

Автотестирование веб-сервиса с Ruby и Rspec

Игорь Любин



О себе

- ▶ Игорь Любин
- ▶ Канд. физ.-мат. наук
- ▶ Тестировщик с 1.12.2007
- ▶ SQAdays 3, 5, 8, 9, 10!

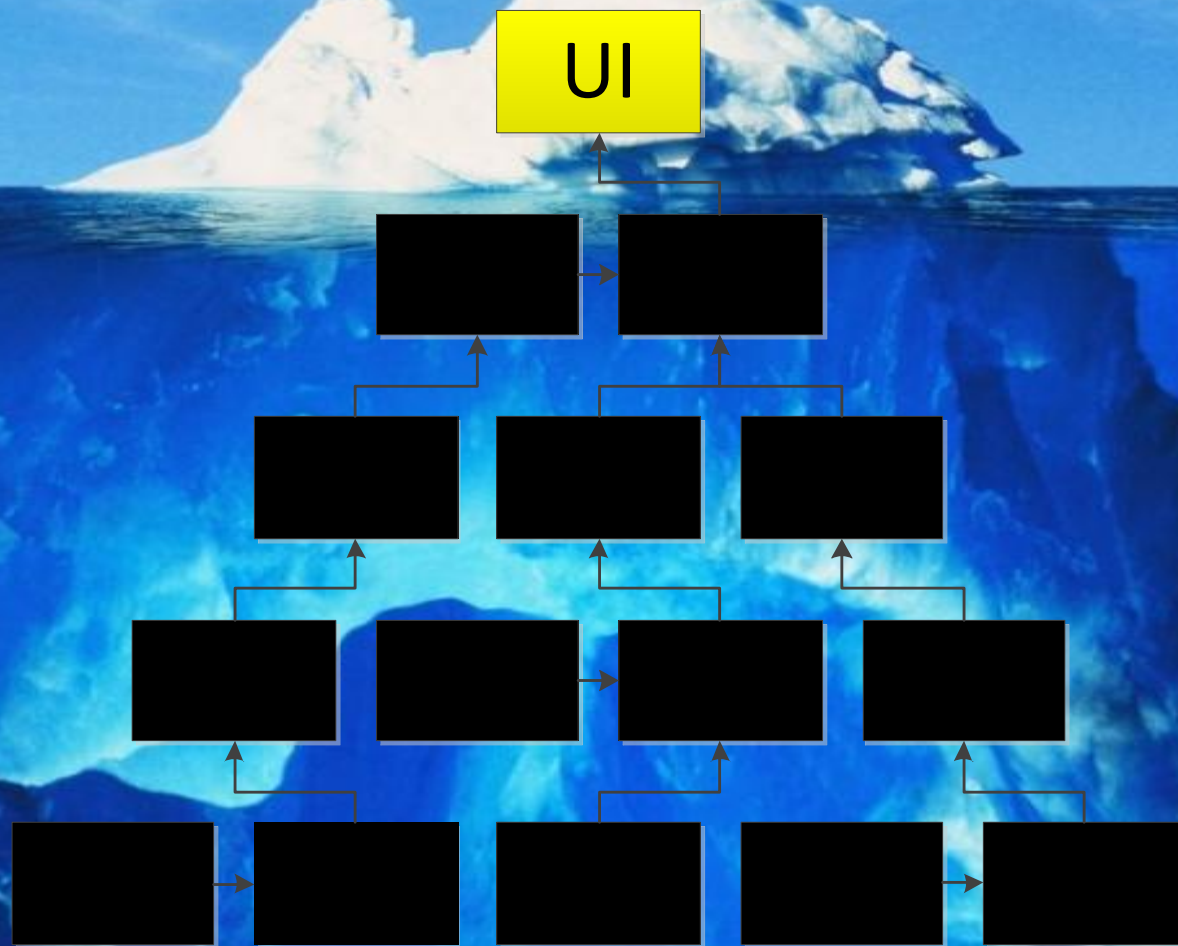


undev.ru

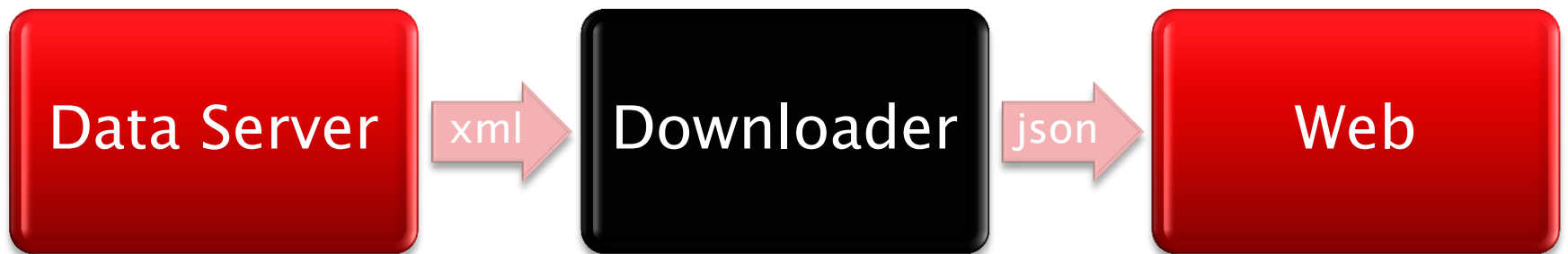


Цель доклада

- ▶ Рассказать о своей работе
- ▶ На примере автоматизации веб-сервиса
- ▶ Получить обратную связь



Постановка задачи



Автоматизация

Язык

Библиотеки

Драйвер

Фреймворк

Из презентации А. Баранцева

Автоматизация. Что есть?

