



# Знакомьтесь FitNess!

Антон Куликов “Active Systems”



# Что такое FitNesse?

- FitNesse это WIKI, которая использует FIT
- FitNesse это веб сервер
- FitNesse это инструмент для тестирования приложений

# Область применения

- Функциональное
- Регресс, Смоук
- Производительность

# А также

- Юнит-тестирование
- Интеграционное
- Cross-System

# Инструмент взаимодействий

- Тестеры
- Заказчик
- Разработчики
- Бизнес аналитик

# FitNess нагледен!



PatientReviews.SettingsPage.

## GetDefaultSettings [add child]

- Test
- Edit
- Properties
- Refactor
- Where Used
- Search
- Files
- Versions
- Recent Changes
- User Guide
- Test History

► [Set Up: .PatientReviews.SettingsPage.SetUp \(edit\)](#)

[Expand All](#) | [Collapse All](#)

**MEMBER HAS REVIEW FEATURE IS TURNED ON**

**MEMBER HAS NOT SET ANY CHANGES ON SETTINGS YET**

**GET "SELECT THE REVIEW RATINGS YOU WANT SESAME TO AUTOMATICALLY PUBLISH" VALUE**

| GetSetSettings |  |
|----------------|--|
| token          | get Automatically Publish Ratings Value? |
| \$authToken    | None                                     |

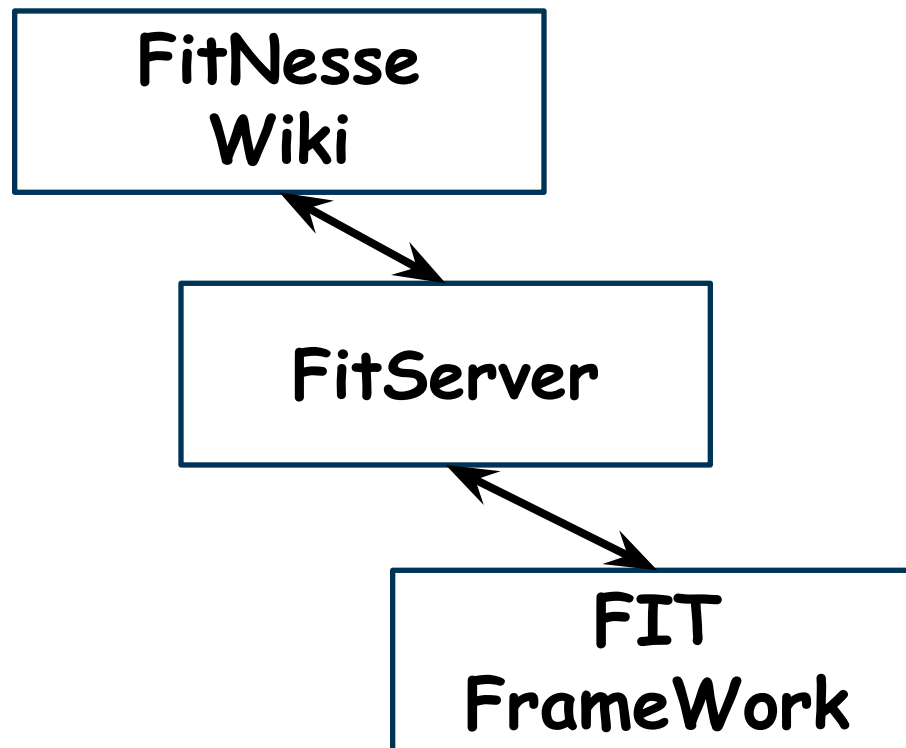
**GET "SELECT WHEN PATIENT REVIEWS SHOULD BE AUTOMATICALLY PUBLISHED" VALUE**

| GetSetSettings |                                       |
|----------------|---------------------------------------|
| token          | get When Automatically Publish Value? |
| \$authToken    | Never                                 |

**GET "SESAME HAS MY PERMISSION TO AUTOMATICALLY MARK REVIEWS WITH 'BAD WORDS' VALUE**

| GetSetSettings |                              |
|----------------|------------------------------|
| token          | get Bad Words setting Value? |
| \$authToken    | Yes                          |

# Простая схема работы





Suite

Edit

Properties

Refactor

Where Used

Search

Files

Versions

Recent Changes

User Guide

Test History

## SendingReminders [add child]

### Contents:

- [TestsConfirmationDependenceSending](#)
- [TestsForCheckSchedulingAndTime2Send](#)
- [TestsForEmailRemindersSending](#)
- [TestsForSmsRemindersSending](#)
- [TestsForVoiceRemindersSending](#)

[Front Page](#) | [User Guide](#)  
[root](#) (for global !path's, etc.)



