



DevFest  
Siberia 2018

# Go contributors workshop

Novosibirsk, November 23

<https://bit.ly/2DSE9a2>

Ссылка на эти слайды

<http://slack.golang-ru.com>

Канал #gocontributing

# Checklist (пункты для выполнения)

- ☒ Установить **go** и **git**
- ☐ Подписать **Google CLA**
- ☐ Настроить **HTTP Credentials** и **git**
- ☐ Поставить [golang.org/x/review/git-codereview](https://golang.org/x/review/git-codereview)
- ☐ `git clone go.googlesource.com/go`
- ☐ Найти задачу, решить её
- ☐ Выслать свой **CL**

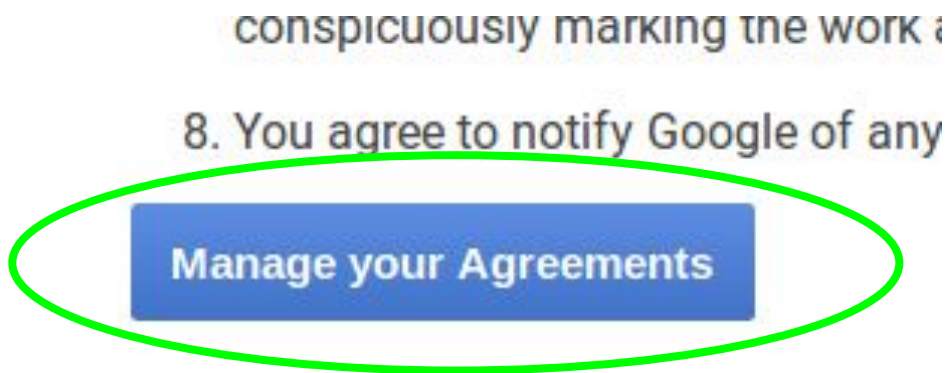
# ВАЖНО

Нужно указать одинаковые email в CLA и **git user.email**. На [github.com](https://github.com) этот email должен быть установлен как основной.

# Оформление Google CLA (1)

<https://cla.developers.google.com/about/google-individual>

Нажать на “Manage your Agreements”



# Оформление Google CLA (2)

Нажать на “Only Yourself”

**Sign a new agreement:**

You plan to submit contributions on behalf of:

**Only Yourself**

**Your Employer**

# Оформление Google CLA (3)

Заполнить форму, нажать “I Agree”

GitHub Account Name

Quasilyte

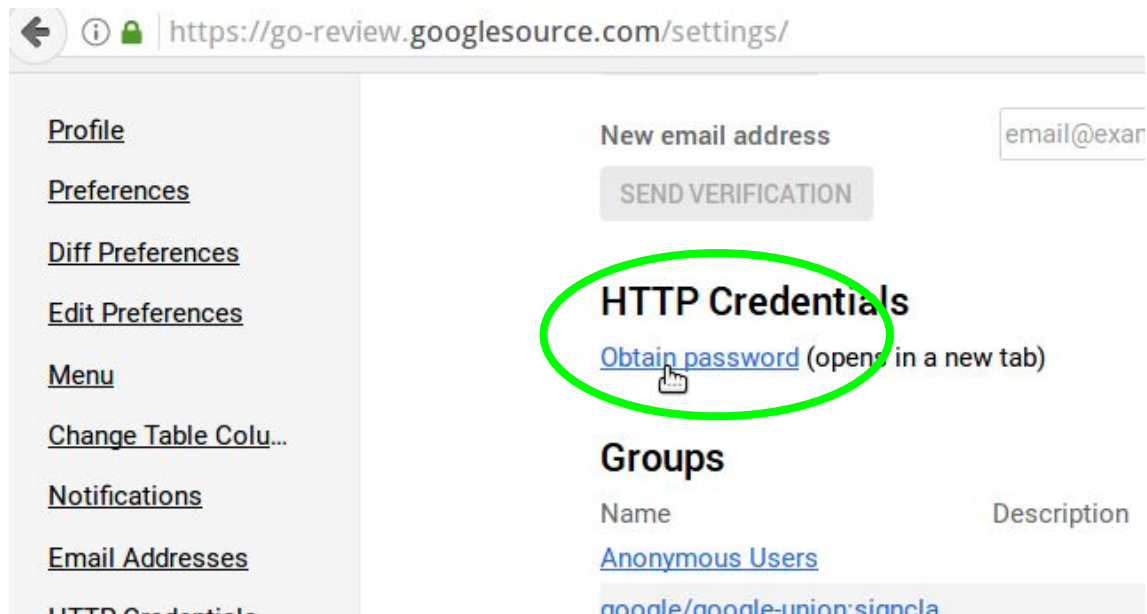
I Agree



# HTTP Credentials (1)

<https://go-review.googlesource.com/settings/>

HTTP Credentials -> Obtain password



# HTTP Credentials (2)

← ⓘ 🔒 <https://www.google.com/new-password?s>



Git

## Configure Git

Configure Git with a cookie for this service by copying this script and p

Windows Users ↴

Authenticate as **quasilyte@gmail.com** to go.google.com

```
set +o history
touch ~/.gitcookies
chmod 0600 ~/.gitcookies

git config --global http.cookiefile ~/.gitcookies
```

Копировать содержимое из блока кода, вставить в командной строке. Для **Linux** и **Windows** инструкции разные, читайте информацию на странице.