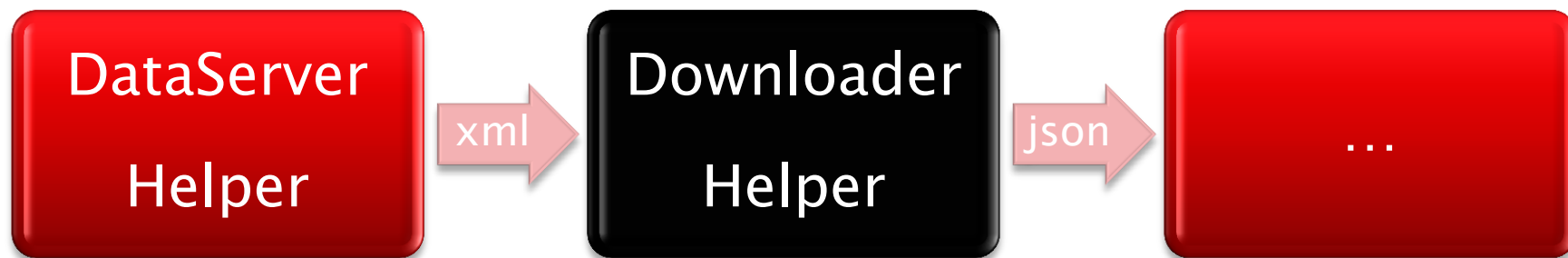
Ruby

Webrick
Iconv

Helpers

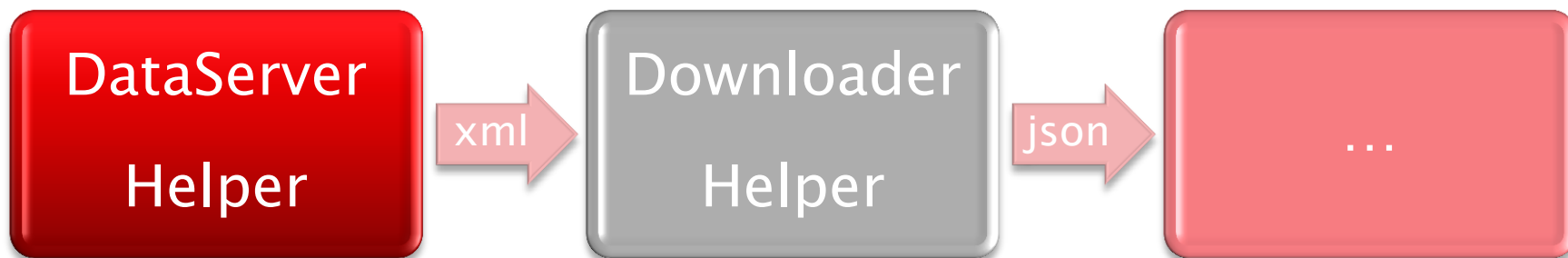
Rspec

Описание тестируемых сервисов



