OZON tech

Allure-Go

Синяев Антон, Ведущий специалист по тестированию

О чем мы сегодня поговорим?

OZON{ech

Оглавление

ozon{tech

Предыстория

Давайте знакомиться

Немного похвастаемся

OZON{ech

200+

звезд на Github
(более 100 уникальных
клонирований ежедневно)

200+

репозиториев-
пользователей
в компании

2 статьи

на Хабре

Комьюнити

официальное признание
от сообщества Allure

Проблематика

Пропасть между Dev & QA

- Основной стек backend разработки на Go, а тесты на Python
- QA тесты — загадка для разработчиков
- Большой разброс по качеству тестов и код базы

Проблематика

OZON{ech

Принятые в компании решения не казались оптимальными для задач тестирования

Python-тесты имели ощутимые **legacy-проблемы**

Попытки написать тесты на GO **до нас «не взлетели»**

Инфраструктура backend **не настроена под тесты QA**

ozon{tech

Что такое Allure-Go?

Allure-Go

Основа основ

OZONtech

Тестовый фреймворк

Allure-provider

Open source библиотека

Кто был до нас?

Исследуем Github

Бибилиотека	Пара слов
gotest2allure	Парсит json, который отдает библиотека testing. Невозможно поддерживать и расширять. Не поддерживает параллельный запуск тестов
GabbyyLS/allure-go-common	Последний коммит 7 лет назад
dailymotion/allure-go	Популярно, но очень не удобно. Высокий порог входа. Много лишних нагромождений и вложений

Dailymotion

```go
func TestAllureSetupTeardown(t *testing.T) {
    allure.BeforeTest(t,
        allure.Description("setup"),
        allure.Action(func() {
            allure.Step(
                allure.Description("Setup step 1"),
                allure.Action(func() {}))
        }))
    allure.Test(t,
        allure.Description("actual test"),
        allure.Action(func() {
            allure.Step(
                allure.Description("Test step 1"),
                allure.Action(func() {}))
        }))
    allure.AfterTest(t,
        allure.Description("teardown"),
        allure.Action(func() {
            allure.Step(
                allure.Description("teardown step 1"),
                allure.Action(func() {}))
        }))
}
```

OzonTech

```go
type MySuite struct {
    suite.Suite
}

func (s *MySuite) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s *MySuite) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s *MySuite) TestSome(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(MySuite))
}
```

Dailymotion

```go
func TestAllureSetupTeardown(t *testing.T) {
    allure.BeforeTest(t,
        allure.Description("setup"),
        allure.Action(func() {
            allure.Step(
                allure.Description("Setup step 1"),
                allure.Action(func() {}))
        }))
    allure.Test(t,
        allure.Description("actual test"),
        allure.Action(func() {
            allure.Step(
                allure.Description("Test step 1"),
                allure.Action(func() {}))
        }))
    allure.AfterTest(t,
        allure.Description("teardown"),
        allure.Action(func() {
            allure.Step(
                allure.Description("teardown step 1"),
                allure.Action(func() {}))
        }))
}
```

OzonTech

```go
type MySuite struct {
    suite.Suite
}

func (s *MySuite) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s *MySuite) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s *MySuite) TestSome(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(MySuite))
}
```

Dailymotion

```go
func TestAllureSetupTeardown(t *testing.T) {
    allure.BeforeTest(t,
        allure.Description("setup"),
        allure.Action(func() {
            allure.Step(
                allure.Description("Setup step 1"),
                allure.Action(func() {}))
        }))
    allure.Test(t,
        allure.Description("actual test"),
        allure.Action(func() {
            allure.Step(
                allure.Description("Test step 1"),
                allure.Action(func() {}))
        }))
    allure.AfterTest(t,
        allure.Description("teardown"),
        allure.Action(func() {
            allure.Step(
                allure.Description("teardown step 1"),
                allure.Action(func() {}))
        }))
}
```

OzonTech

```go
type MySuite struct {
    suite.Suite
}

func (s *MySuite) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s *MySuite) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s *MySuite) TestSome(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(MySuite))
}
```

Dailymotion

```go
func TestAllureSetupTeardown(t *testing.T) {
    allure.BeforeTest(t,
        allure.Description("setup"),
        allure.Action(func() {
            allure.Step(
                allure.Description("Setup step 1"),
                allure.Action(func() {}))
        }))
    allure.Test(t,
        allure.Description("actual test"),
        allure.Action(func() {
            allure.Step(
                allure.Description("Test step 1"),
                allure.Action(func() {}))
        }))
    allure.AfterTest(t,
        allure.Description("teardown"),
        allure.Action(func() {
            allure.Step(
                allure.Description("teardown step 1"),
                allure.Action(func() {}))
        }))
}
```

OzonTech

```go
type MySuite struct {
    suite.Suite
}

func (s *MySuite) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s *MySuite) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s *MySuite) TestSome(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(MySuite))
}
```

ozon{tech

Основные фичи

Модуль Pkg/Allure

OZON{ech

1. Предоставляет Allure-модели (в том числе для интеграции)
2. Предоставляет функционал для работы с этими моделями
3. Отвечает за создание Allure-артефактов в системе

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON tech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON,   := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON{ech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON tech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON tech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZON tech

```go
type JSONStruct struct {
    Message string `json:"message"`
}

type AttachmentTestDemoSuite struct {
    suite.Suite
}

func (s *AttachmentTestDemoSuite) TestAttachment(t provider.T) {
    // Example A
    attachmentText := `THIS IS A TEXT ATTACHMENT`
    attachmentA := allure.NewAttachment(
      "Text Attachment",
      allure.Text,
      []byte(attachmentText))
    t.WithAttachments(attachmentA)

    // Example B
    step := allure.NewSimpleStep("Step A")
    var ExampleJson = JSONStruct{"this is JSON message"}
    attachmentJSON, _ := json.Marshal(ExampleJson)
    attachmentB := allure.NewAttachment(
      "Json Attachment for Step A",
      allure.JSON,
      attachmentJSON)
    step.WithAttachments(attachmentB)
    t.Step(step)
}
```

Модуль Pkg/Allure

OZONtech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZONtech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON tech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZONtech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON{ech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON tech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZONtech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZONtech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON{ech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON tech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON tech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON tech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZONtech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON tech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON tech

```go
type LabelsDemoSuite struct {
    suite.Suite
}

func (s *LabelsDemoSuite) BeforeEach(t provider.T) {
    t.Epic("Demo")
    t.Feature("Labels")
    t.Story("Story Label")

    t.Owner("John Doe")
    t.Lead("John Doe's Boss")

    t.Tag("EachTestTag")
}

func (s *LabelsDemoSuite) TestLabelsExample1(t provider.T) {
    t.Title("Labels Demo Example 1")
    t.Description(`
        This Test will have all labels from SetupTest function
        Unique labels:
            ID = "example1"
            Severity = "blocker"
            Unique tag = "Example1"
        Also this test has additional "suite" label`)

    t.ID("Id example1")
    t.Severity(allure.BLOCKER)

    t.AddSuiteLabel("AnotherSuite")

    t.Tag("Example1")
}
```

Модуль Pkg/Allure

OZON tech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Allure

OZON tech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Allure

OZONtech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Allure

OZON tech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Allure

OZON tech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Allure

OZON tech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Allure

OZON tech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Allure

OZON tech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Allure

OZON tech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Allure

OZON tech

```go
func (s *StepDemoSuite) TestInnerStep(t provider.T) {
    t.Epic("Demo")
    t.Feature("Steps")
    t.Title("Add child steps to existed step.")
    t.Description(`
        Step A is parent step for Step B and Step C
        Step D is parent step for Step E and Step F
        Call order will be saved in allure report
        A -> (B, C), D -> (E, F)`)

    t.Tags("Steps", "Nesting")

    // Example A
    stepA := allure.NewSimpleStep("Step A")
    stepB := allure.NewSimpleStep("Step B")
    stepC := allure.NewSimpleStep("Step C")
    stepA.WithChild(stepB)
    stepA.WithChild(stepC)
    t.Step(stepA)

    // Example B
    stepD := allure.NewSimpleStep("Step D")
    t.Step(stepD)
    stepD.WithChild(allure.NewSimpleStep("Step E"))
    stepF := allure.NewStep("Step F", // Step's Name
        allure.Passed,                          // Step Status
        allure.GetNow(),                        // Step Start
        allure.GetNow(),                        // Step Finish
        allure.NewParameters("paramF", "value")) // Step Parameters
    stepF.WithParent(stepD)
}
```

Модуль Pkg/Framework

OZON{ech

1. **Предоставляет функционал для сбора и запуска тестов**
   - Suite
   - Run
   - Runner

2. **Предоставляет широкий набор возможностей по работе с Allure**
   - Заполнение всех базовых полей
   - Оборачивание ассертов в шаги на уровне фреймворка
   - Полноценная параллельность в тестах
   - Табличные (параметризованные) тесты

3. **Предоставляет интерфейсы для интеграций (например, библиотека Ozontech/Cute)**

Модуль Pkg/Framework

OZONtech

```
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZON{ech

```go
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZON{ech

```go
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZON{ech

```go
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZONtech

```
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZON{ech

```go
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZON{ech

```go
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZONtech

```go
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZONtech

```go
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZON{ech

```go
runner.Run(t, "My test", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("runner.RunTest")

        t.Title("Some Sample test")
        t.Description("allure-go allows you to use allure without suites")
        t.WithParameters(allure.NewParameter("host", "localhost"))

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "No provider initialization")
}
```

Модуль Pkg/Framework

OZON{ech

```go
type AllureGoBeforesAfters struct {
    suite.Suite
}

func (s AllureGoBeforesAfters) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s AllureGoBeforesAfters) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s AllureGoBeforesAfters) BeforeAll(t provider.T) {
    t.NewStep("Setup test step 2")
}

func (s AllureGoBeforesAfters) AfterAll(t provider.T) {
    t.NewStep("Teardown test step 2")
}

func (s AllureGoBeforesAfters) TestBeforesAfters(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(AllureGoBeforesAfters))
}
```

Модуль Pkg/Framework

OZON{ech

```go
type AllureGoBeforesAfters struct {
    suite.Suite
}

func (s AllureGoBeforesAfters) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s AllureGoBeforesAfters) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s AllureGoBeforesAfters) BeforeAll(t provider.T) {
    t.NewStep("Setup test step 2")
}

func (s AllureGoBeforesAfters) AfterAll(t provider.T) {
    t.NewStep("Teardown test step 2")
}

func (s AllureGoBeforesAfters) TestBeforesAfters(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(AllureGoBeforesAfters))
}
```

Модуль Pkg/Framework

OZON{ech

```go
type AllureGoBeforesAfters struct {
    suite.Suite
}

func (s AllureGoBeforesAfters) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s AllureGoBeforesAfters) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s AllureGoBeforesAfters) BeforeAll(t provider.T) {
    t.NewStep("Setup test step 2")
}

func (s AllureGoBeforesAfters) AfterAll(t provider.T) {
    t.NewStep("Teardown test step 2")
}

func (s AllureGoBeforesAfters) TestBeforesAfters(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(AllureGoBeforesAfters))
}
```

Модуль Pkg/Framework

OZON{ech

```go
type AllureGoBeforesAfters struct {
    suite.Suite
}

func (s AllureGoBeforesAfters) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s AllureGoBeforesAfters) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s AllureGoBeforesAfters) BeforeAll(t provider.T) {
    t.NewStep("Setup test step 2")
}

func (s AllureGoBeforesAfters) AfterAll(t provider.T) {
    t.NewStep("Teardown test step 2")
}

func (s AllureGoBeforesAfters) TestBeforesAfters(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(AllureGoBeforesAfters))
}
```

Модуль Pkg/Framework

OZON{ech

