



Java is a trademark of Sun Microsystems, Inc.

JavaOneSM

Ajaxifying Legacy Web Applications

Anas Mughal

Bluenog

<http://www.bluenog.com>

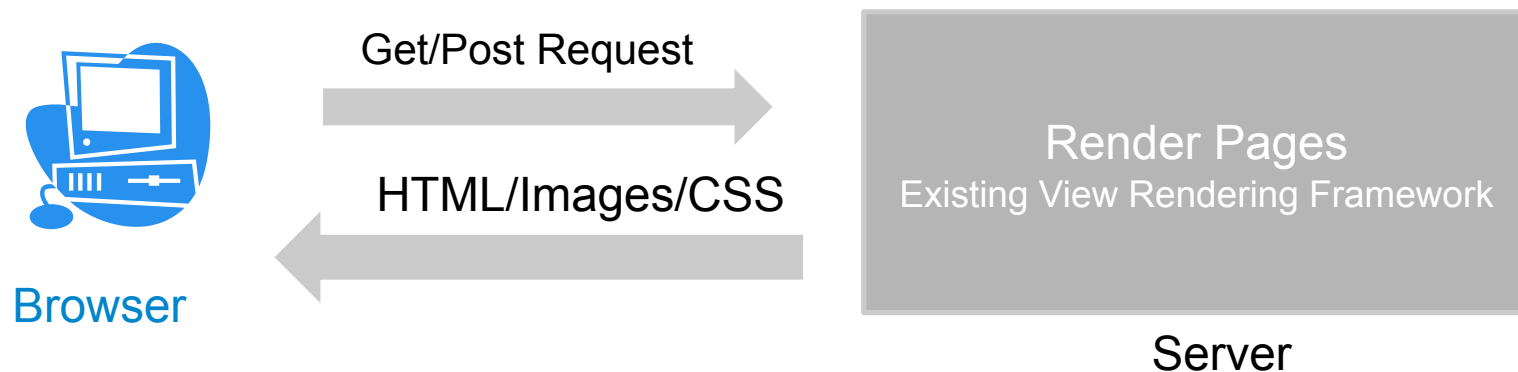


Topics

- > Overview: Legacy vs. RIA
- > Design Approach
- > Data Formats
- > Tools
- > Demo
- > Ajaxifying Portal
- > Portal Demo
- > QA

Web 1.0 Applications

- > Pages are rendered server-side
- > Client-side is stateless
- > Server-side MVC
- > Full-page refresh



Ajax Applications

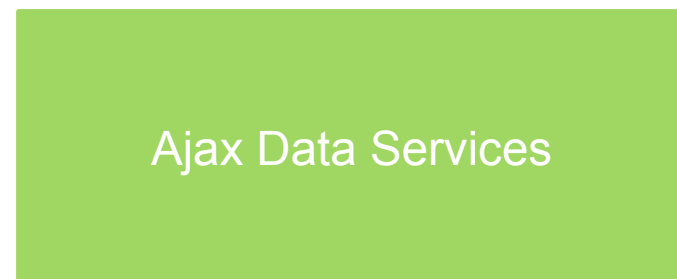
- > State-full client
- > Server-side provides data services
- > Client-side MVC framework
- > No full-page refresh



