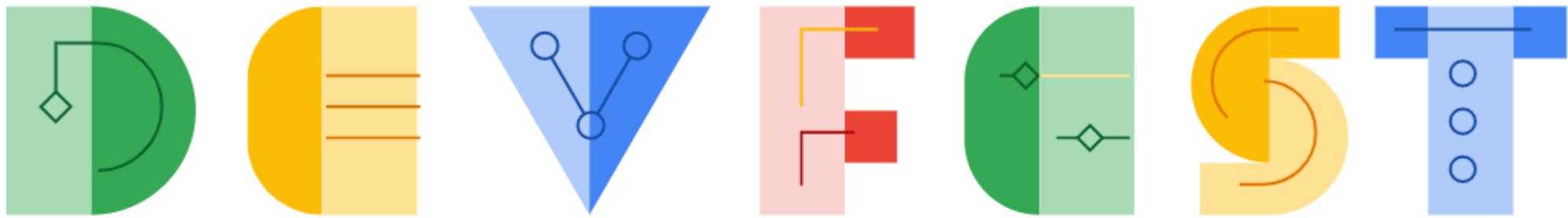


GDG

Nizhny Novgorod





2019



devfest.gdgnn.ru



Nizhny Novgorod. November 15-16, 2019
Be a hero. Be a GDG!

VIEW HIGHLIGHTS

BECOME A SPEAKER

Social Networks



<https://t.me/gdgnn>



<https://www.facebook.com/gdgnizhny>



<https://www.vk.com/gdgnizhny>



<https://www.instagram.com/gdgnizhny>

Golang Workshop

Why Go?



Чем хорош Go в web

- Статически типизированный язык
- Обширная стандартная библиотека
- Корутины - горутины
- Обратная совместимость
- Один бинарный файл
- Кросс-компиляция

Для начала

- Установить Go с сайта golang.org
- IDE для удобства
 - Visual Studio Code
 - Goland
 - Другая на Ваш выбор
- Установка пакетов `go get ...`

HTTP сервер из коробки

```
package main
import (
    "fmt"
    "net/http"
    "time"
)
func main() {
    http.ListenAndServe("127.0.0.1:8001", http.HandlerFunc(
        func(w http.ResponseWriter, r *http.Request) {
            fmt.Fprintf(w, "Welcome to web at %s!", time.Now().Format("03:04:05"))
        }))
}
```

Роутинг и шаблоны

```
func main() {
    tmpl := template.Must(template.New("index").Parse(`<html>
        <head><link rel="stylesheet" href="/static/awsm.min.css"></head>
        <body><h1>Welcome to web at {{.Format "03:04:05"}}!</h1></body>
    </html>`))
    router := http.NewServeMux()
    router.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        w.Header().Add("Content-Type", "text/html")
        w.WriteHeader(http.StatusOK)
        tmpl.Execute(w, time.Now())
    })
    fs := http.FileServer(http.Dir("static/"))
    router.Handle("/static/", http.StripPrefix("/static/", fs))
    http.ListenAndServe("127.0.0.1:8002", router)
}
```

Подключение к SQL БД

```
import _ "github.com/mattn/go-sqlite3"

func main() {
    db, err := sql.Open("sqlite3", "file:db.sqlite?cache=shared")
    if err != nil {
        log.Fatalln(err)
    }
    defer db.Close()
    if _, err := db.Exec("CREATE TABLE IF NOT EXISTS demo (id INTEGER PRIMARY KEY
AUTOINCREMENT, url TEXT, time TIMESTAMP)"); err != nil {
        log.Fatalln(err)
    }
}
```

Middleware

```
func main() {
    // ...
    router := http.NewServeMux()
    middleware := func(w http.ResponseWriter, r *http.Request) {
        path, now := r.URL.String(), time.Now()
        if _, err := db.Exec("INSERT INTO demo (url, time) VALUES (?,?)",
            path, now); err != nil {
            log.Println(err)
        } router.ServeHTTP(w, r)
    }
    // ...
    http.ListenAndServe("127.0.0.1:8003", http.HandlerFunc(middleware))
}
```

