

# Утки и Тесты



# Содержание

- Interfaces
- Dependency injection
- Unit testing
- Integration testing
- Functional testing
- Заключение
- Ссылки

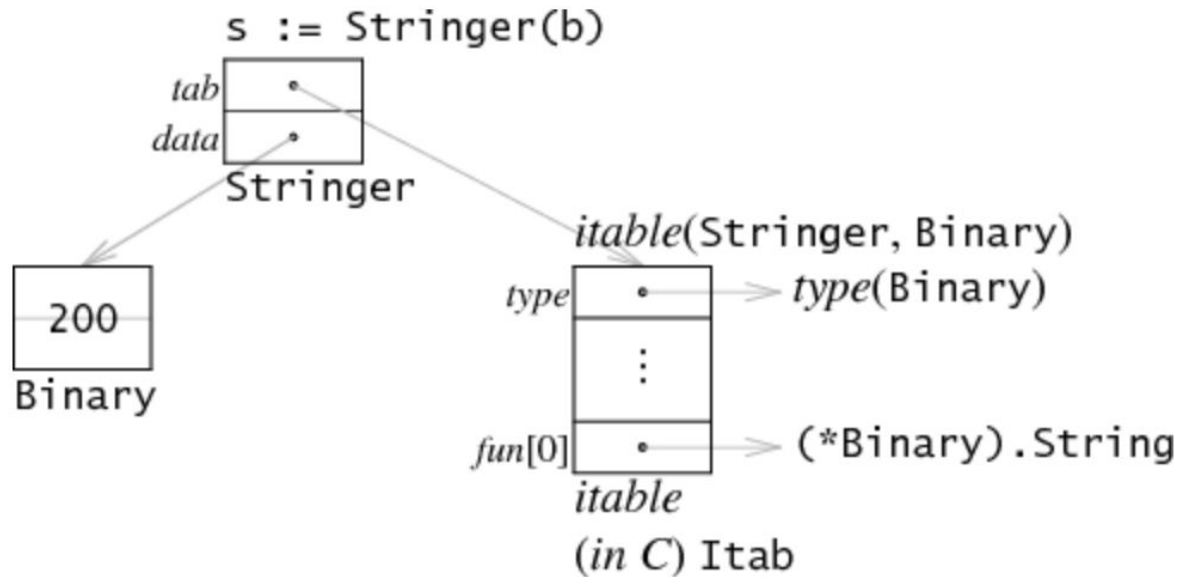
# Interfaces

```
type Binary uint64

func (i Binary) String() string {
    return strconv.Uitob64(i.Get(), 2)
}

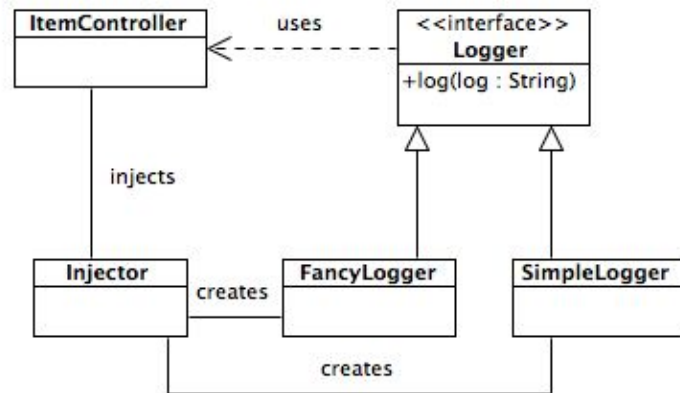
func (i Binary) Get() uint64 {
    return uint64(i)
}
```

# Interfaces



# Dependency injection

*Dependency injection* (внедрение зависимости) — компонование сущностей, таким образом, что одна сущность(зависимость) становится частью состояния другой (родительской сущности). Родительская сущность затем использует зависимость при необходимости.



# Unit

Модульное тестирование, или юнит-тестирование (англ. unit testing) — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.

Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

# Unit

- Stub
- Fake
- Spy
- Mock

# Integration

Интеграционные тесты начинаются с того места, где заканчивается модульное тестирование. В то время как модульное тестирование гарантирует, что каждый отдельный модуль системы работает изолированно, интеграционное тестирование гарантирует правильное взаимодействие различных модулей.

