

Everything as a Service

Jelly 201: Getting Started with Jelly Script for UI Development

Tyler Jones

Engineering Manager, General UI
ServiceNow

Will Leingang

Sr. Software Engineer - UI
ServiceNow



Tyler Jones

- Manager, Design and Experience
- Formerly ServiceNow CMS customer at BYU (2008-2011)
- ServiceNow since 2011
- Projects
 - CMS
 - Tablet
 - Mobile
 - UI14, UI15, UI16



Will Leingang

- Senior Developer, Design and Experience
- Joined ServiceNow in 2013
- Projects
 - Visual Task Boards
 - UI14, UI15
- Trying to decide which of his 3 children to sell for Apple Watch
- Once crashed Fred Luddy's drone into the side of the ServiceNow San Diego office



Agenda

What is Jelly?

Basics of Jelly Syntax

Build a Jelly UI Page

Jelly Basics

What is Jelly

- Java **E**lements originates from the Apache Foundation
 - <http://commons.apache.org/jelly>
- ServiceNow uses Jelly to process XML documents containing static HTML and special Jelly tags
- The output is HTML
 - You can't eat it

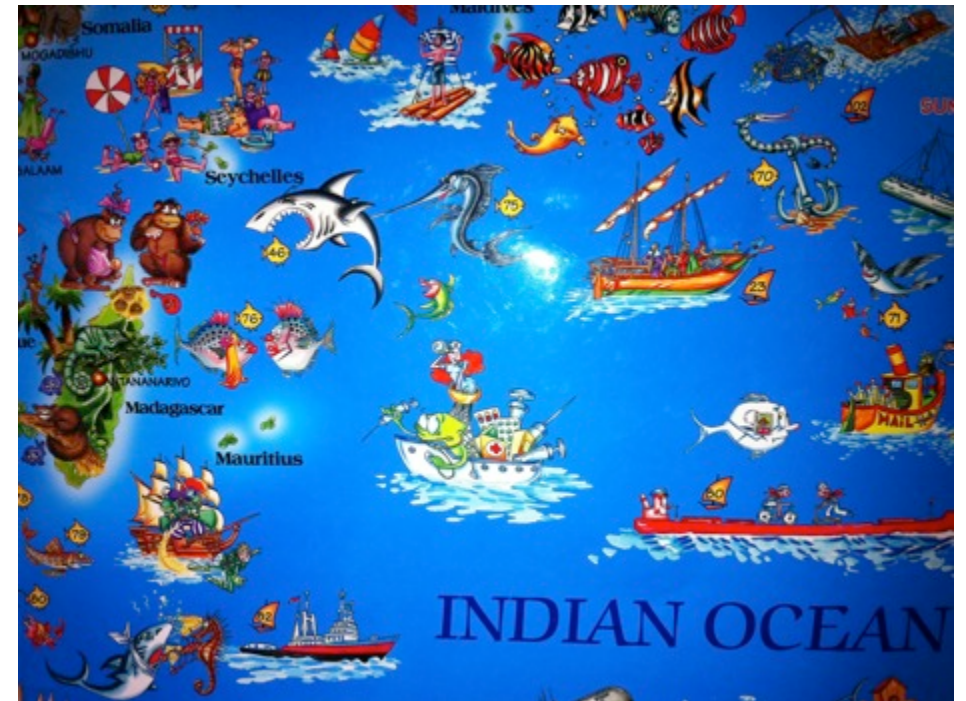
Why Learn Jelly?

- Deepen your understanding of the ServiceNow platform
- Jelly is used in UI Pages and UI Macros
 - There are ~180 UI Pages and ~210 UI Macros baseline
- But a word of caution...



