构建简单健壮的Web自动化测试的模式

路宁

Shipping Instructions - New

Details * | Container and Cargo * | Preview

My Favorite Templates: Select Go

* Required. If you could not finish SI form in one hour, we strongly suggest you save draft as you go.

Carrier: * OOCL

Shipper *

Desired Text on B/L: Tip

Dummy Service Provider
Dummy Address Test 001

Booking Number *

Add Delete

TEST_BK_2

Bill of Lading Number

Export References

Carrier Rate Type: Service Contract Number

Carrier Rate Reference Number: PE064836

User Reference Type: Select (if 'Other, please specify)

User Reference Number: Add Delete

Forwarder Reference - 8786549633

Consignee *

To Order

Desired Text on B/L: Tip

Dummy Service Provider
Dummy Address Test 001

Forwarder

Desired Text on B/L: Tip

Black Decker U.S. Inc
Highway 301 South
test com1, North Carolina - 28306
United States

FMC Number:

Origin of Goods

Notify Party

Desired Text on B/L: Tip

Black Decker U.S. Inc
Highway 301 South
test com1, North Carolina - 28306
United States

Also Notify Party

1. Desired Text on B/L: Tip

Black Decker U.S. Inc
Highway 301 South
test com1, North Carolina - 28306
United States

2. Desired Text on B/L: Tip

Hyundai Merchant Marine Co., Ltd
451167 Hudson Way
Decatur, Georgia - 30033
United States

Pre-Carriage by

Place of Receipt

Desired Text on B/L: Ningbo, Zhejiang, China

Loading Pier/Terminal

Originals to be Released At

Vessel, Voyage & Direction *

Desired Text on B/L: SHAN HE 130W

Port of Load *

Desired Text on B/L: Ningbo

Port of Discharge *

Desired Text on B/L: Chittagong

Final Destination

Desired Text on B/L: Chittagong, Bangladesh

Traffic Mode *

FCL/FCL

Other Instructions on the B/L

Requested B/L Date: Select

Ocean Freight: * Prepaid Collect

Remarks on B/L:

B/L Handling Instructions

Draft B/L

Receiving Party: Select | Receive By: Select | (if 'Fax', please specify) [Country] - [Area] - [Local]

Final B/L

B/L Type: * Select | Receive By: * Select | (If 'Other, please specify)

B/L Distribution: * Tip

	Shipper		Forwarder		Consignee		Notify Party		Also Notify Party
	B/L Type	Copy	B/L Type	Copy	B/L Type	Copy	B/L Type	Copy	Copy
Non-Freighted									
Freighted (All Charges)									
Freighted (Prepaid Only)									
Freighted (Collect Only)									

Other B/L Handling Instructions:

Shipping Instructions - New

Details* | Container and Cargo* | Preview

My Favorite Templates: Select Go

* Required.

Container and Cargo Details

Add Summary Details

Container

Container 1

Associated Bkg #: * TEST_BK_2

Container #: CNTR_3-1

Shipper Owned

Container Size Type: 20' x 8' x 8' General Purpose Container

Total Cargo Gross Weight: 2 Tons

Seal Type: High Security / Select / Select

(please specify)

Seal #:

Cargo

Marks and Numbers	Quantity	Cargo Nature / Description	Weight / Volume	Actions
Marks and Numbers: Select Add to favorite	Quantity: * 98 Package Type: * Container	Cargo Nature: * Reefer Cargo Description: Select test Add to Favorite	Gross Weight: 122.9 Pounds Net Weight: 11.3 Kilograms Volume: 20.7 Cubic Meters	Copy Cargo to

Add Cargo

Number of containers to be added 1 Add to List

Details * | Container and Cargo * | Preview

Cancel | Validate SI

Save as Template | Save as Draft

thebuzz
faster, easier, better

home page | about us | news & info | FAQs | contact us

login to manage my policy

ask your question here... ask

car insurance | home insurance | claims | manage my policy

Control your price

Only pay for what you need!

Backed by Insurance Australia Group

find out more

get a quick quote in 3 easy steps

car | home

1 Car's postcode and suburb: 2000 BARANGAROO NSW

2 Year: 2011

Make: AUDI

Model: A7 HATCH 4G

Type: SPORTBACK 3.0 TDI QUATTRO

Engine: 3.0L DIESEL

3 DOB: 11 Oct 1979

show my quote

quick quote in 60 seconds

manage your policy online

renew or pay your policy

3 easy steps

1 my quote
- my car
- my details
- my insurance

2 my policy
- my car
- owner driver details
- my insurance policy

3 my payment
- payment details

Not completed

1 my quote — in progress

my car — completed

details

Year * 2011

Make * AUDI

Model * A7 HATCH 4G

Type * SPORTBACK 3.0 TDI QUAT

Engine * 3.0L DIESEL

use

My car is used for? *

- (•) Private
- ( ) Courier, demonstration, delivery or transport of goods, courtesy vehicle, hire, paid driving instruction, security patrol, taxi or limousine
- ( ) Other business use

finance

Any finance owing on the car? *

- (•) No
- ( ) Yes

next

my details — in progress

my insurance — completed edit

2 my policy — in progress

3 my payment — not started

my quote

~~$1,865.49~~

Details used to calculate the quote have changed. Click OK to recalculate the quote total

EMAIL QUOTE

live chat with The Buzz

live chat now

Got a question?

ask your question here...