```go
type AllureGoBeforesAfters struct {
    suite.Suite
}

func (s AllureGoBeforesAfters) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s AllureGoBeforesAfters) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s AllureGoBeforesAfters) BeforeAll(t provider.T) {
    t.NewStep("Setup test step 2")
}

func (s AllureGoBeforesAfters) AfterAll(t provider.T) {
    t.NewStep("Teardown test step 2")
}

func (s AllureGoBeforesAfters) TestBeforesAfters(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(AllureGoBeforesAfters))
}
```

Модуль Pkg/Framework

OZON{ech

```go
type AllureGoBeforesAfters struct {
    suite.Suite
}

func (s AllureGoBeforesAfters) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s AllureGoBeforesAfters) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s AllureGoBeforesAfters) BeforeAll(t provider.T) {
    t.NewStep("Setup test step 2")
}

func (s AllureGoBeforesAfters) AfterAll(t provider.T) {
    t.NewStep("Teardown test step 2")
}

func (s AllureGoBeforesAfters) TestBeforesAfters(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(AllureGoBeforesAfters))
}
```

Модуль Pkg/Framework

OZON{ech

```go
type AllureGoBeforesAfters struct {
    suite.Suite
}

func (s AllureGoBeforesAfters) BeforeEach(t provider.T) {
    t.NewStep("Setup test step 1")
}

func (s AllureGoBeforesAfters) AfterEach(t provider.T) {
    t.NewStep("Teardown test step 1")
}

func (s AllureGoBeforesAfters) BeforeAll(t provider.T) {
    t.NewStep("Setup test step 2")
}

func (s AllureGoBeforesAfters) AfterAll(t provider.T) {
    t.NewStep("Teardown test step 2")
}

func (s AllureGoBeforesAfters) TestBeforesAfters(t provider.T) {
    t.Epic("Compare with allure-go")
    t.NewStep("Test step 1")
}

func TestAllureGoBeforesAfters(t *testing.T) {
    suite.RunSuite(t, new(AllureGoBeforesAfters))
}
```

Модуль Pkg/Framework

OZON{ech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Модуль Pkg/Framework

OZONtech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Модуль Pkg/Framework

OZONtech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Модуль Pkg/Framework

OZONtech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Модуль Pkg/Framework

OZON{ech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Модуль Pkg/Framework

OZONtech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Модуль Pkg/Framework

OZON tech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Модуль Pkg/Framework

OZON{ech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Модуль Pkg/Framework

OZON tech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Модуль Pkg/Framework

OZONtech

```go
func TestOtherSampleDemo(realT *testing.T) {
    r := runner.NewRunner(realT, realT.Name())

    r.BeforeEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is before test step for %s", t.Name()))
    })
    r.BeforeAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is BeforeAll test step for %s", t.Name()))
    })
    r.AfterEach(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterEach test step for %s", t.Name()))
    })
    r.AfterAll(func(t provider.T) {
        t.NewStep(fmt.Sprintf("This is AfterAll test step for %s", t.Name()))
    })

    r.NewTest("My test 1", func(t provider.T) {
        t.Epic("Only Provider Demo")
        t.Feature("T.Run()")

        t.Title("Some Other Sample test")
        t.Description("allure-testify allows you to use allure without suites")

        t.WithNewStep("Some nested step", func(ctx provider.StepCtx) {
            ctx.WithNewStep("Some inner step 1", func(ctx provider.StepCtx) {

            })
            ctx.WithNewStep("Some inner step 2", func(ctx provider.StepCtx) {

            })
        })
    }, "Sample", "Provider-only", "with provider initialization")

    r.RunTests()
}
```

Параметризация

OZON{ech

```go
type ParametrizedSuite struct {
    suite.Suite
    ParamCities []string
}

func (s *ParametrizedSuite) BeforeAll(t provider.T) {
    for i := 0; i < 10; i++ {
        s.ParamCities = append(s.ParamCities, fake.City())
    }
}

func (s *ParametrizedSuite) TableTestCities(t provider.T, city string)
{   t.Parallel()
    t.Require().NotEmpty(city)
}

func TestNewParametrizedDemo(t *testing.T) {
    suite.RunSuite(t, new(ParametrizedSuite))
}
```

Параметризация

OZON{ech

```go
type ParametrizedSuite struct {
    suite.Suite
    ParamCities []string
}

func (s *ParametrizedSuite) BeforeAll(t provider.T) {
    for i := 0; i < 10; i++ {
        s.ParamCities = append(s.ParamCities, fake.City())
    }
}

func (s *ParametrizedSuite) TableTestCities(t provider.T, city string)
{   t.Parallel()
    t.Require().NotEmpty(city)
}

func TestNewParametrizedDemo(t *testing.T) {
    suite.RunSuite(t, new(ParametrizedSuite))
}
```

Параметризация

OZONtech

```go
type ParametrizedSuite struct {
    suite.Suite
    ParamCities []string
}

func (s *ParametrizedSuite) BeforeAll(t provider.T) {
    for i := 0; i < 10; i++ {
        s.ParamCities = append(s.ParamCities, fake.City())
    }
}

func (s *ParametrizedSuite) TableTestCities(t provider.T, city string)
{   t.Parallel()
    t.Require().NotEmpty(city)
}

func TestNewParametrizedDemo(t *testing.T) {
    suite.RunSuite(t, new(ParametrizedSuite))
}
```

Параметризация

OZONtech

```go
type ParametrizedSuite struct {
    suite.Suite
    ParamCities []string
}

func (s *ParametrizedSuite) BeforeAll(t provider.T) {
    for i := 0; i < 10; i++ {
        s.ParamCities = append(s.ParamCities, fake.City())
    }
}

func (s *ParametrizedSuite) TableTestCities(t provider.T, city string)
{   t.Parallel()
    t.Require().NotEmpty(city)
}

func TestNewParametrizedDemo(t *testing.T) {
    suite.RunSuite(t, new(ParametrizedSuite))
}
```

Параметризация

OZONtech

```go
type ParametrizedSuite struct {
    suite.Suite
    ParamCities []string
}

func (s *ParametrizedSuite) BeforeAll(t provider.T) {
    for i := 0; i < 10; i++ {
        s.ParamCities = append(s.ParamCities, fake.City())
    }
}

func (s *ParametrizedSuite) TableTestCities(t provider.T, city string)
{   t.Parallel()
    t.Require().NotEmpty(city)
}

func TestNewParametrizedDemo(t *testing.T) {
    suite.RunSuite(t, new(ParametrizedSuite))
}
```

Параметризация

OZONtech

```go
type ParametrizedSuite struct {
    suite.Suite
    ParamCities []string
}

func (s *ParametrizedSuite) BeforeAll(t provider.T) {
    for i := 0; i < 10; i++ {
        s.ParamCities = append(s.ParamCities, fake.City())
    }
}

func (s *ParametrizedSuite) TableTestCities(t provider.T, city string)
{   t.Parallel()
    t.Require().NotEmpty(city)
}

func TestNewParametrizedDemo(t *testing.T) {
    suite.RunSuite(t, new(ParametrizedSuite))
}
```

Параметризация

OZON{ech

```go
type ParametrizedSuite struct {
    suite.Suite
    ParamCities []string
}

func (s *ParametrizedSuite) BeforeAll(t provider.T) {
    for i := 0; i < 10; i++ {
        s.ParamCities = append(s.ParamCities, fake.City())
    }
}

func (s *ParametrizedSuite) TableTestCities(t provider.T, city string)
{   t.Parallel()
    t.Require().NotEmpty(city)
}

func TestNewParametrizedDemo(t *testing.T) {
    suite.RunSuite(t, new(ParametrizedSuite))
}
```

Параметризация

OZONtech

```go
type ParametrizedSuite struct {
    suite.Suite
    ParamCities []string
}

func (s *ParametrizedSuite) BeforeAll(t provider.T) {
    for i := 0; i < 10; i++ {
        s.ParamCities = append(s.ParamCities, fake.City())
    }
}

func (s *ParametrizedSuite) TableTestCities(t provider.T, city string)
{   t.Parallel()
    t.Require().NotEmpty(city)
}

func TestNewParametrizedDemo(t *testing.T) {
    suite.RunSuite(t, new(ParametrizedSuite))
}
```

Параметризация

OZONtech

```go
type ParametrizedSuite struct {
    suite.Suite
    ParamCities []string
}

func (s *ParametrizedSuite) BeforeAll(t provider.T) {
    for i := 0; i < 10; i++ {
        s.ParamCities = append(s.ParamCities, fake.City())
    }
}

func (s *ParametrizedSuite) TableTestCities(t provider.T, city string)
{   t.Parallel()
    t.Require().NotEmpty(city)
}

func TestNewParametrizedDemo(t *testing.T) {
    suite.RunSuite(t, new(ParametrizedSuite))
}
```

Ассерты

Детализируем отчеты!

provider.T предоставляет методы Require() и Assert() для оборачивания проверок в шаги

Require - «hard assert» (провал проверки приводит к завершению теста)

Assert - «soft assert» (провал проверки **НЕ** приводит к завершению теста)

Ассерты

Пример

```go
func (s *FailsDemoSuite) TestAssertionFail(t provider.T)
{   t.Title("This test failed by assert with message")
    t.Description(`
        This Test will be failed with assert Error.
        Error text: Assertion Failed`)
    t.Tags("fail", "assertions")

    t.Require().Equal(1, 2, "Assertion Failed")
}
```

TestFails/FailsDemoSuite/TestAssertionFail

Failed **This test failed by assert with message**

Overview History Retries

```
Assertion Failed
```

```
Error Trace:    wrapper.go:105
                                        helper.go:37
                                        wrapper.go:101
                                        helper.go:32
                                        fails_test.go:29
Error:          Not equal:
                expected: 1
                actual  : 2
Test:           This test failed by assert with message
Messages:       Assertion Failed
```

Tags: fail assertions

Categories: Product defects

Severity: normal

Duration: 0s

Description

```
This Test will be failed with assert Error.
Error text: Assertion Failed
```

Execution

Test body

REQUIRE: Assertion Failed

2 parameters

Expected	1
Actual	2

```
Assertion Failed
```

TestFails/FailsDemoSuite/TestAssertionFail

Failed **This test failed by assert with message**

Overview History Retries

```
Assertion Failed
```

```
Error Trace:    wrapper.go:105
                                        helper.go:37
                                        wrapper.go:101
                                        helper.go:32
                                        fails_test.go:29
Error:          Not equal:
                expected: 1
                actual  : 2
Test:           This test failed by assert with message
Messages:       Assertion Failed
```

Tags: fail assertions

Categories: Product defects

Severity: normal

Duration: 0s

Description

```
This Test will be failed with assert Error.
Error text: Assertion Failed
```

Execution

Test body

REQUIRE: Assertion Failed
2 parameters

Expected	1
Actual	2

```
Assertion Failed
```

TestFails/FailsDemoSuite/TestAssertionFail

Failed **This test failed by assert with message**

Overview History Retries

```
Assertion Failed
```

```
Error Trace:    wrapper.go:105
                                helper.go:37
                                wrapper.go:101
                                helper.go:32
                                fails_test.go:29
Error:          Not equal:
                expected: 1
                actual  : 2
Test:           This test failed by assert with message
Messages:       Assertion Failed
```

Tags: fail assertions

Categories: Product defects

Severity: normal

Duration: 0s

Description

```
This Test will be failed with assert Error.
Error text: Assertion Failed
```

Execution

Test body

REQUIRE: Assertion Failed

2 parameters

Expected	1
Actual	2

```
Assertion Failed
```

WithTestSetup/WithTestTeardown

OZON{ech

Как подняли, так и уронили

Setup/Teardown для конкретного теста в рамках тестового набора (только в отчетах)

t.WithTestSetup(func(t provider.T)) - добавит все записи в блок SetUp в allure отчете

t.WithTestTeardown(func(t provider.T)) - добавит все записи в блок TearDown в allure отчете

WithTestSetup/WithTestTeardown

OZONtech

Пример