Browser



XML/JSON

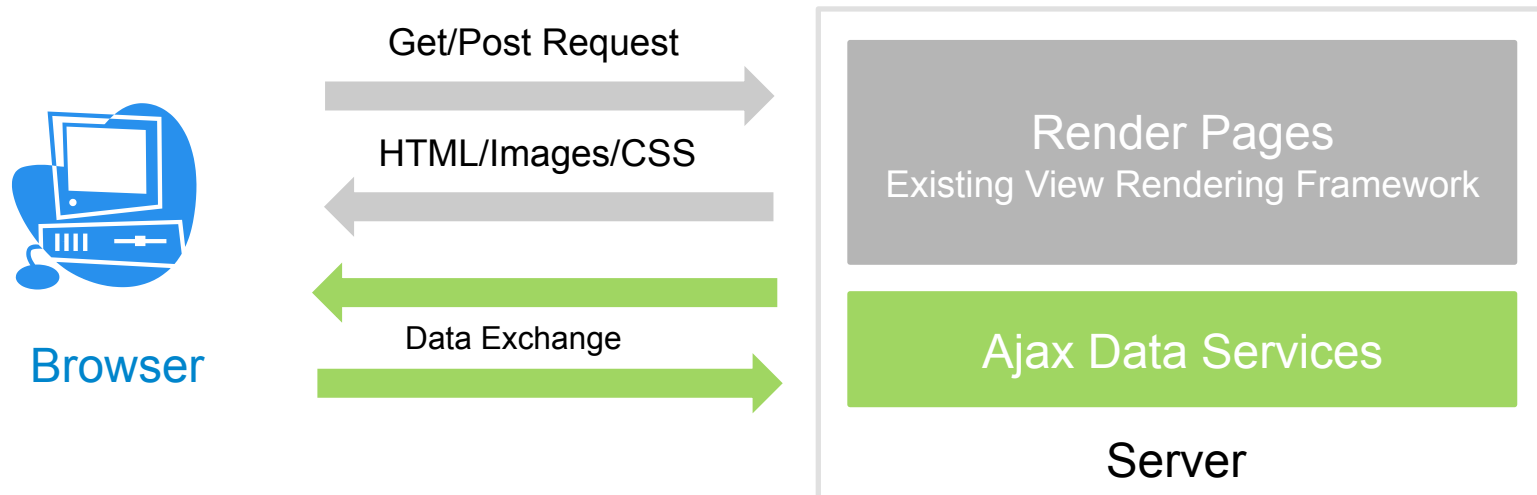


Ajax Data Services

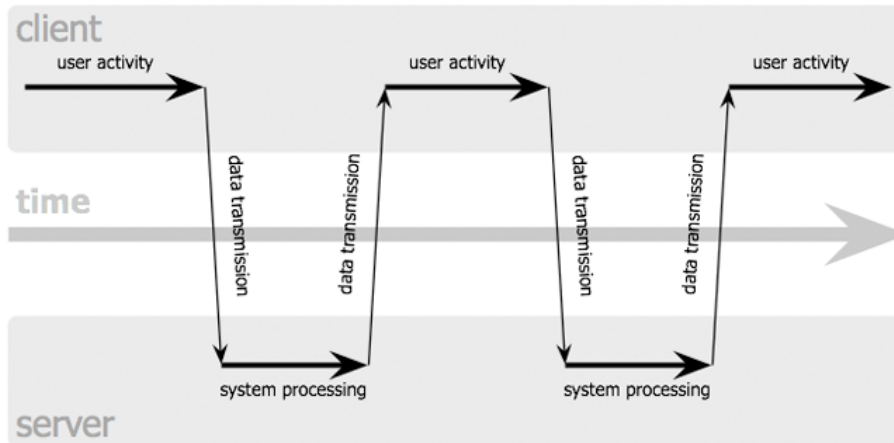
Server

Suggested Approach (for Ajaxifying legacy Applications)

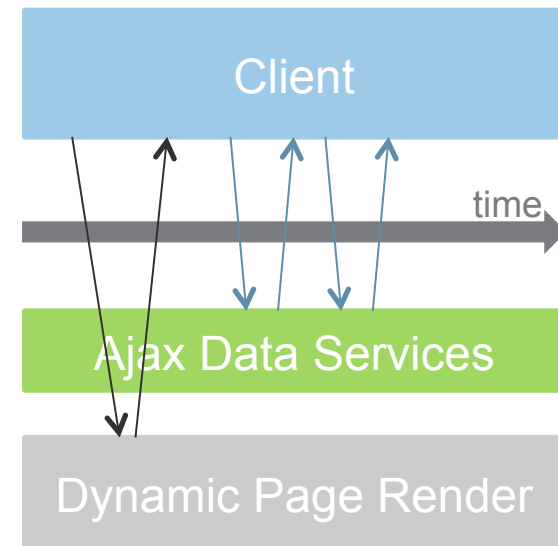
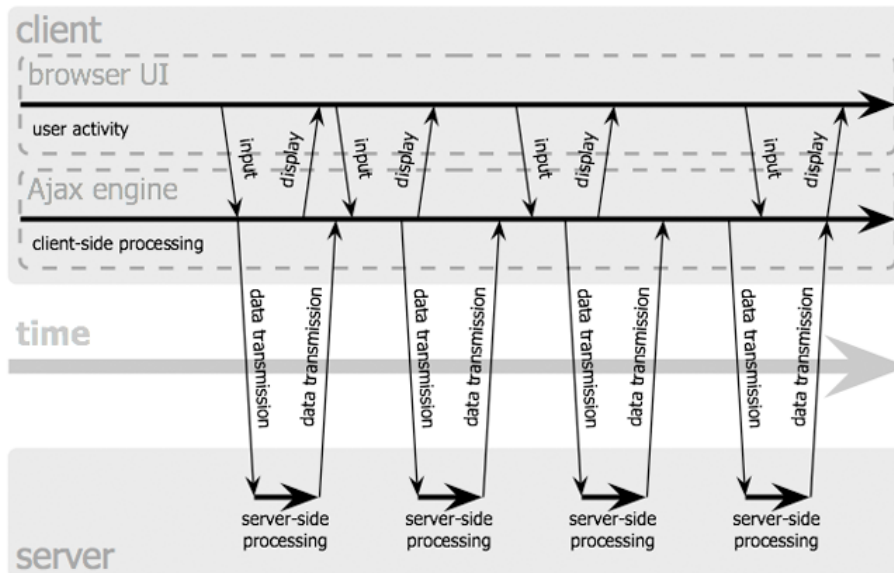
- > Pages rendered server-side
- > Keep Server-side MVC
- > Add Ajax **controls/grids/forms** on client-side
- > Add Ajax Data Services on server-side