# Платформы

- Java
- .NET
- C++
- Delphi
- Python
- Ruby
- Smalltalk
- Perl



Как это работает?  
К примерам!

> #

# 0 fixture

- Column Fixture
- Row Fixture

# Пример с Column Fixture

- Используется для проверки конкретного еденичного значения
- True /false
- Точное ожидаемое значение

# Создаем таблицу

```
!| info.fitness.fixturegallery.ColumnFixtureTest |  
| firstPart | secondPart | together? | totalLength? |  
| Hello | World | Hello, World | 10 |  
| Houston | We Have a Problem | Houston, We Have a  
Problem | 24 |
```

# Все не так сложно



## SimpleTest [\[add child\]](#)

Test

Edit

Properties

Refactor

Where Used

info.fitness.fixturegallery.ColumnFixtureTest

| firstPart | secondPart        | together?                  | totalLength? |
|-----------|-------------------|----------------------------|--------------|
| Hello     | World             | Hello, World               | 10           |
| Houston   | We Have a Problem | Houston, We Have a Problem | 24           |

# Успешный тест

info.fitness.fixturegallery.ColumnFixtureTest

| firstPart | secondPart        | together?                  | totalLength? |
|-----------|-------------------|----------------------------|--------------|
| Hello     | World             | Hello, World               | 10           |
| Houston   | We Have a Problem | Houston, We Have a Problem | 24           |

# Неуспешный тест

info.fitness.fixturegallery.ColumnFixtureTest

| firstPart | secondPart        | together?  | totalLength?       |
|-----------|-------------------|--|--------------------|
| Hello     | World             | Hello, World   | 10                 |
| Houston   | We Have a Problem | [Houston, We Have a Problem] expected [Houston, We Have Problem] | [24] expected [30] |



# Код fixture

```
package info.fitnessse.fixturegallery;

import fit.ColumnFixture;

public class ColumnFixtureTest extends ColumnFixture {
    public String firstPart;
    public String secondPart;
    private int length;
    public String together() {
        length=firstPart.length()+secondPart.length();
        return firstPart+ ", "+secondPart;
    }
    public int totalLength() {
        return length;
    }
}
```

# Пример с Row Fixture

- Используется для проверки массива данных

# Табличное представление



## SimpleTest2 [\[add child\]](#)

|            |  |         |
|------------|--|---------|
| Test       | info.fitnessse.fixturegallery.RowFixtureTest |         |
| Edit       | fname  | lname   |
| Properties | Anton  | Kulikov |
| Refactor   | Vasya  | Pupkin  |

# Простой класс

```
package info.fitnessse.fixturegallery;
import java.util.ArrayList;
import java.util.List;

public class Person{
    public String fname;
    public String lname;

    public Person(){
    }
    public Person (String fname, String lname) {
        this.fname = fname;
        this.lname = lname;
    }

    public static List<Person> persons=new ArrayList<Person>();
    public static void addPerson(String fname, String lname){
        persons.add(new Person(fname, lname));
    }
}
```

# Код самой fixture

```
package info.fitnessse.fixturegallery;

import fit.RowFixture;

public class RowFixtureTest extends RowFixture {
    public Class<?> getTargetClass() {
        return Person.class;
    }
    public Object[] query() throws Exception{
        Person.addPerson("Anton", "Kulikov");
        Person.addPerson("Andrey", "Genievskiy");
        return Person.persons.toArray();
    }
}
```

# Результат выполнения

## SimpleTest2

TEST RESULTS [\[history\]](#)

Assertions: 2 right, 2 wrong, 0 ignored, 0 exceptions

info.fitness.fixturegallery.RowFixtureTest

fname

lname

Anton

Kulikov

Vasya *missing*

Pupkin

Andrey *surplus*

Genievskiy

# Пример из жизни

- Тестирование Веб-Сервисов
- Оформление теста

# Страница теста в wiki



PatientReviews. DoNotPublishReviews.

