Selector-based View Composition

William Zeller and Edward Felten
Princeton University
http://svc.from.bz

What is SVC?

- Modification to Model-View-Controller (MVC) architecture
- Provides automatic progressive enhancement (Ajaxification) of web pages.

State of Ajax on the Web

- Allows interactivity
- Expected by users
- Only works if browser supports JavaScript

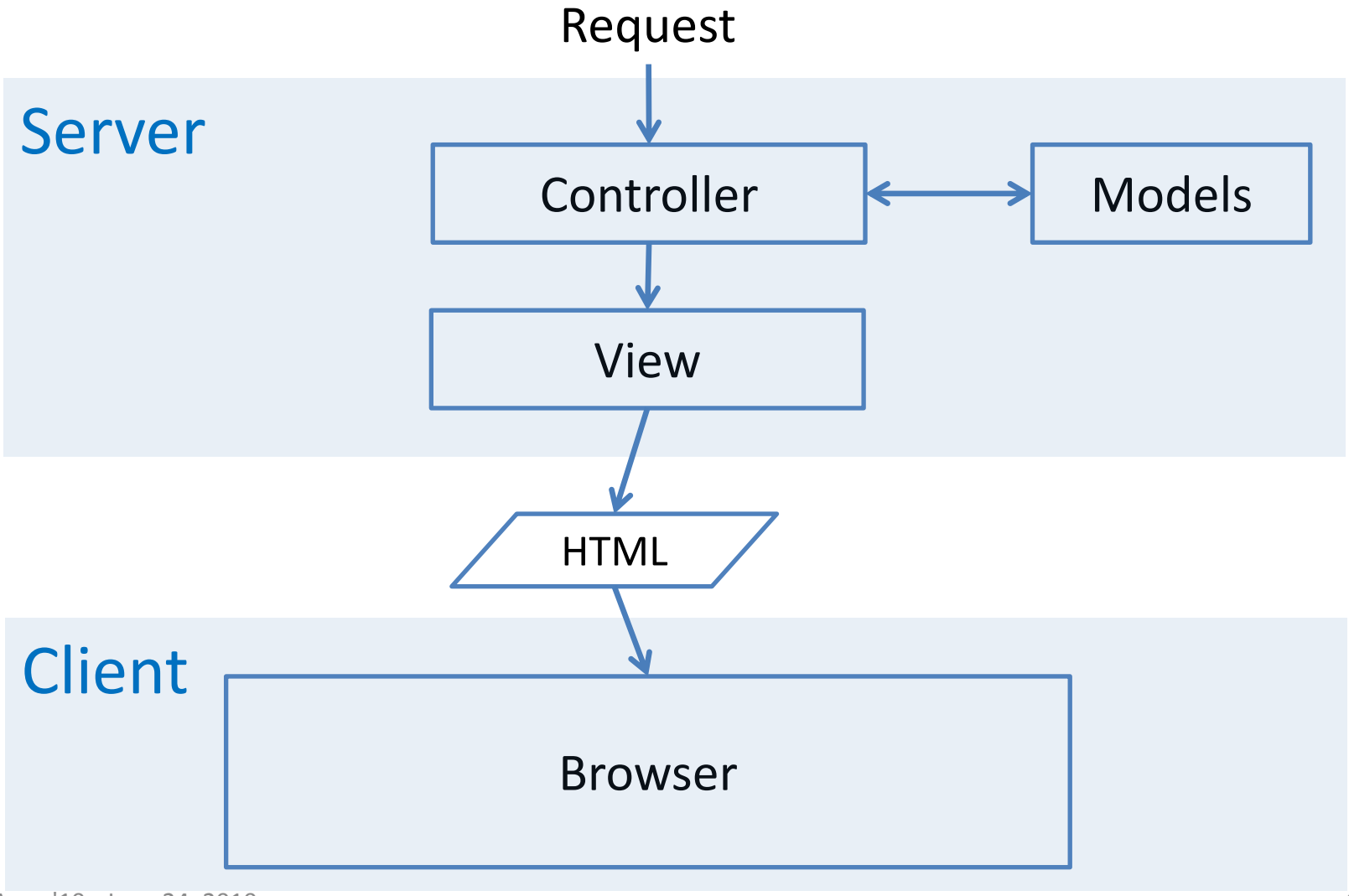
Ajax and Non-JavaScript Browsers

- Why support non-JavaScript browsers?
  - JS may be disabled
  - Users with disabilities
  - **Search engines**
  - Other annoyances if only JS supported
    - E.g., “Open in new tab” may not work