classic web application model (synchronous)



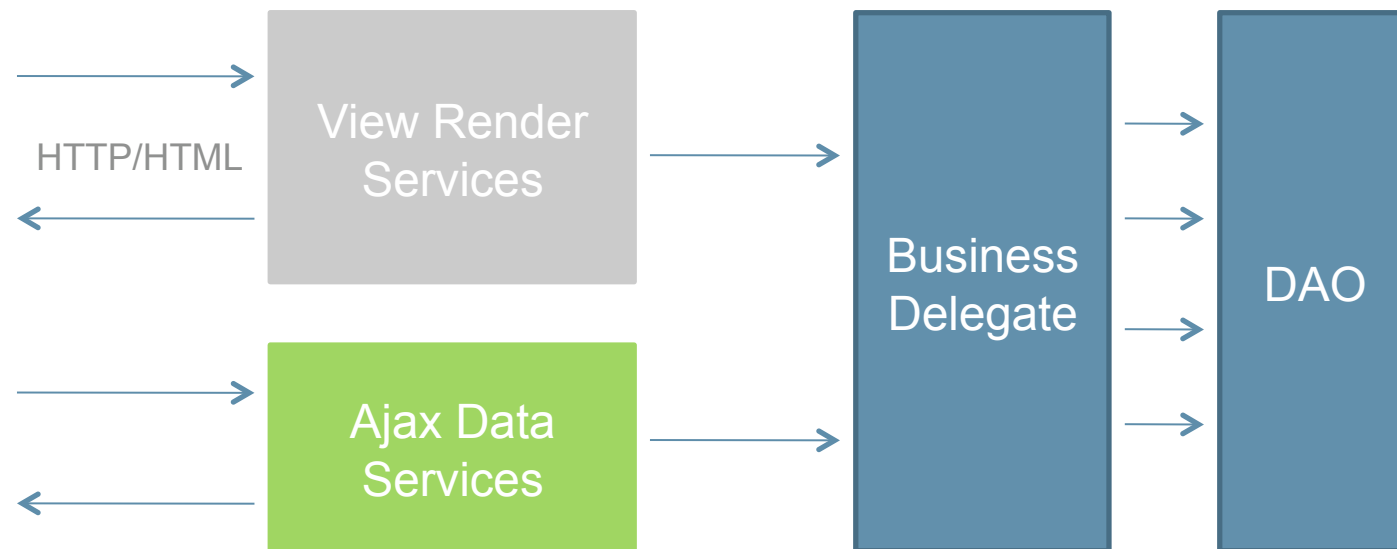
Ajax web application model (asynchronous)



Suggested Model

Design Approach

- > Incorporate Ajax data services layer
- > Integrate Ajax layer with business delegates



Data Formats

> JSON (JavaScript Object Notation)

- Native JavaScript structure.
- Lightweight / compact
- Retains type information

```
{"names": ["Anna Maria", "Maurice"], "count": 3 , "price", 4.75}
```

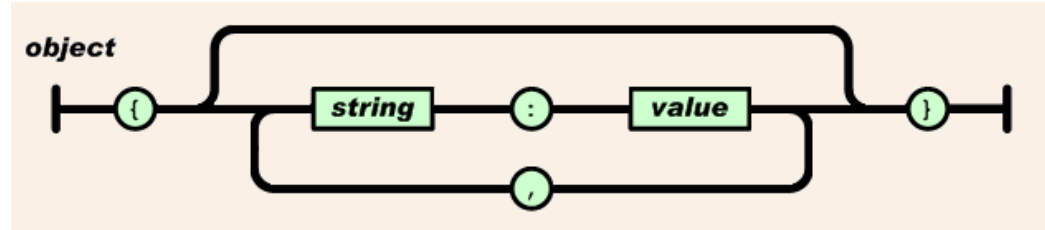
> XML

- Suited for data transformation needs.
- Parsing is CPU intensive.
- DOM may require large memory on client-side.
- Does not retain type information.