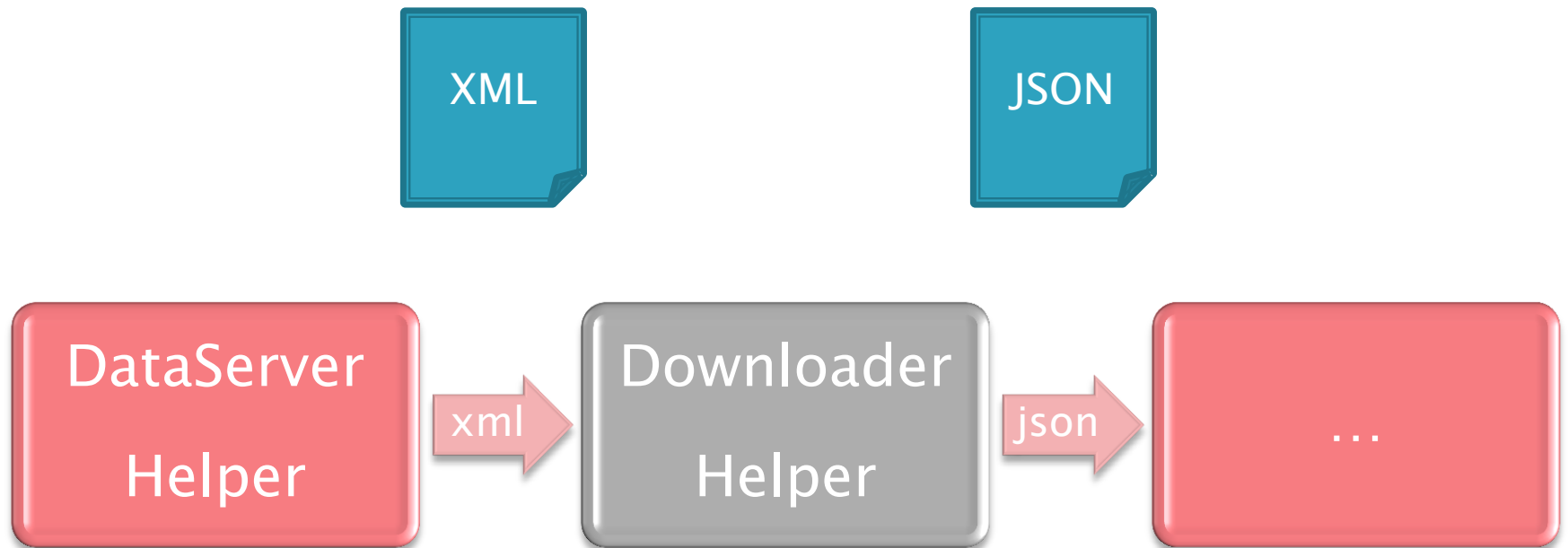
Запуск web-сервера

```
WEBrick::HTTPServer.new(:DocumentRoot=>dir).start
```



