

RubyによるiOSアプリのテスト

松本宗太郎 @soutaro   



The Addison-Wesley Signature Series

XUNIT TEST PATTERNS

REFACTORING
TEST CODE

GERARD MESZAROS



Foreword by Martin Fowler



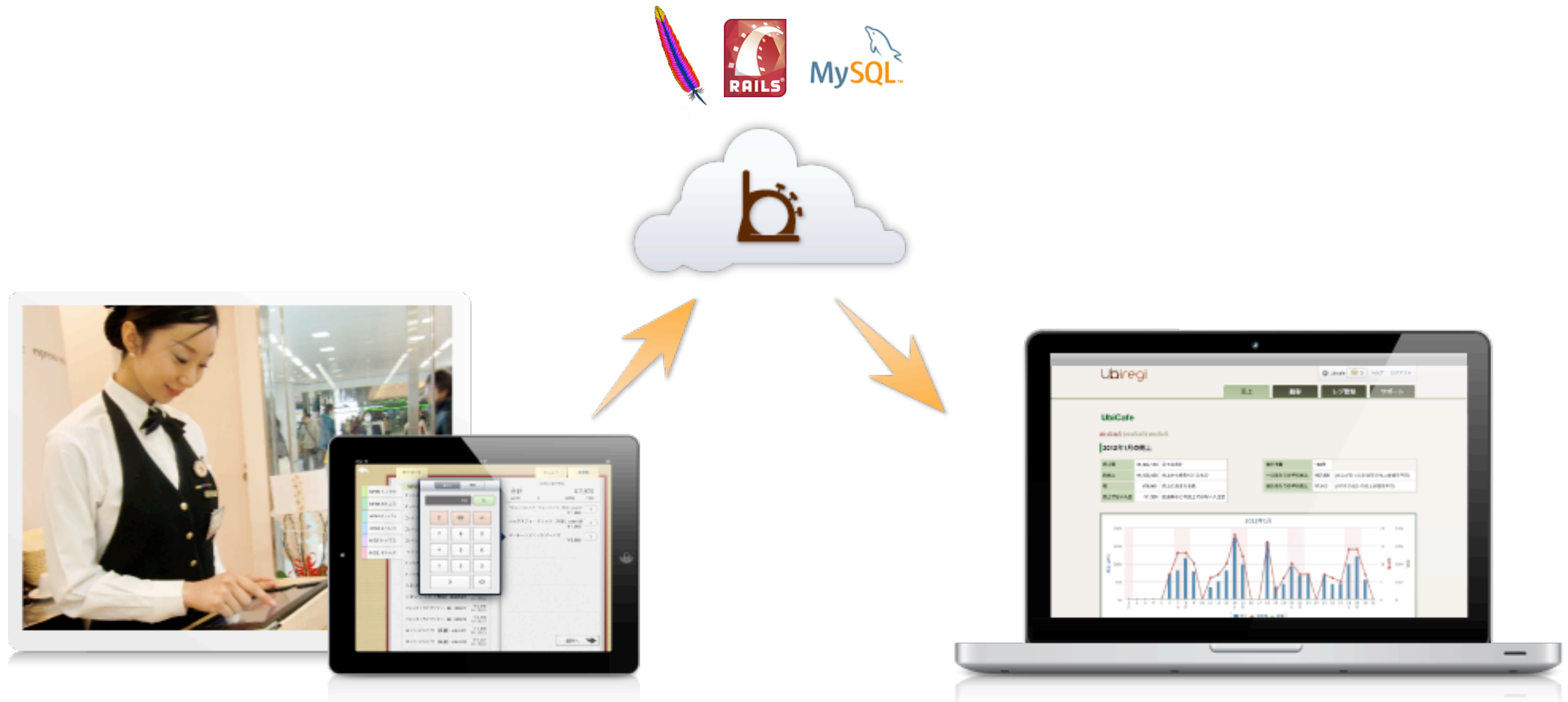
XUNIT TEST PATTERNS

MESZAROS



Addison
Wesley

インテグレーションテスト



RubyによるiOSアプリのテスト

松本宗太郎 @soutaro   

FrankによるiOSアプリの インテグレーションテスト

松本宗太郎 @soutaro   

TestingWithFrank

[Home](#)

[Getting Started](#)

[Writing Tests](#)

[Included Steps](#)

[Writing Your Own Steps](#)

[User Contributed Steps](#)

[More Information](#)

[Presentations](#)

[Screencasts](#)

[FAQ](#)

[Wire Protocol \(Frankly\)](#)

[Get Involved](#)

[Mailing Lists](#)

[Contributing](#)

Painless iOS Testing With Cucumber

Frank allows you to write structured text test/acceptance tests/requirements (using [Cucumber](#)) and have them execute against your iOS application.

An example acceptance test:

```
Feature: Login to the app
Scenario: Successful login
  Given I launch the app
  When I log in with a valid userid and password
  Then I am on the start view
```

Frank also includes a powerful “app inspector” (called [Symbiote](#)) that you can use to get detailed information on your running app.

Features

- **Easy Setup** - Getting your iOS app setup for Frank should take less than 10 minutes.
- **Record Video** - record video of your test runs to show the app in action
- **Run everywhere** - Run your tests on both the Simulator and Device
- **Integrates with CI** - Run your tests on every checkin
- **Great Community** - A strong community of helpful people

- CucumberというRubyの受け入れテストのフレームワークでテストを書けるよ！
- Viewのインスペクターが付いてくるよ！
- CIと一緒に使うと良いよ！

TestingWithFrank

[Home](#)

[Getting Started](#)

[Writing Tests](#)

[Included Steps](#)

[Writing Your Own Steps](#)

[User Contributed Steps](#)

[More Information](#)

Painless iOS Testing With Cucumber

Frank allows you to write structured text test/acceptance tests/requirements (using [Cucumber](#)) and have them execute against your iOS application.