```
func (s *SetupSuite) TestMyOtherTest(t provider.T) {
    t.Title("Just Test WithSetup")
    t.Description(`
        Test will prepare some data at TestSetup.`)
    t.Tags("Parallel", "Setup", "BeforeAfter")

    t.Parallel()
    var (
        name string
        age  int
    )

    t.WithTestSetup(func(t provider.T) {
        t.WithNewStep("init args", func(sCtx provider.StepCtx) {
            name = fake.FullName()
            age = fake.Day()
            sCtx.WithNewParameters("name", name, "age", age)
        })
    })

    t.Assert().NotEmpty(name)
    t.Assert().NotEmpty(age)

    defer t.WithTestTeardown(func(t provider.T) {
        t.WithNewStep("clear args", func(sCtx provider.StepCtx)
{           name = ""
            age = 0
            sCtx.WithNewParameters("name", name, "age", age)
        })
    }
}
```

WithTestSetup/WithTestTeardown

OZON tech

Пример

```
func (s *SetupSuite) TestMyOtherTest(t provider.T) {
    t.Title("Just Test WithSetup")
    t.Description(`
        Test will prepare some data at TestSetup.`)
    t.Tags("Parallel", "Setup", "BeforeAfter")

    t.Parallel()
    var (
        name string
        age  int
    )

    t.WithTestSetup(func(t provider.T) {
        t.WithNewStep("init args", func(sCtx provider.StepCtx) {
            name = fake.FullName()
            age = fake.Day()
            sCtx.WithNewParameters("name", name, "age", age)
        })
    })

    t.Assert().NotEmpty(name)
    t.Assert().NotEmpty(age)

    defer t.WithTestTeardown(func(t provider.T) {
        t.WithNewStep("clear args", func(sCtx provider.StepCtx)
{           name = ""
            age = 0
            sCtx.WithNewParameters("name", name, "age", age)
        })
    }
}
```

WithTestSetup/WithTestTeardown

OZONtech

Пример

```
func (s *SetupSuite) TestMyOtherTest(t provider.T) {
    t.Title("Just Test WithSetup")
    t.Description(`
        Test will prepare some data at TestSetup.`)
    t.Tags("Parallel", "Setup", "BeforeAfter")

    t.Parallel()
    var (
        name string
        age  int
    )

    t.WithTestSetup(func(t provider.T) {
        t.WithNewStep("init args", func(sCtx provider.StepCtx) {
            name = fake.FullName()
            age = fake.Day()
            sCtx.WithNewParameters("name", name, "age", age)
        })
    })

    t.Assert().NotEmpty(name)
    t.Assert().NotEmpty(age)

    defer t.WithTestTeardown(func(t provider.T) {
        t.WithNewStep("clear args", func(sCtx provider.StepCtx)
{           name = ""
            age = 0
            sCtx.WithNewParameters("name", name, "age", age)
        })
    }
}
```

WithTestSetup/WithTestTeardown

OZONtech

Пример

TestRunner/SetupSuite/TestMyOtherTest

Passed **Just Test WithSetup**

Overview History Retries

Tags: BeforeAfter Parallel Setup

Severity: normal

Duration: 1ms

Description

```
Test will prepare some data at TestSetup.
```

Execution

Set up

init args 2 parameters

name	Jane Hamilton
age	23

Test body

ASSERT: Not Empty
1 parameter

ASSERT: Not Empty
1 parameter

Tear down

clear args 2 parameters

name	
age	0

WithTestSetup/WithTestTeardown

Пример

TestRunner/SetupSuite/TestMyOtherTest

Passed **Just Test WithSetup**

Overview History Retries

Tags: BeforeAfter Parallel Setup

Severity: normal

Duration: 1ms

Description

```
Test will prepare some data at TestSetup.
```

Execution

Set up

init args 2 parameters

name	Jane Hamilton
age	23

Test body

ASSERT: Not Empty
1 parameter

ASSERT: Not Empty
1 parameter

Tear down

clear args 2 parameters

name	
age	0

Параллельность тестов

Распредели это

t.Parallel() - метод, используемый библиотекой testing для параллелизации тестов

С момента **вызова** t.Parallel() снимается ограничение с рутины и тест становится асинхронным

Больше никаких действий для параллелизации не требуется

Параллельность

Пример

```go
func (s *AsyncSuiteStepDemo) TestAsyncStepDemo1(t provider.T) {
    t.Title("Async Test with async steps 1")

    t.Parallel()

    t.WithNewAsyncStep("Async Step 1", func(ctx provider.StepCtx) {
        ctx.WithNewParameters("Start", fmt.Sprintf("%s", time.Now()))
        time.Sleep(3 * time.Second)
        ctx.WithNewParameters("Stop", fmt.Sprintf("%s", time.Now()))
    })

    t.WithNewAsyncStep("Async Step 2", func(ctx provider.StepCtx) {
        ctx.WithNewParameters("Start", fmt.Sprintf("%s", time.Now()))
        time.Sleep(3 * time.Second)
        ctx.Logf("Step 2 Stopped At: %s", fmt.Sprintf("%s", time.Now()))
        ctx.WithNewParameters("Stop", fmt.Sprintf("%s", time.Now()))
    })
}
```

Параллельность

Пример

```go
func (s *AsyncSuiteStepDemo) TestAsyncStepDemo1(t provider.T) {
    t.Title("Async Test with async steps 1")

    t.Parallel()

    t.WithNewAsyncStep("Async Step 1", func(ctx provider.StepCtx) {
        ctx.WithNewParameters("Start", fmt.Sprintf("%s", time.Now()))
        time.Sleep(3 * time.Second)
        ctx.WithNewParameters("Stop", fmt.Sprintf("%s", time.Now()))
    })

    t.WithNewAsyncStep("Async Step 2", func(ctx provider.StepCtx) {
        ctx.WithNewParameters("Start", fmt.Sprintf("%s", time.Now()))
        time.Sleep(3 * time.Second)
        ctx.Logf("Step 2 Stopped At: %s", fmt.Sprintf("%s", time.Now()))
        ctx.WithNewParameters("Stop", fmt.Sprintf("%s", time.Now()))
    })
}
```

Параллельность

Пример

```go
func (s *AsyncSuiteStepDemo) TestAsyncStepDemo1(t provider.T) {
    t.Title("Async Test with async steps 1")

    t.Parallel()

    t.WithNewAsyncStep("Async Step 1", func(ctx provider.StepCtx) {
        ctx.WithNewParameters("Start", fmt.Sprintf("%s", time.Now()))
        time.Sleep(3 * time.Second)
        ctx.WithNewParameters("Stop", fmt.Sprintf("%s", time.Now()))
    })

    t.WithNewAsyncStep("Async Step 2", func(ctx provider.StepCtx) {
        ctx.WithNewParameters("Start", fmt.Sprintf("%s", time.Now()))
        time.Sleep(3 * time.Second)
        ctx.Logf("Step 2 Stopped At: %s", fmt.Sprintf("%s", time.Now()))
        ctx.WithNewParameters("Stop", fmt.Sprintf("%s", time.Now()))
    })
}
```

Параллельность

Пример

TestSuiteStepAsyncRunner/AsyncSuiteStepDemo/TestAsyncStepDemo1

Passed **Async Test with async steps 1**

Overview History Retries

Tags: async suite steps

Severity: normal

Duration: 3s 002ms

Execution

Test body

Async Step 1
2 parameters 3s 001ms

Start	2023-04-12 16:25:33.258649 +0300 MSK m=+0.005422543
Stop	2023-04-12 16:25:36.259925 +0300 MSK m=+3.006693376

Async Step 2
2 parameters 3s 002ms

Start	2023-04-12 16:25:33.258609 +0300 MSK m=+0.005383168
Stop	2023-04-12 16:25:36.260456 +0300 MSK m=+3.007224209

Параллельность

Пример

TestSuiteStepAsyncRunner/AsyncSuiteStepDemo/TestAsyncStepDemo1

Passed **Async Test with async steps 1**

Overview History Retries

Tags: async suite steps

Severity: normal

Duration: 3s 002ms

Execution

Test body

Async Step 1
2 parameters — 3s 001ms

Start	2023-04-12 16:25:33.258649 +0300 MSK m=+0.005422543
Stop	2023-04-12 16:25:36.259925 +0300 MSK m=+3.006693376

Async Step 2
2 parameters — 3s 002ms

Start	2023-04-12 16:25:33.258609 +0300 MSK m=+0.005383168
Stop	2023-04-12 16:25:36.260456 +0300 MSK m=+3.007224209

Параллельность

Пример

TestSuiteStepAsyncRunner/AsyncSuiteStepDemo/TestAsyncStepDemo1

Passed **Async Test with async steps 1**

Overview History Retries

Tags: async suite steps

Severity: normal

Duration: 3s 002ms

Execution

Test body

Async Step 1
2 parameters 3s 001ms

Start	2023-04-12 16:25:33.258649 +0300 MSK m=+0.005422543
Stop	2023-04-12 16:25:36.259925 +0300 MSK m=+3.006693376

Async Step 2
2 parameters 3s 002ms

Start	2023-04-12 16:25:33.258609 +0300 MSK m=+0.005383168
Stop	2023-04-12 16:25:36.260456 +0300 MSK m=+3.007224209

Параллельность

Пример

TestSuiteStepAsyncRunner/AsyncSuiteStepDemo/TestAsyncStepDemo1

Passed **Async Test with async steps 1**

Overview History Retries

Tags: async suite steps

Severity: normal

Duration: 3s 002ms

Execution

Test body

Async Step 1
2 parameters 3s 001ms

Start	2023-04-12 16:25:33.258649 +0300 MSK m=+0.005422543
Stop	2023-04-12 16:25:36.259925 +0300 MSK m=+3.006693376

Async Step 2
2 parameters 3s 002ms

Start	2023-04-12 16:25:33.258609 +0300 MSK m=+0.005383168
Stop	2023-04-12 16:25:36.260456 +0300 MSK m=+3.007224209

Управление состоянием теста

OZON{ech

Контролируй это

t.Fail()/t.FailNow() - методы, отмечающие тест как упавший

t.Broken()/t.BrokenNow() - методы, отмечающие тест как сломанный

t.XSkip() - метод, отмечающий тест как пропущенный только в случае падения ассерта (добавляет к имени префикс [XSkip])

Управление состоянием теста

OZON{ech

Контролируй это... и описывай

t.Fatal(args...)t.Fatalf(string, args...) - метод, отмечающий тест как упавший и записывающий сообщение в отчет

t.Break(args...)/t.Breakf(string, args...) - методы, отмечающие тест как сломанный и записывающий сообщение в отчет

t.Skip(args...)/t.Skipf(string, args...) - метод, отмечающий тест как пропущенный и записывающий сообщение в отчет

ozon{ech

Как работает?

provider.T и provider.StepCtx

Что это?

1. Основной интерфейс для работы с фреймворков
2. Является оберткой над *testing.T
3. Предоставляет возможности по работе с Allure и тестами

Allure Report

Из чего состоит?

OZONtech

uuid-container.json

- Контейнер для Before-After-степов всего сьюта

uuid-container.json

- Контейнер для Before-After-степов конкретного теста

uuid-result.json

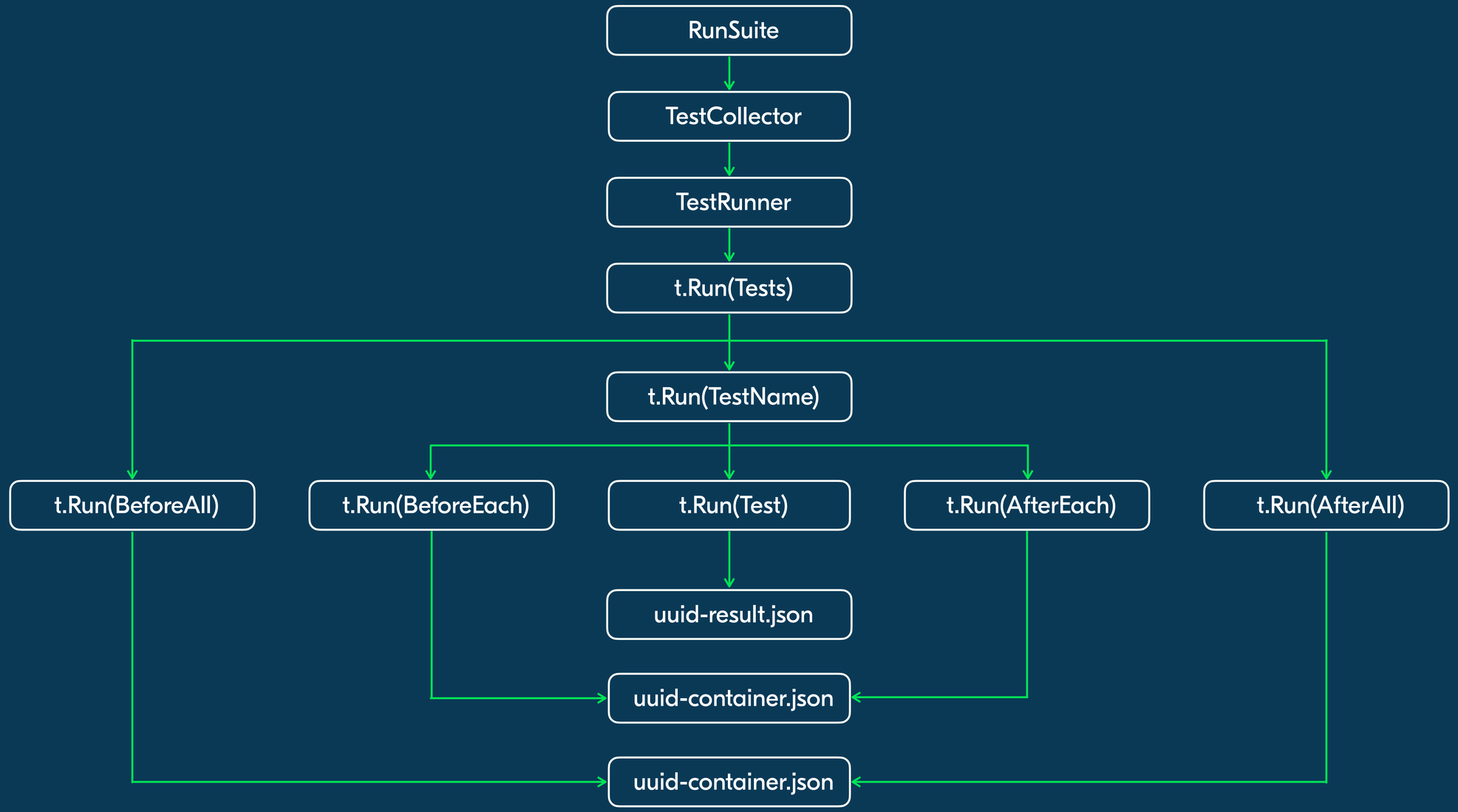
- Отчет по тесту

uuid-attachment.*

- Аттачмент для шагов

Общая схема RunSuite

Разберемся на простом



Шаг 1.
Инициализация и подготовка

Шаг 1. Инициализация и подготовка

OZON{ech

1. Описываем структуру, расширяем ее структурой suite.Suite