Before You Dive In

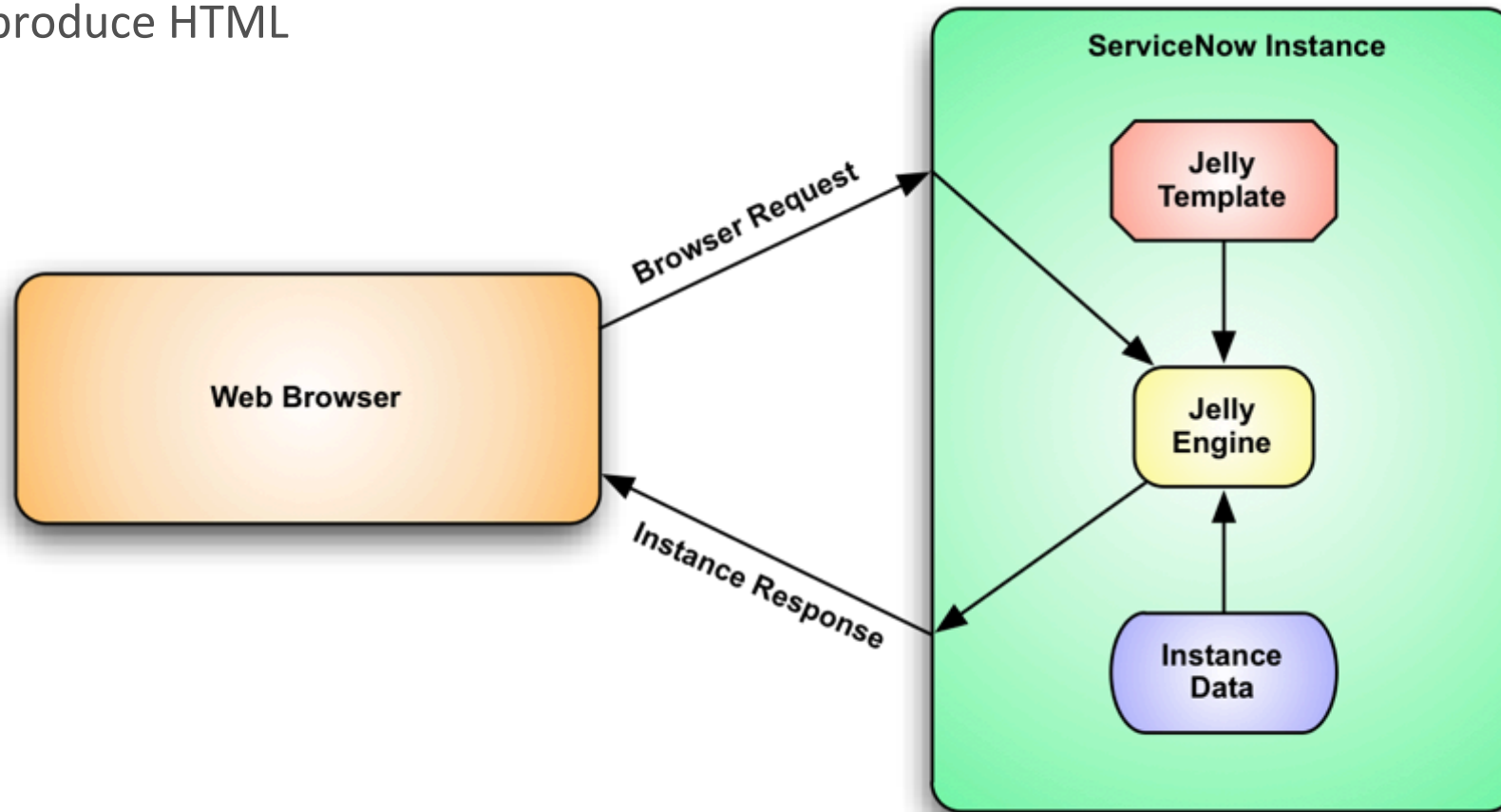
- Treat baseline UI Pages and UI Macros as read-only
 - If you modify them, you will not receive upgrades to the original script
- Jelly != JavaScript
 - Debugging and maintaining Jelly script is challenging
 - Jelly developers need to be aware of 'two-phase processing'



“Here be dragons”