An example acceptance test:

```
Feature: Login to the app
Scenario: Successful login
  Given I launch the app
  When I log in with a valid userid and password
```


Frankを組み込んだアプリ

RubyプログラムでViewを操作



HTTP



```
ターミナル — zsh — 80x24
> frankly_map('view:"UITextField" index:1',
               "setText:", "test@example.com")
=> [nil]
> frankly_map('view:"UITextField" index:0',
               "setText:", "topsecret")
=> [nil]
> frankly_map('button marked:"Login"',
               "touch")
=> [true]
```

Ruby

```
frankly_map('view:"UITextField" index:1',  
            "setText:", "test@example.com")
```



HTTP/JSON

```
{ "query": "view:\"UITextField\" index:1",  
  "operation": {  
    "method_name": "setText:",  
    "arguments": ["test@example.com"]  
  } }
```



iOS

```
NSArray* views = selectViewsToSelector(selector);  
for (UIView* view in views) {  
    SEL sel = NSSelectorFromString(methodName);  
    [view performSelector:sel withObject:arguments];  
}
```



```
ターミナル — zsh — 80x24

> frankly_map('view:"UITextField" index:1',
               "setText:", "test@example.com")
=> [nil]
> frankly_map('view:"UITextField" index:0',
               "setText:", "topsecret")
=> [nil]
> frankly_map('button marked:"Login"',
               "touch")
=> [true]
```




```
class Account < ActiveRecord::Base
  has_many :recipes
  has_many :api_sessions

  has_secure_password

  validates_uniqueness_of :email
  validates_presence_of :password_digest, :on => :create

  def self.authenticate(email, password)
    account = Account.where(:email => email).first
    account.try do |a|
      a.authenticate(password)
    end
  end
end
....
```



```
ターミナル — zsh — 80x24

> frankly_map('view:"UITextField" index:1',
               "setText:", "test@example.com")
=> [nil]
> frankly_map('view:"UITextField" index:0',
               "setText:", "topsecret")
=> [nil]
> frankly_map('button marked:"Login"',
               "touch")
=> [true]
```



サーバがバグってた

ソーシャルグラフのセットアップ

アカウントは作成されているか？

検索すべきデータのセットアップ

パスワードはどうなっている？

有料会員の期限

アカウント名
の重複



検索結果の検証

レシピはちゃんと
投稿されたか？


パスワードのリセット

テストごとにリセットしないと

サーバに接続できなかった場合のテスト



アプリの操作
状態のテスト



```
class Account < ActiveRecord::Base
  has_many :recipes
  has_many :api_sessions

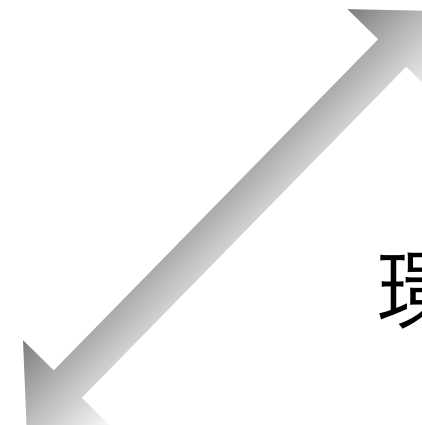
  has_secure_password

  validates_uniqueness_of :email
  validates_presence_of :password_digest, :on => :create

  def self.authenticate(email, password)
    account = Account.where(:email => email).first
    account.try do |a|
      a.authenticate(password)
    end
  end
end
....
```



テスト



環境のセットアップ
状態のテスト

```
ターミナル — zsh — 80x24

> frankly_map('view:"UITextField" index:1',
               "setText:", "test@example.com")
=> [nil]
> frankly_map('view:"UITextField" index:0',
               "setText:", "topsecret")
=> [nil]
> frankly_map('button marked:"Login"',
               "touch")
=> [true]
```

Frankでテストをする

- Frankのインストール
- Frank対応版のアプリをビルド
- アプリを起動
- Viewを調べる
- Rubyで操作してみる

\$ gem install frank-cucumber

\$ frank setup

\$ frank build

\$ frank launch

\$ frank inspect

\$ frank console

デモ



他との比較

	KIF	UIA	nativedriver	Frank
開発者	Square Inc.	Apple	Google	Pete Hodgson @ThoughtWorks
テスト実行	アプリ内	Instruments	外部	外部
テスト記述	Objective C	JS	Javaなど	Rubyなど
プロトコル	-	不明	オープン (多分)	オープン

まとめ

- ✓ サーバと通信するアプリのインテグレーションテスト
 - サーバのセットアップ、テスト
- ✓ Frankやnatedriverのような、iOSアプリからテスト実行が分離された、オープンなプロトコルのツールを使うと、楽
 - cf. UIAutomation, KIF
- ✓ FrankはHTTPでリクエスト送るだけなので、Rubyじゃなくても、PHPでもPerlでも好きな言語でクライアント書けば良い
(Java版はあるよ！)

リンク

- Frank

<http://www.testingwithfrank.com>

- ソースコード (今日中くらいにやります)

<https://github.com/soutaro/chefpad>

- Purty Social Icon Set by Purty Pixels

<http://www.purtypixels.com/purty-social-icon-set/>