```
type SuiteStruct struct {
    suite.Suite
}
```

Шаг 1. Инициализация и подготовка

OZON{ech

1. Описываем структуру, расширяем ее структурой suite.Suite

```
type SuiteStruct struct {
    suite.Suite
}

func (s *SuiteStruct) TestFunction(t provider.T) {
    // test something
}
```

Шаг 1. Инициализация и подготовка

OZON{ech

1. Описываем структуру, расширяем ее структурой suite.Suite

```go
type SuiteStruct struct {
    suite.Suite
}

func (s *SuiteStruct) BeforeEach(t provider.T) {
    // before each
}

func (s *SuiteStruct) AfterEach(t provider.T) {
    // after each
}

func (s *SuiteStruct) TestFunction(t provider.T) {
    // test something
}
```

Шаг 1. Инициализация и подготовка

OZON{ech

1. Описываем структуру, расширяем ее структурой suite.Suite

```go
type SuiteStruct struct {
    suite.Suite
}

func (s *SuiteStruct) BeforeEach(t provider.T) {
    // before each
}

func (s *SuiteStruct) AfterEach(t provider.T) {
    // after each
}

func (s *SuiteStruct) BeforeAll(t provider.T) {
    // before all
}

func (s *SuiteStruct) AfterAll(t provider.T) {
    // after all
}

func (s *SuiteStruct) TestFunction(t provider.T) {
    // test something
}
```

Шаг 1. Инициализация и подготовка

OZON{ech

2. Описываем запускающий тест

```go
type SuiteStruct struct {
    suite.Suite
}

func (s *SuiteStruct) BeforeEach(t provider.T) {
    // before each
}

func (s *SuiteStruct) AfterEach(t provider.T) {
    // after each
}

func (s *SuiteStruct) BeforeAll(t provider.T) {
    // before all
}

func (s *SuiteStruct) AfterAll(t provider.T) {
    // after all
}

func (s *SuiteStruct) TestFunction(t provider.T) {
    // test something
}

func TestRunner(t *testing.T) {

}
```

Шаг 1. Инициализация и подготовка

OZON{ech

3. Инициализируем структуру с тестами

```go
type SuiteStruct struct {
    suite.Suite
}

func (s *SuiteStruct) BeforeEach(t provider.T) {
    // before each
}

func (s *SuiteStruct) AfterEach(t provider.T) {
    // after each
}

func (s *SuiteStruct) BeforeAll(t provider.T) {
    // before all
}

func (s *SuiteStruct) AfterAll(t provider.T) {
    // after all
}

func (s *SuiteStruct) TestFunction(t provider.T) {
    // test something
}

func TestRunner(t *testing.T) {
  mySuite := new(SuiteStruct)
}
```

Шаг 1. Инициализация и подготовка

OZON{ech

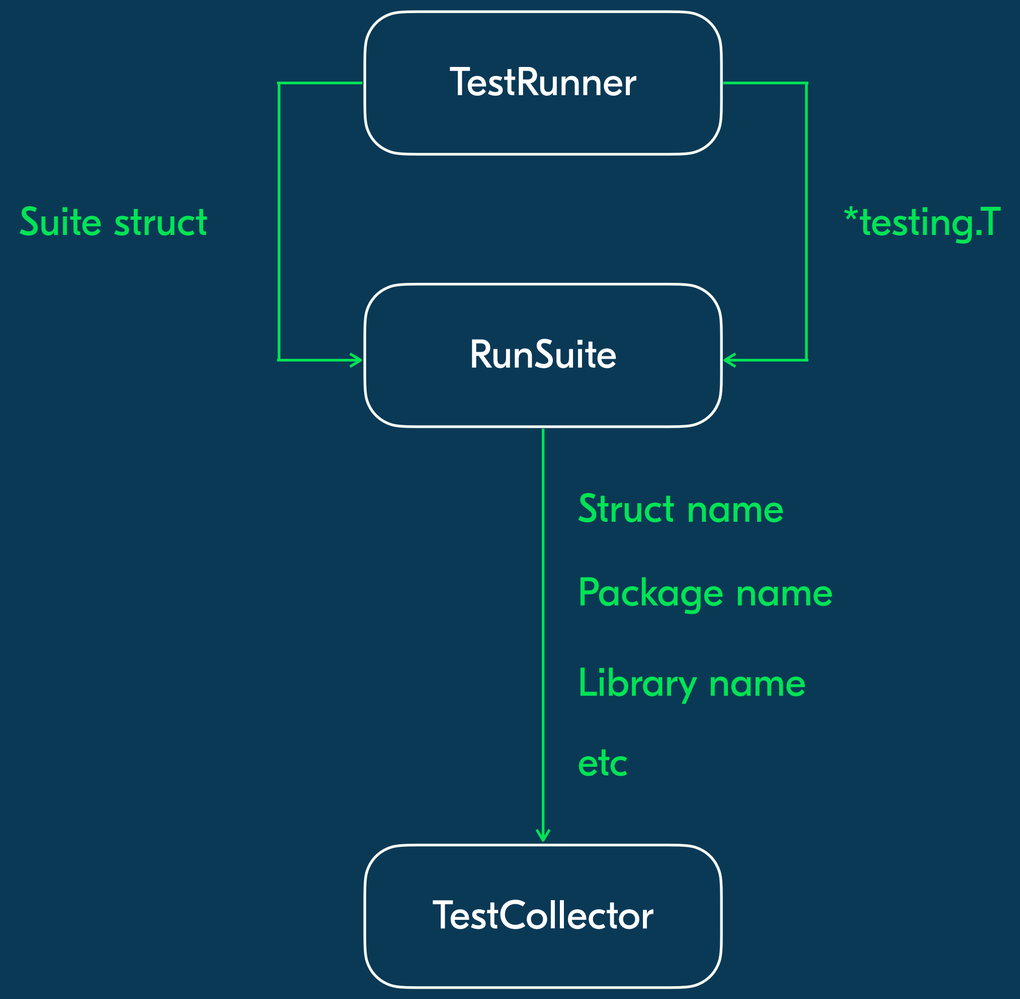
4. Вызываем метод suite.RunSuite или suite.RunNamedSuite