Big Picture

- Jelly tags define dynamic content that is processed to produce HTML



Jelly Context: UI Pages

- Fundamental building blocks of the user interface

The image displays three overlapping user interface mockups from ServiceNow:

- Service Catalog > Top Results:** Shows a product card for a "Laptop preconfigured for development". The card includes an image of a laptop, a description, and a status indicator that says "Exists in categories".
- Notification Preferences:** A modal dialog with a title bar. It contains buttons for "New Device", "Cancel", and "Save". The main text explains notification settings: "Get notified whenever actions that involve you or scheduling to receive only the most important notifications, or unsubscribe entirely from messages. You can also create new notifications or notification devices." It features a "Primary email" field with a plus icon and a "Request Opened on Behalf" section with a minus icon.
- Team Todo Board:** A Kanban-style board titled "Team Todo Board" with a user profile for "System Administrator". It has filters for "Defect" (red), "High Priority" (blue), "Feature" (green), and "Idea" (purple). The board is divided into columns: "To Do" (with a count of 1 and a plus icon) and "Doing". A task card is visible in the "To Do" column with the title "Visual Task Boards" and ID "PTSK0001001" with a timestamp "just now". A "Add Task" button is at the bottom.

Jelly Context: UI Macros

- Reusable UI components rendered using Jelly
 - Can be embedded in UI Pages

The screenshot displays a ServiceNow interface. At the top, a 'Related Search Results' dropdown is visible. Below it, a list of search results is shown, including 'Getting around in Windows', 'Childcare Vouchers', and 'Windows: Should I...'. Each result has a 'Preview' and 'Attach' button. On the right side, a 'Caller' field shows 'Carol Coughlin' with a red box highlighting the search, warning, and info icons. Below the caller field is a 'Location' field. At the bottom, a workflow bar shows the stages: Draft, Scoping, Awaiting approval, Work in Progress, Testing/QA, Deployment, and Closed. The 'Draft' stage is currently selected.

Related Search Results ▾

Getting around in Windows you can customize: Right-click any **app** and the Application bar will appear at the bottom, with options for the application you have selected. To add a...

Childcare Vouchers Childcare Vouchers can significantly help to reduce working parents' childcare costs Savings up to £933 can be made by exchanging the maximum of £243 for Basic...

Windows: Should I Should I upgrade to Windows 8.x? Windows 8.x is designed for using touch, mouse, and keyboard together on hardware...

Preview Attach

Preview Attach

Caller

Carol Coughlin

Location

Draft Scoping Awaiting approval Work in Progress Testing/QA Deployment Closed

Jelly Tags

- Two kinds of tags
 - Core tags in the jelly:core namespace (**j** tags')
 - ServiceNow custom tags in the glide namespace (**g** tags')

```
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null" xmlns:g2="null">
```

- Examples:

Core Tags

<j:if>
<j:set>
<j:while>
<j:choose>

Glide tags

<g:evaluate>
<g:breakpoint>
<g:macro_invoke>
<g:call>

Jelly Phases

Original

```
<j:set var='jvar_first' value='Hello' />
<j2:set var='jvar_second' value='World' />

<p> ${jvar_first} </p>
<p> ${jvar_second} </p>
<p> Phases don't faze me </p>
```

After Phase 1

```
<j2:set var='jvar_second' value='World' />

<p> Hello </p>
<p> ${jvar_second} </p>
<p> Phases don't faze me </p>
```

After Phase 2

```
<p> Hello </p>
<p> World </p>
<p> Phases don't faze me </p>
```



Let's Build Something!

Polling App

- Ask a question
 - Display question and related answers
 - Wait for form submit
 - Save results on submit
- Once a user votes
 - Display their answer

How satisfied are you with the Jelly lab?