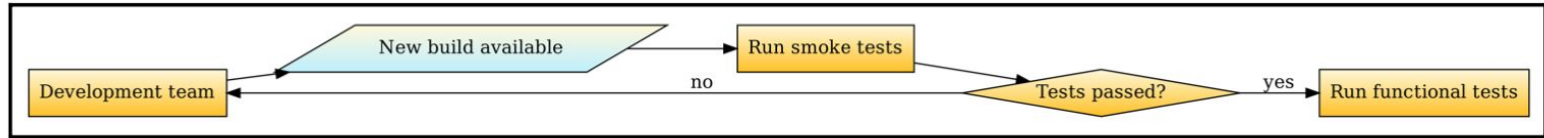


# Functional || E2E

Функциональные или сквозные тесты выводят системное тестирование на совершенно новый уровень. Основная цель функционального тестирования — убедиться, что вся система работает должным образом. С этой целью функциональные тесты предназначены для моделирования сложных сценариев взаимодействия, включающих несколько компонентов системы. Очень распространенным вариантом использования функциональных тестов является проверка сквозной правильности путем имитации перемещения пользователя по системе.

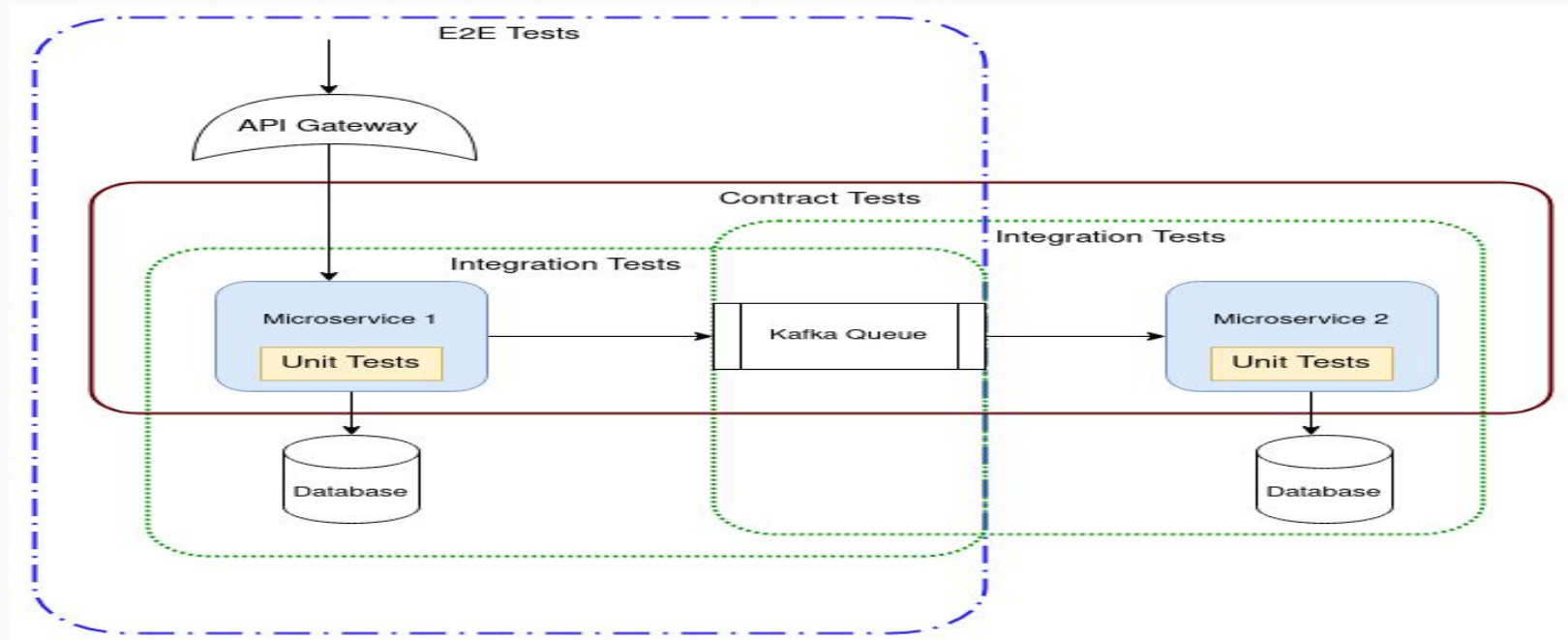
# Functional || E2E

- Smoke test



- Chaos test

# Заключение



# Ссылки

- <https://research.swtch.com/interfaces>
- <https://gobyexample.com/interfaces>
- <https://www.packtpub.com/product/hands-on-software-engineering-with-golang/9781838554491>
- [https://github.com/Zulbukharov/lectures/tree/master/interfaces\\_and\\_tests](https://github.com/Zulbukharov/lectures/tree/master/interfaces_and_tests)
- <https://github.com/stretchr/testify>
- <https://github.com/gojuno/minimock>
- <https://github.com/DATA-DOG/go-sqlmock>