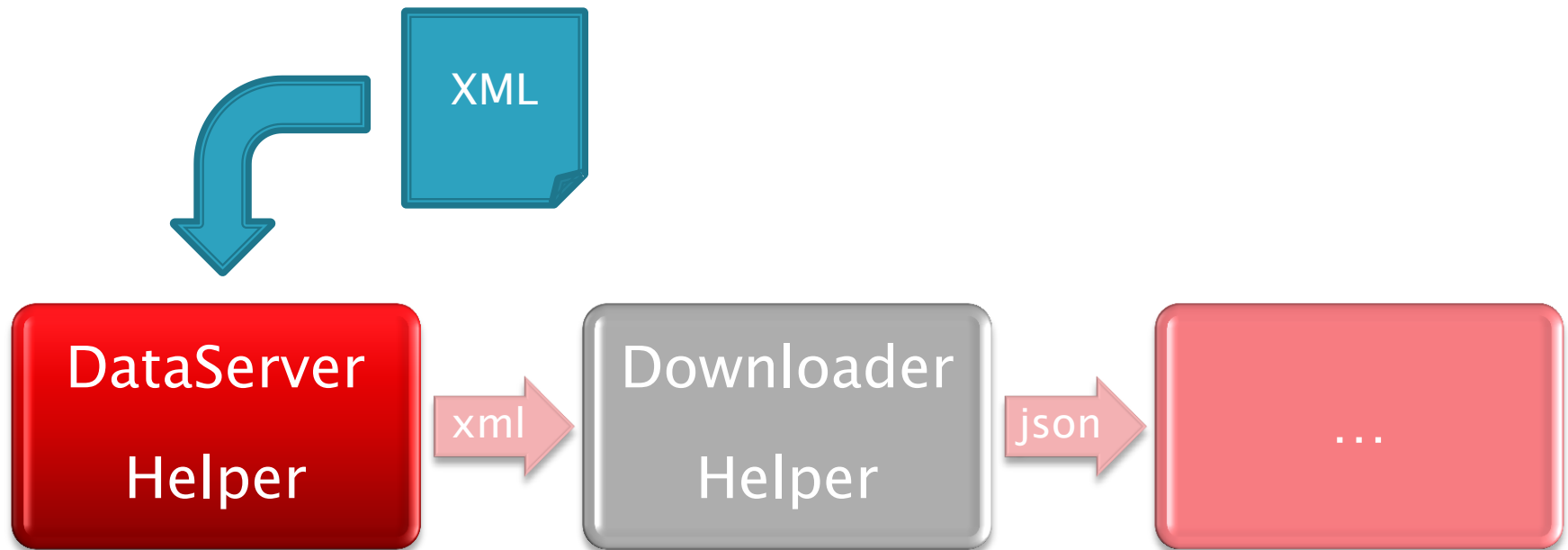
```
@data_server.start
```

Подготовка данных



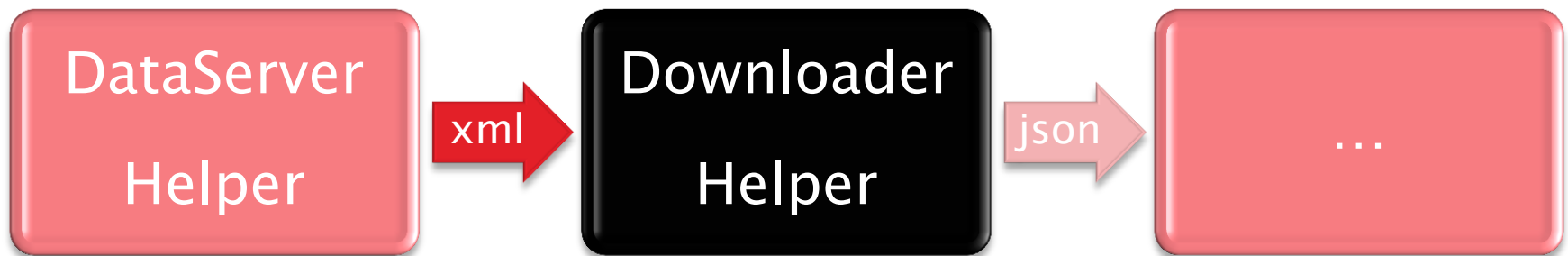
```
schedule = Schedule.new(Time.now + 1.day)
```