ask

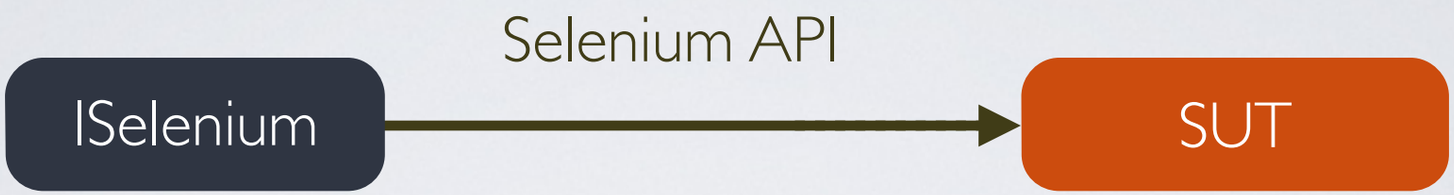
复杂的表单

jQuery

Ajax

Animation

让测试健壮


Selenium API
ISelenium
SUT

消灭
和大部分

Thread.Sleep
WaitForCondition

- 慢
- 不稳定
- 丑陋

避免同步等待
使用一致的异步等待方法

jQuery.ajax.active == 0

Ajax.activeRequestCount == 0

dojo.io.XMLHTTPTransport.inFlight.length == 0

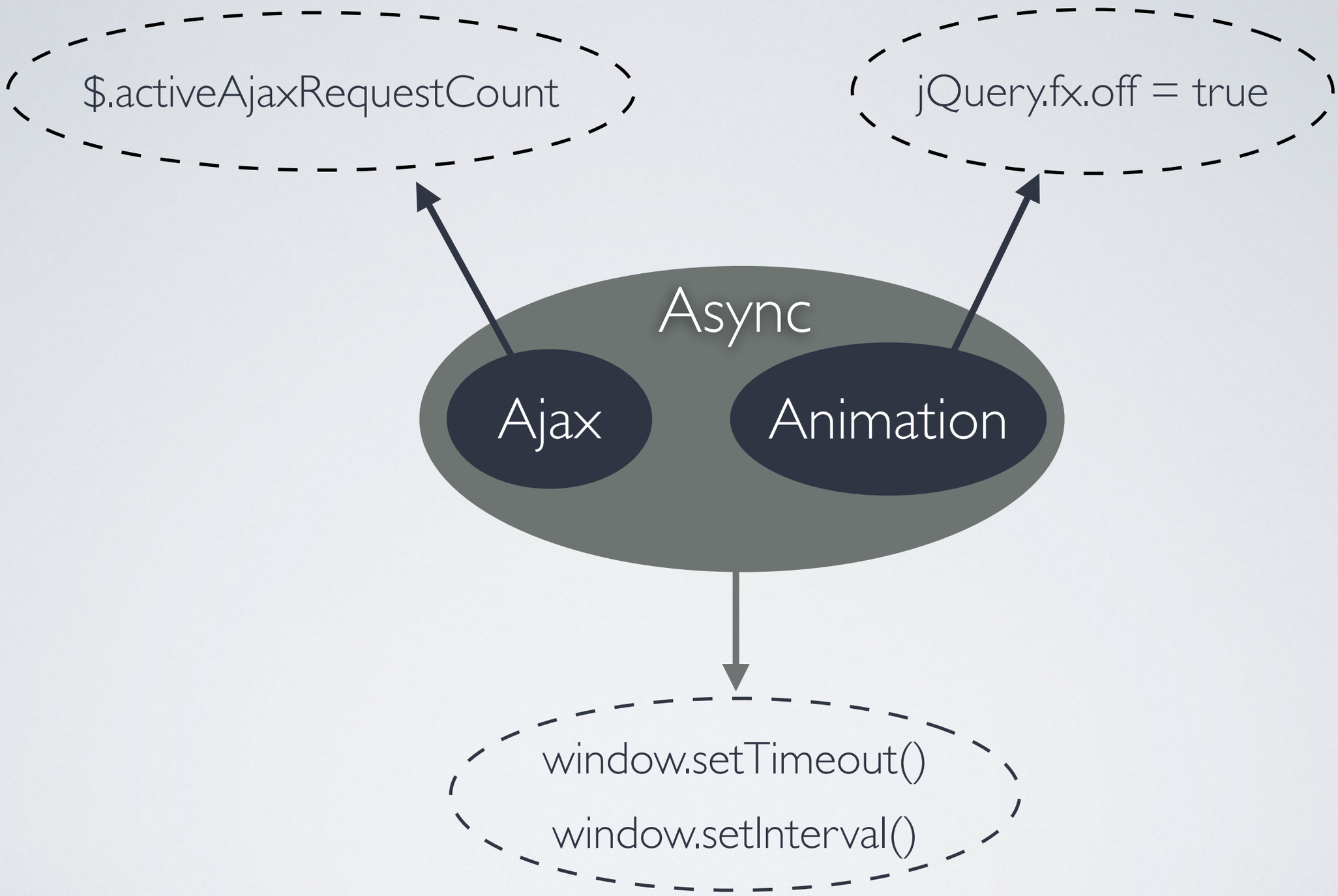
等待Ajax call完成

```javascript
/* javascript code in client side */
// extend jQuery ajax with the capability of remembering
// the count of active ajax requests
$.activeAjaxRequestCount = 0;

$().ajaxSend(function() {
  $.activeAjaxRequestCount++;
  // register lazily to make sure it's the last handler of
  // ajaxError event and get executed after all production handlers
  if(!$._ajaxErrorHandlerAdded) {
    $().ajaxError(function() {
      $.activeAjaxRequestCount--;
    });
    $._ajaxErrorHandlerAdded = true;
  }
})
.ajaxSuccess(function() {
  $.activeAjaxRequestCount--;
});
```

```
28  // an overload with 60 seconds as default timeout,
29  // this is the mostly used version
30  public void Click(string locator)
31  {
32    Click(locator, 60000);
33  }
34  // this overload has more control on timeout
35  public void Click(string locator, int timeout)
36  {
37    // decorate selenium call with waiting logic
38    WaitComplete(() => selenium.Click(locator), timeout);
39  }
40  // generic decorating helper for selenium calls
41  private void WaitComplete(Action action, int timeout)
42  {
43    // do default selenium action
44    action();
45    // wait for active ajax request count becomes zero,
46    // none ajax selenium action will return immediately
47    selenium.WaitForCondition("selenium.browserbot.getCurrentWindow();
48                               window.jQuery.activeAjaxRequestCount===0;",
49                               timeout.ToString());
50  }
51
52  /* do the same trick to Select, Type, FireEvent and more selenium methods */
53
```

等待Animation完成?

```
24  // run this js code somehow before selenium test
25  // to turn off animation of jQuery
26  $.fx.off = true;
```


$.activeAjaxRequestCount
jQuery.fx.off = true
Async
Ajax
Animation
window.setTimeout()
window.setInterval()