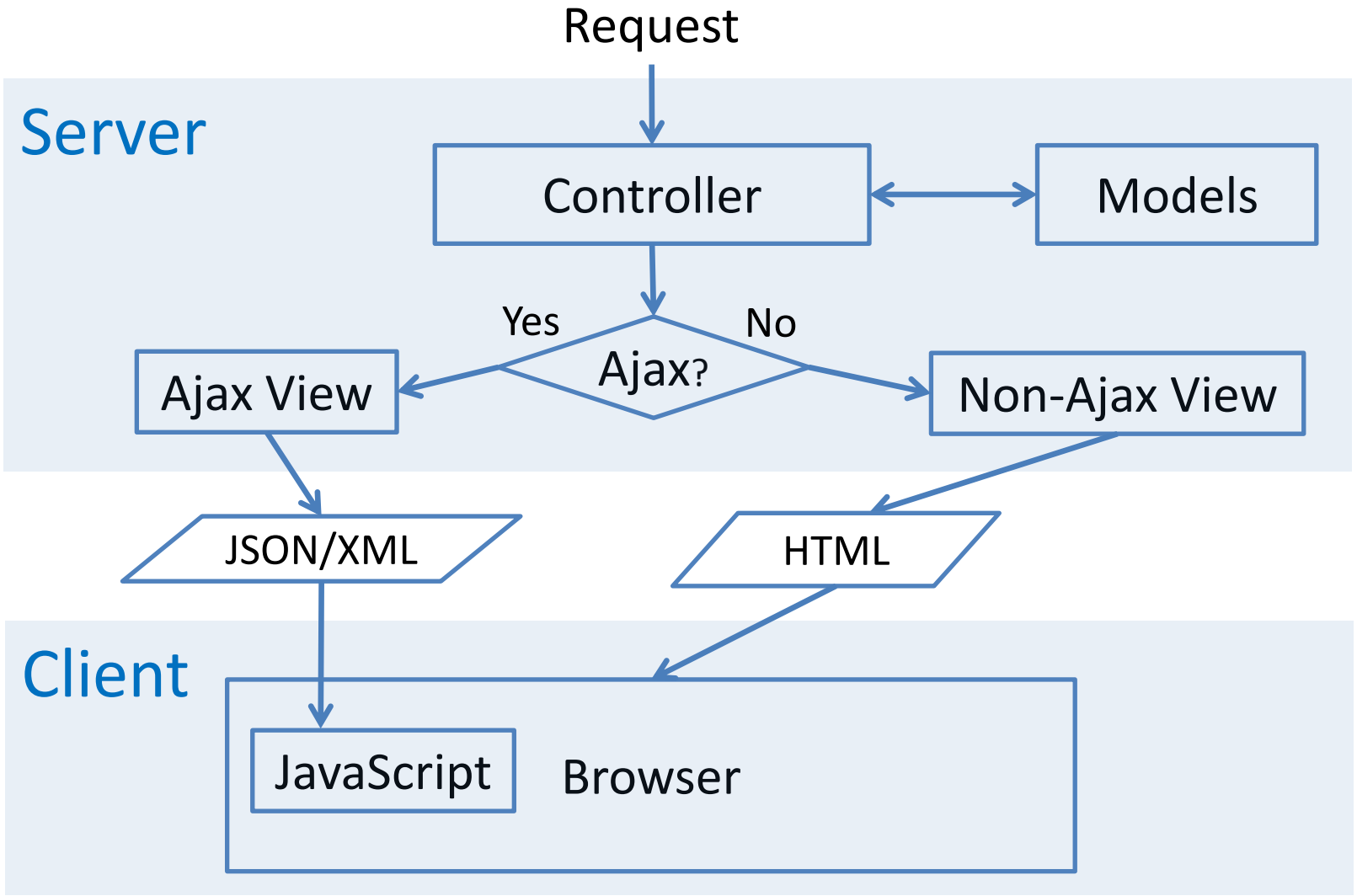
Progressive Enhancement

- Standard "best practice" for supporting both JS and non-JS browsers
- Idea:
  1. Create working, non-JS version of site
  2. Add JavaScript to "enhance site" by adding interactivity
- Difficult
  - All Ajax functionality needs to be duplicated

MVC before Progressive Enhancement



MVC after Progressive Enhancement



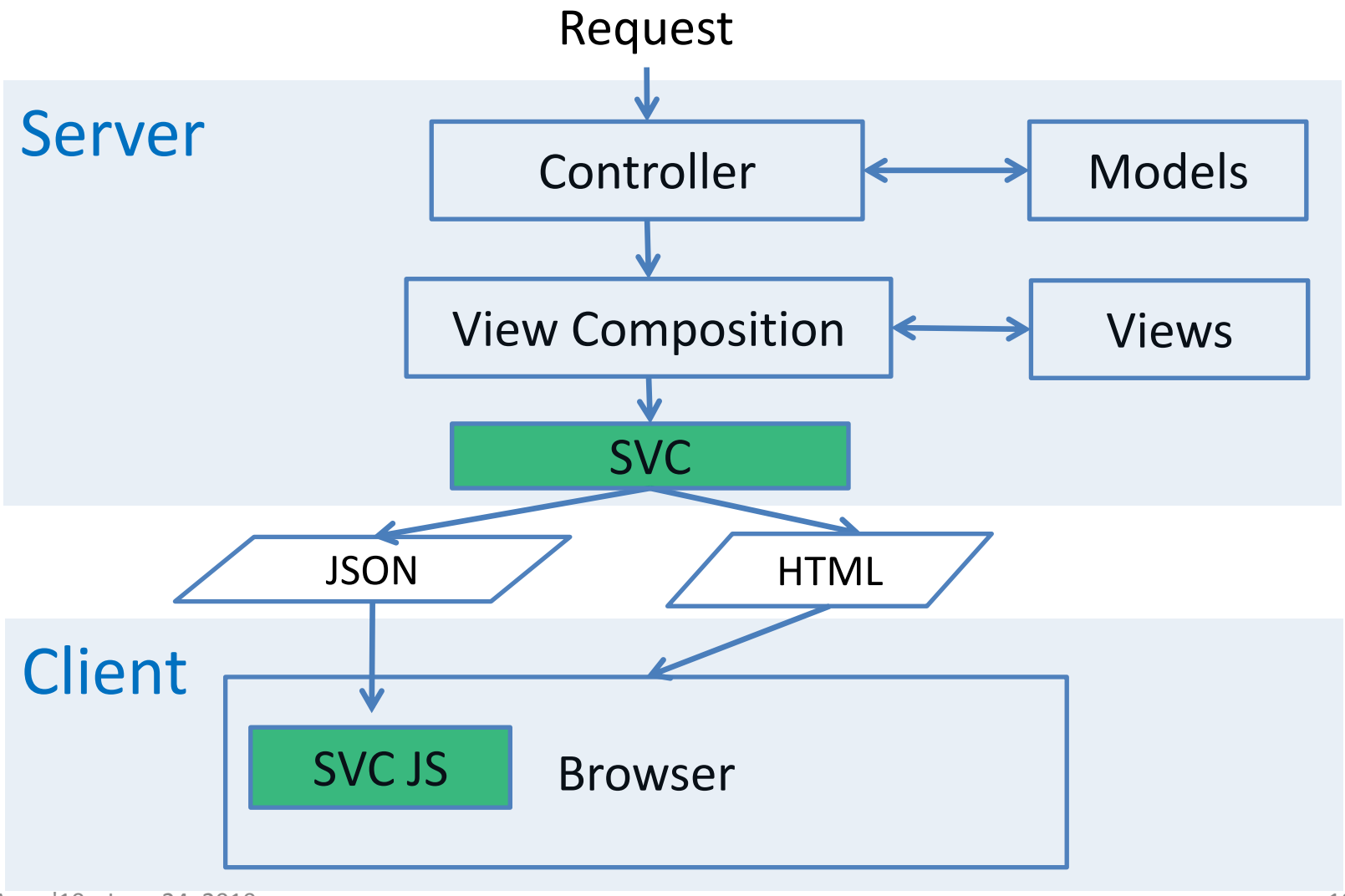
SVC

- Provided as server-side API
- API is used to *compose views* together
- (CSS) *selectors* are used to identify the point of composition
- Advantages:
  - Write view update code once
  - No need to write JavaScript (for actions supported by SVC)
  - Language agnostic
  - Existing template code and JavaScript unaffected

Why Selectors?

- Selectors, e.g.,
  - `#foo` (element with id "foo")
  - `.foo` (elements with class "foo")
  - `div > a` (select all 'a' elements that are children of 'div' elements)
- Familiar to developers (CSS)
- Becoming more common in JS frameworks
  - Closure, Dojo, jQuery, MooTools, Prototype, YUI

SVC


Request
Server
Controller
Models
View Composition
Views
SVC
JSON
HTML
Client
SVC JS
Browser

What does SVC look like?

```
$svc->initial('example'); // assume example is
                          //<p id="msg"></p>
$svc->text('#msg', 'My message');
$svc->addClass('p', 'highlight');
```