Структурируем параметры шаблона

```
import (...)

type templateDataRecord struct {
    ID int
    Time time.Time
    URL string
}

type templateData struct {
    Now time.Time
    Error string
    Records []templateDataRecord
}

func main() {
```

Добавим в шаблон вывод лога запросов

```
func main() {
    // ...
    tmpl := template.Must(template.New("index").Parse(`<html> <head><link
rel="stylesheet" href="/static/awsm.min.css"></head> <body> <h1>Welcome to
web at {{.Now.Format "03:04:05"}}!</h1> {{if .Error}}<h2 style="color:red;">Error:
{{.Error}}</h2>{{- end}} <table> {{range
.Records}}<tr><td>{{.ID}}</td><td>{{.URL}}</td> <td>{{.Time.Format "2006.01.02
03:04:05"}}</td></tr>{{- end}} </table> </body> </html>`))
    // ...
}
```

Добавим данных

```
router.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
    //...
    data := templateData{Now: time.Now()}
    if rows, err := db.Query("SELECT id, url, time FROM demo ORDER BY time DESC LIMIT 15"); err != nil {
        data.Error = err.Error()
    } else {
        defer rows.Close()
        for rows.Next() {
            var record templateDataRecord
            if err := rows.Scan(&record.ID, &record.URL, &record.Time); err != nil {
                data.Error = err.Error()
                break
            }
            data.Records = append(data.Records, record)
        }
    }
    tmpl.Execute(w, data)
})
```

Как сделать проще?

Используем библиотеки

- go get -u github.com/jmoiron/sqlx
- go get -u github.com/labstack/echo/...

Подключаем SQL Бд

```
import (
    // ...
    "github.com/jmoiron/sqlx"
)

func main() {
    db, err := sqlx.Open("sqlite3", "file:db.sqlite?cache=shared")
    if err != nil {
        log.Panicln(err)
    }
    defer db.Close()
    if _, err := db.Exec("CREATE TABLE IF NOT EXISTS demo (id INTEGER PRIMARY
KEY AUTOINCREMENT, url TEXT, time TIMESTAMP)"); err != nil {
        // ...
    }
}
```

Echo

High performance, extensible, minimalist Go web framework



main

```
> go run main.go
```



```
⇒ http server started on :1323
```



Сервер

```
func main() {
    //...
    e := echo.New()
    //...
    e.Static("/static", "static")
    e.GET("/", func(c echo.Context) error {
        data := templateData{Now: time.Now()}
        if err := db.Select(&data.Records, "SELECT * FROM demo ORDER BY time
DESC LIMIT 15"); err != nil {
            data.Error = err.Error()
        }
        return c.Render(http.StatusOK, "index", data)
    })
    e.Logger.Fatal(e.Start("127.0.0.1:8004"))
}
```

Middleware

```
func main() {
    //...
    e.Use(func(next echo.HandlerFunc) echo.HandlerFunc {
        return func(c echo.Context) error {
            path, now := c.Request().URL.String(), time.Now()
            if _, err := db.Exec("INSERT INTO demo (url, time) VALUES
(?,?)", path, now); err != nil {
                c.Logger().Error(err)
            }
            return next(c)
        }
    })
    // ...
}
```

Шаблоны

```
const indexTemplate = `...`  
type templateRenderer struct {  
    templates *template.Template  
}  
  
func (t *templateRenderer) Render(w io.Writer, name string, data  
interface{}, c echo.Context) error {  
    return t.templates.ExecuteTemplate(w, name, data)  
}  
  
func main() {  
    //...  
    e := echo.New()  
    e.Renderer = &templateRenderer{  
        template.Must(template.New("index").Parse(indexTemplate))  
    }  
    // ...  
}
```

Работа с JSON/XML

```
func main() {
    type ColorGroup struct {
        ID int
        Name string
        Colors []string
    }
    group := ColorGroup{ID: 1, Name: "Reds", Colors:
        []string{"Crimson", "Red", "Ruby", "Maroon"},}
    b, err := json.Marshal(group)
    if err != nil {
        fmt.Println("error:", err)
    }
    os.Stdout.Write(b)
}
```

