

Sign up



cosmtrek / air Public

Live reload for Go apps

GPL-3.0 license

9k stars 595 forks

☆ Star

🔔 Notifications

<> Code Issues 74 Pull requests 20 Discussions Actions Projects Security

🔑 master



cosmtrek docs: update README ...

Jan 14, 2023 ⌚ 303

[View code](#)

☰ README.md

☁ Air - Live reload for Go apps

🔗 release passing 🔧 code quality B 📊 go report A+ 📈 codecov 70%

```
~/repos/go/src/github.com/cosmtrek/airtest main* 43s
> air

  /_/\  | | | | |
 /_/--\ | | | | | v1.41.0, built with Go 1.19.2

watching .
!exclude tmp
building...
running...
[GIN] 2023/01/13 - 23:38:00 | 200 | 115.166µs | 127.0.0.1 | GET | "/air"
main.go has changed
building...
running...
[GIN] 2023/01/13 - 23:38:58 | 200 | 123.916µs | 127.0.0.1 | GET | "/air"
```

English | 简体中文

🔗 Motivation

When I started developing websites in Go and using [gin](#) framework, it was a pity that gin lacked a live-reloading function. So I searched around and tried [fresh](#), it seems not much flexible, so I intended to rewrite it better. Finally, Air's born. In addition, great thanks to [pilu](#), no fresh, no air :)

Air is yet another live-reloading command line utility for developing Go applications. Run `air` in your project root directory, leave it alone, and focus on your code.

Note: This tool has nothing to do with hot-deploy for production.

🔗 Features

- Colorful log output
- Customize build or any command
- Support excluding subdirectories
- Allow watching new directories after Air started
- Better building process

🔗 🌟 beta feature

Support air config fields as arguments:

If you want to config build command and run command, you can use like the following command without the config file:

```
air --build.cmd "go build -o bin/api cmd/run.go" --build.bin "./bin/api"
```

Use a comma to separate items for arguments that take a list as input:

```
air --build.cmd "go build -o bin/api cmd/run.go" --build.bin "./bin/api" --  
build.exclude_dir "templates,build"
```

🔗 Installation

🔗 Prefer install.sh

```
# binary will be $(go env GOPATH)/bin/air  
curl -sSfL https://raw.githubusercontent.com/cosmtrek/air/master/install.sh | sh  
  
# or install it into ./bin/  
curl -sSfL https://raw.githubusercontent.com/cosmtrek/air/master/install.sh | sh  
  
air -v
```

🔗 Via go install

With go 1.16 or higher:

```
go install github.com/cosmtrek/air@latest
```

🔗 Docker

Please pull this docker image [cosmtrek/air](#).

```
docker run -it --rm \
  -w "<PROJECT>" \
  -e "air_wd=<PROJECT>" \
  -v $(pwd):<PROJECT> \
  -p <PORT>:<APP SERVER PORT> \
  cosmtrek/air
  -c <CONF>
```

► For example

🔗 Usage

For less typing, you could add `alias air='~/air'` to your `.bashrc` or `.zshrc`.

First enter into your project

```
cd /path/to/your_project
```

The simplest usage is run

```
# firstly find `air.toml` in current directory, if not found, use defaults
air -c .air.toml
```

You can initialize the `.air.toml` configuration file to the current directory with the default settings running the following command.

```
air init
```

After this, you can just run the `air` command without additional arguments and it will use the `.air.toml` file for configuration.

```
air
```

For modifying the configuration refer to the [air_example.toml](#) file.

🔗 Runtime arguments

You can pass arguments for running the built binary by adding them after the `air` command.

```
# Will run ./tmp/main bench
air bench

# Will run ./tmp/main server --port 8080
air server --port 8080
```

You can separate the arguments passed for the `air` command and the built binary with `--` argument.

```
# Will run ./tmp/main -h
air ---h

# Will run air with custom config and pass -h argument to the built binary
air -c .air.toml -- -h
```

🔗 Docker-compose

```
services:
  my-project-with-air:
    image: cosmtrek/air
    # working_dir value has to be the same of mapped volume
    working_dir: /project-package
    ports:
      - <any>:<any>
    environment:
      - ENV_A=${ENV_A}
      - ENV_B=${ENV_B}
      - ENV_C=${ENV_C}
    volumes:
      - ./project-relative-path/:/project-package/
```

🔗 Debug

`air -d` prints all logs.