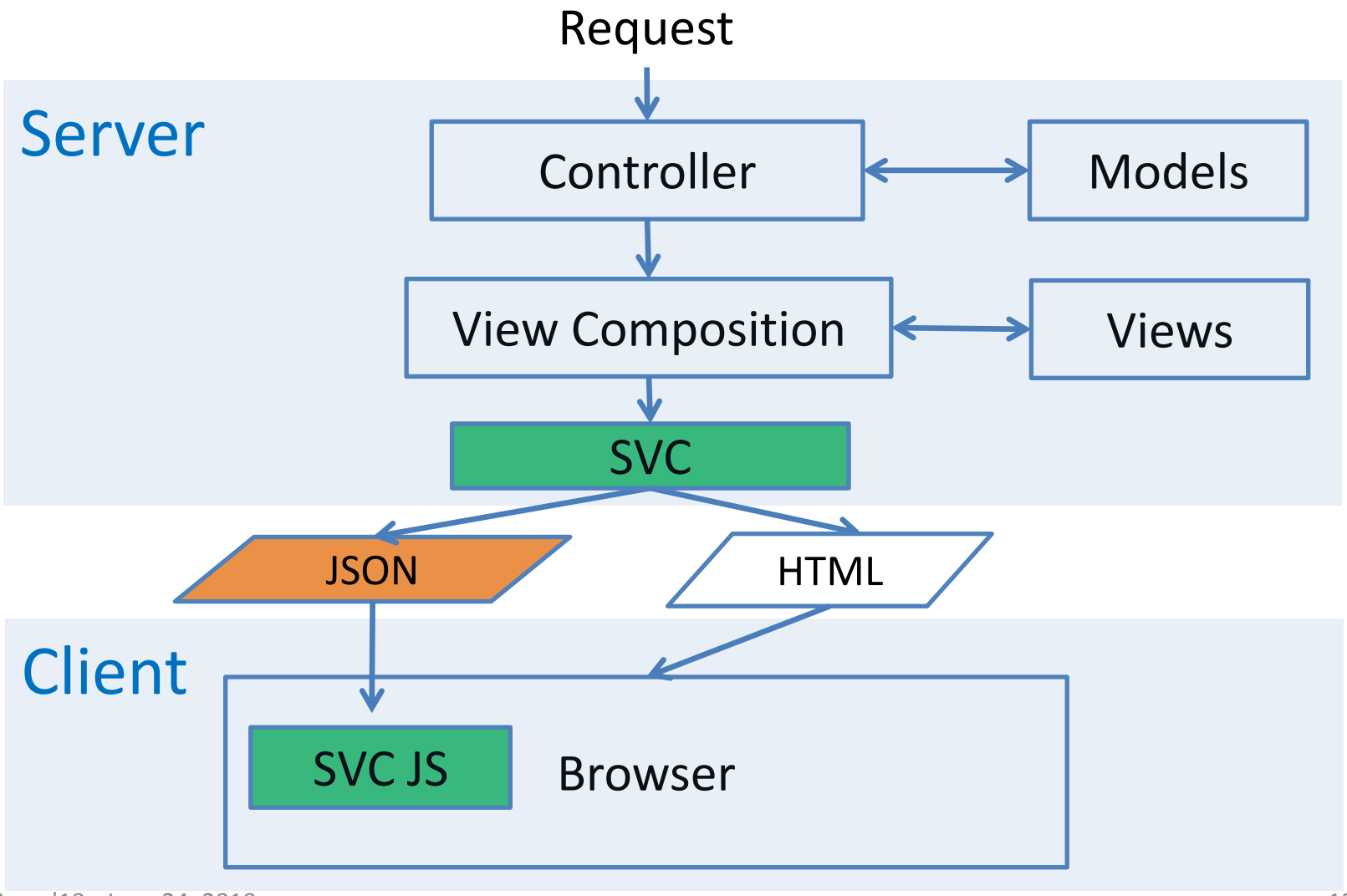
- Non-Ajax request response (HTML):

```
<p id="msg" class="highlight">My message</p>
```

- Ajax Request response (JSON)

```
[['text', ['#msg', 'My message']],
 ['addClass', ['p', 'highlight']]]
```

SVC


Request
Server
Controller
Models
View Composition
Views
SVC
JSON
HTML
Client
SVC JS
Browser

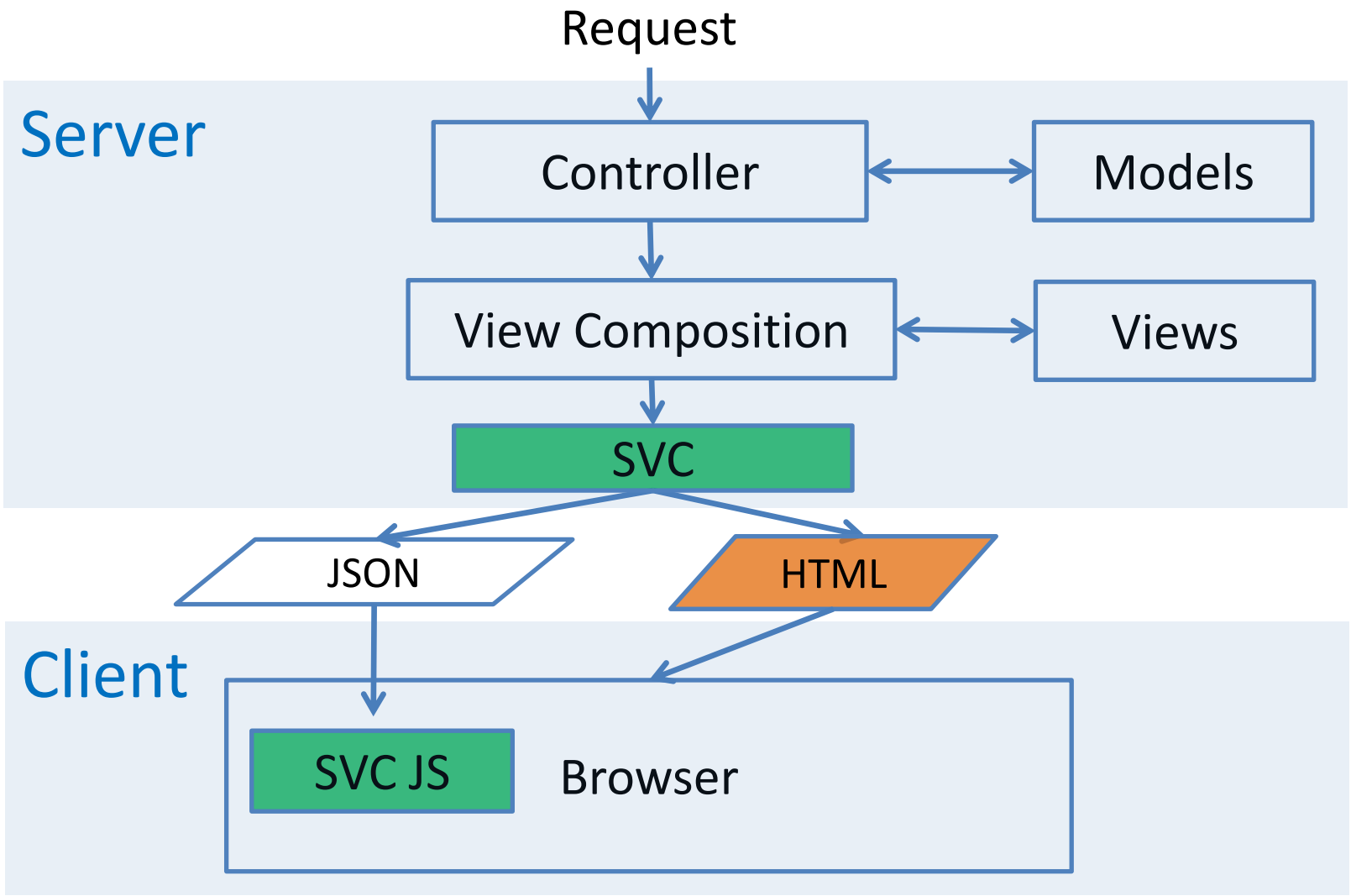
SVC Client-side JavaScript

- SVC client-side JS applies JSON to current document
- In previous example:

```
[['text', ['#msg', 'My message']],
['addClass', ['p', 'highlight']]]
```

  - Sets the text of any element with ID 'msg'
  - Adds class "highlight" to any paragraph tag.
- Size: 1.7KB, requires jQuery (could be written on top of other libraries as well)

SVC


Request
Server
Controller
Models
View Composition
Views
SVC
JSON
HTML
Client
SVC JS
Browser

How does SVC perform actions on server-side?

- JavaScript is used to update view on client-side, but what about server?
- Need to compose view on server-side for non-JS requests
  - Requires selector support and DOM manipulation
- Our version of SVC uses WebKit (WebCore)
  - Actions implemented in C++ extension to PHP

How does SVC Progressively Enhance?