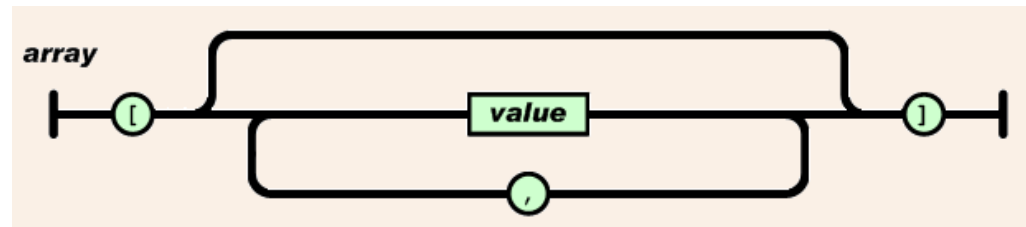
```
<xmldata>
  <names>
    <name>Anna Maria</name>
    <name>Fitzwilliam</name>
    <name>Maurice</name>
  </names>
  <count>3</count>
</xmldata>
```


JSON

- > A collection of name/value pairs.



- > An ordered list of values.



JSON Support on the Server-Side

- > Spring MVC Framework
 - Spring-json View (<http://spring-json.sourceforge.net>)
 - Supports: SOJO and JSON-lib
- > Struts Framework
 - Struts JSON Plugin (<http://tinyurl.com/b87ndu>)
 - Serializes entire action class variables
 - Provides incoming request interceptor
- > Build your own
 - Convert data structures using a JSON library
 - Set content-type to “application/json”

Spring-JSON View

- > Include spring-json library
- > Add view to Spring assembly

```
<bean name="jsonView" class="org.springframework.w.servlet.view.json.JsonView">  
    <property name="jsonWriter"><ref bean="sojoJsonWriter"/></property>  
    ...  
</bean>
```

- > Assign json view in Controller:

```
public ModelAndView  
handleRequest(HttpServletRequest, HttpServletResponse) {  
    Map model = new HashMap();  
    model.put("claims", claimDelegate.findClaims());  
    return new ModelAndView("jsonView", model);  
}
```

Tools

- > Server-side Serializers: (<http://json.org>)
 - SOJO
 - json-lib
- > Formatter: jsonformatter.curiousconcept.com
- > Visualizer: chris.photobooks.com/json/default.htm
- > Validator: www.jsonlint.com , www.jslint.com
- > Editor: jsoneditor.net
- > Editor, Minifier, Tree View: jsoneditor.appspot.com
- > Javascript Editor: SPKet Eclipse Plugin

JS Loader

- > Consistent loading
- > Easy patching of bugs
- > Avoid duplicate loading of libraries

```
<script src="http://myserver/jsloader/jsloader.js"></script>

<script>
    JSLoader.load("extjs", "ext", "2.2.1-g");
    JSLoader.load("myLibs", "validation", "1.0");
</script>
```

Prerequisites

- > Clean your HTML -- HTML Validator
- > Select appropriate JSON library: (<http://json.org>)
 - SOJO, or
 - JSON-LIB
- > Select appropriate Ajax framework:
 - ExtJS
 - Yahoo UI
 - GWT

Demo & Code Walk-thru

- > Display Results using Ajax Grid
(Data is embedded in HTML)
- > Grid Fetching Results Async
(Results not embedded in HTML)
- > Ext Form and Fetching Grid Data Async
(Async Results)
- > Grid Supporting Pagination and Sorting
(Subset of data is fetched based on pagination and sorting)
- > Editable Grid

<http://localhost:9090/ajax-demo1/>

Ajaxifying Portals

- > Layouts
 - Render using panels and layout managers
- > Portlets
 - Securing Ajax Communication
 - Inter-portlet Communication
 - JSLoader

