

Backend Golang

Урок 1

Введение в Go и настройка окружения

Цель занятия

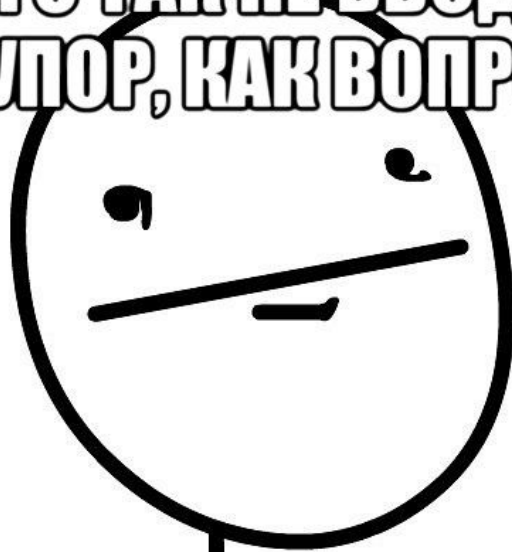
- Познакомиться
- Понять, зачем нужен Golang
- Подготовить рабочее окружение

Преподаватель

Мурзаев Алихан

- Выпускник Satbayev University
- Выпускник 4 потока OneLab
- Пишу на Go с 2020 года
- Работаю над проектом ForteBusiness

**НИЧЕГО ТАК НЕ ВВОДИТ В
СТУПОР, КАК ВОПРОС**



**РАССКАЖИ ЧТО-НИБУДЬ О
СЕБЕ**

Зачем нужен OneLab?

- Чтобы обучить и вырастить будущих специалистов
- Чтобы расширить команду One Technologies

May 2021	May 2020	Change	Programming Language	Ratings	Change
1	1		C	13.38%	-3.68%
2	3	⬆️	Python	11.87%	+2.75%
3	2	⬇️	Java	11.74%	-4.54%
4	4		C++	7.81%	+1.69%
5	5		C#	4.41%	+0.12%
6	6		Visual Basic	4.02%	-0.16%
7	7		JavaScript	2.45%	-0.23%
8	14	⬆️⬆️	Assembly language	2.43%	+1.31%
9	8	⬇️	PHP	1.86%	-0.63%
10	9	⬇️	SQL	1.71%	-0.38%
11	15	⬆️⬆️	Ruby	1.50%	+0.48%
12	17	⬆️⬆️	Classic Visual Basic	1.41%	+0.53%
13	10	⬇️	R	1.38%	-0.46%
14	38	⬆️⬆️	Groovy	1.25%	+0.96%
15	13	⬇️	MATLAB	1.23%	+0.06%
16	12	⬇️⬇️	Go	1.22%	-0.05%
17	23	⬆️⬆️	Delphi/Object Pascal	1.21%	+0.60%
18	11	⬇️⬇️	Swift	1.14%	-0.65%
19	18	⬇️	Perl	1.04%	+0.16%
20	34	⬆️⬆️	Fortran	0.83%	+0.51%

hh

kz

Работодателям

Помощь

Поиск

Создать резюме

Войти

golang

Найти

Вакансии

Резюме

Компании

27 вакансий «golang»

Постоянная работа

Подработка

Подработка

Свежие

Удаленная работа

Нет опыта

по соответствию

весь период

Списком

На карте

Изменить запрос

Регион

Алматы

Соседние города

Бишкек

1

Метро

Первая

5

Уровень дохода

Указан доход

9

от 499 900 KZT

7

от 699 900 KZT

4

Еще 3

Профобласть

Внешний менеджмент

1

Отрасль компании

Опыт работы

График работы

Будьте первыми

Golang разработчик

Умение разбираться в чужом коде. Опыт разработки сложных проектов. Уверенное знание GitLab. Знание SQL (в частности PostgreSQL) и умение оптимизировать...

Откликнуться

5 мая

Будьте первыми

Middle/Senior Go Developer

350 000 – 1 200 000 KZT

ТОО One technologies

Алматы

Разрабатывать новые и улучшать существующие сервисы. Активно участвовать в процессе разработки, от проектирования архитектуры до релиза. Способность самостоятельно выполнять задачи. Опыт разработки Golang от 2х лет. Опыт разработки микро-сервисных приложений. Участие в разработке высоконагруженного сервиса. Понимание и опыт...