- SVC can now respond to both Ajax and non-Ajax requests
- But how are Ajax requests generated?
- Developer has used SVC on server
- Writes HTML without JavaScript in mind (e.g., ...a href="/foo">Link)

How does SVC Progressively Enhance?

- On server-side, SVC adds the class "svc_rewrite" to any element which should use JavaScript to load request
- On client-side, SVC adds `onclick` or `onsubmit` event handlers to any element with the class "svc_rewrite"

When does SVC Progressively Enhance Pages?

- SVC does not know when loading a link using Ajax is appropriate
  - We do not want to transmit page to server to see if manipulating the DOM makes sense
- SVC provides API to define which links should be rewritten to use Ajax

```
$svc->rewrite('a', 'b')
```

(On page "b", rewrite all links to "a")

What can be done with SVC?

- We implemented the following actions:
  - addClass
  - append
  - attr
  - css
  - html
  - prepend
  - remove
  - removeClass
  - text

What can be done with SVC?

- SVC can be easily extended.
- Could be written once and distributed as SVC plugins
- Only requirement is that action can be performed on client and server.

How are Actions Added to SVC?

- Implement action on client and server.
- E.g., add `before` action
  - On server, use PHP or scripting language to perform update to view
  - On client, use JavaScript to perform equivalent action
  - SVC users can then call `before` in any of their code.

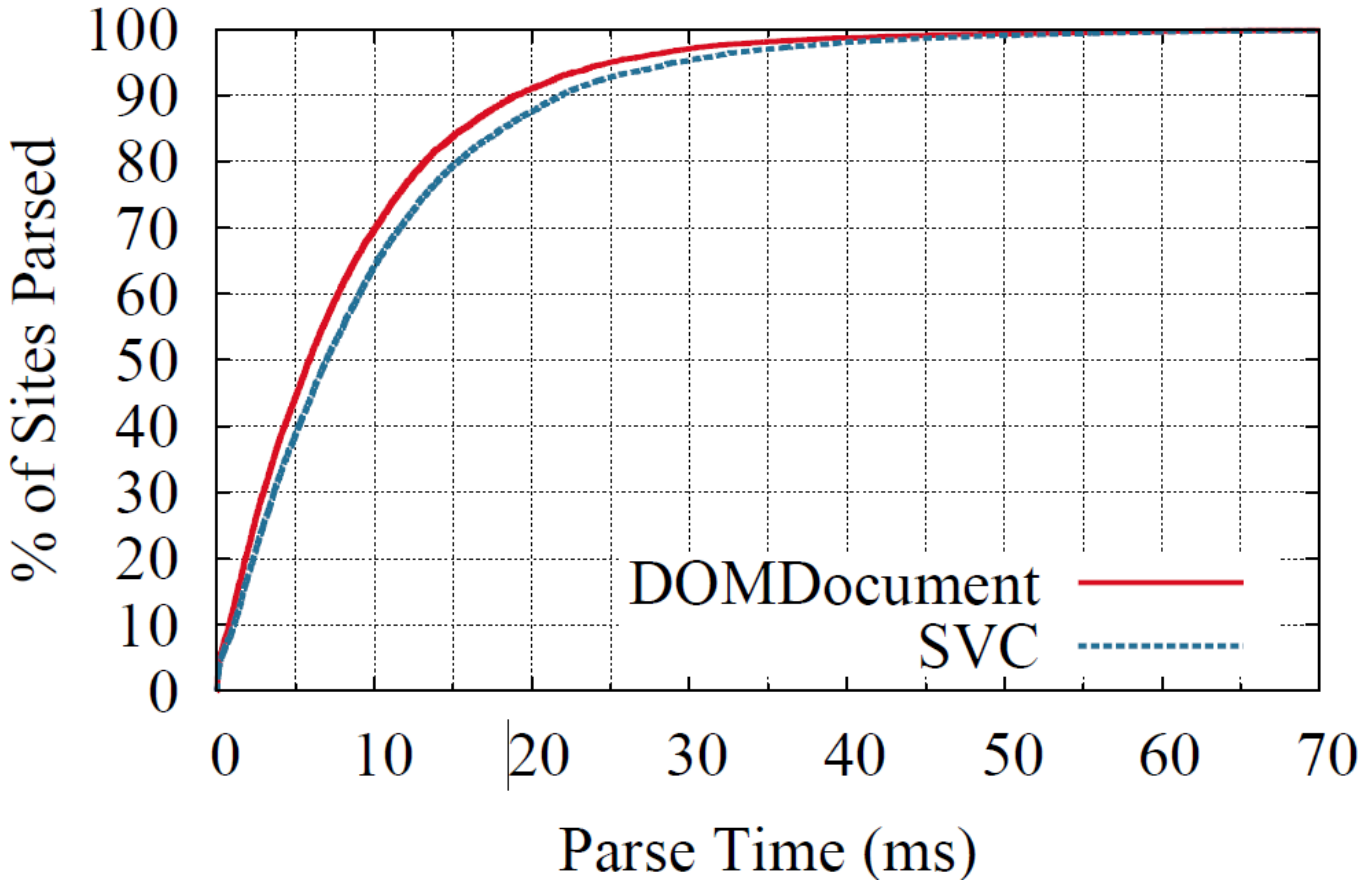
Adding non-DOM Actions to SVC

- Actions we've seen manipulate the DOM
- SVC actions not limited to DOM manipulation
  - Only requirement is to server/client equivalence
- E.g., redirect
  - On server, output HTTP header
  - On client, call window.location

SVC Overhead

- Non-Ajax requests (initial and non-JS requests)
  - Page needs to be constructed on server
  - Requires parsing, running selectors and actions
  - DOM output can be cached.
- Ajax requests
  - List of SVC actions sent directly to client
  - Modifications performed by client JS
  - Essentially no server overhead