创建你自己的 testability_extension.js

```
/* javascript code in client side */
// extend jQuery ajax with the capability of remembering
// the count of active ajax requests
$.activeAjaxRequestCount = 0;

$().ajaxSend(function() {
  $.activeAjaxRequestCount++;
  // register lazily to make sure it's the last handler of
  // ajaxError event and get executed after all production handlers
  if(!$._ajaxErrorHandlerAdded) {
    $().ajaxError(function() {
      $.activeAjaxRequestCount--;
    });
    $._ajaxErrorHandlerAdded = true;
  }
})
.ajaxSuccess(function() {
  $.activeAjaxRequestCount--;
});
```

❶提高可测试性

❷被Selenium测试代码使用，简化测试

❸仅在测试环境中使用

包装你自己的 Selenium API

```
28  // an overload with 60 seconds as default timeout,
29  // this is the mostly used version
30  public void Click(string locator)
31  {
32    Click(locator, 60000);
33  }
34  // this overload has more control on timeout
35  public void Click(string locator, int timeout)
36  {
37    // decorate selenium call with waiting logic
38    WaitComplete(() => selenium.Click(locator), timeout);
39  }
40  // generic decorating helper for selenium calls
41  private void WaitComplete(Action action, int timeout)
42  {
43    // do default selenium action
44    action();
45    // wait for active ajax request count becomes zero,
46    // none ajax selenium action will return immediately
47    selenium.WaitForCondition("selenium.browserbot.getCurrentWindow();
48                               window.jQuery.activeAjaxRequestCount===0;",
49                               timeout.ToString());
50  }
51
52  /* do the same trick to Select, Type, FireEvent and more selenium methods */
53
```

❶扩展Seleniun方法

❷封装以提供一致的行为，隐藏噪音

任何Selenium操作失败后都抛出指定异常并保存浏览器上下文

- Screenshot
- HTML Body
- Location
- Page Title

```
1  public class MySelenium
2  {
3    private readonly ISelenium _selenium;
4    // ... blocks of code omitted
5
6    public string GetValue(string locator)
7    {
8      return TrySeleniumFunction(s => s.GetValue(locator),
9        string.Format("Error getting value of element {0}", locator));
10   }
11
12   private T TrySeleniumFunction<T>(Func<ISelenium, T> function
13                                    , string errorMessage)
14   {
15     T value = default(T);
16     TrySeleniumAction(s => value = function(s), errorMessage);
17     return value;
18   }
19
20   private void TrySeleniumAction(Action<ISelenium> action, string errorMessage)
21   {
22     try
23     {
24       action(_selenium);
25     }
26     catch (Exception ex)
27     {
28       throw CreateMySeleniumException(errorMessage, ex);
29     }
30   }
```

To be continued...

Continued from previous page

```
31
32    public MySeleniumException CreateMySeleniumException(string errorMessage
33                                                         , Exception innerException)
34    {
35      try
36      {
37        var context = BrowserContext.Save(_selenium);
38        return new MySeleniumException(errorMessage, innerException, context);
39      }
40      catch (SeleniumException exceptionWhileSavingScreenshot)
41      {
42        var message = string.Format("{0} [Failed to save screenshot: {1}]",
43                                    errorMessage, exceptionWhileSavingScreenshot.Message);
44          return new MySeleniumException(message, innerException);
45      }
46    }
47  }
```

```
37          var context = BrowserContext.Save(_selenium);
```



```
1   selenium.CaptureScreenshot(filename);
2
3   string html = selenium.GetHtmlSource();
4
5   string location = selenium.GetEval("selenium.browserbot.getCurrentWindow();"
6                                       + " window.location.href;") ?? "unknown";
7
8   string title = selenium.GetEval("selenium.browserbot.getCurrentWindow();"
9                                     + " window.document.title;") ?? "unknown";
```

重新思考所有Wait完成的条件
并通过MySelenium管理起来

WaitForPageToLoad的完成条件

selenium.WaitForPageToLoad()返回

所有Animation结束

所有document ready事件处理函数执行完成

所有Ajax访问已返回并处理完毕

可以进行用户交互的条件

可以进行用户交互的条件

```
1  private void WaitUntilReadyForUserInteraction(int timeout)
2  {
3    try
4    {
5      TryTurnOffAllJQueryAnimations();
6      _selenium.WaitForCondition("selenium.browserbot.getCurrentWindow();"
7                + "window.jQuery? (window.jQuery.readyList===null"
8                + "&& (!window.jQuery.activeAjaxRequestCount"
9                +  " || window.jQuery.activeAjaxRequestCount===0)):true;",
10       timeout.ToString());
11   }
12   catch (SeleniumException ex)
13   {
14     var details = GetEvalWithJQuery(
15             "window.jQuery.waitForUserInteractionFailureDetails();");
16     var message = string.Format("Failed to wait until ready for UI,"
17                 + "{0}, original exception = {1}", details, ex.Message);
18     CreateMySeleniumException(message, ex);
19   }
20 }
21
22 private void TryTurnOffAllJQueryAnimations()
23 {
24   const string script = "selenium.browserbot.getCurrentWindow(); " +
25                         "if (window.jQuery) window.jQuery.fx.off = true;";
26   _selenium.GetEval(script);
27 }
```

所有document ready事件处理函数执行完成

所有Ajax访问已返回并处理完毕

关闭Animation

~~WaitForElementPresent~~

WaitForElementVisible的完成条件

selenium.isElementPresent()

selenium.isVisible()

