

Русский



Урок Спринт 3.1

Опция для запуска всех тестов в пакете

Опция `./...` позволяет запускать тесты во всех пакетах внутри текущего модуля. Давайте попробуем запустить тесты во всех пакетах внутри текущего модуля:

```
go test -v ./...
```

Мы видим, что тесты проходят успешно:

```
ivan.petrov ~/project $ go test -v ./...
?      module3_1-lesson      [no test files]
=== RUN   TestPrintHello
--- PASS: TestPrintHello (0.00s)
=== RUN   TestPrintHelloIvan
--- PASS: TestPrintHelloIvan (0.00s)
PASS
ok      module3_1-lesson/printer    0.096s
```

Разницы между командами `go test -v` и `go test -v ./...` нет. Они запускают одни и те же тесты. Теперь давайте добавим папку `counter` внутри папки `internal` и добавим туда 2 файла:

– `main.go`:

```
package counter

var counter int

func Increment() {
    counter++
}
```

– `main_test.go`:

```
package counter

import "testing"

func TestIncrement(t *testing.T) {
    Increment()
}
```

```
    if counter != 1 {  
        t.Error("counter is expected to be incremented")  
    }  
}
```

Структура директорий должна выглядеть примерно так:

```
ivan.petrov ~/project $ tree  
.  
├── README.md  
├── go.mod  
├── internal  
│   └── counter  
│       ├── main.go  
│       └── main_test.go  
├── main.go  
└── main_test.go
```

Запустим тесты во всех пакетах внутри текущего модуля:

```
go test -v ./...
```

Мы видим, что тесты проходят успешно:

```
ivan.petrov ~/project $ go test -v ./...  
=== RUN    TestPrintHello  
--- PASS: TestPrintHello (0.00s)  
=== RUN    TestPrintHelloIvan  
--- PASS: TestPrintHelloIvan (0.00s)  
PASS  
ok        module3_1-lesson      0.090s  
=== RUN    TestIncrement  
--- PASS: TestIncrement (0.00s)  
PASS  
ok        module3_1-lesson/internal/counter 0.147s
```

Запустим `go test -v`:

```
ivan.petrov ~/project $ go test -v  
=== RUN    TestPrintHello  
--- PASS: TestPrintHello (0.00s)  
=== RUN    TestPrintHelloIvan  
--- PASS: TestPrintHelloIvan (0.00s)  
PASS  
ok        module3_1-lesson      0.240s
```

Мы видим, что тесты в пакете `counter` не запускаются. Это происходит, потому что `go test` запускает тесты только в текущем пакете `project`.

В реальной жизни код проекта разложен по директориям и пакетам. Поэтому важно уметь запускать тесты во всех пакетах внутри текущего модуля.

Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»