SVC Server-side Parsing Overhead

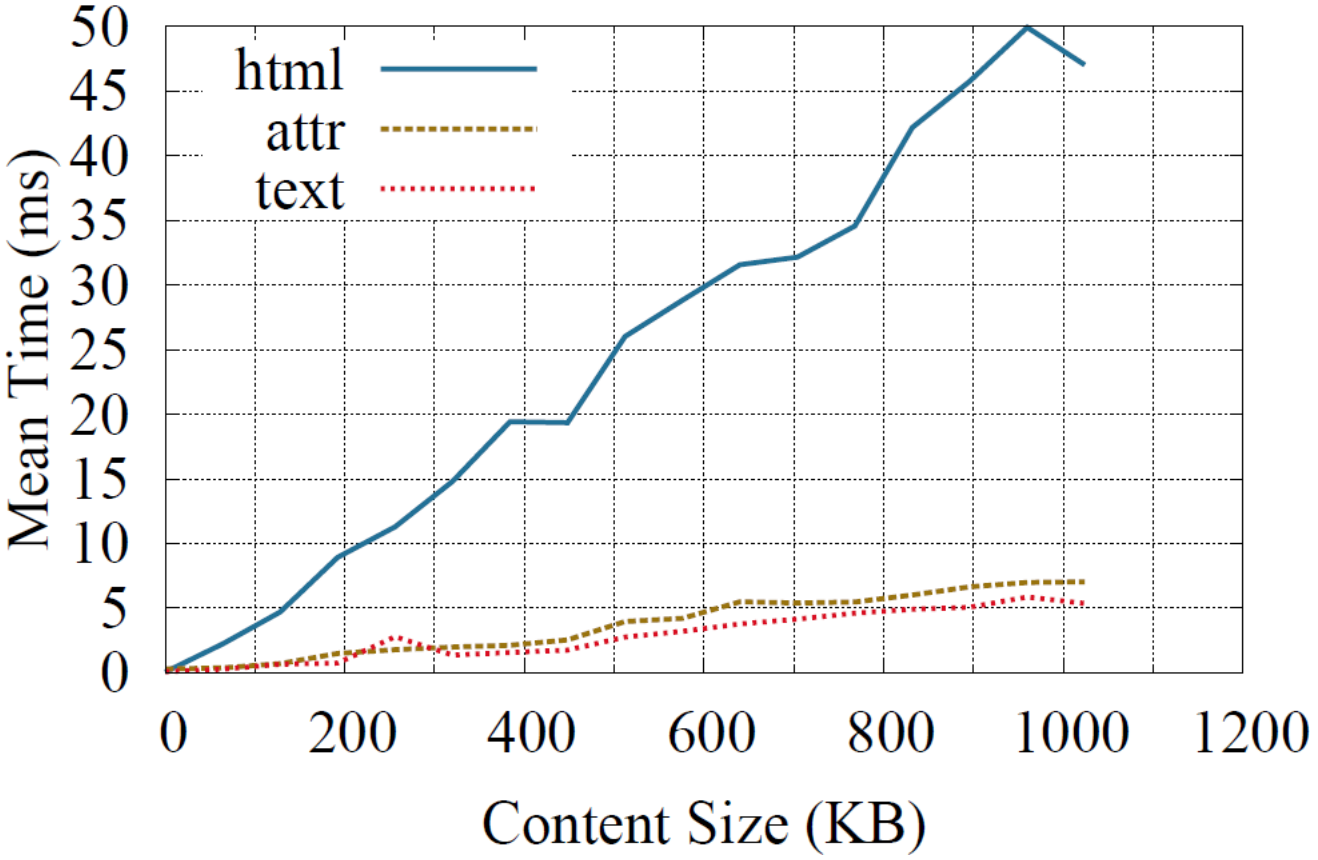


- E.g., SVC parsed 80% of the sites in 15ms
- DomDocument is an alternative PHP HTML parser
- 10,000 most popular websites parsed

SVC Server-side Selector Overhead

- Our SVC implementation uses WebCore's querySelectorAll
- Translated SlickSpeed test to PHP
  - Ran 40 selectors against standard 108KB webpage
  - Ran test 1000 times, mean time of 1.49ms per selector
  - More details in paper

SVC Server-side Action Overhead



- Mean time to insert 0-1MB (64KB increments)
- 100 runs per size

SVC Client-side Overhead

- Our SVC JavaScript wraps jQuery functionality
  - E.g., the literal implementation of addClass

```
addClass: function(selector, className){
    $(selector).addClass(className);
},
```

- Depends on browser, underlying JS library, action complexity

Alternatives to SVC

- GWT, Cappuccino, SproutCore and RJS (Rails JS)
  - Allow server-side creation of JS
  - Do not provide non-JS browser support
- Not aware of any other solution that provides automatic progressive enhancement of pages
- More comparisons in paper

Extending SVC

- Additional Language Support
  - Python prototype has been written
    - Simpler than PHP – only requires lxml library
- Additional client-side libraries
  - Currently based on jQuery
  - Versions could be written for Prototype, Closure, Dojo, etc.

Conclusion

- SVC provides automatic progressive enhancement and compatibility with older browsers
- SVC allows composition of views using selectors
- More info: http://svc.from.bz

Thanks! Questions?

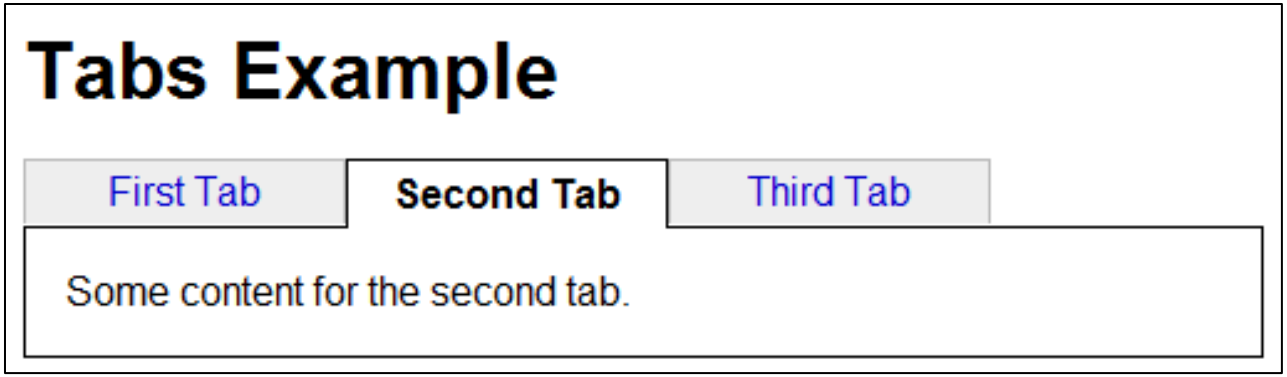
Addendum Example: Tabs

- Page consisting of multiple tabs
- Tabs should load using Ajax when available
  - But work without JavaScript
- Changing tab requires changing:
  - Tab content
  - CSS class in tab header
  - Page title