Откликнуться

9 мая

Июнь 2021

hh

kz

Работодателям

Помощь

Поиск

Создать резюме

Войти

golang

Найти

Вакансии

Резюме

Компании

36 вакансий «golang»

Постоянная работа

Подработка

Подработка

Свежие

Удаленная работа

Нет опыта

По соответствию

За всё время

Поиск

Меню

Регион

Казахстан

36

Алматы

25

Нур-Султан

10

Костанай

1

Показать все

Уровень дохода

Не имеет значения

от 759 300 KZT

9

от 1 118 600 KZT

7

от 2 196 600 KZT

2

Своя зарплата

от

Указан доход

13

Отклик без резюме

Senior Golang

3 000 – 6 000 USD

Алматы

Переписывание существующего бэкенда с Java на golang . Участие в проектировании архитектуры и интерфейсов. Написание тестов для повышения надежности бэкенда. Опыт Go от 2 лет. Знание gin web framework. Grpc, nacos, MySQL, MongoDB, redis.

Работодатель сейчас онлайн

Откликнуться

Показать контакты

16 октября

Отклик без резюме

Senior Golang

3 000 – 6 000 USD

Нур-Султан

Переписывание существующего бэкенда с Java на golang . Участие в проектировании архитектуры и интерфейсов. Написание тестов для повышения надежности бэкенда. Опыт Go от 2 лет. Знание gin web framework. Grpc, nacos, MySQL, MongoDB, redis.

Работодатель сейчас онлайн

Откликнуться

Показать контакты

16 октября

Октябрь 2021

Где используется Go?

- Backend
- Системные утилиты
- DevOps
- Сетевое программирование

Где НЕ используется Go?

- Разработка игр (клиентская часть)
- Научные вычисления
- Встраиваемые устройства

Что написано на Go?

- Docker
- Kubernetes
- Grafana
- ...

Что написано на Go в One Technologies?

- ForteBusiness
- ForteMarket
- ForteTravel
- Эквайринг E-Commerce

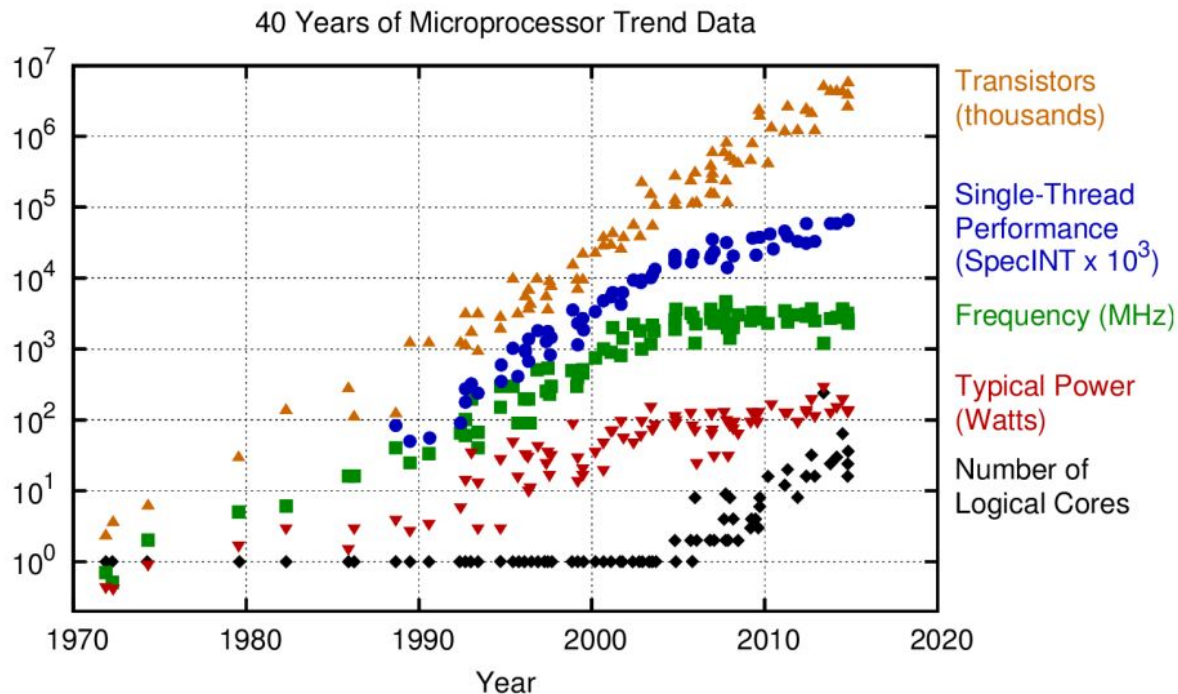
Описание Go

Go — **компилируемый многопоточный** язык программирования, разработанный внутри компании **Google**. Разработкой занимались Роберт Гризмер, Роб Пайк и Кен Томпсон.