🔗 Installation and Usage for Docker users who don't want to use air image

Dockerfile

```
# Choose whatever you want, version >= 1.16
FROM golang:1.19-alpine

WORKDIR /app

RUN go install github.com/cosmtrek/air@latest

COPY go.mod go.sum ./
RUN go mod download

CMD ["air", "-c", ".air.toml"]
```

docker-compose.yaml

```
version: "3.8"
services:
  web:
    build:
      context: .
      # Correct the path to your Dockerfile
      dockerfile: Dockerfile
    ports:
      - 8080:3000
    # Important to bind/mount your codebase dir to /app dir for live reload
    volumes:
      - ./:/app
```

Q&A

"command not found: air" or "No such file or directory"

```
export GOPATH=$HOME/xxxxx
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
export PATH=$PATH:$(go env GOPATH)/bin <---- Confirm this line in your profile!!!
```

Error under wsl when ' is included in the bin

Should use `\`` to escape the ``` in the bin. related issue: [#305](#)

Development

Please note that it requires Go 1.16+ since I use `go mod` to manage dependencies.

Fork this project

```
# Clone it
mkdir -p $GOPATH/src/github.com/cosmtrek
cd $GOPATH/src/github.com/cosmtrek
git clone git@github.com:<YOUR USERNAME>/air.git

# Install dependencies
cd air
make ci

# Explore it and happy hacking!
make install
```

Pull requests are welcome.

🔗 Release

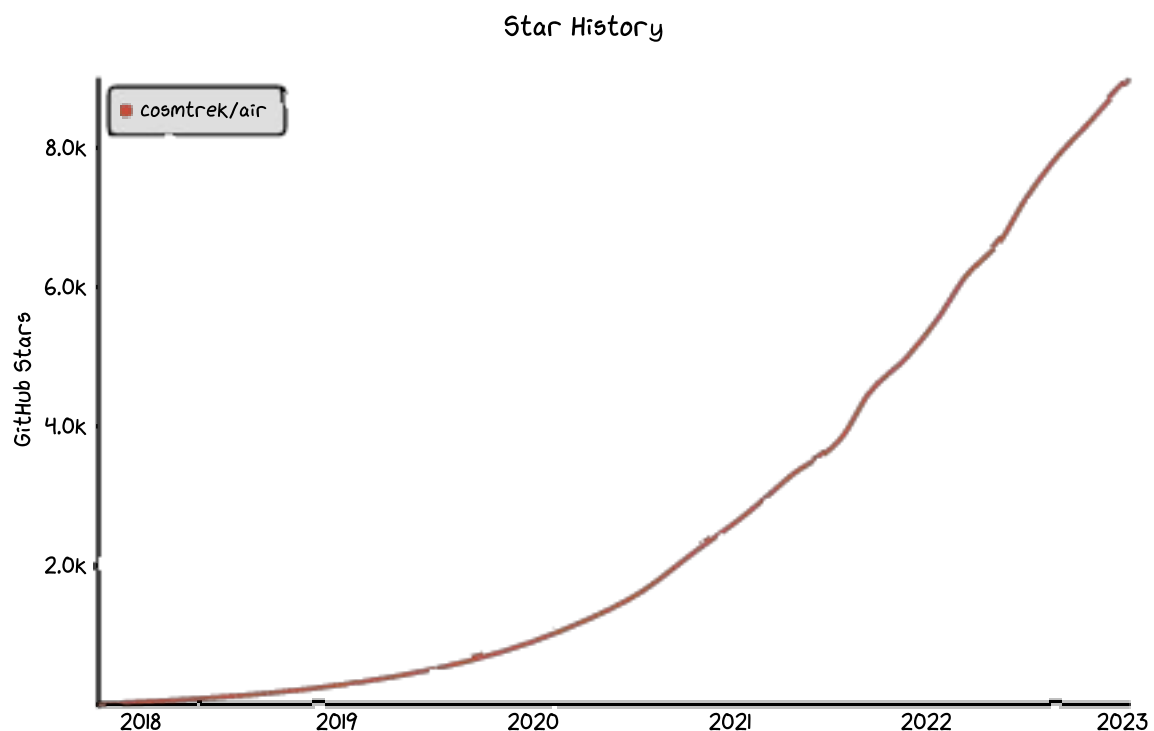
```
# Checkout to master
git checkout master

# Add the version that needs to be released
git tag v1.xx.x

# Push to remote
git push origin v1.xx.x

# The CI will process and release a new version. Wait about 5 min, and you can fe
```

🔗 Star History



Date

[Sponsor](#)




Give huge thanks to lots of supporters. I've always been remembering your kindness.

[License](#)

[GNU General Public License v3.0](#)

Releases 29

 **v1.41.0** Latest
Jan 13, 2023

[+ 28 releases](#)

Sponsor this project



Packages

No packages published

Used by 8



Contributors 52



[+ 41 contributors](#)

Languages



● **Go** 84.8% ● **Shell** 12.6% ● **Makefile** 1.5% ● **Other** 1.1%

[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)



© 2023 GitHub, Inc.