RWT

Rails Web Toolkit

Author: Sérgio M. Brant

Revision: 07/09/2009

Introduction

Explain by Test

The idea here is to explain the rwt API by explaining the rwt Rspec tests. I'll start with the most basic rwt class, the Component class.

Lets take a look at our first test of our component_spec.rb test suite:

```
 it "should generate javascript in the buffer" do
  component(:test=>'test')
#   puts Rwt.code
  Rwt.code.should include("test:'test'")
 end
```

If we uncomment the puts line we can see what is being generated in the Rwt.code buffer:

```
var v1=new Ext.Component({v:'v1',test:'test'});
```

This is an instatiation of a Ext.Component with the *test* parater set to 'test'. The *component* method of the Rwt module instantiates a Rwt::Component class that generates (during its initialization) the corresponding javascript instatiation code of the ExtJs Ext.Component class. The additional *v* parameter is just an auxiliary place to store the name of the javascript variable that will point to the javascript object created at run-time.

So, whenever we call *component* in a rwt ruby view we will get new code generated in the buffer that later will be used by rwt_render to send it to be executed in the client browser.

The next test exercises the capacity of constructing a hierarchy of components with rwt:

```
 it "should be able to construct a hierachy of components" do
  a_gf=component('grandfather') do |gf| # gf here only needed because of reference in son
   component('father') do |f|
    f.owner.config[:text].should == 'grandfather'
    component('son') do |s|
     s.owner.config[:text].should == 'father'
     s.on(“create”) do
      Rwt << "#{gf}.show();"
     end
    end
   end
  end
#   puts Rwt.code
#   puts a_gf.vid

  Rwt.code.should include("#{a_gf}.show()")
 end
```

Wich led to the folowing javascript code:

```
var v4=new Ext.Component({v:'v4',text:'son'});
v4.on('create',function(){v2.show();});
var v3=new Ext.Component({v:'v3',items:[v4],text:'father'});
var v2=new Ext.Component({v:'v2',items:[v3],text:'grandfather'});
```

As you can see, the creation is done in reverse order to avoid errors in javascript execution.

The << method of Rwt inserts text directly in the code buffer. In the case showed here this is done in the moment of event code creation.