Официально язык был представлен в ноябре **2009** года.

Компилируется для **FreeBSD, OpenBSD, Linux, macOS, Windows, DragonFly BSD, Plan 9, Solaris, Android, AIX**.

Развитие процессоров



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Какие проблемы решает Go?

- Медленная сборка проекта
- Сложности контроля зависимостей
- Сложности деплоя и распространения ПО
- Плохая утилизация процессорного времени

Что нужно для обучения?

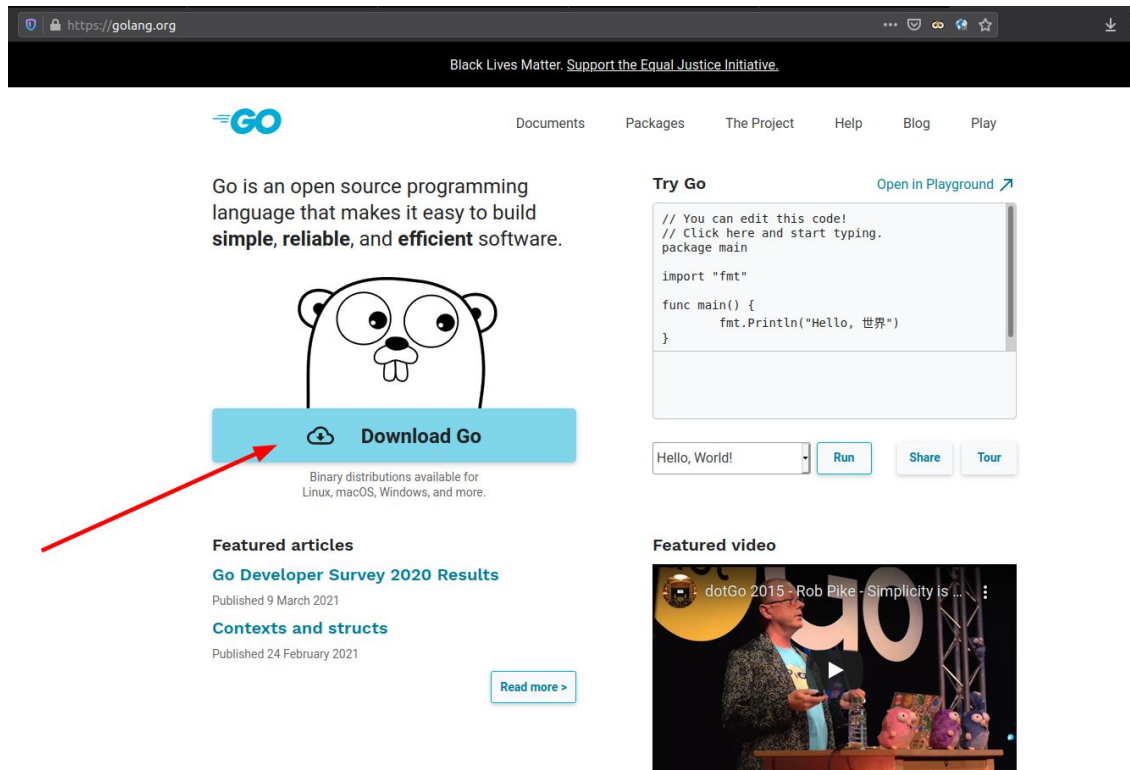
1. Компьютер обыкновенный
2. Интернет
3. ...

Что нужно для обучения?


1. Компьютер обыкновенный
2. Интернет
3. Компилятор
4. Любой редактор кода IDE Goland от JetBrains

Установка компилятора


<https://golang.org>




Black Lives Matter. [Support the Equal Justice Initiative.](#)

 Documents Packages The Project Help Blog Play

Go is an open source programming language that makes it easy to build **simple, reliable, and efficient** software.



 **Download Go**

Binary distributions available for Linux, macOS, Windows, and more.

Featured articles

[Go Developer Survey 2020 Results](#)
Published 9 March 2021

[Contexts and structs](#)
Published 24 February 2021


[Read more >](#)

Try Go [Open in Playground](#)

```
// You can edit this code!  
// Click here and start typing.  
package main  
  
import "fmt"  
  
func main() {  
    fmt.Println("Hello, 世界")  
}
```

Hello, World! [Run](#) [Share](#) [Tour](#)

Featured video

 dotGo 2015 - Rob Pike - Simplicity is ...