Работа с JSON/XML

```
func main() {
    var jsonBlob = []byte(`[ {"Name": "Platypus", "Order": "Monotremata"}, {"Name": "Quoll", "Order": "Dasyuromorphia"} ]`)
    type Animal struct {
        Name string
        Order string
    }
    var animals []Animal
    err := json.Unmarshal(jsonBlob, &animals)
    if err != nil {
        fmt.Println("error:", err)
    }
    fmt.Printf("%+v", animals)
}
```

Работа в echo с JSON/XML

Счетчик запросов

```
package main
import (
    // ...
    "sync/atomic"
    "github.com/labstack/echo"
)
type response struct {
    Number int64
    Time time.Time
}
var responseCounter int64

func newResponse() response {
    return response{atomic.AddInt64(&responseCounter, 1), time.Now()}
}
```

Работа в echo с JSON/XML

Счетчик запросов

```
func main() {
    e := echo.New()
    e.GET("/json", func(c echo.Context) error {
        return c.JSONPretty(http.StatusOK, newResponse(), " ")
    })
    e.GET("/xml", func(c echo.Context) error {
        return c.XMLPretty(http.StatusOK, newResponse(), " ")
    })
    e.Logger.Fatal(e.Start("127.0.0.1:8005"))
}
```

echo и JSON/XML/FORM

```
type User struct {
    Name string `json:"name" form:"name" query:"name"`
    Email string `json:"email" form:"email" query:"email"`
}

func Handler(c echo.Context) (err error) {
    u := new(User)
    if err = c.Bind(u); err != nil {
        return err
    }
    return c.JSON(http.StatusOK, u)
}
```

Сборка проекта и deploy

- `go build cmd/main.go`
- В Go работает кросс-компиляция, включается переменными окружения `GOOS` и `GOARCH`
- Инструменты автоматизирующие сборку
 - Makefile
 - Task github.com/go-task/task
 - Mage magefile.org
 - Releaser goreleaser.com - пакеты для разных OS, Docker образы

Пишем REST сервис

- Используем библиотеки роутинга `gorilla/mux`, `chi`, `fasthttprouter`
- Используем легкие фреймворки `echo`, `gin`, `macaron`
- Используем фреймворки `beego`, `buffalo`
- Используем кодогенераторы `goa`, `go-swagger`
- Используем библиотеки для построения микросервисов `go-kit`, `gizmo`

Тестирование

```
// файл main_test.go
package main
import (
    "encoding/xml"
    "net/http"
    "net/http/httpptest"
    "testing"
    "time"
)
type testResponse struct {
    Number int64
    Time time.Time
}
// в main.go func createServer() *echo.Echo { /*...*/ }
```

Тестирование

```
func TestServer(t *testing.T) {
    server := httptest.NewServer(createServer())
    defer server.Close()
    now := time.Now()
    resp, err := http.Get(server.URL + "/xml")
    if err != nil {
        t.Errorf("%s", err)
        t.FailNow()
    }
    var r response
    if err := xml.NewDecoder(resp.Body).Decode(&r); err != nil {
        t.Errorf("%s", err)
    }
    if r.Time.Before(now) {
        t.Errorf("response time %v is before %v", r.Number, now)
    }
}
```

Тестирование - фреймворки

- [testify](#) - общего назначения
- [ginkgo](#) - TDD фреймворк
- [httpexpect](#) - тестирование REST API
- [GoConvey](#) - тестирование в browser

WebSocket

Варианты работы с WebScoket

- golang.org/x/net/websocket
- github.com/gorilla/websocket

```
var upgrader = websocket.Upgrader{}
func (s *Service) WebSocket(c echo.Context) error {
    conn, err := upgrader.Upgrade(c.Response(), c.Request(), nil)
    if err != nil { return err }
    defer conn.Close()
    for { mt, message, err := c.ReadMessage()
        if err := c.WriteMessage(mt, message); err != nil { //... }
    }
    return nil
}
```

Что почитать

- [Effective Go](https://golang.org/doc/effective_go.html) (https://golang.org/doc/effective_go.html)
- [Go by Example](https://gobyexample.com/) (<https://gobyexample.com/>)
- [Go database/sql tutorial](http://go-database-sql.org/index.html) (<http://go-database-sql.org/index.html>)



RealWorld example apps

<https://realworld.io/>

conduit

A place to share your knowledge.