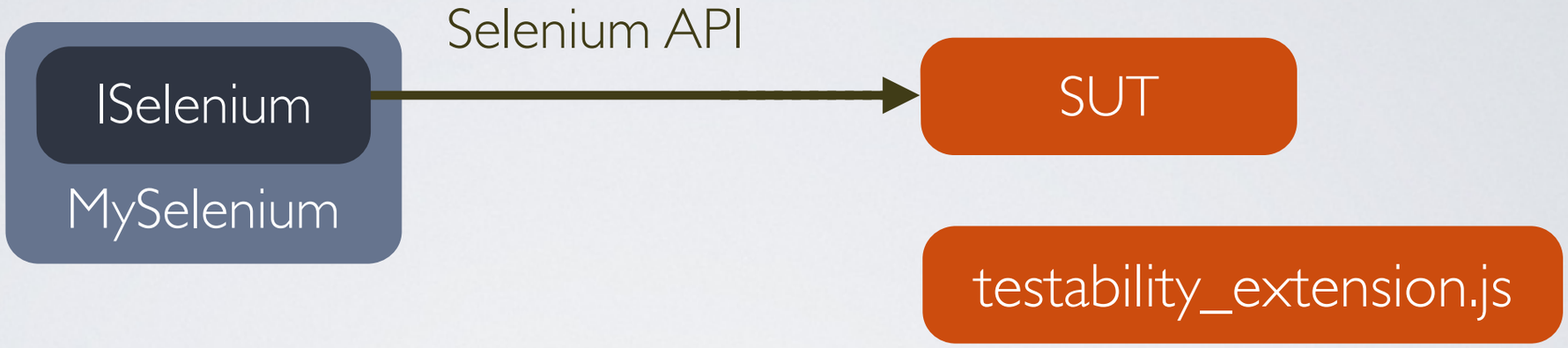
Selenium Core API

WaitUntilReadyForUserInteraction() in MySelenium

```
public void WaitForElementVisible(string locator, int timeout)
{
  TrySeleniumAction(s => s.WaitForCondition(string.Format(
     "selenium.isElementPresent('{0}') && selenium.isVisible('{0}')", locator),
        timeout.ToString())
     ,string.Format("Error while waiting for element '{0}' to be visible."
     , locator));

  WaitUntilReadyForUserInteraction(timeout);
}
```

在Ajax上下文下执行Selenium命令

```
public void Click(string locator)
{
  PerformAjaxAction(() => _selenium.Click(locator), DEFAULT_TIMEOUT);
}

public void Type(string locator, string value)
{
  PerformAjaxAction(() => _selenium.Type(locator, value), DEFAULT_TIMEOUT);
}

private void PerformAjaxAction(Action action, int timeout)
{
  TrySeleniumAction(s => action(),
            string.Format("Error performing action: {0}", action));

  WaitForAjaxToFinish(timeout);
}

private void WaitForAjaxToFinish(int timeout)
{
  const string condition = "selenium.browserbot.getCurrentWindow();"
          + "window.jQuery?window.jQuery.activeAjaxRequestCount===0:true;";

  TrySeleniumAction(s => s.WaitForCondition(condition,timeout.ToString()),
                    "Error waiting for AJAX to finish");
}
```


Selenium API
ISelenium
MySelenium
SUT
testability_extension.js

绕过弹出的windows窗口

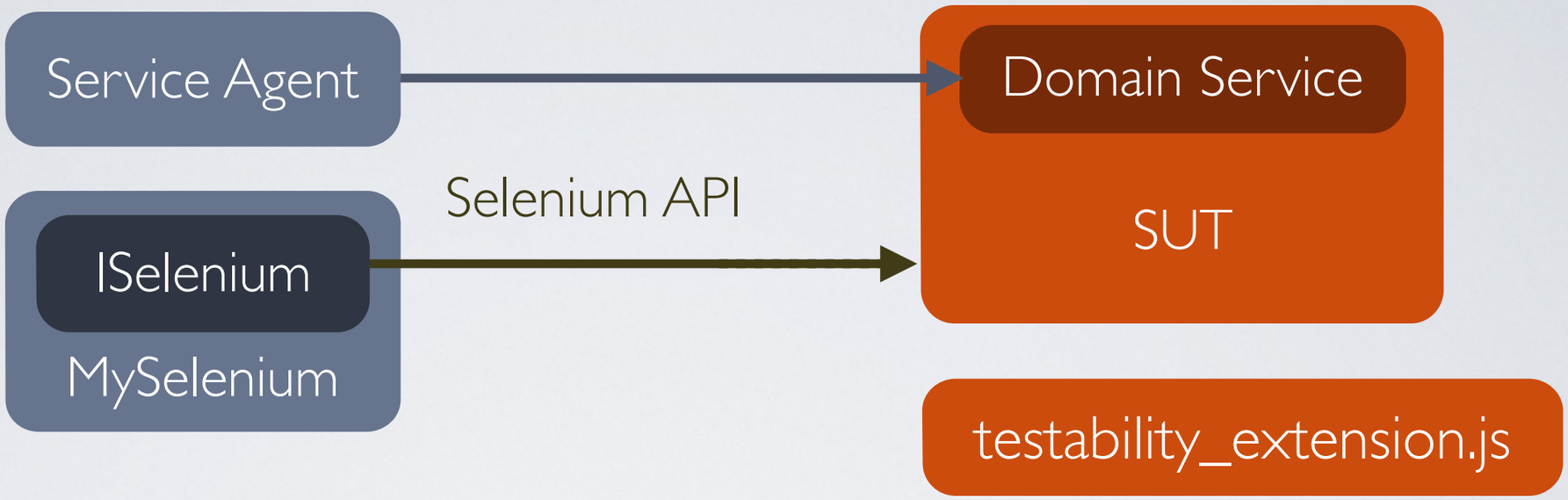
处理浏览器兼容性问题

管理测试数据

- Model Object
  - In Code
  - YAML
  - Excel

加速单个测试的上下文准备

- 调用生产系统的Service
  - 开放接口
  - 避免直接访问数据库
- 避免重用数据实例
  - ID generator


Service Agent
Domain Service
Selenium API
ISelenium
SUT
MySelenium
testability_extension.js

让测试简单

```java
@Test
public void createFullRequest_priview_backEditing_checkData(){
    goToCreationPage();
    browser.click(ADD_APPOINTMENT);
    browser.click("lorryTrucking1");
    browser.select("containerSizeType1_1", "label=20' Platform");
    typeAndBlur("containerQuantity1_1", "1");
    browser.select("cargoPackageType1_1", "label=Bar");
    browser.click("containerTrucking1");
    browser.type("containerQuantity1_1", "1");
    browser.click("containerIsSOC1_1");
    browser.select("containerSizeType1_2", "label=20' x 8' x 8' 6
                   Hanger Container");
    browser.type("containerQuantity1_2","1");

    // omitted 100+ lines.
}
```

type(), select()... 太过繁冗，尤其是对于复杂的页面

测试数据与代码交织，不利于重用

thread.sleep()和waitForCondition()让测试丑陋且不稳定

id, locator到处都是，重复，难以重构

Problems

测试逻辑被各种噪音代码包围

如何让测试变得简洁并富有表达能力

Page Object?