☐ Completely satisfied

☐ Very satisfied

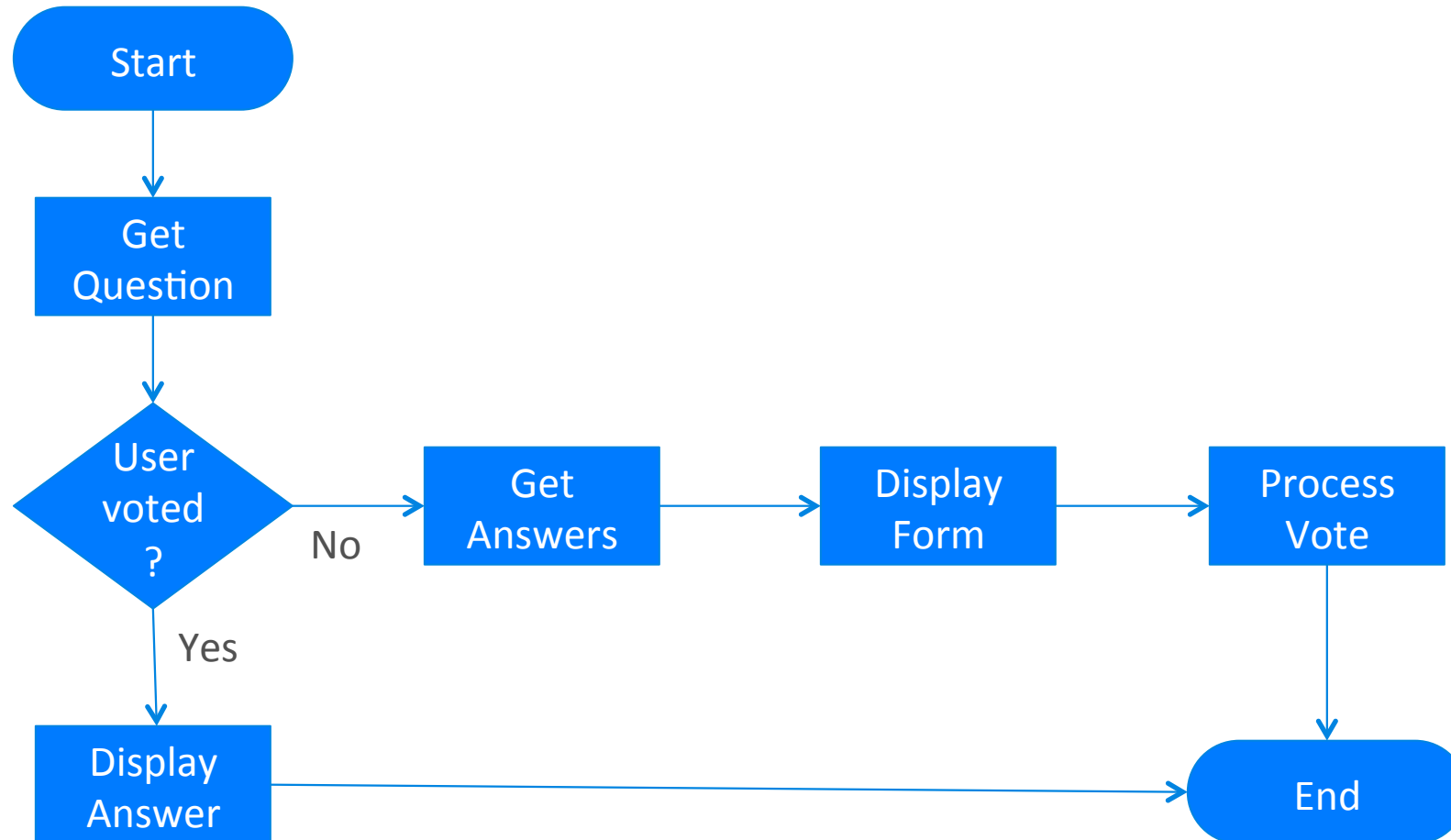
☐ Fairly well satisfied