## GetDoNotPublishReview

TEST RESULTS [\[history\]](#)



[Output](#)  
[Captured](#)

Test: **Assertions: 64 right, 0 wrong, 0 ignored, 0 exceptions**

▶ [Set Up: .PatientReviews.DoNotPublishReviews.SetUp \(edit\)](#) [Expand All](#) | [Collapse All](#)

**MEMBER HAS REVIEWS FEATURE**

**ONLY ONE (FIRST OFFICE) OFFICE ENABLED FOR REVIEW**

**MEMBER ENABLED AND SET FOLLOW SETTINGS:**

▶ [Set settings section](#) [Expand All](#) | [Collapse All](#)

**FOLLOW REVIEWS SUBMITTED BY PATIENTS**

▶ [Fill DB with reviews](#) [Expand All](#) | [Collapse All](#)

**CALL SERVICE GETREVIEWS[?] AND CHECK RESPONSE RESULTS**

| Query: Get Reviews | SauthToken->[s7c8tKuNa8wJ5RKnyWcNw==] | do not publish |                |               |          |            |
|--------------------|---------------------------------------|----------------|----------------|---------------|----------|------------|
| Title              | Comment                               | Rating         | State          | Office Name   | Category | isFeatured |
| Good dentist       | Thank you                             | 4              | do not publish | First Office  | None     | false      |
| Ftitle15           | comment15                             | 3              | do not publish | Second Office | None     | false      |

▶ [Tear Down: .PatientReviews.DoNotPublishReviews.TearDown \(edit\)](#) [Expand All](#) | [Collapse All](#)