需要太多的编织代码

jQuery.parcel

github.com/luning/jquery.parcel

Goal - OO地封装客户端代码

"Side Effect" - 简化了Selenium测试

A "hello world" sample

jquery-1.7.1.js
jquery.print.js (optional)
jquery.parcel.js

```html
<div id="person">
  <input name="name" type="text" />
  <select name="age">
    <option>please select</option>
    <option>18</option>
  </select>
  ...
</div>
```

```javascript
// get state
$("#person").state();

// return JSON
{
  name: "",
  age: "please select",
  ...
}

// set state
$("#person").state({name: "luning", age: "30"});
```


```
new Person{
   name = "luning",
   age = "please select",
   // ...
}
```

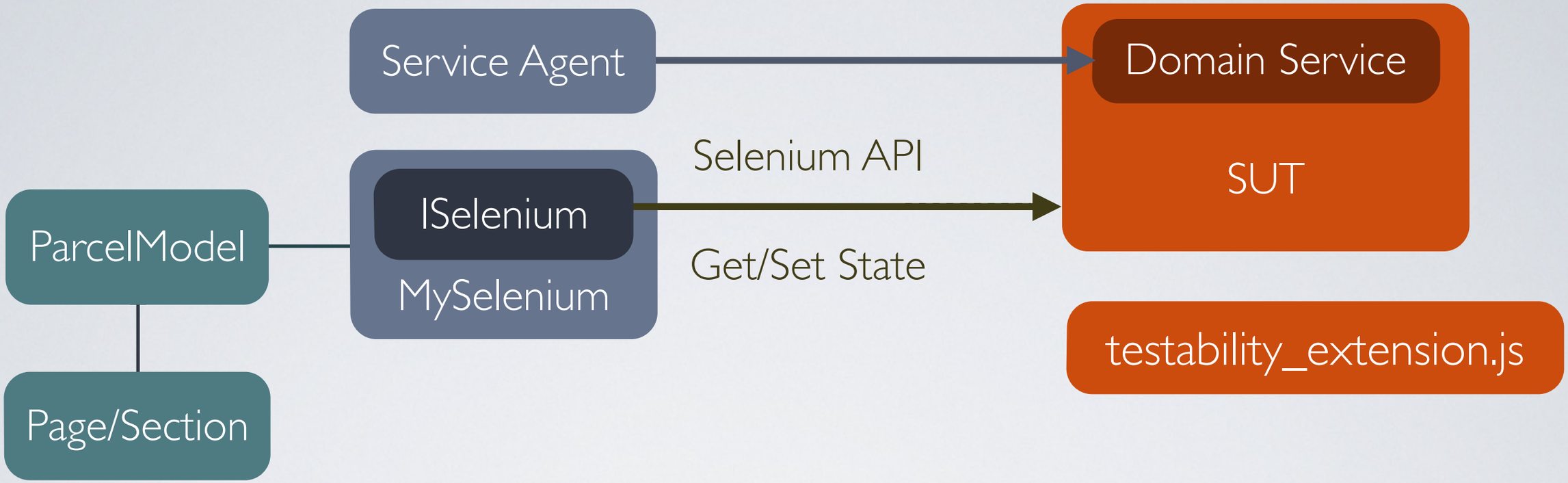
```
{
   name:"luning",
   age:"please select",
   // ...
}
```

Naming Convention

```
1  public class Person
2  {
3      public string name;
4      public string age;
5      ...
6  }
7
8  Person person = new Person{
9                          name = "luning",
10                         age = "30"
11                     };
12
13 string json = new JavaScriptSerializer().Serialize(person);
```

```
1  void Populate(string containerSelector, object model)
2  {
3      string json = new JavaScriptSerializer().Serialize(model);
4      selenium.RunScript(string.Format("$('{0}').state({1});"
5                                       , containerSelector, json));
6  }
7
8  Populate("#person", new Person{
9                          name = "luning",
10                         age = "30"
11                     });
```

```
T GetState(string containerSelector)
{
    var json = selenium.GetEval(string.Format("$('{0}').state();"
                                              , containerSelector));
    return new JavaScriptSerializer().Deserialize<T>(json);
}

Person person = GetState<Person>("#person");
```


Service Agent
Domain Service
SUT
Selenium API
ISelenium
ParcelModel
MySelenium
Get/Set State
testability_extension.js
Page/Section

```
var page = new Person();
page.Populate(new Person{
                    name = "luning",
                    age = "please select",
                    // ...
                });
page.ClickOK();
page.GetState();
Assert.AreEqual("some value", page.anotherField);
```

```
1  public abstract class ParcelModel
2  {
3    private ISelenium selenium;
4
5    internal void PopulateWith(string containerSelector, string state) { /*...*/ }
6
7    protected T GetStateFromUI<T>(string containerSelector)
8                where T : ParcelModel { /*...*/ }
9
10   protected void CopyState<T>(T copyFrom) where T : ParcelModel { /*...*/ }
11 }
```

```
13  public class Person : ParcelModel
14  {
15    public string name;
16    public string age;
17    // other fields...
18
19    private const string SECTIONID = "#person";
20    private const string OKBUTTON = "submit";
21
22    public string Selector()
23    {
24      return SECTIONID;
25    }
26
27    public void Populate(object state)
28    {
29      PopulateWith(Selector(), state.AsJSON());
30    }
31
32    public void GetState()
33    {
34      CopyState(GetStateFromUI<QuickQuote>(SECTIONID));
35    }
36
37    public void ClickOK()
38    {
39      selenium.Click(OKBUTTON);
40    }
41
42    public void verifyXXX() { /*...*/ }
43
44    public void verifyYYY() { /*...*/ }
45  }
```

Fields

Locators

Get/Set State

Actions

Complex Verifications

```cs
public class Person : ParcelModel
{
  public string name;
  public string age;
  public DOB dob;
  // ...
}
```

