternatively, you can create symlinks using symlink.sh script in order to avoid running update.sh after every gocode update.

**symlink.sh** script does the following:

#!/bin/sh

cd "${0%/\*}"

ROOTDIR=`pwd`

mkdir -p "$HOME/.vim/autoload"

mkdir -p "$HOME/.vim/ftplugin/go"

ln -s "$ROOTDIR/autoload/gocomplete.vim" "$HOME/.vim/autoload/"

ln -s "$ROOTDIR/ftplugin/go/gocomplete.vim" "$HOME/.vim/ftplugin/go/"

1. Make sure vim has filetype plugin enabled. Simply add that to your **.vimrc**:

filetype plugin on

1. Autocompletion should work now. Use <C-x><C-o> for autocompletion (omnifunc autocompletion).

Alternatively take a look at the vundle/pathogen friendly repo: <https://github.com/Blackrush/vim-gocode>.