# Настройка git

```
$ git config --global user.name nickname
```

```
$ git config --global user.email a@gmail.com
```

```
$ git config user.name
```

```
nickname
```

```
$ git config user.email
```

```
a@gmail.com
```

# Установка git-codereview

```
$ go get -v golang.org/x/review/git-codereview  
$ DST=$(git --exec-path)  
$ cp $GOPATH/bin/git-codereview $DST  
# Стала доступна подкоманда codereview:  
$ git codereview help
```

# Забираем Go исходники

## Внимание!

Если вам не очень комфортно при мыслях о **GOROOT\_BOOTSTRAP**, рекомендуется иметь стабильный Go в **~/go** (Linux) или **%username%/go** (Windows).

# Забираем Go исходники

## Схема инсталляции:

- `~/go` - стабильный Go
- `~/gocontrib` - Go tip + наши изменения

# Забираем Go исходники

```
$ mkdir -p ~/gocontrib
```

```
$ REPO=https://go.googlecode.com/go
```

```
$ git clone $REPO ~/gocontrib
```

```
$ cd ~/gocontrib
```

```
# См. FAQ секцию для объяснений по GOROOT.
```

```
$ export GOROOT=~/gocontrib
```

## Решение задачи

Изменять Go нужно внутри директории `~/golang`. Вносите нужные изменения, затем запускайте тесты. Если тесты проходят, то можно готовить и высылать коммит (change list / CL). Не забудьте выполнять **gofmt** при сохранении Go файлов, если ваш редактор не делает это за вас.



# Запуск тестов, сборка Go

```
$ cd src
```

```
# Windows:
```

```
$ CGO_ENABLED=0 ./all.bat
```

```
# Linux:
```

```
$ ./all.bash
```

```
# Запросит GOROOT_BOOTSTRAP, если ~/go пуст.
```

# Отправление изменений

```
$ git add .
```

```
$ git codereview change branch_name
```

```
# Запросит commit message.
```

```
# См. ссылку или FAQ раздел презентации.
```

```
# https://golang.org/doc/contribute.html#commit\_messages
```

# Забираем Go исходники

```
$ git codereview mail
```

```
remote: Processing changes: new: 1, done
```

```
remote:
```

```
remote: New Changes:
```

```
remote: https://go-review.googlesource.com/#/c/scratch/+/? quasilyte: add main.go
```

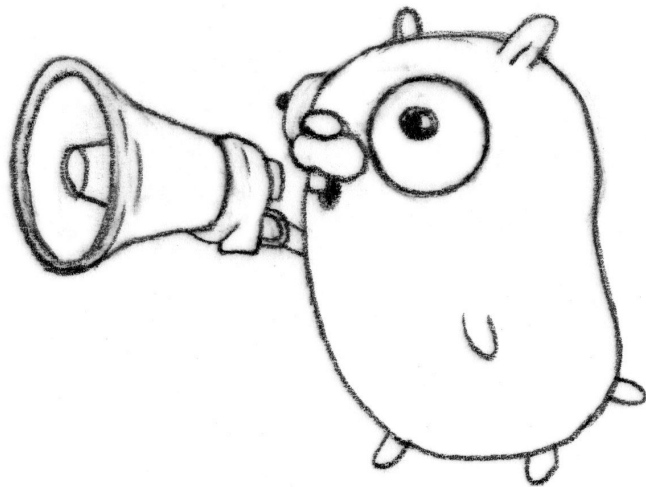
```
remote:
```

Вместо “?” в конце ссылки будет ID изменения.

Ссылка ведёт на ревью патча.

Если ID=123, то патч именуют CL123, etc.

*I demand changes!*



На ревью могут запросить изменения

## Внесение изменений

Производите изменения на ветке, с которой отправлялся CL (branch\_name при “change”).

```
$ git add .
```

```
$ git codereview mail
```

Теперь по прежней ссылке на CL будет доступен patch set #2.

<https://bit.ly/2PLj4Vz>

Таблица для ведения статуса

# FAQ #@!

Если что-то не получается, ниже несколько слайдов, которые описывают основные понятия и могут помочь разобраться самостоятельно.

## Contribution гайды

<https://golang.org/doc/contribute.html>

Есть также устаревший перевод, но, возможно, будет полезен:

[https://github.com/Konstantin8105/Contribution  
Guide\\_RU](https://github.com/Konstantin8105/Contribution_Guide_RU)



## Зачем устанавливать **GOROOT**?

Когда у вас в системе несколько Go дистрибутивов, например, один с tip (master), другой стабильный, нужно указать, какой использовать при компиляции. Если вам нужен tip, нужно переопределять **GOROOT**.

## Commit message (вкратце)

Как минимум 1 строка - summary.

Начинается с пути пакета, затем двоеточие, затем описание “что было сделано”.

Опционально, ставите одну пустую строку и добавляете 1-N строк детального описания.

Можно использовать “Updates #XXX”.

# Терминология

- **Gerrit** - место, где происходит ревью
- **Change** - аналог Pull Request в github
- **CL** - Change List, синоним **change**
- **Patch Set** - версия **CL** коммита

Первый “git codereview mail” отправляет **patch set 1**. Следующие mail увеличивают счётчик.

# Вместо отчаяния

Подборка материалов на эту тему:

<http://goo.gl/nbPfXw> Daniel Martí guide

<https://golang.org/s/gophercon2017>

<https://bit.ly/2Kqp28l> KFU-Go-2018 guide