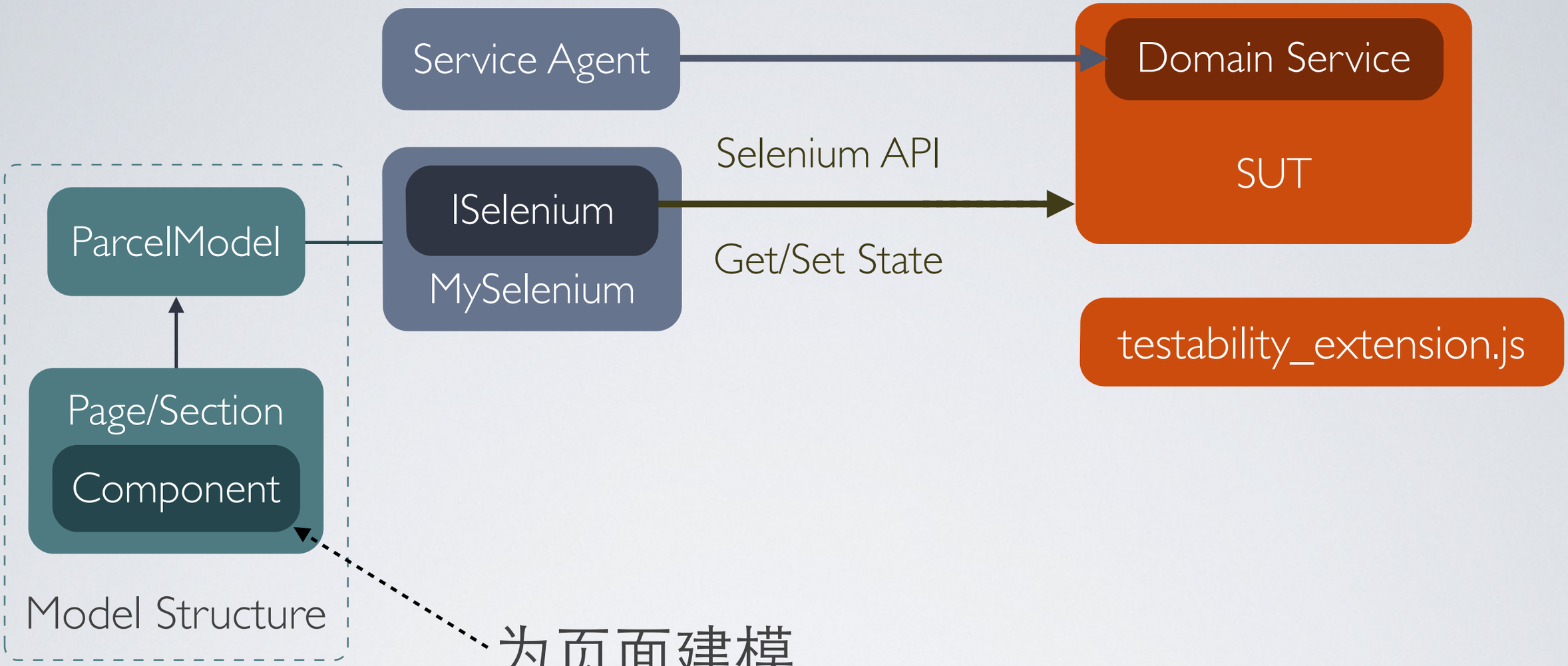
Component

```cs
public class DOB
{
  public string day;
  public string month;
  public string year;

  public override string ToString()
  {
    return string.Format("{0}-{1}-{2}", day, month, year);
  }
}
```

More than "hello world"

Composite Parcel Model

```
{
  title: "Mr",
  firstname: "John",
  lastname: "Smith",
  contact:
  {
    number: "+61 2 9555 0123",
    type: "work"
  }
}

{
  title: "Mr",
  firstname: "John",
  lastname: "Smith",
  email:["a@my.com", "b@my.com"]
}
```



为页面建模

ParcelField(自定义的DOM属性)

篡改window.setTimeout

```
function(code){
  $.isFunction(code) ? code() :  eval(code);
  return -1;
};
```