Создание расписания



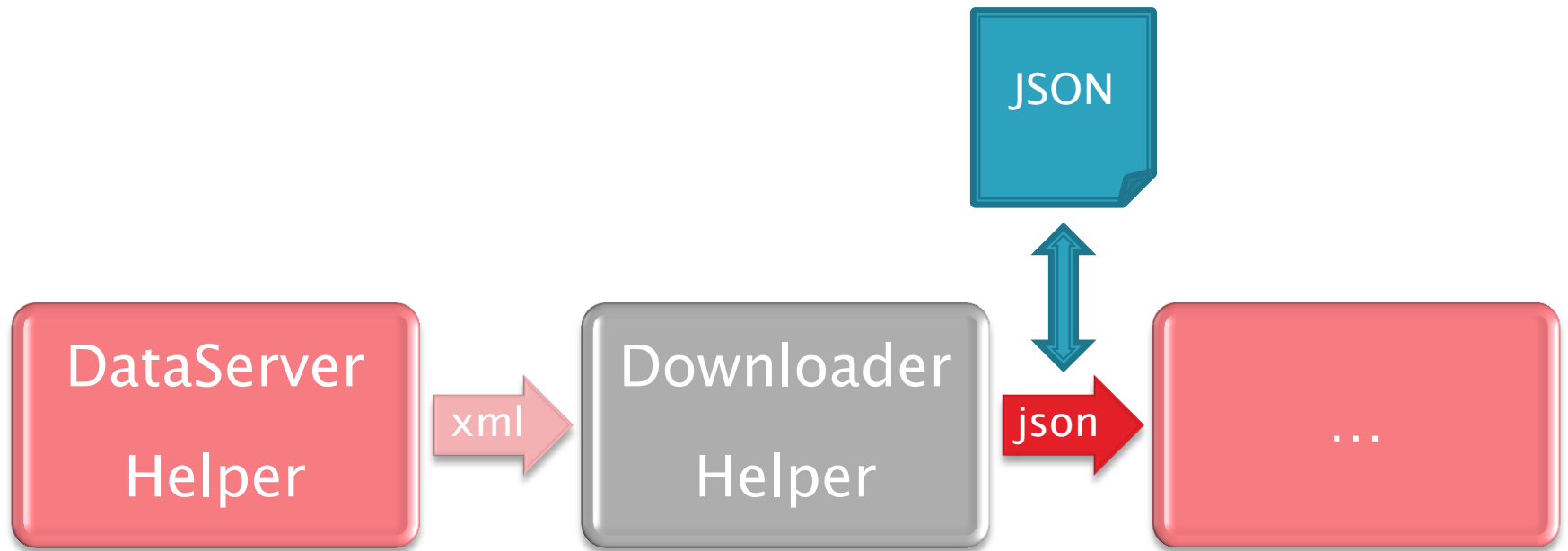
```
@data_server.create_schedule(schedule)
```

Скачивание расписания



```
act_schedule = @downloader.download_schedule(schedule)
```

Проверка



```
act_schedule.should == schedule.expected
```


Rspec

```
describe "downloader" do
  before :all do
    @data_server = DataServerHelper.new
    @downloader = DownloaderHelper.new
    @data_server.start
  end
  before :each do
    @data_server.delete_all_schedules
    @downloader.clear_jobs_queue
  end
  it "download schedule for tomorrow" do
    schedule = Schedule.new(Time.now + 1.day)
    @data_server.create_schedule(schedule)
    act_schedule = @downloader.download_schedule(schedule)
    act_schedule.should == schedule.expected
  end
  after :all do
    @data_server.stop
  end
end
```

Rspec

```
describe "downloader" do
  before :all do
    @data_server = DataServerHelper.new
    @downloader = DownloaderHelper.new
    @data_server.start
  end

  before :each do
    @data_server.delete_all_schedules
    @downloader.clear_jobs_queue
  end

  it "download schedule for tomorrow" do
    schedule = Schedule.new(Time.now + 1.day)
    @data_server.create_schedule(schedule)
    act_schedule = @downloader.download_schedule(schedule)
    act_schedule.should == schedule.expected
  end

  after :all do
    @data_server.stop
  end
end
```

Rspec

```
describe "downloader" do
  before :all do
    @data_server = DataServerHelper.new
    @downloader = DownloaderHelper.new
    @data_server.start
  end
  before :each do
    @data_server.delete_all_schedules
    @downloader.clear_jobs_queue
  end
  it "download schedule for tomorrow" do
    schedule = Schedule.new(Time.now + 1.day)
    @data_server.create_schedule(schedule)
    act_schedule = @downloader.download_schedule(schedule)
    act_schedule.should == schedule.expected
  end
  after :all do
    @data_server.stop
  end
end
```

Rspec

```
describe "downloader" do
  before :all do
    @data_server = DataServerHelper.new
    @downloader = DownloaderHelper.new
    @data_server.start
  end
  before :each do
    @data_server.delete_all_schedules
    @downloader.clear_jobs_queue
  end
  it "download schedule for tomorrow" do
    schedule = Schedule.new(Time.now + 1.day)
    @data_server.create_schedule(schedule)
    act_schedule = @downloader.download_schedule(schedule)
    act_schedule.should == schedule.expected
  end
  after :all do
    @data_server.stop
  end
end
```