Downloads

After downloading a binary release suitable for your system, please follow the [installation instructions](#).

If you are building from source, follow the [source installation instructions](#).

See the [release history](#) for more information about Go releases.

As of Go 1.13, the go command by default downloads and authenticates modules using the Go module mirror and Go checksum database run by Google. See <https://proxy.golang.org/privacy> for privacy information about these services and the [go command documentation](#) for configuration details including how to disable the use of these servers or use different ones.

Featured downloads

Microsoft Windows

Windows 7 or later, Intel 64-bit processor

[go1.16.3.windows-amd64.msi](#) (119MB)

Apple macOS

macOS 10.12 or later, Intel 64-bit processor

[go1.16.3.darwin-amd64.pkg](#) (125MB)

Linux

Linux 2.6.23 or later, Intel 64-bit processor

[go1.16.3.linux-amd64.tar.gz](#) (123MB)

Source

[go1.16.3.src.tar.gz](#) (20MB)

Stable versions

go1.16.3 ▼

Проверка установки

```
alikhhan@cerebro:~$ go version
go version go1.16.3 linux/amd64
alikhhan@cerebro:~$ which go
/usr/local/go/bin/go
```

Установка GoLand

<https://www.jetbrains.com/help/go/installation-guide.html#toolbox>

The screenshot shows a web browser window displaying the JetBrains Toolbox installation guide for Linux. The URL in the address bar is <https://www.jetbrains.com/help/go/installation-guide.html#toolbox>. The page has a dark theme. At the top, there are tabs for 'Windows', 'macOS', and 'Linux', with 'Linux' selected and highlighted by a red rectangle. The main heading is 'Install the Toolbox App'. Below it, there are two numbered steps: 1. Download the tarball `.tar.gz` from the [Toolbox App web page](#). 2. Extract the tarball to a directory that supports file execution. An example command is provided in a code block: `$ sudo tar -xzf jetbrains-toolbox-1.17.7391.tar.gz -C /opt`. Below the code block, there is a paragraph explaining that after running the `jetbrains-toolbox` binary, the Toolbox App will be installed and its icon will be added to the main menu. At the bottom of the screenshot, there is a preview of the JetBrains Toolbox application window. The window has a title bar with standard Linux window controls and a system tray showing the time as 12:56. The application interface shows the 'Tools' tab selected, with a list of available tools under the 'Available' section. The tools listed are GoLand, MPS, and RubyMine. GoLand is highlighted, and a dropdown menu is open next to it, showing a list of versions: 2019.2 Dev, 2019.2 Nightly, 2019.2 EAP, 2019.1 Nightly, 2019.1.3, and 2018.3.6. Each version has a corresponding version number to its right. An 'Install' button is visible next to the version list.

Windows macOS **Linux**

Install the Toolbox App

1. Download the tarball `.tar.gz` from the [Toolbox App web page](#).
2. Extract the tarball to a directory that supports file execution.

For example, if the downloaded version is 1.17.7391, you can extract it to the recommended `/opt` directory using the following command:

```
$ sudo tar -xzf jetbrains-toolbox-1.17.7391.tar.gz -C /opt
```

Execute the `jetbrains-toolbox` binary from the extracted directory to run the Toolbox App and select which product and version you want to install. After you run the Toolbox App for the first time, it will automatically add the Toolbox App icon to the main menu.

JetBrains Toolbox

Projects **Tools**

Available

	GoLand A clever IDE to Go	Install
	MPS Create your own domain language	
	RubyMine The most intelligent Ruby	

2019.2 Dev 192.4488.32
2019.2 Nightly 192.4488.26
2019.2 EAP 192.4205.56
2019.1 Nightly 191.7479.39
2019.1.3 191.7479.32
2018.3.6 183.6156.12

GOPATH & GOROOT

- **GOROOT** - путь к исходникам Go, компилятору
- **GOPATH** - путь к workspace, где будут находиться исходники, бинарники ваших программ