Securing Portlet Ajax Communication

- > JSR 168 vs JSR 286
- > Jetspeed2 -- resourceURL

```
<portlet:renderURL var="resourceURL">  
  <portlet:param name="org.apache.jetspeed.portlet.resource.url" value="av"/>  
</portlet:renderURL>
```

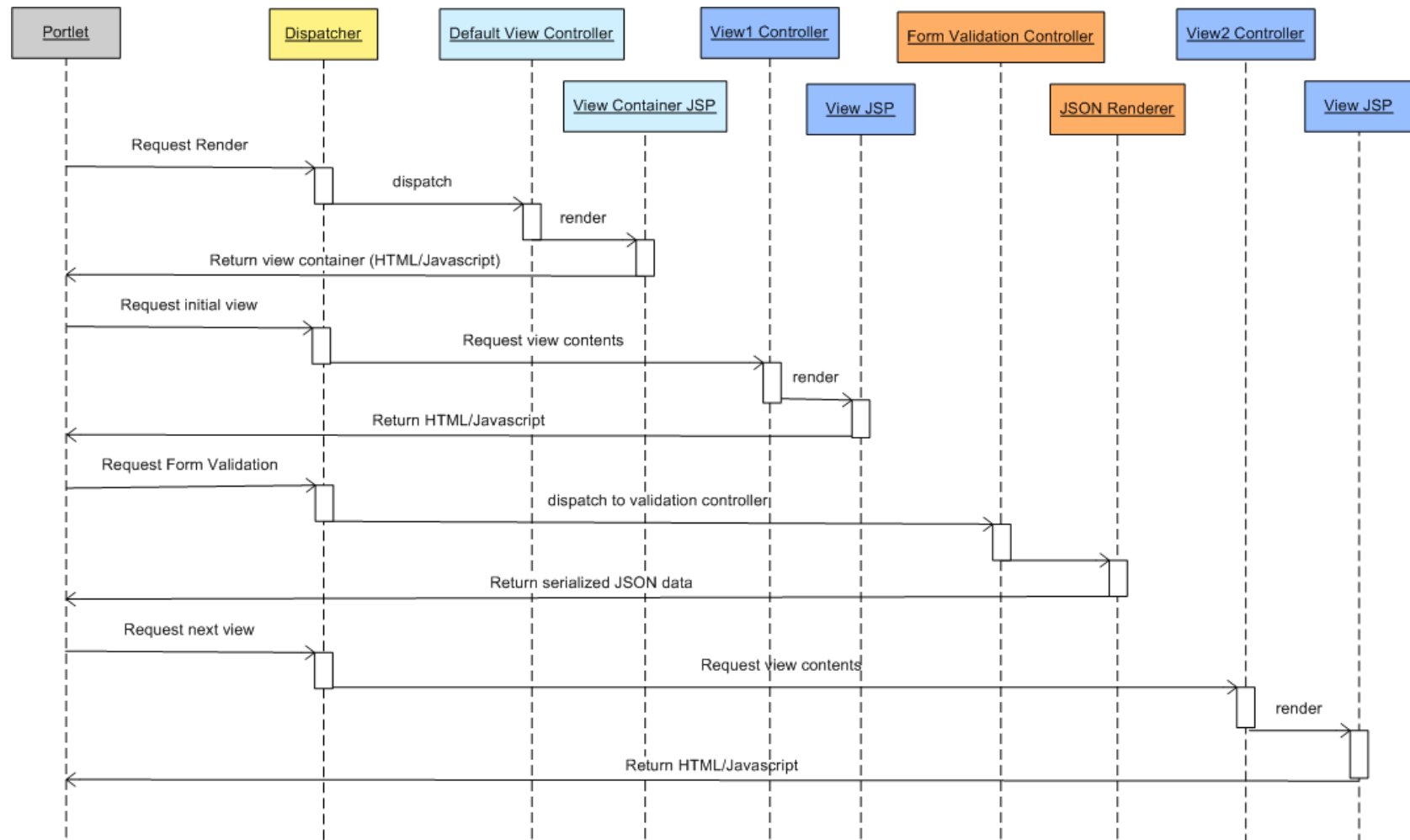
Inter-portlet Communication

- > JavaScript framework-based eventing system
- > Tibco PageBus

Portlet View Manager

- > Persist resourceURL for portlet
- > Load and manage views
- > Fetch views using Ajax
- > History support

View Manager Sequence Diagram



Ajaxifying Portlets Demo

<http://localhost:8480/bluenog>



JavaOneSM

Thank You

Anas Mughal
anas.mughal@bluenog.com

Bluenog