```go
type SuiteStruct struct {
    suite.Suite
}

func (s *SuiteStruct) BeforeEach(t provider.T) {
    // before each
}

func (s *SuiteStruct) AfterEach(t provider.T) {
    // after each
}

func (s *SuiteStruct) BeforeAll(t provider.T) {
    // before all
}

func (s *SuiteStruct) AfterAll(t provider.T) {
    // after all
}

func (s *SuiteStruct) TestFunction(t provider.T) {
    // test something
}

func TestRunner(t *testing.T) {
  mySuite := new(SuiteStruct)
  suite.RunSuite(t, mySuite)
}
```

Шаг 2.
Сбор тестов

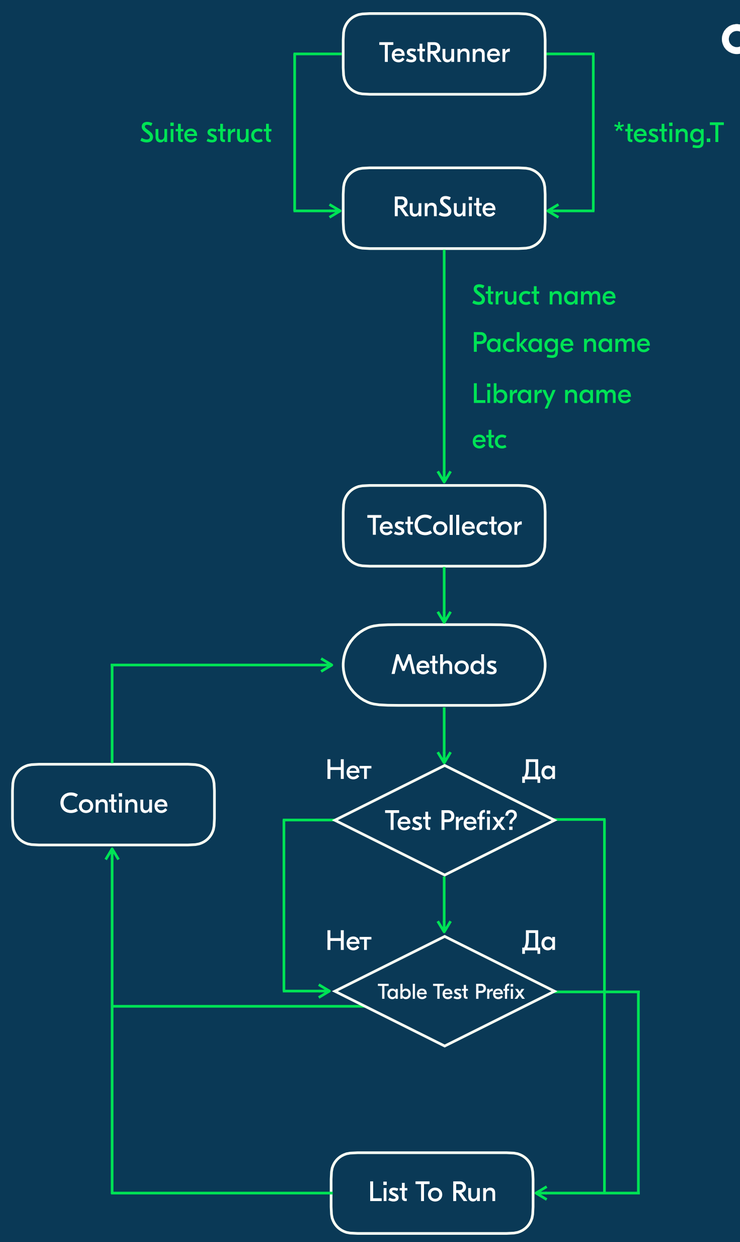
Шаг 2. Сбор тестов

1. Собираем всю информацию, которая нам доступна на этом этапе (имя структуры, пакета, версия библиотеки и тд)



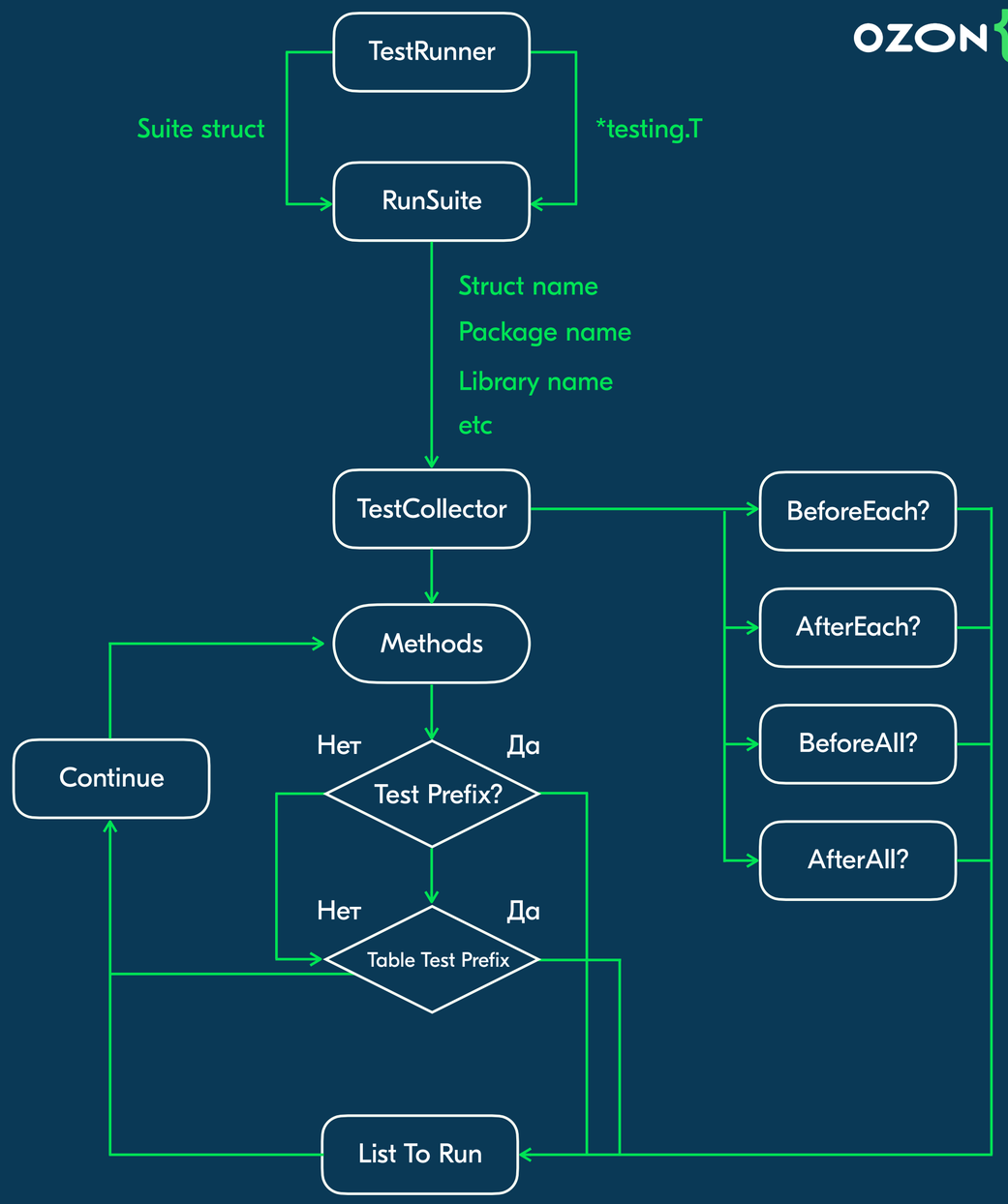
Шаг 2. Сбор тестов

2. С помощью рефлексии проходимся по всем методам структуры и с помощью регулярного выражения ищем те, которые называются как Test* или TableTest*



Шаг 2. Сбор тестов

3. Проверяем, имплементирует ли структура интерфейсы Before/After Each, Before/After All. Если да, так же их запоминаем в объект раннера