Sync - 是否禁用animation并同步执行Ajax调用

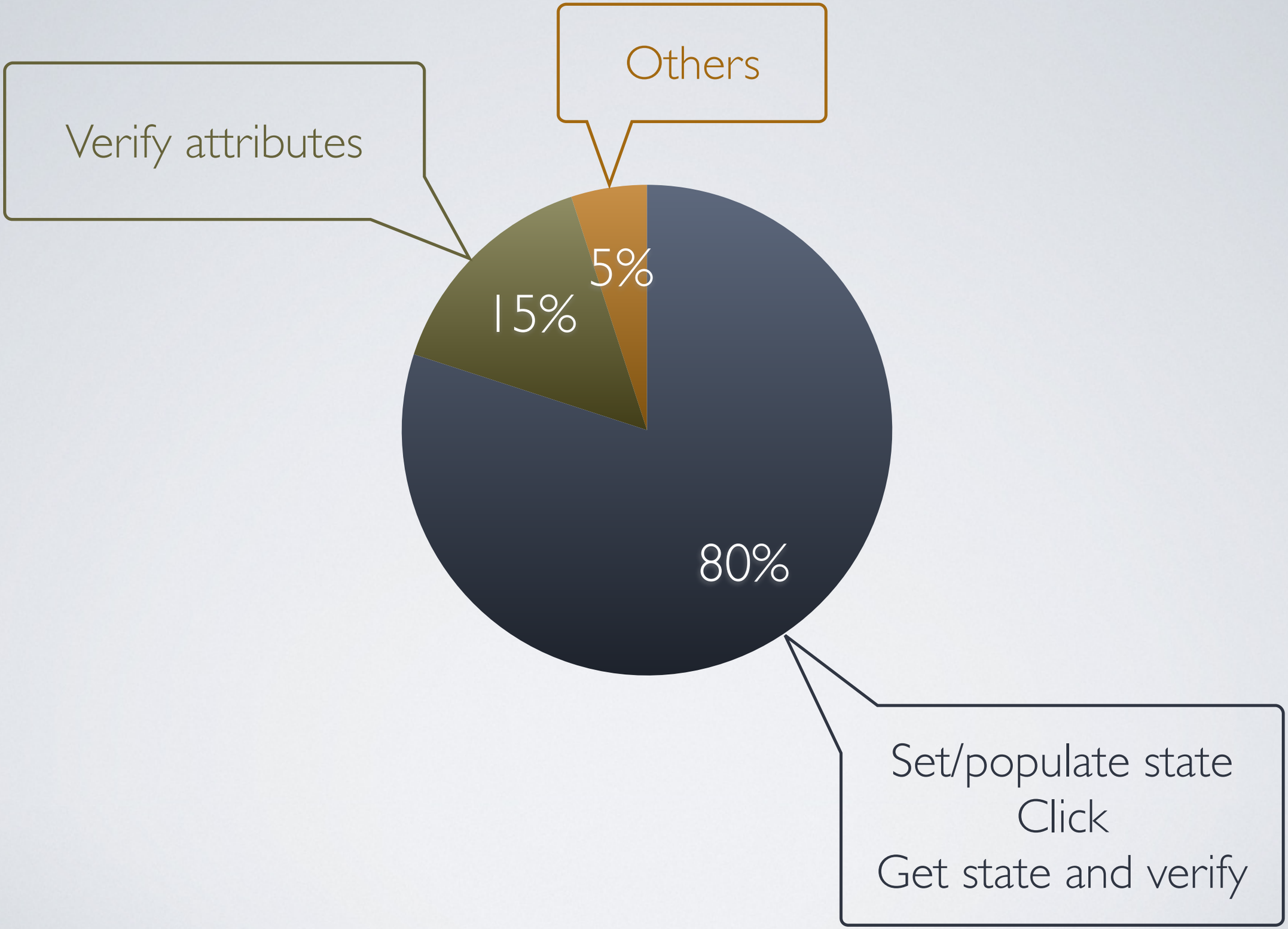
Editable - 是否检查field为visible并且enabled

Verify - 是否检查field如期发生变化

Exist - 是否检查所有属性都有对应的field

Set State时需要更多控制

```
1  $("#person").state({name: "luning"}, {sync: true, editable: true
2                                       , exist: true, verify: true});
```


Verify attributes
Others
5%
15%
80%
Set/populate state
Click
Get state and verify

Verify attributes

```
1  Assert.IsTure(person.GetAttributes("dob.day").Visible);
```

In testing code

```
1  public class UIElementAttributes
2  {
3    public bool Disabled;
4    public bool Visible;
5    public string Class;
6    public string Style;
7
8    public bool HasClass(string c)
9    {
10     return Class.IndexOf(c) != -1;
11   }
12
13   public bool HasStyle(string s)
14   {
15     return Style.IndexOf(s) != -1;
16   }
17
18   public string OptionsOfSelect;
19
20   public string DatepickerMinDate; //yyyy-mm-dd
21   public string DatepickerMaxDate;
22
23   public string AnchorLink;
24 }
```

Common attributes of DOM

Attributes of specific INPUTs

In testability_extension.js

```
1  $.getAttributes = function(selector, fieldLocator) {
2    try {
3      var field;
4      // set field to a parcel field
5
6      var attrs = {
7        Disabled: field.attr("disabled") === undefined ?
8                          false : field.attr("disabled"),
9        Visible: field.is(":visible"),
10       Class: field.attr("class") || "",
11       Style: field.attr("style") || ""
12     };
13
14     if (field.is("select")) {
15       // append attributes for select
16     }
17     // append attributes for other type of field
18
19     return $.toJSON(attrs);
20   } catch (ex) {
21     return "ERROR:" + ex.toString();
22   }
23 };
```

In ParcelModel.cs

```
1   public UIElementAttributes GetAttributes(string fieldLocator)
2   {
3     return GetAttributes(Selector(), fieldLocator);
4   }
5
6   public UIElementAttributes GetAttributes(string containerSelector,
7                                            string fieldLocator)
8   {
9     var json = selenium.GetEvalWithJQuery("window.jQuery.getAttributes("
10                + "'" + containerSelector + "', '" + fieldLocator + "');");
11    if(json.StartsWith("ERROR"))
12    {
13      throw new ArgumentException(json, fieldLocator);
14    }
15
16    return new JavaScriptSerializer().Deserialize<UIElementAttributes>(json);
17  }
```

Get errors on page

In ParcelModel.cs

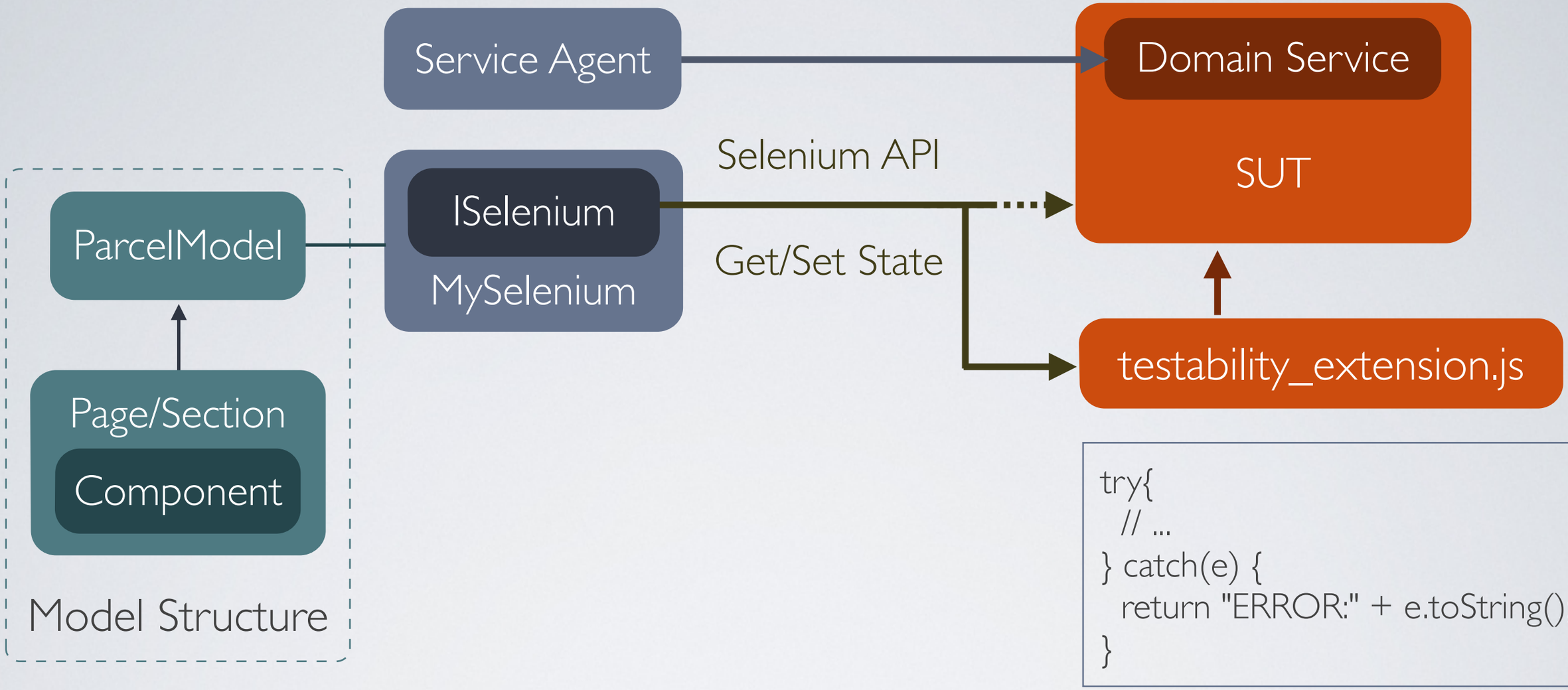
```
1  public string ErrorsOnPage()
2  {
3    return selenium.GetEvalWithJQuery("window.jQuery.getErrorsInPage();");
4  }
```