Tabs Screenshots

- Tab 1

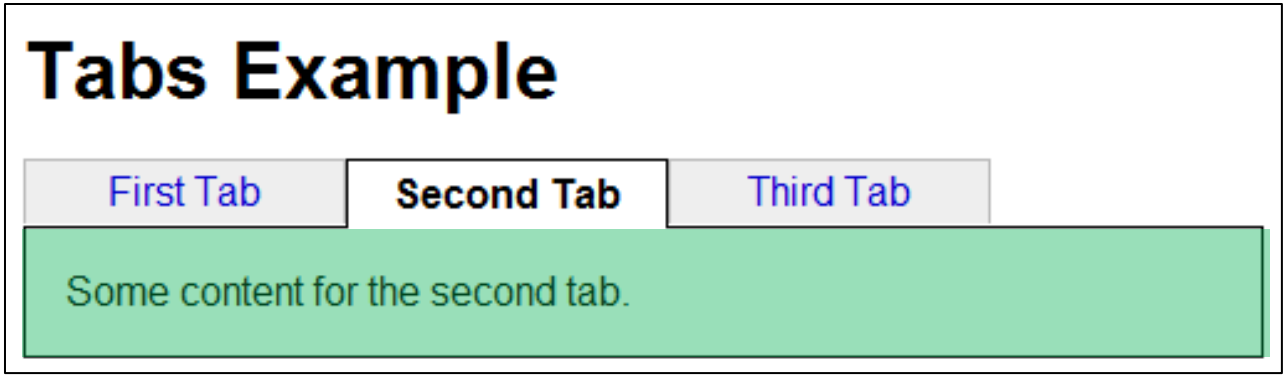
- Tab 2



Tabs Screenshots

- Tab 1

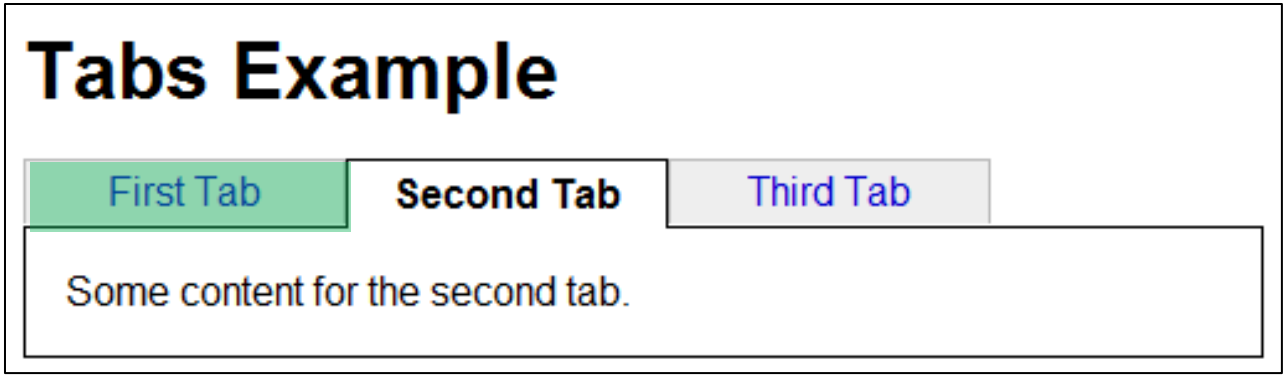
- Tab 2



Tabs Screenshots

- Tab 1

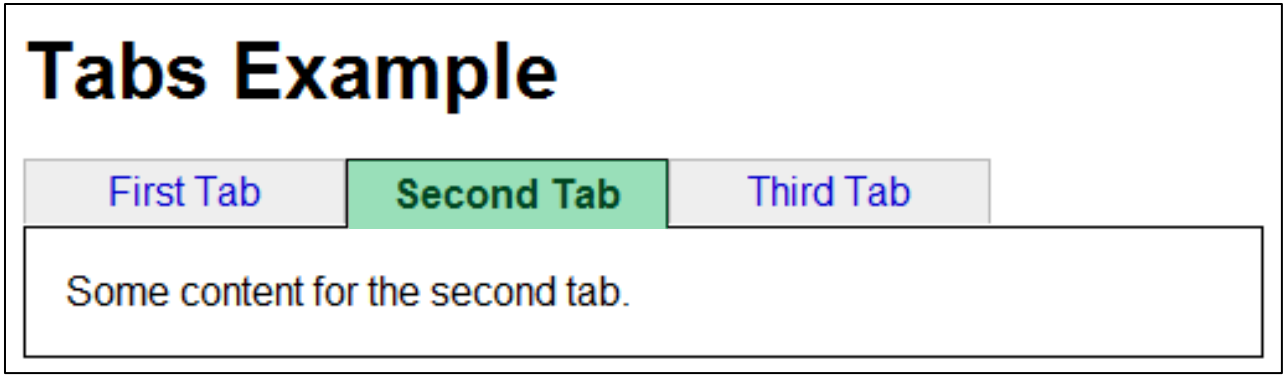
- Tab 2



Tabs Screenshots

- Tab 1

- Tab 2



First Tab Page HTML

```
<html>
  <head>
    <title>First tab page</title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
<body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
    <li class="selected"><a href="/tab1">First Tab</a></li>
    <li><a href="/tab2">Second Tab</a></li>
    <li><a href="/tab3">Third Tab</a></li>
  </ul>
</div>
(continued on next page...)
```

First Tab Page HTML


```
<html>
  <head>
    <title>First tab page</title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
<body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
    <li class="selected"><a href="/tab1">First Tab</a></li>
    <li><a href="/tab2">Second Tab</a></li>
    <li><a href="/tab3">Third Tab</a></li>
  </ul>
</div>
(continued on next page...)
```

First Tab Page HTML

```
<html>
  <head>
    <title>First tab page</title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
<body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
    <li class="selected"><a href="/tab1">First Tab</a></li>
    <li><a href="/tab2">Second Tab</a></li>
    <li><a href="/tab3">Third Tab</a></li>
  </ul>
</div>
(continued on next page...)
```

Tabs header

First Tab Page HTML

```
<html>
  <head>
    <title>First tab page</title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
<body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
    <li class="selected"><a href="/tab1">First Tab</a></li>
    <li><a href="/tab2">Second Tab</a></li>
    <li><a href="/tab3">Third Tab</a></li>
  </ul>
</div>
(continued on next page...)
```

CSS class showing selected tab

First Tab Page HTML

```
(continued...)
<div id="content">
  First tab content <br/><br/>

  <b>Bold</b> first tab message.
</div>

</body>
</html>
```

First Tab Page HTML

Tab content

```
(continued...)
<div id="content">
   First tab content <br/><br/>

   <b>Bold</b> first tab message.
</div>

</body>
</html>
```

Tabs Page without SVC

- Create site without JavaScript
- Use JavaScript to add Ajax support
  - This is the Progressive Enhancement step

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
  }
  function tab2() {
  }
  function tab3() {
  }
}
```

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
  }
  function tab2() {
  }
  function tab3() {
  }
}
```

tab1() runs when /tab1 is requested, etc

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
  }
  function tab2() {
  }
  function tab3() {
  }
}
```

We don't show tab3() due to space limitations.

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
  }
  function tab2() {
  }
}
```

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                  'content' => view('tab1_content'));
  }
  function tab2() {
  }
}
```

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                  'content' => view('tab1_content'));
  }
  function tab2() {
  }
}
```

Define array with data which will be sent to the view

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                  'content' => view('tab1_content'));
  }
  function tab2() {
    $data = array('title' => 'Second tab page',
                  'tab_num' => 2,
                  'content' => view('tab2_content'));
  }
}
```

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                  'content' => view('tab1_content'));
     return view('tab_template', $data);
  }
  function tab2() {
    $data = array('title' => 'Second tab page',
                  'tab_num' => 2,
                  'content' => view('tab2_content'));
     return view('tab_template', $data);
  }
}
```

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                  'content' => view('tab1_content'));
     return view('tab_template', $data);
  }
  funct
    $da
                  'tab_num' => 2,
                  'content' => view('tab2_content'));
     return view('tab_template', $data);
  }
}
```

Call view "tab_template" with data array.

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                  'content' => view('tab1_content'));
    return view('tab_template', $data);
  }
  functio     () {
    $                                 ond tab page',

                  'content' => view('tab2_content'));
    return view('tab_template', $data);
  }
}
```

Output interpreted view.

Tabs Controller

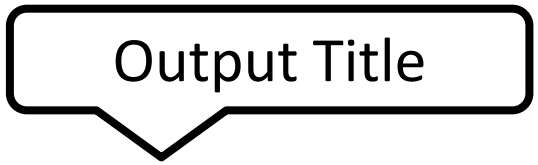
```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                   'content' => view('tab1_content'));
     return view('tab_template', $data);
  }
  function tab2() {
    $data = array('title' => 'Second tab page',
                  'tab_num' => 2,
                  'content' => view('tab2_content'));
     return view('tab_template', $data);
  }
}
```

Template (tab_template)

```
<html>
  <head>
    <title><?=$title?></title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
<body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
    <li <? if ($tab_num == 1) { ?> class="selected" <? } ?>
      id="tab_1"><a href="/tab1">First Tab</a></li>
    <li <? if ($tab_num == 2) { ?> class="selected" <? } ?>
      id="tab_2"><a href="/tab2">Second Tab</a></li>
```

(continued on next page...)

