

# Shoulda 2.0 Cheat Sheet

## AR Associations

<code>should_have_and_belong_to_many</code>	<code>:schools</code>
<code>should_have_many</code>	<code>:posts</code>
<code>should_have_one</code>	<code>:profile</code>
<code>should_belong_to</code>	<code>:user</code>

## AR Validations

<code>should_ensure_length_at_least</code>	<code>:name, 3</code>
<code>should_ensure_length_in_range</code>	<code>:password, (6..20)</code>
<code>should_ensure_length_is</code>	<code>:ssn, 9</code>
<code>should_ensure_value_in_range</code>	<code>:age, (0..100)</code>
<code>should_require_unique_attributes</code>	<code>:ssn</code>
<code>should_require_acceptance_of</code>	<code>:terms_of_use</code>
<code>should_only_allow_numeric_values_for</code>	<code>:years_of_experience</code>
<code>should_require_attributes</code>	<code>:name, :age</code>
<code>should_allow_values_for*</code>	<code>:phone, "(123) 999-3049", "201-0023"</code>

## AR Class Attributes

<code>should_have_class_methods</code>	<code>:find, :destroy</code>
<code>should_have_instance_methods</code>	<code>:name=, :email</code>
<code>should_have_named_scope</code>	Any <code>named_scope</code> options
<code>should_have_readonly_attributes</code>	<code>:password, :is_admin</code>
<code>should_protect_attributes</code>	<code>:password, :ssn</code>
<code>should_have_db_column*</code>	<code>:post_id</code>
<code>should_have_index*</code>	<code>:url</code>

\*Indicates that a negative form of the assertion exists, e.g. **should\_not\_allow\_values\_for**, **assert\_does\_not\_contain**, etc.

## Controllers

<code>should_route</code>	<code>:post, '/posts', :controller =&gt; :posts, :action =&gt; :create</code>
<code>should_change*</code>	<code>"Post.count", :from =&gt; 0, :to =&gt; 1</code>
<code>should_change*</code>	<code>"Post.count", :by =&gt; 1</code>
<code>should_assign_to*</code>	<code>:post, :equals =&gt; "@post"</code>
<code>should_assign_to*</code>	<code>:post, :class =&gt; Post</code>
<code>should_filter_params</code>	<code>:password, :ssn</code>
<code>should_render_with_layout*</code>	<code>"special"</code>
<code>should_respond_with_content_type</code>	<code>:rss</code>
<code>should_return_from_session</code>	<code>:user_id, "@user.id"</code>
<code>should_set_the_flash_to*</code>	<code>/created/i</code>
<code>should_redirect_to</code>	<code>"users_url(@user)"</code>
<code>should_render_template</code>	<code>:new   "new.html.erb"</code>
<code>should_render_a_form</code>	Asserts that view contains a form
<code>should_respond_with</code>	<code>:success   :redirect</code>

## Added Assertions

<code>assert_same_elements</code>	<code>[1, 2, 3], [3, 2, 1]</code>
<code>assert_contains*</code>	<code>['a', '1'], /a/</code>
<code>assert_good_value</code>	<code>User, :email, "bob@email.com"</code>
<code>assert_bad_value</code>	<code>User, :birthdate, "Older than you!"</code>
<code>assert_save</code>	<code>User.new(params[:user])</code>
<code>assert_valid</code>	<code>Post.new(params[:post])</code>
<code>assert_sent_email do  email </code>	
	<code>email.subject =~ /activated/ &amp;&amp; email.to.include?('bob@email.com')</code>
<code>end</code>	