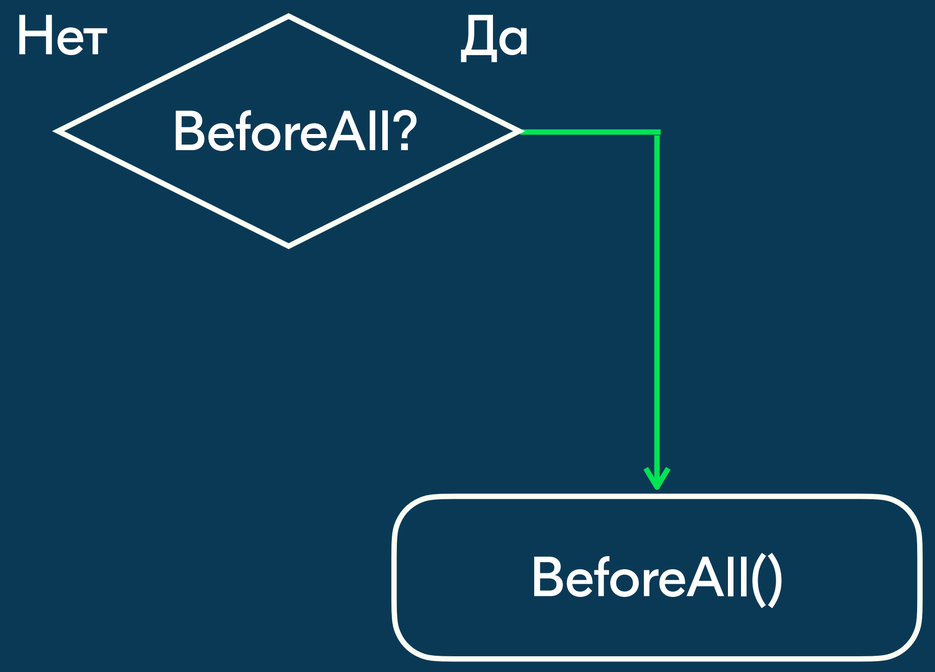


Шаг 3.
Подготавливаем запуск тестов

Шаг 3. Подготавливаем запуск тестов

OZON{ech

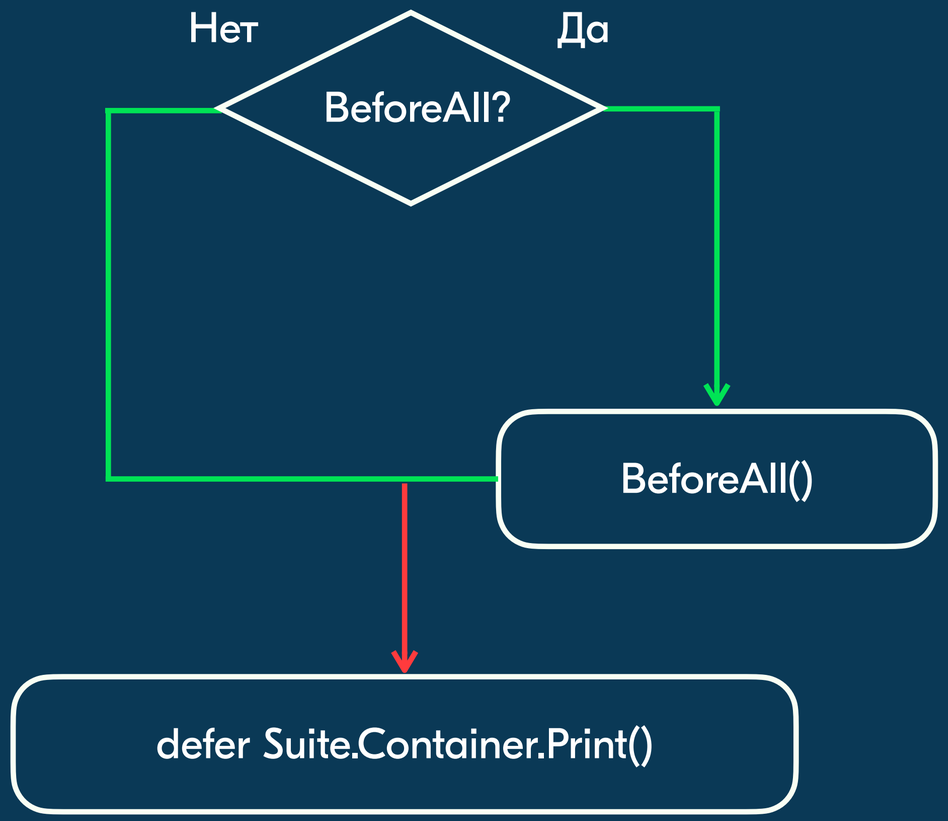
1. Запускаем BeforeAll



Шаг 3. Подготавливаем запуск тестов

OZON{ech

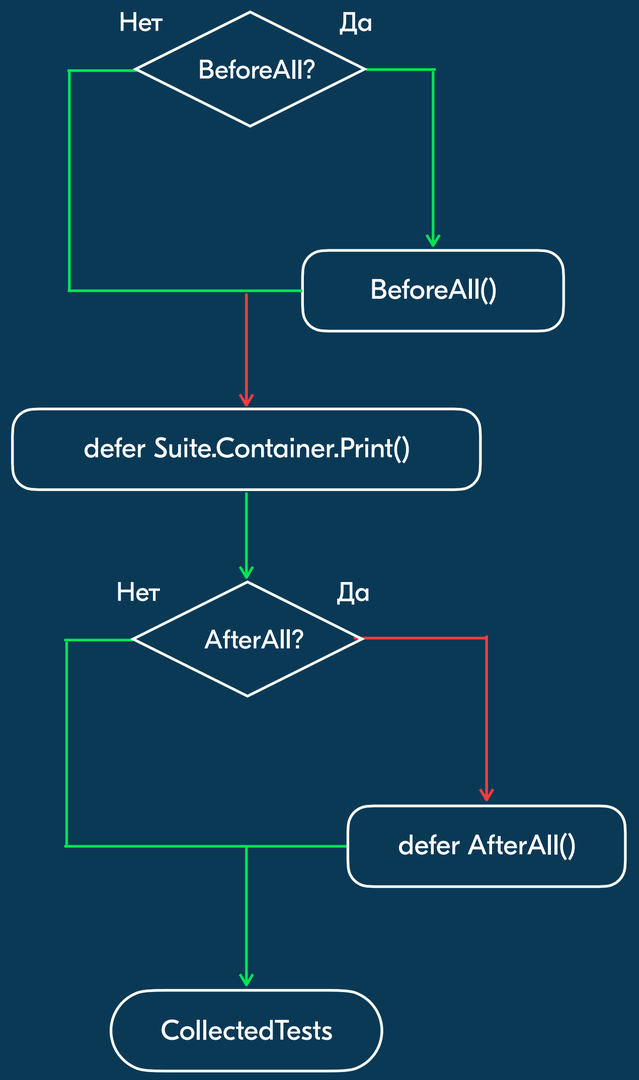
2. Через defer готовимся создавать файл uuid-container.json (уровень комплекта тестов)



Шаг 3. Подготавливаем запуск тестов

OZON{ech

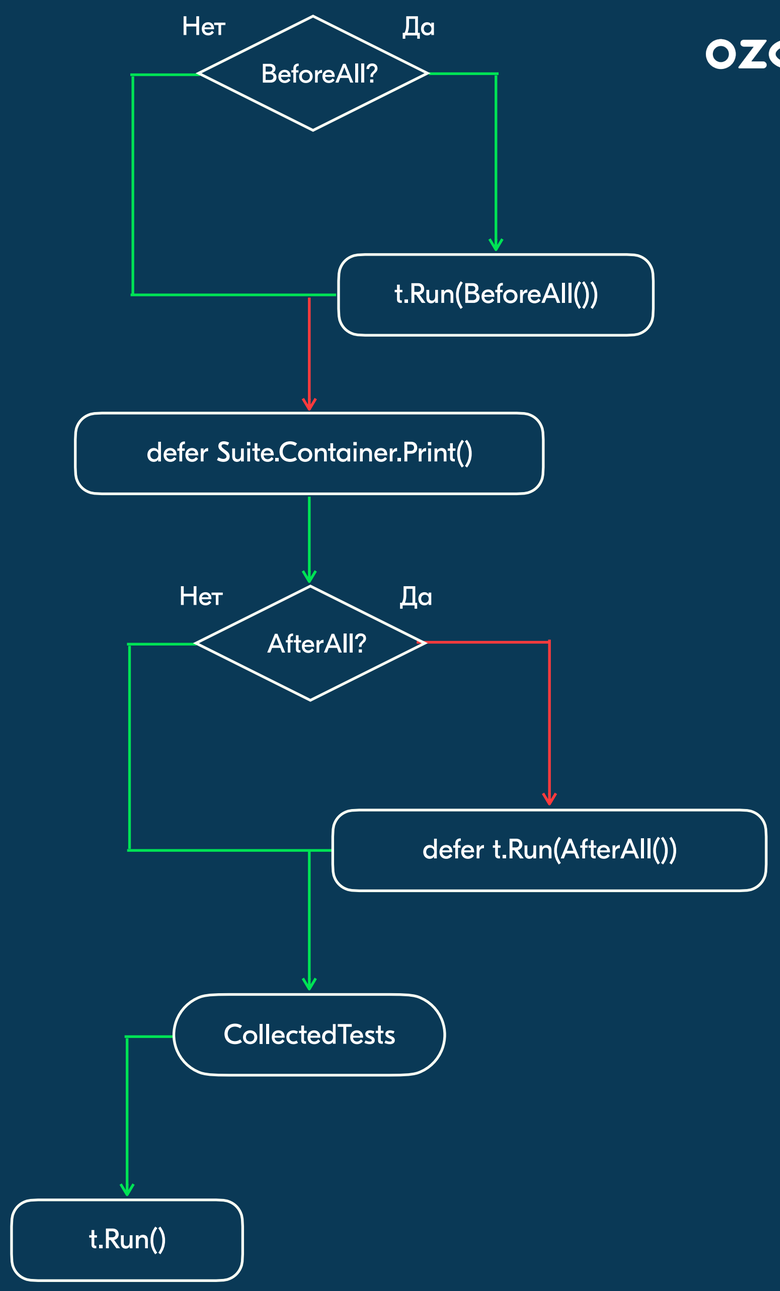
3. Через defer запускаем AfterAll



Шаг 4.
Запуск тестов

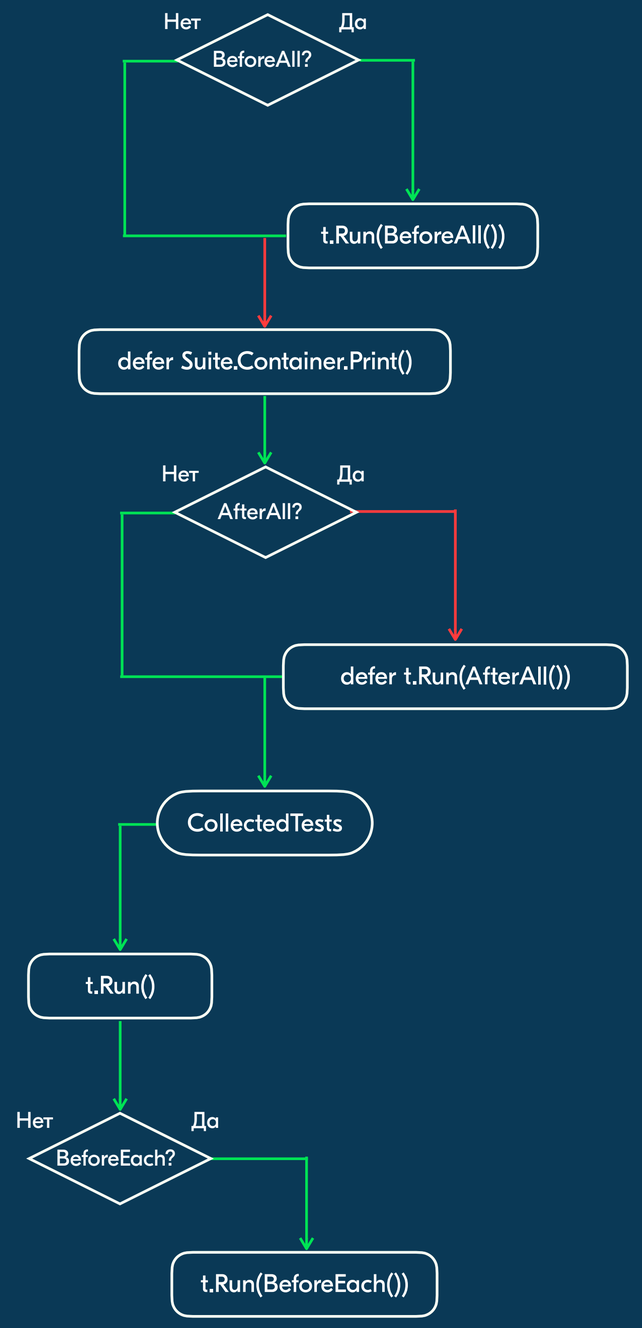
Шаг 4. Запуск тестов

1. Проходимся по всему списку собранных тестов и запускаем родительский тест для каждого



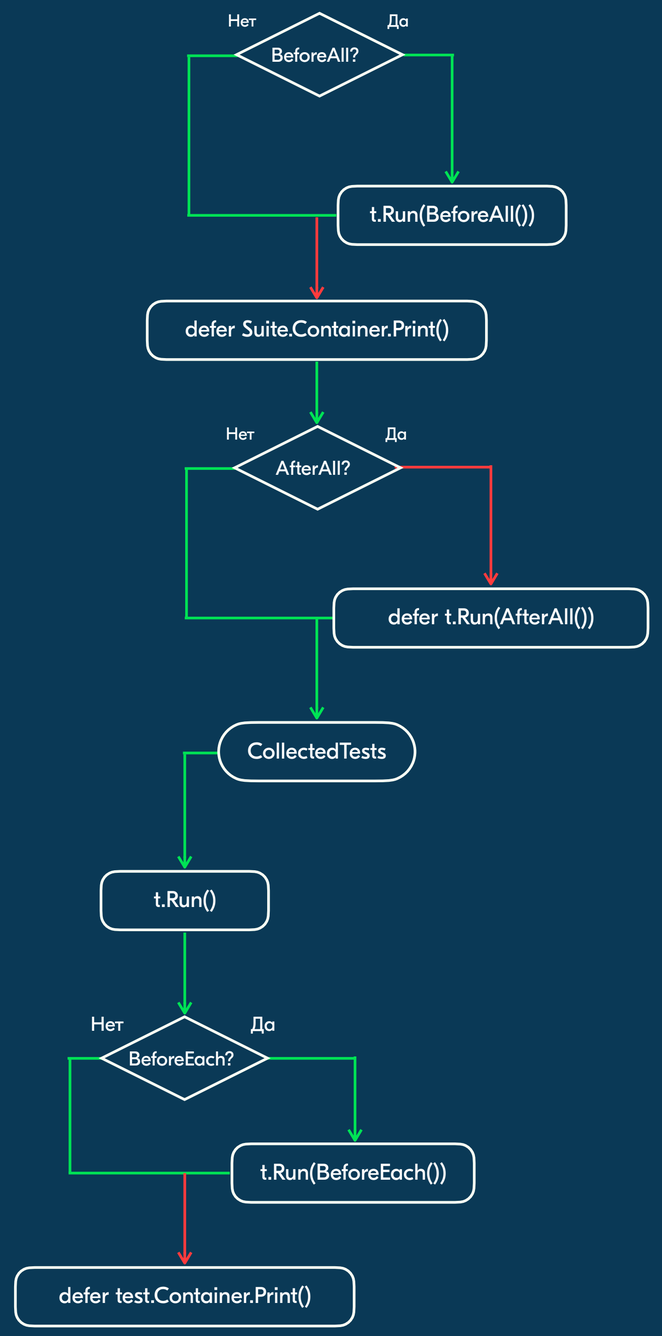
Шаг 4. Запуск тестов

2. Запускаем BeforeEach



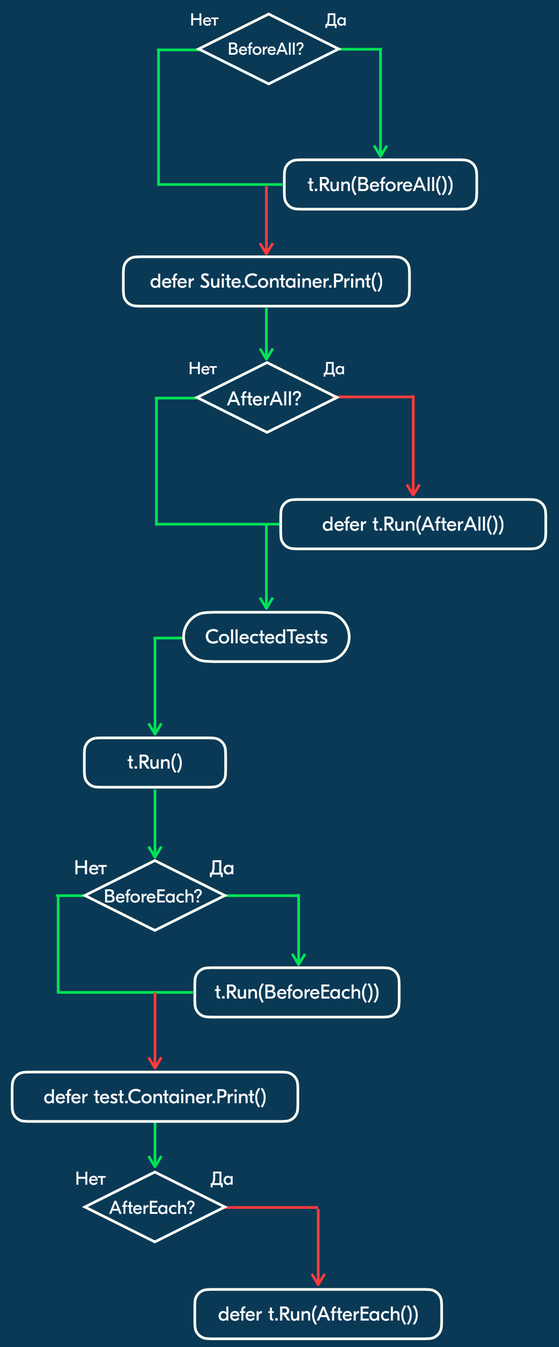
Шаг 4. Запуск тестов

3. Через defer готовимся создавать файл uuid-container.json (уровень конкретного теста)



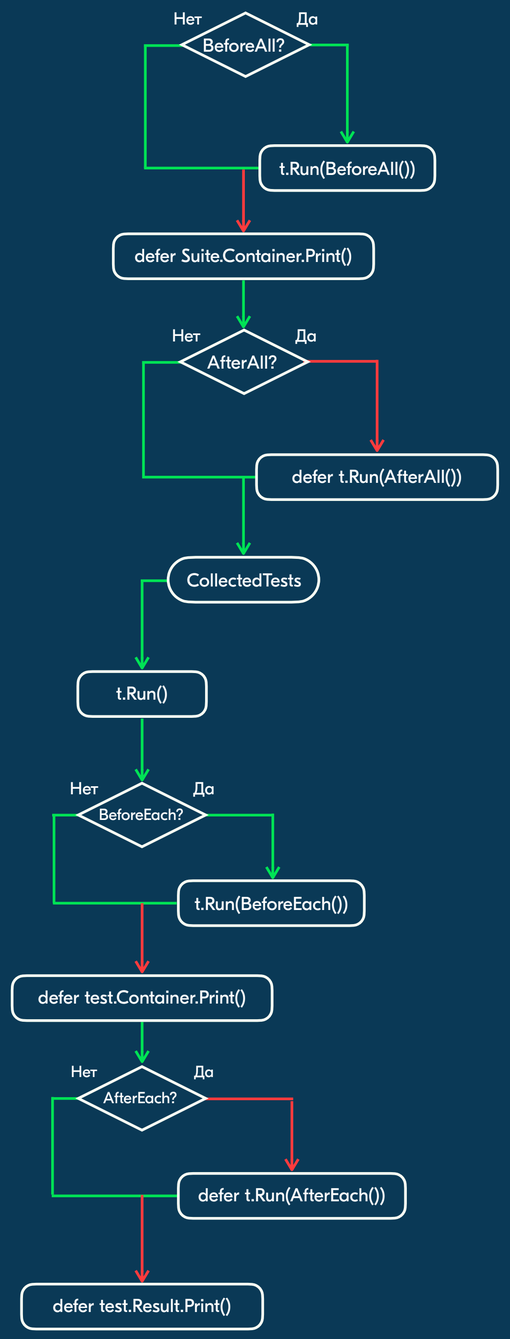
Шаг 4. Запуск тестов

4. Через Defer запускаем AfterEach