# Fixture

```
// -----Constructors-----
public GetReviews(String token, String which) {}
// -----

public void setToken(String token) {}
public void setWhich(String which) {}
public void setSortBy(String sortBy) {}
public void setOffset(Integer offset) {}
public void setLength(Integer length) {}

private List<Object> list(Object... objects) {}

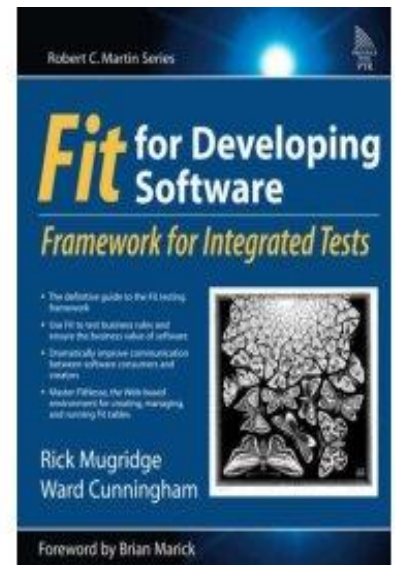
private String getLocationNameByOfficeId(Long off_id) {}

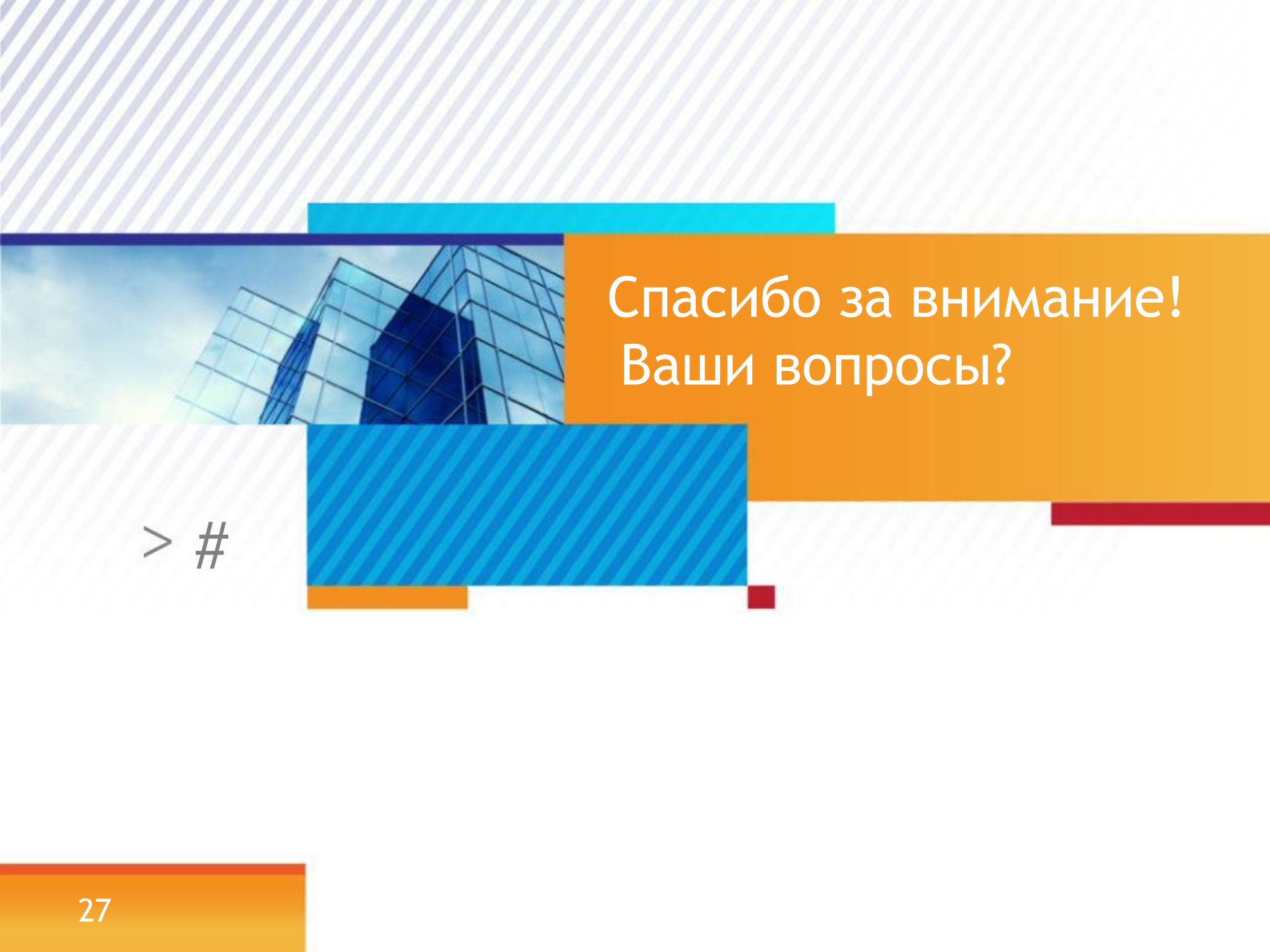
private String getCategoryNameById(Byte categoryId) throws InvalidTokenException {}

public List<Object> query() throws InvalidTokenException {
    List<Object> table = new ArrayList<Object>();
    ReviewsResult reviewsResult = reviewsWS.getReviews(token, which, sortBy, offset, length);
    List<Review> responseReviews = reviewsResult.getReview();
    for (int i = 0; i < responseReviews.size(); i++) {
        Review responseReview = responseReviews.get(i);
        List<Object> row = new ArrayList<Object>();
        row.add(list("Title", responseReview.getTitle()));
        row.add(list("Comment", responseReview.getComment()));
        row.add(list("Rating", responseReview.getRating()));
        row.add(list("State", responseReview.getState()));
        row.add(list("Office Name", getLocationNameByOfficeId(responseReview.getOfficeID())));
        row.add(list("Category", getCategoryNameById(responseReview.getCategoryID())));
        row.add(list("isFeatured", responseReview.isFeatured()));
        table.add(row);
    }
    return table;
}
```

# Ресурсы

- Сайт FitNesse homepage
  - <http://fitnesse.org/>
  
- FitNesse Yahoo Group
  - <http://tech.groups.yahoo.com/group/fitnesse/>
  
- Книга Fit for Developing Software:  
Framework for Integrated Tests
  - Robert C. Martin Series





Спасибо за внимание!  
Ваши вопросы?

> #