Где будем хранить исходники?

```
alikhhan@cerebro:~$ tree -L 1 ~/go
/home/alikhhan/go
├── bin
├── pkg
└── src

3 directories, 0 files
```

go/bin

Скомпилированные бинарники

```
alikhhan@cerebro:~$ tree -L 1 ~/go/bin/  
/home/alikhhan/go/bin/  
├── b7m  
├── benchcmp  
├── benchstat  
├── benchviz  
├── commander  
├── currency_conversion_api  
├── dlv  
├── faq  
├── gdlv  
├── go101  
├── go-callvis  
├── gocode  
├── gocode-gomod  
├── godef  
├── go-erd  
├── gokit-stringsvc  
├── golint  
├── go-outline  
├── gopkgs  
├── gops  
├── goreturns  
├── hello  
├── m5  
├── r2d2  
├── test  
├── tests  
└── warrior
```

go/pkg

Скачанные пакеты (зависимости)

```
alikhhan@cerebro:~$ tree -L 2 ~/go/pkg/  
/home/alikhhan/go/pkg/  
├── linux_amd64  
│   ├── git.dar.kz  
│   ├── github.com  
│   ├── golang.org  
│   ├── google.golang.org  
│   ├── gopkg.in  
│   ├── go.uber.org  
│   └── layeh.com  
├── linux_amd64_race  
│   └── git.dar.kz  
├── mod  
│   ├── cache  
│   ├── git.dar.kz  
│   ├── github.com  
│   ├── golang.org  
│   ├── google.golang.org  
│   ├── go.opentelemetry.io  
│   ├── gopkg.in  
│   └── go.uber.org  
└── sumdb  
    └── sum.golang.org  
  
21 directories, 0 files
```


go/src

Исходные коды программ

```
— github.com
  — aarzilli
    — gdlv
  — alecthoimas
    — template
  — alikhanmurzayev
    — currency_conversion_api
    — go-design-patterns
    — gokit-stringsvc
    — melody-test
    — onelab-golang
    — onetech_internship_test
    — onetech_internship_test_solved
  — andybalholm
    — brotli
    — cascadia
  — armon
    — go-socks5
  — aymerick
    — raymond
  — beevik
    — ntp
  — beorn7
    — perks
  — blang
    — semver
  — boj
    — redistore
```

Hello, world!

```
package main

import (
    "fmt"
)

func main() {
    fmt.Println("Golang рулит")
}
```

```
$ go run main.go
```

Объявление переменных

```
package main

import (
    "fmt"
)

func main() {
    var a int = 5
    var b = 6
    c := 7
    fmt.Printf("%d, %d, %d\n", a, b, c)
    c, d := 9, 132
    b = 15
    fmt.Printf("%d, %d, %d, %d\n", a, b, c, d)
}
```

Константы

```
package main

import (
    "fmt"
)

func main() {
    const (
        a = 1 // a == 1
        b = 2 // b == 2
        c
        // c == 2
        d
        // d == 2
    )
    fmt.Printf("%d %d %d %d", a, b, c, d)
}
```

Константы iota

```
package main

import (
    "fmt"
)

func main() {
    const (
        a = iota // a == 0
        b = iota // b == 1
        c = iota // c == 2
        d      // d == 3 (implicitly d = iota)
    )
    fmt.Printf("%d %d %d %d", a, b, c, d)
}
```

Условный оператор

```
package main

import "fmt"

func main() {
    a := 8
    b := 7
    if a > b {
        fmt.Println("yes")
    } else {
        fmt.Println("no")
    }
}
```

Присвоение внутри условного оператора

```
package main

func main() {
    if a, b := 5, 7; a > b {
        println("yes")
    } else {
        println("no")
    }
}
```

Цикл **for**

```
package main

import "fmt"

func main() {
    for i := 0; i < 10; i++ {
        fmt.Println(i)
    }
}
```


Цикл **while** (ахах)

```
package main

import "fmt"

func main() {
    i := 0
    for i < 10 {
        fmt.Println(i)
        i++
    }
}
```

Вечный цикл

```
package main

import "fmt"

func main() {
    i := 0
    for {
        fmt.Println(i)
        i++
        if i > 10 {
            break
        }
    }
}
```

Итерация по массиву

```
package main

func main() {
    a := [...]int{81, 54, 43, 66}
    for i, v := range a {
        println(i, v)
    }
}
```

Switch

```
package main

func main() {
    a := 15
    switch a {
    case 10:
        println("One")
    case 15:
        println("Two")
    case 30:
        println("Three")
    default:
        println("DEFAULT!")
    }
}
```

На этом пока всё

часть презентации была бессовестно украдена

Вопросы?

ДЗ

1. Посмотреть [Как 200 строк на Go помогли нам освободить 15 серверов – Паша Мурзаков \(Badoo\)](#).
2. Прочитать [Всё, что вы хотели знать про GOPATH и GOROOT](#).
3. Научиться настраивать GOPATH и GOROOT в Goland.
4. Научиться устанавливать разные версии языка в Goland для разных проектов.
5. Изучить сочетания клавиш Goland (открыть терминал, открыть project tool window, запустить программу и т.д.)
6. Поиграться с базовыми конструкциями языка из презентации.