Template (tab_template)




```
<html>
  <head>
    <title><?=$title?></title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
<body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
    <li <? if ($tab_num == 1) { ?> class="selected" <? } ?>
      id="tab_1"><a href="/tab1">First Tab</a></li>
    <li <? if ($tab_num == 2) { ?> class="selected" <? } ?>
      id="tab_2"><a href="/tab2">Second Tab</a></li>
```

(continued on next page...)

Template (tab_template)

```
<html>
  <head>
    <title><?=$title?></title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
<body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
    <li <? if ($tab_num == 1) { ?> class="selected" <? } ?>
      id="tab_1"><a href="/tab1">First Tab</a></li>
    <li <? if ($tab_num == 2) { ?> class="selected" <? } ?>
      id="tab_2"><a href="/tab2">Second Tab</a></li>
```

Set CSS class if appropriate

(continued on next page...)

Template (tab_template)

```
(continued...)
    <li <? if ($tab_num == 3) { ?> class="selected" <? } ?>
      id="tab_3"><a href="/tab3">Third Tab</a></li>
 </ul>
</div>

<div id="content"><?=$content?></div>

</body>
</html>
```

Template (tab_template)

```
(continued...)
    <li <? if ($tab_num == 3) { ?> class="selected" <? } ?>
      id="tab_3"><a href="/tab3">Third Tab</a></li>
 </ul>
</div>

<div id="content"><?=$content?></div>

</body>
</html>
```

Set CSS class if appropriate

Template (tab_template)

```
(continued...)
    <li <? if ($tab_num == 3) { ?> class="selected" <? } ?>
      id="tab_3"><a href="/tab3">Third Tab</a></li>
 </ul>
</div>

<div id="content"><?=$content?></div>

</body>
</html>
```

Output content

Tabs Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                   'content' => view('tab1_content'));
     return view('tab_template', $data);
  }
  function tab2() {
    $data = array('title' => 'Second tab page',
                  'tab_num' => 2,
                  'content' => view('tab2_content'));
     return view('tab_template', $data);
  }
}
```

Content Templates

tab1_content

```
First tab content <br/><br/>

<b>Bold</b> first tab message.
```

tab2_content

```
Some content for the second tab.
```

Non-Ajax Tabs Example Complete

- Site works (but no Ajax)
- Let's enhance!

Adding Ajax (Main Template)

```
<html>
  <head>
    <title><?=$title?></title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
<body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
    <li <? if ($tab_num == 1) { ?> class="selected" <? } ?>
      id="tab_1"><a href="/tab1">First Tab</a></li>
...
```

Adding Ajax (Main Template)

Add scripts to progressively enhance

```
<html>
  <head>
    <title><?=$title?></title>
    <script type="text/javascript" src="jQuery.js"></script>
    <script type="text/javascript" src="tabs.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
<body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
    <li <? if ($tab_num == 1) { ?> class="selected" <? } ?>
      id="tab_1"><a href="/tab1">First Tab</a></li>
...
```

Add Ajax to Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                   'content' => view('tab1_content'));
     return view('tab_template', $data);
  }

   ...
}
```

Add Ajax to Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                   'content' => view('tab1_content'));
    if (is_ajax()) {
      return json_encode($data);
    } else {
      return view('tab_template', $data);
    }
  }

   ...
}
```

Add Ajax to Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                   'content' => view('tab1_content'));
    if (is_ajax()) {
      return json_encode($data);
    } else {
      retur
    }
  }

    ...
}
```

If Ajax request, return JSON

Add Ajax to Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                   'content' => view('tab1_content'));
    if (is_ajax()) {
      return json_encode($data);
    } else {
      return view('tab_template', $data);
    }
  }
    ...
}
```

Otherwise, output HTML

Example: Adding Ajax (tabs.js)

```
$(function() {

});
```

Example: Adding Ajax (tabs.js)

```
$(function() {
  var onTabClk = function() {
    // ...
  };

  $('#tab_header a').click(onTabClk);
});
```

Add click handler to tab links

Example: Adding Ajax (tabs.js)

Function to be called when tab link is clicked

```
$(function() {
  var onTabClk = function() {
    // ...
  };

  $('#tab_header a').click(onTabClk);
});
```

Example: Adding Ajax (tabs.js)

```
$(function() {
  var onTabClk = function() {
    return false;
  };

  $('#tab_header a').click(onTabClk);
});
```

Example: Adding Ajax (tabs.js)

```
$(function() {
  var onTabClk = function() {
    var href = $(this).attr('href');
    return false;
  };

  $('#tab_header a').click(onTabClk);
});
```

Extract link URL

Example: Adding Ajax (tabs.js)

```
$(function() {
  var onTabClk = function() {
    var href = $(this).attr('href');
    $.getJSON(href, function(data) {
      // ... process JSON response
    });
    return false;
  };

  $('#tab_header a').click(onTabClk);
});
```

Make JSON request

Example: Adding Ajax (tabs.js)

```
$(function() {
  var onTabClk = function() {
    var href = $(this).attr('href');
    $.getJSON(href, function(data) {
      document.title = data.title;
    });
    return false;
  };

  $('#tab_header a').click(onTabClk);
});
```

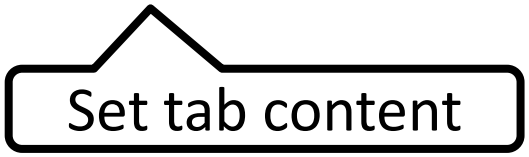
Example: Adding Ajax (tabs.js)

```
$(function() {
  var onTabClk = function() {
    var href = $(this).attr('href');
    $.getJSON(href, function(data) {
      document.title = data.title;
      $('#tab_' + data.tabNum).addClass('selected');
    });
    return false;
  };

  $('#tab_header a').click(onTabClk);
});
```

Set CSS class of new tab

Example: Adding Ajax (tabs.js)

```
$(function() {
  var onTabClk = function() {
    var href = $(this).attr('href');
    $.getJSON(href, function(data) {
      document.title = data.title;
      $('#tab_' + data.tabNum).addClass('selected');
      $('#content').html(data.content);
    });
    return false;
  };

  $('#tab_header a').click(onTabClk);
});
```



Example: Adding Ajax (tabs.js)

```
$(function() {
  var onTabClk = function() {
    var href = $(this).attr('href');
    $.getJSON(href, function(data) {
      document.title = data.title;
      $('.selected').removeClass('selected');
      $('#tab_' + data.tabNum).addClass('selected');
      $('#content').html(data.content);
    });
    return false;
  };

  $('#tab_header a').click(onTabClk);
});
```

Remove "selected" CSS class from all elements

Example: Adding Ajax (tabs.js)

```
$(function() {
  var onTabClk = function() {
    var href = $(this).attr('href');
    $.getJSON(href, function(data) {
      document.title = data.title;
      $('.selected').removeClass('selected');
      $('#tab_' + data.tabNum).addClass('selected');
      $('#content').html(data.content);
    });
    return false;
  };

  $('#tab_header a').click(onTabClk);
});
```

Tabs Example Now Progressively Enhanced

- Tabs will load using Ajax in JS browsers

Cost of Progressive Enhancement

- Requires duplication of view updates.
  - Server-side using templates.
  - Client-side using JavaScript

Examples of Duplication: Page Title

- Server-side code

```
<title><?=$title?></title>
```

- Client-side code

```
document.title = data.title;
```

Examples of Duplication: Tab Headers

- Server-side code