☐ Somewhat satisfied

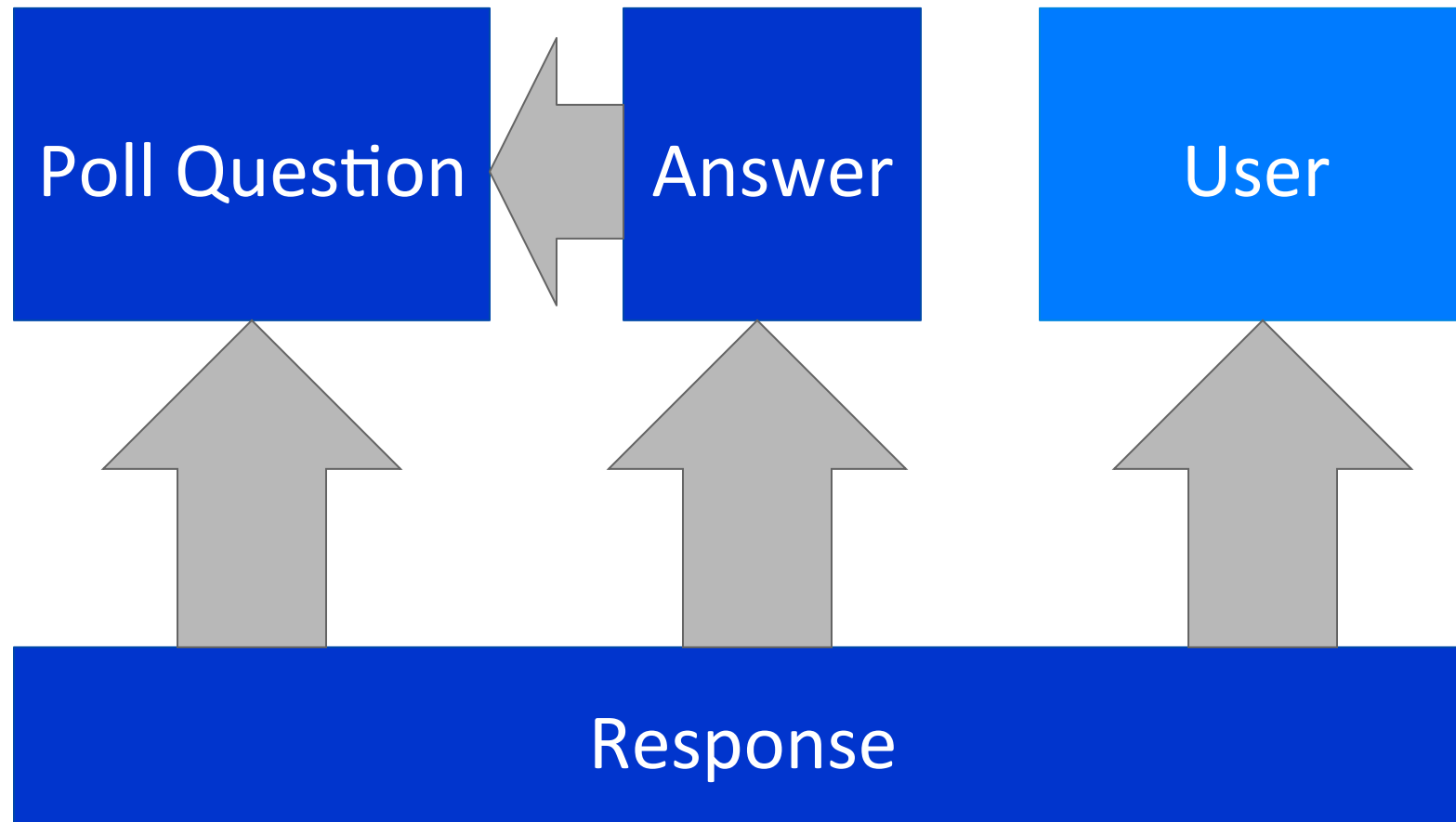
☐ Very dissatisfied

Vote!