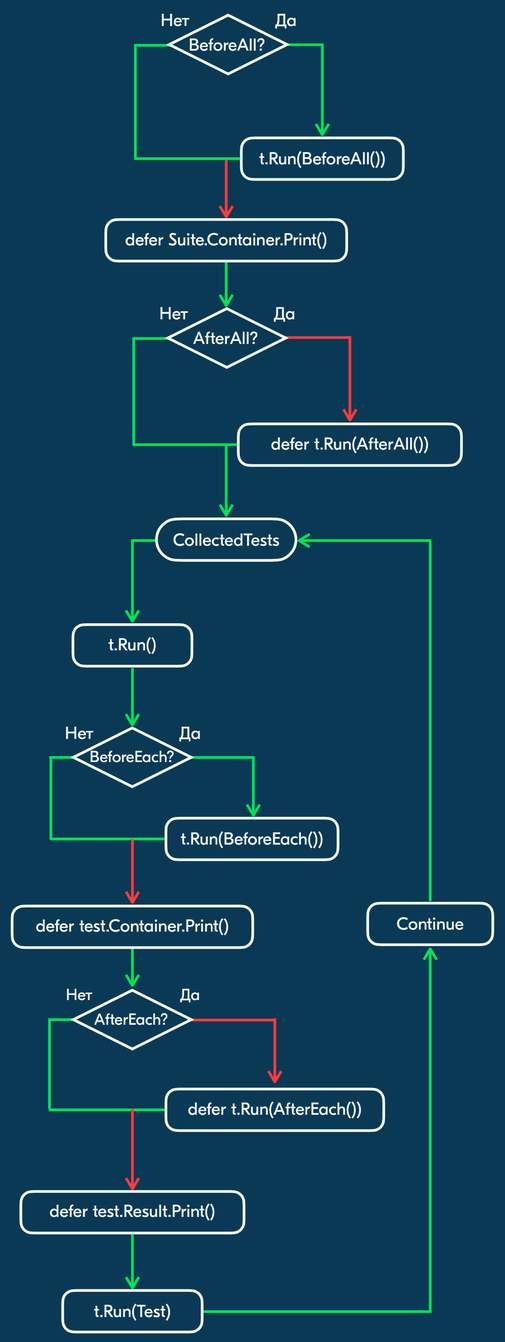
Шаг 4. Запуск тестов

5. Через Defer готовимся создавать файл uuid-result.json



Шаг 4. Запуск тестов

6. Запускаем конкретный тест



Шаг 4. Запуск тестов

ozon{tech

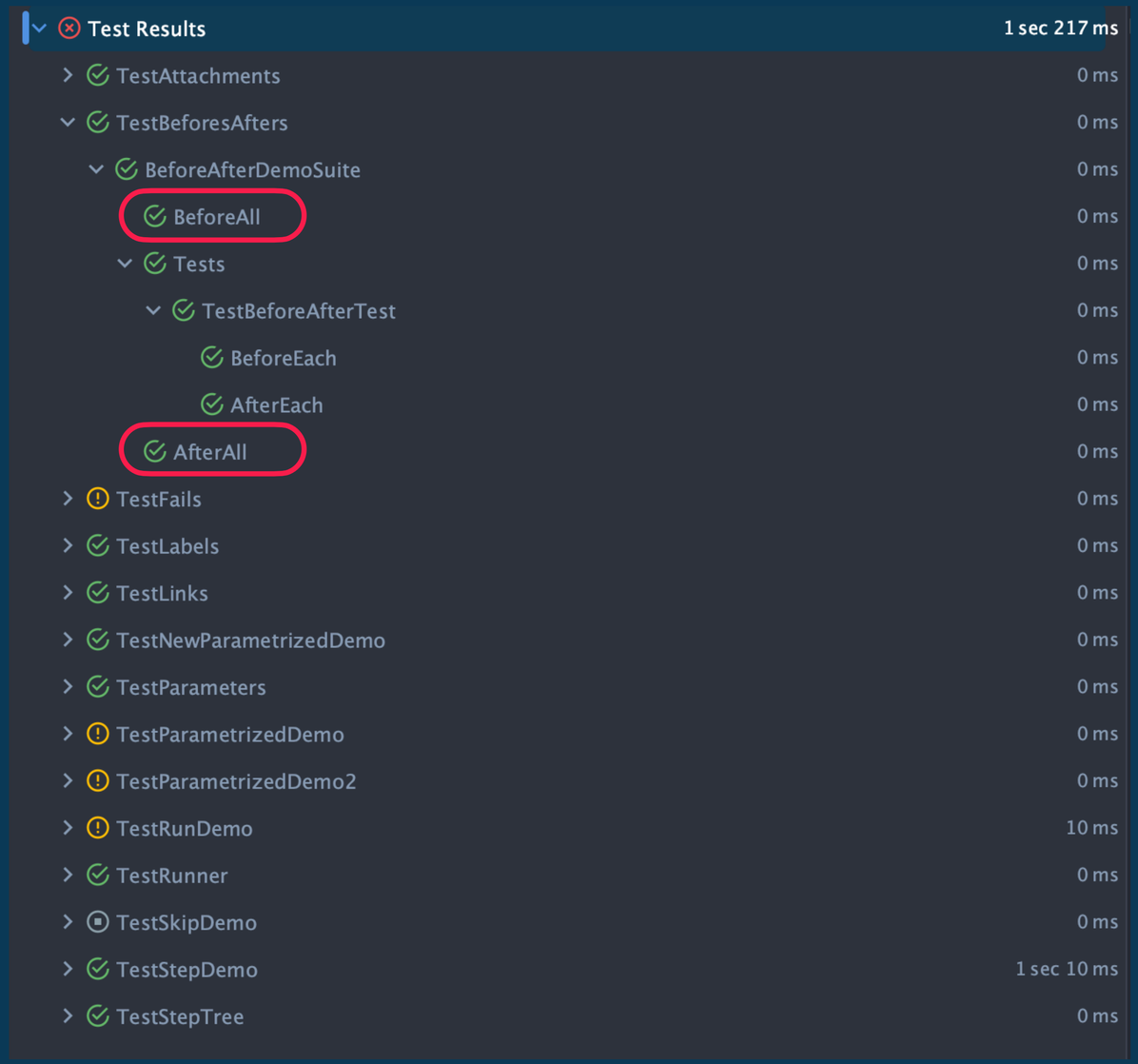
Как выглядит в IDE?

Test Results	1 sec 217 ms
TestAttachments	0 ms
TestBeforesAfters	0 ms
BeforeAfterDemoSuite	0 ms
BeforeAll	0 ms
Tests	0 ms
TestBeforeAfterTest	0 ms
BeforeEach	0 ms
AfterEach	0 ms
AfterAll	0 ms
TestFails	0 ms
TestLabels	0 ms
TestLinks	0 ms
TestNewParametrizedDemo	0 ms
TestParameters	0 ms
TestParametrizedDemo	0 ms
TestParametrizedDemo2	0 ms
TestRunDemo	10 ms
TestRunner	0 ms
TestSkipDemo	0 ms
TestStepDemo	1 sec 10 ms
TestStepTree	0 ms

Шаг 4. Запуск тестов

ozon{tech

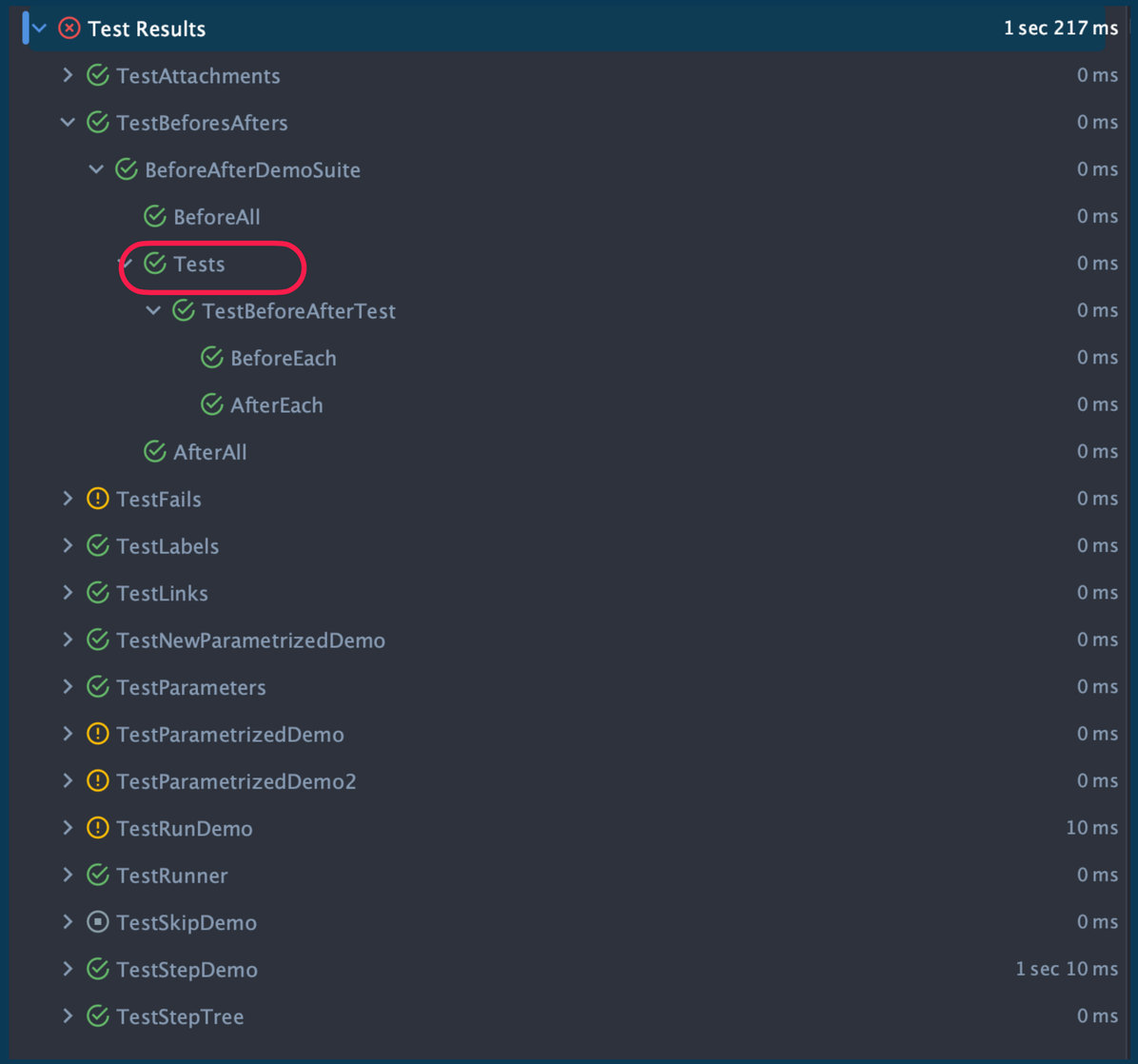
Как выглядит в IDE?



Шаг 4. Запуск тестов

ozon{tech

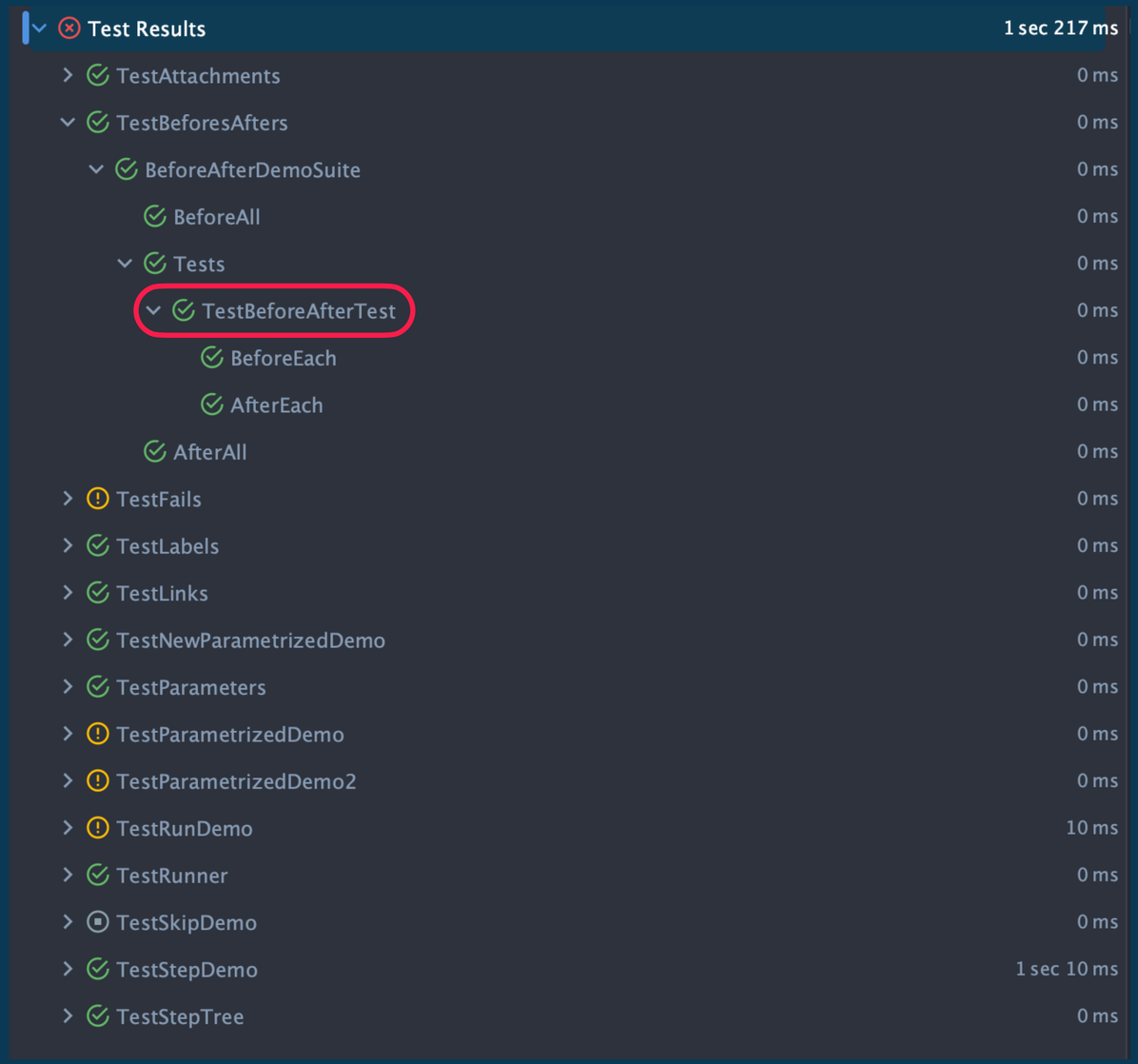
Как выглядит в IDE?



Шаг 4. Запуск тестов

ozon{tech

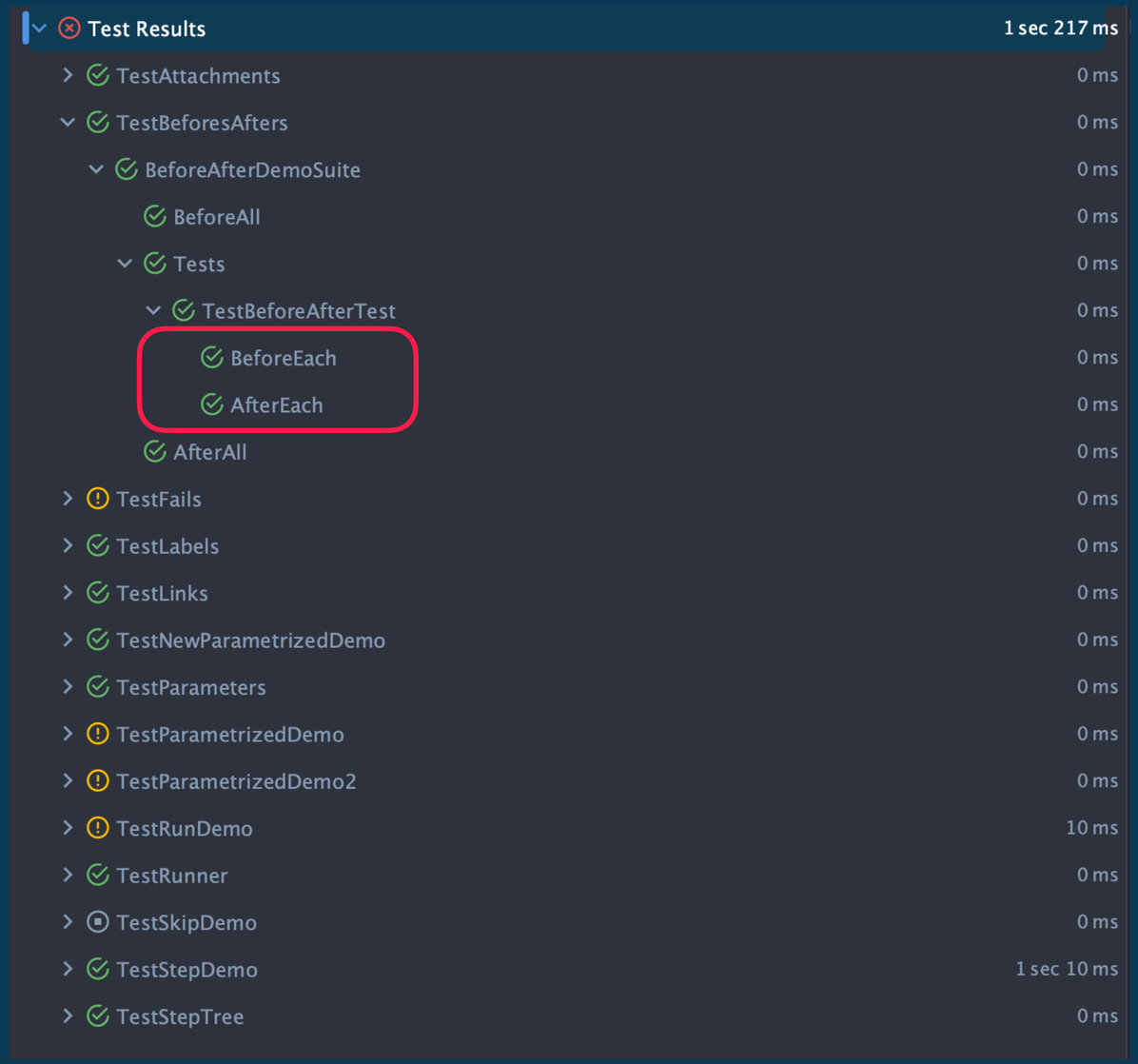
Как выглядит в IDE?



Шаг 4. Запуск тестов

ozon{tech

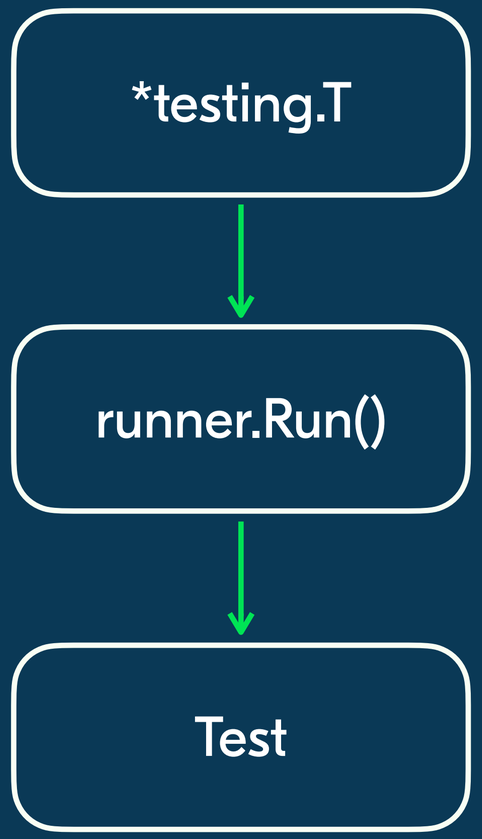
Как выглядит в IDE?



Шаг 4. Запуск тестов

OZON{ech

1. Прокидываем testing.T в функцию runner.Run
2. Собираем всю доступную информацию по тесту
3. Запускаем тест



Общая схема работы TestRunner

OZONtech

1. Набираем тесты в объект TestRunner
2. Проставляем, если требуется Before/After Each/All
3. Запускаем RunTests
4. Запускаем прокинутые before/after each/all
5. Запускаем тест
6. Создаем файлы

Планы развития

В ближайшее время хотим

OZON{ech

1. Провести масштабный рефакторинг пакета pkg/framework (версия 1.0)
2. Предоставить функционал и интерфейсы для разработки плагинов
3. Переработать механизмы сбора и запуска тестов
4. Разработать плагин для Goland для работы с Allure-Go

Выводы

Выводы

OZON{ech

1. Историю создания Allure-Go
2. Основной функционал Allure-Go и некоторые малоизвестные фичи
3. Познакомились с принципом работы сбора и запуска тестов
4. Посмотрели примеры тестов
5. Поговорили о планах на будущее

OZON tech

Спасибо за внимание

Синяев Антон, Ведущий специалист по тестированию

belomorovich@gmail.com