```
<li <? if ($tab_num == 1) { ?> class="selected" <? } ?>
      id="tab_1"><a href="/tab1">First Tab</a></li>
<li <? if ($tab_num == 2) { ?> class="selected" <? } ?>
      id="tab_2"><a href="/tab2">Second Tab</a></li>
<li <? if ($tab_num == 3) { ?> class="selected" <? } ?>
      id="tab_3"><a href="/tab3">Third Tab</a></li>
```

- Client-side code

```
$('.selected').removeClass('selected');
$('#tab_' + data.tabNum).addClass('selected');
```

Examples of Duplication: Page Content

- Server-side code

```
<div id="content"><?=$content?></div>
```

- Client-side code

```
$('#content).html(data.content);
```

Tabs Example Using SVC

- Instead of duplicating view update code, use SVC API to describe how to compose views

Example: Non-SVC Controller

```
<?php
class TabsExample extends Controller {
  function tab1() {
    $data = array('title' => 'First tab page',
                  'tab_num' => 1,
                   'content' => view('tab1_content'));
     return view('tab_template', $data);
  }
  function tab2() {
    $data = array('title' => 'Second tab page',
                  'tab_num' => 2,
                  'content' => view('tab2_content'));
     return view('tab_template', $data);
  }
}
```

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
     //
  }

  function tab2() {
    // ...
  }
}
```

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
     //
  }

  function tab2() {
    // ...
  }
}
```

tabbase() creates an SVC list consisting of only the template "tabbase_template"

Example: Template (tabbase)

```
<html><head>
    <title></title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
</head><body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
      <li id="tab_1"><a href="/tab1">First Tab</a></li>
      <li id="tab_2"><a href="/tab2">Second Tab</a></li>
      <li id="tab_3"><a href="/tab3">Third Tab</a></li>
 </ul>
</div>
<div id="content"></div>
</body></html>
```

Example: Template (tabbase)

```
<html><head>
    <title></title>
    <link rel="stylesheet" t                    ss"/>
</head><body>
<div id="tab_header">
  <h1>Tabs Example</h1>
  <ul>
      <li id="tab_1"><a href="/tab1">First Tab</a></li>
      <li id="tab_2"><a href="/tab2">Second Tab</a></li>
      <li id="tab_3"><a href="/tab3">Third Tab</a></li>
 </ul>
</div>
<div id="content"></div>
</body></html>
```

Like "tab_template", but all content is empty.

Example: SVC Controller

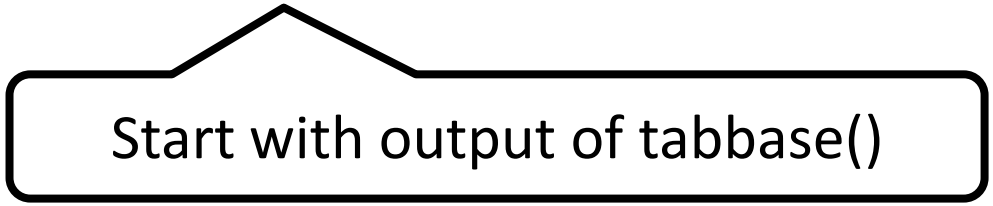
```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
     //
  }

  function tab2() {
    // ...
  }
}
```

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
  }

  function tab2() {
    // ...
  }
}
```

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
  }

  function tab2() {
    // ...
  }
}
```

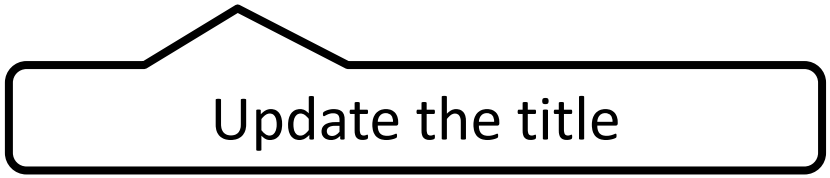


Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
  }

  function tab2() {
    // ...
  }
}
```

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
  }

  function tab2() {
    // ...
  }
}
```



Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
    $this->svc->removeClass('.selected', 'selected');
  }
  ...
}
```

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
    $this->svc->removeClass('.selected', 'selected');
  }
  ...
}
```

Remove the "selected"' class from all elements to clear tab highlighting

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
    $this->svc->removeClass('.selected', 'selected');
    $this->svc->addClass('#tab_1', 'selected');
  }
  ...
}
```

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
    $this->svc->removeClass('.selected', 'selected');
    $this->svc->addClass('#tab_1', 'selected');
  }
  ...
}
```

Add "selected" class to the first tab.

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
    $this->svc->removeClass('.selected', 'selected');
    $this->svc->addClass('#tab_1', 'selected');
    $this->svc->html('#content', view('tab1_content'));
  }
  ...
}
```

Example: SVC Controller

```
<?php
class TabsExample extends Controller {
  function tabbase() {
    $this->svc->initial(view('tabbase_template'));
  }

  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
    $this->svc->removeClass('.selected', 'selected');
    $this->svc->addClass('#tab_1', 'selected');
    $this->svc->html('#content', view('tab1_content'));
  }
  ...
}
```

Update tab content

Example: SVC Controller

```
  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
    $this->svc->removeClass('.selected', 'selected');
    $this->svc->addClass('#tab_1', 'selected');
    $this->svc->html('#content', view('tab1_content'));
  }
 function tab2() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'Second tab page');
    $this->svc->removeClass('.selected', 'selected');
    $this->svc->addClass('#tab_2', 'selected');
    $this->svc->html('#content', view('tab2_content'));
  }
}
```

Example: SVC Controller

```
  function tab1() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'First tab page');
    $this->svc->removeClass('.selected', 'selected');
    $this->svc->addClass('#tab_1', 'selected');
    $this->svc->html('#content', view('tab1_content'));
  }
function tab2() {
    $this->svc->initial('tabbase');
    $this->svc->text('title', 'Second tab page');
    $this->svc->removeClass('.selected', 'selected');
    $this->svc->addClass('#tab_2', 'selected');
    $this->svc->html('#content', view('tab2_content'));
  }
}
```

Update second tab in the same way

Example: SVC Controller

Duplicated code! We can factor it out.

```
function tab1() {
  $this->svc->initial('tabbase');
  $this->svc->text('title', 'First tab page');
  $this->svc->removeClass('.selected', 'selected');
  $this->svc->addClass('#tab_1', 'selected');
  $this->svc->html('#content', view('tab1_content'));
}
function tab2() {
  $this->svc->initial('tabbase');
  $this->svc->text('title', 'Second tab page');
  $this->svc->removeClass('.selected', 'selected');
  $this->svc->addClass('#tab_2', 'selected');
  $this->svc->html('#content', view('tab2_content'));
}
}
```

Example: SVC Controller

```
  function tab1() {
    this->_updateTab('First tab page', 1, 'tab1_content');
  }
  function tab2() {
    $this->_updateTab('Second tab page', 1, 'tab2_content');
  }
  function _updateTab($title, $num, $content) {
    $this->svc->initial('tabbase');
    $this->svc->text('title', $title);
    $this->svc->removeClass('.selected', 'selected');
    $this->svc->addClass('#tab_' . $num, 'selected');
    $this->svc->html('#content', view($content));
  }
}
```

SVC Tabs Example: Done!

- SVC framework will automatically Ajaxify tab loads.
- SVC outputs JSON in response to Ajax requests
  - Transformations are applied on client by SVC JS code
- SVC outputs HTML in response to non-Ajax requests
  - Transformations are applied on server-side by SVC code.
- No JavaScript needs to be written