Client-Side Frameworks
(starring AngularJS and Ruby on Rails)
Bloomberg BGov

- Launched January 2011 as traditional Rails app
- Transitioned to AngularJS Fall 2013-Present
- Large, diverse, data-intensive
<table>
<thead>
<tr>
<th>Member Name</th>
<th>Party</th>
<th>Percentage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank R. Wolf</td>
<td>R</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Barbara J. Lee</td>
<td>D</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Lynn C. Woolsey</td>
<td>D</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Wally Herger</td>
<td>R</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Ed Whitfield</td>
<td>R</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Pete J. Visclosky</td>
<td>D</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Dan Burton</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

House HRES465 Providing for consideration of the bill (H.R. 7) to prohibit taxpayer funded abortions, and providing for consideration of the conference report to accompany the bill (H.R. 2642) to provide for the reform and continuation of agricultural and other programs of the Department of Agriculture through fiscal year 2018, and for other purposes.

Vote 26: On Ordering the Previous Question 222-194
Obama Seeking Pena Nieto’s Help in Pressing for Opening of Cuba

President Obama wants to enlist Mexican President Enrique Pena Nieto, who arrives today for his first White House meeting, in pressuring Cuba to move toward democracy.
What’s a client-side framework?

- Render JSON server-side, construct views client-side
- Often avoid page reloads
- Lots of unrelated jargon, sacred cows, misinformation
Client-Side MVC
Why use a client-side framework?
New Survey

Name
Rails Survey

Question remove
How many applications do you have in production?

Answer None remove
Answer 1 remove
Answer 2 remove
Answer 3 or more remove

Add answer

Question remove
Which JavaScript framework do you prefer?

Answer Prototype remove
<fieldset>
  <label for="survey_questions_attributes_0_answers_attributes_0_content">Answer</label>
  <input id="survey_questions_attributes_0_answers_attributes_0_content" name="survey[questions_attributes][0][answers_attributes][0][content]" size="30" type="text">
  <input id="survey_questions_attributes_0_answers_attributes_0__destroy" name="survey[questions_attributes][0][answers_attributes][0][__destroy]" type="hidden" value="false">
  <a href="#" class="remove_fields">remove</a>
</fieldset>
Form Submission

Form Data

- utf8: ✓
- _method: put
- authenticity_token: hDl1Fox1KyXq1ev3RvhT7euijxj3CJSxGP9jxkgCMmQ=
- survey[name]: What's your favorite Pizza?
- survey[questions_attributes][0][content]: What's your favorite kind of cheese?
- survey[questions_attributes][0][_destroy]: 0
- survey[questions_attributes][0][answers_attributes][0][content]: mozzarella
- survey[questions_attributes][0][answers_attributes][0][_destroy]: false
- survey[questions_attributes][0][answers_attributes][0][id]: 10
- survey[questions_attributes][0][id]: 4
“Solution (js)”

```javascript
function remove_fields(link) {
    $(link).prev("input[type=hidden]").val("1");
    $(link).closest(".fields").hide();
}

function add_fields(link, association, content) {
    var new_id = new Date().getTime();
    var regexp = new RegExp("new_"+association, "g")
    $(link).parent().before(content.replace(regexp, new_id));
}
```
"Solution" (ruby)

module ApplicationHelper
  def link_to_remove_fields(name, f)
    f.hidden_field(:_destroy) + link_to_function(name, "remove_fields(this)")
  end

  def link_to_add_fields(name, f, association)
    new_object = f.object.class.reflect_on_association(association).klass.new
    fields = f.fields_for(association, new_object, :child_index => "new_#{association}"), do |builder|
      render(association.to_s.singularize + "_fields", :f => builder)
    end
    link_to_function(name, "add_fields(this, "#{association}", "#{escape_javascript(fields)}")")
  end
end
```
content = $this.data('association-insertion-template'),
insertionMethod = $this.data('association-insertion-method') || $this.data('association-insertion-position') || 'be
insertionNode = $this.data('association-insertion-node'),
insertionTraversal = $this.data('association-insertion-traversal'),
count = parseInt($this.data('count'), 10),
regexp_braced = new RegExp('\\[new_ + assoc + '\]|(.*?\\s)', 'g'),
regexp_underscored = new RegExp('_new_ + assoc + '_((\\w*)', 'g'),
new_id = create_new_id(),
new_content = content.replace(regexp_braced, newcontent_braced(new_id)),
new_contents = [];

if (new_content == content) {
  regexp_braced = new RegExp('\\[new_ + assocs + '\]|(.*?\\s)', 'g');
  regexp_underscored = new RegExp('_new_ + assocs + '_((\\w*)', 'g');
  new_content = content.replace(regexp_braced, newcontent_braced(new_id));
}
new_content = new_content.replace(regexp_underscored, newcontent_underscored(new_id));
new_contents = [new_content];
count = (isNaN(count) ? 1 : Math.max(count, 1));
count -= 1;
while (count) {
  new_id = create_new_id();
  new_content = content.replace(regexp_braced, newcontent_braced(new_id));
  new_content = new_content.replace(regexp_underscored, newcontent_underscored(new_id));
  new_contents.push(new_content);
```
Solution! (AngularJS)

http://plnkr.co/edit/yhAzTguiJKLRvHL8cXVX?p=info

```html
<fieldset ng-repeat="answer in question.answer_attributes">
  <input ng-model="answer.content" />
  <a href="" ng-click="question.answer_attributes.remove(answer)"">remove</a>
</fieldset>
<a href="" ng-click="question.answer_attributes.push([{}])">Add Field</a>
```
When not to go client-side

- No need to reinvent the wheel
- Many apps don’t require the additional complexity
- Frameworks are much less mature

It's a perversion to think that responding to Ajax with HTML fragments instead of JSON is somehow dirty. It's simple, clean, and fast.
CRUD App

- Build a Blog
- MVC
- REST

POSTS

<table>
<thead>
<tr>
<th>All Posts</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Headline</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGov News Story</td>
<td>Edit</td>
<td>Delete</td>
</tr>
<tr>
<td>Another Post</td>
<td>Edit</td>
<td>Delete</td>
</tr>
<tr>
<td>Blog!</td>
<td>Edit</td>
<td>Delete</td>
</tr>
</tbody>
</table>
Gotchas

- Huge controllers
- Abuse of $watch/$emit/$broadcast
- Scope confusion
- Lack of Community Standards / Fundamentals
- Rapidly evolving; backwards-incompatible
- “Magical”
Directives

- Isolate re-usable functionality
- Drop down to jQuery
- Expressive HTML
- Drag/Drop: http://jsfiddle.net/mrajcok/5Mzda/
- Like javascript, but awesome
- New Rails default
Bower

- Package management
- “Bundler for javascript”
- Still just a javascript file
Thanks!

- Lee Richmond – lrichmond1@bloomberg.net
- Wade Tandy – wtandy@bloomberg.net