In testability_extension.js

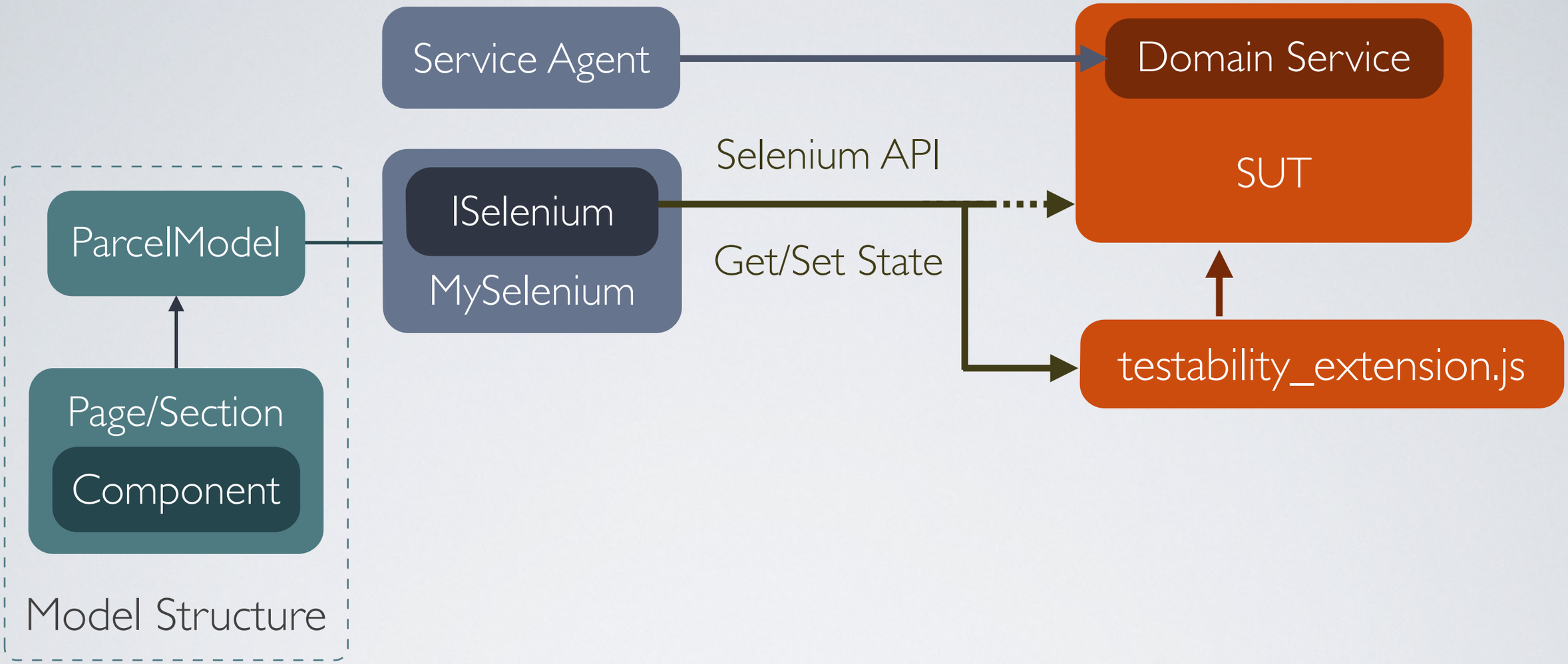
```
1  $.getErrorsInPage = function() {
2    var errors = $.grep($("label.error:visible,div.error:visible,"
3                        + ".errorMessage:visible,span.error:visible"),
4                        function(err) {
5                          return $(err).text() !== "";
6                        });
7
8    if (errors.length > 0) {
9      return "[ \"" + $.map(errors, function(e) {
10                                     return $(e).text();
11                                   }).join("\", \"") + "\" ]";
12   }
13
14   return "";
15 };
```

```
var page = new Person();

var err = page
  .Populate(new Person{
                name = "luning",
                age = "please select",
                // ...
            })
  .ClickOK()
  .ErrorsOnPage();

Assert.AreEqual("age should be selected", err);

page.Populate(new Person{
                fieldA = "A"
              })
  .GetState()
  .VerifyEmails();
```

Catch all JavaScript exceptions




```
try{
  // ...
} catch(e) {
  return "ERROR:" + e.toString();
}
```


Service Agent
Domain Service
SUT
Selenium API
ISelenium
MySelenium
Get/Set State
ParcelModel
testability_extension.js
Page/Section
Component
Model Structure

Get/set state替代type, select, 提高了抽象和表达能力
利用命名习惯，极少使用ID和Locator，拼写错误易得提示
极大程度减少了维护性代码
更快，因为更少的调用
方便地感知JavaScript异常
测试数据(JSON)极易管理
仅需维护少量的编织代码

优势

依赖所使用的js库
依赖具体的应用
依赖GetEval
引入仅在测试时使用的代码

问题

Thanks!

路宁