Global Feed



longxiao

October 26, 2019

44

555

[Read more...](#)

2



ido2

October 26, 2019

some atrchile

sadkads

[Read more...](#)

1

Popular Tags

butt test dragons training
tags as coffee animation
cars flowers baby caramel
money japan happiness clean
sugar sushi well cookies

Slim  RealWorld example app 0 Star 436 Fork 112	Rails  RealWorld example app 0 Star 449 Fork 80	Kotlin / Spring  RealWorld example app 0 Star 394 Fork 196
Rust  RealWorld example app 0 Star 203 Fork 15	Koa / Knex  RealWorld example app 0 Star 164 Fork 36	Rust / Rocket  RealWorld example app 0 Star 185 Fork 12
Clojure / Polylith  RealWorld example app 0 Star 128 Fork 11	Serverless AWS Lambda  RealWorld example app 0 Star 101 Fork 23	Go Clean Architecture  RealWorld example app 0 Star 105 Fork 15
Scala & Play Framework  RealWorld example app 0 Star 87 Fork 20	CakePHP  RealWorld example app 0 Star 78 Fork 16	GCP Cloud Functions + Datastore  RealWorld example app 0 Star 77 Fork 25
Haskell / Yesod  RealWorld example app 0 Star 79 Fork 11	Go / Echo  RealWorld example app 0 Star 67 Fork 27	Molecular  RealWorld example app 0 Star 66 Fork 21
Hapi.js  RealWorld example app 0 Star 62 Fork 13	F#  RealWorld example app 0 Star 51 Fork 9	Kotlin + Javalin + Exposed  RealWorld example app 0 Star 54 Fork 11
Functional ASP.NET Core  RealWorld example app 0 Star 53 Fork 12	Crystal (Onyx)  RealWorld example app 0 Star 32 Fork 5	Node / hapi pal  RealWorld example app 0 Star 26 Fork 12
QEWD.js  RealWorld example app 0 Star 17 Fork 3	Express w/ Bookshelf.js  RealWorld example app 0 Star 19 Fork 3	Ruby & Colmena  RealWorld example app 0 Star 16 Fork 0

Backends

Vue  RealWorld example app 0 Star 2.2k Fork 596	React / MobX   RealWorld example app 0 Star 902 Fork 196	PureScript + Halogen   RealWorld example app 0 Star 390 Fork 33
AngularJS  RealWorld example app 0 Star 315 Fork 245	Svelte / Sapper  RealWorld example app 0 Star 365 Fork 54	ClojureScript + re-frame  RealWorld example app 0 Star 268 Fork 32
Angular + ngrx + nx    RealWorld example app 0 Star 221 Fork 62	Aurelia  RealWorld example app 0 Star 169 Fork 20	AppRun  RealWorld example app 0 Star 73 Fork 16
ClojureScript + Keechma  RealWorld example app 0 Star 46 Fork 5	Hyperapp 1  RealWorld example app 0 Star 32 Fork 2	Stencil.js  RealWorld example app 0 Star 30 Fork 3
Dojo 2  RealWorld example app	Imba  RealWorld example app	Crizmas MVC  RealWorld example app

Frontends

Ссылки

- Swagger UI <https://editor.swagger.io>
- JSON to Go structs <https://mholt.github.io/json-to-go/>
- API specs
<https://github.com/gothinkster/realworld/blob/master/api/swagger.json>
- JWT <https://jwt.io>
- RealWorld Go + Gin <https://github.com/gothinkster/golang-gin-realworld-example-app>

A photograph of a terracotta head sculpture, possibly a mask, positioned in the lower-left foreground. It has a textured, reddish-brown surface and a small ear visible on the left side. Behind it is a large window or glass wall that looks out onto a city skyline featuring several modern buildings under a clear blue sky.

Вопросы

i@aleksei.co
github.com/a5i

Практическая часть

<https://github.com/a5i/go-workshop>