Rspec отчет

```
$ rspec spec -f doc
```

download

download schedule for today

download schedule for week

not download yesterday schedule

not download schedule after 7th day

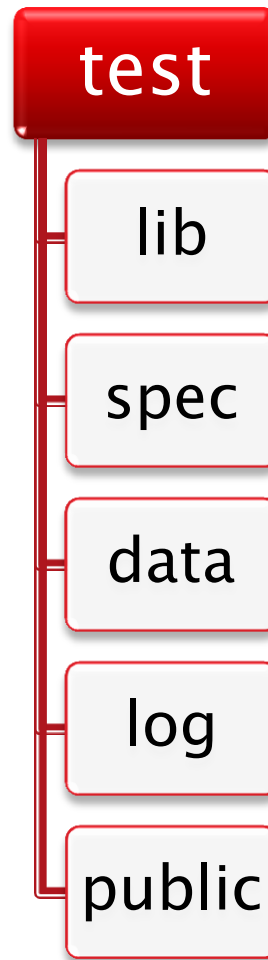
download schedule if it was updated

download schedule with different event names

Finished in 120.089 seconds

6 examples, 1 failures

Структура проекта



XML расписания

```
<?xml version="1.0" encoding="windows-1251"?>
```

```
<Grid>
```

```
  <Event>...</Event>
```

} Event

```
  <Event>...</Event>
```

```
  <Event>...</Event>
```

```
  <Event>...</Event>
```

```
  <Event>...</Event>
```

```
  <Event>...</Event>
```

```
  ...
```

```
</Grid>
```

Iconv

XML расписания

```
<Event>
  <ID>33894863</ID>
  <DayNo> ... </DayNo>
  <Start>__DAY__ 07:00:00</Start>
  <Finish>__DAY__ 07:15:00</Finish>
  <Star>0</Star>
  <Name>Test Name</Name>
  <Info id="89724" type="2"> ... </Info>
  <Flag><ID>4</ID><Name>Инфо</Name></Flag>
  <Location>
    <ID> ... </ID>
    <Type>1</Type>
  </Location>
  <Gate> ...
    <Info>Test Description</Info>
  ... </Gate>
  <Type ... >other</Type>
</Event>
```

__DAY__

Event

JSON расписания

```
[  
  {  
    "name": "Test Name",  
    "id": "33894863",  
    "beginning": "__DAY__ 07:00:00",  
    "ending": "__DAY__ 07:15:00",  
    "description": "Test Description",  
    "type": "other"  
  },  
  ...  
]
```

Event

Тестирование данных

Данные	Тестируем как
name, description	текстовые поля
beginning – ending	интервал времени
id	числовое поле

Пример. Тестирование name

- ▶ Нормальное название
- ▶ Длинное название
- ▶ Стандартные разделители
 - “ ‘ ` | / \ , ; : & < > ^ * ! ? « »
- ▶ Пустое название
- ▶ Пробелы
- ▶ ЧуВСтВиТельнОсТЬ К рЕГИСТрУ
- ▶ Окончания строк
 - ^M, \n, \r
- ▶ Диакритические знаки
 - àáâãäå
- ▶ Плохие символы
 - ☺ ♂ ♠ ♪
- ▶ Греческий, арабский, китайский
- ▶ Другая кодировка
- ▶ Плохое выражение
 - “[|]’~<!--@/*\$%^&#*/()?>,.*/\-->

Финальные тезисы

1. Учить язык программирования
2. Использовать готовые библиотеки
3. Писать обертки
4. Отделять данные от тестов
5. Не забывать о «тестировании по выходу»

Спасибо за внимание

▶ Ruby

- Вики-учебник <http://ru.wikibooks.org/wiki/Ruby>
- Упражнения <http://rubymonk.com>

▶ Rspec

- <http://kerryb.github.com/iprug-rspec-presentation>

▶ Контакты

- E-mail: ilyubin@yandex.ru
- Skype: igor.lyubin