Process Flow



Data/Table Structure



Lab 1: Create a New UI Page

- Login to your instance as admin
- Navigate to **System UI > UI Pages**
- Click New
- Enter the following information:
 - Name: poll_view
 - Description: A poll manager UI Page
- The following appears in the HTML field:

```
<?xml version="1.0" encoding="utf-8" ?>  
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null" xmlns:g2="null">  
  
</j:jelly>
```

Lab 1: Create a UI Page

- Between the <j:jelly> and </j:jelly> tags, enter a simple string
- Click “Try It”

```
<?xml version="1.0" encoding="utf-8" ?>  
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null" xmlns:g2="null">  
Poll View 1.0<br />  
</j:jelly>
```

- Click your browser “Back” button to return to your UI page definition

Lab 2: Get the User's ID and Test

```
<?xml version="1.0" encoding="utf-8" ?>
<j:jelly trim="false" xmlns:j="jelly:core" xmlns:g="glide" xmlns:j2="null" xmlns:g2="null">
Poll View 1.0<br />
<!--
    Get the current user's ID
    Concepts:
        Simple g:evaluate
        Passing a JavaScript value to Jelly variable
        Using the Jelly variable in a JEXL expression
-->
<g:evaluate var="jvar_userid" expression="gs.getUserID;" />
Welcome user id: ${jvar_userid}<br />

    <!-- REPLACE THIS LINE WITH LAB 3 -->
</j:jelly>
```

Lab 3: Get an Active Poll ID

```
<g:evaluate var="jvar_poll_id">

  var p1 = new GlideRecord('u_poll');

  p1.addQuery('active', true);
  p1.orderByDesc('sys_updated_on');
  p1.query();

  if (p1.next()) {
    result = p1.getValue('sys_id');
  }

  result;
</g:evaluate>
The latest poll is: ${jvar_pollid}<br/>

<!-- REPLACE THIS LINE WITH LAB 4 -->
```

Lab 4: Get the Question Text

Sidebar: Debugging

```
<g:evaluate>

  var p2 = new GlideRecord('u_poll');
  var questionText= '';

  if (p2.get(${jvar_poll_id}) {
    questionText = p2.getValue('u_question');

  } else {
    gs.print('>>>poll_view: Error reading question');
  }

</g:evaluate>
<div class="poll_question">${questionText}</div>

<!-- REPLACE THIS LINE WITH LAB 5 -->
```

Lab 5: Has This User Voted Yet?

```
<g:evaluate>
  var response = new GlideRecord('u_poll_response');
  var result;

  response.addQuery('u_user', ${jvar_userid});
  response.addQuery('u_poll', ${jvar_poll_id});
  response.query();
  result = response.next();

  result;
</g:evaluate>

User has voted: ${jvar_has_voted}<br/>

<!-- REPLACE THIS LINE WITH LAB 6 -->
```

Lab 6: If the User HAS Voted...

```
<j:if test="\${jvar_has_voted eq true}">  
  Your response was: ${response.u_answer.u_text}<br />  
</j:if>  
  
<!-- REPLACE THIS LINE WITH LAB 7 -->
```


Lab 7: If the User has NOT Voted...

```
<j:if test="!${jvar_has_voted}">  
  <g:evaluate>  
    var ans = new GlideRecord('u_poll_answer');  
  
    ans.addQuery('u_poll', jelly.jvar_poll_id);  
    ans.orderBy('order');  
    ans.query();  
  </g:evaluate>  
  We should have an answer object now ${ans.hasNext()}<br />  
  
  <!-- REPLACE THIS LINE WITH LAB 8 -->  
  
</j:if>
```

Lab 8: Ensure the Question has Answers

```
<!--  
    Check if this is a question without answers  
    Concepts:  
        Conditional statement on a JavaScript variable/method  
        gs.getMessage()  
  
    <j:if test="${!ans.hasNext()}">  
        <div class="poll_error">  
            ${gs.getMessage('poll_no_answers_found');}  
        </div><br />  
    </j:if>  
  
    <!-- REPLACE THIS LINE WITH LAB 9 -->
```

Lab 9: Construct the Form

```
<j:if test="${ans.hasNext()}">
  <g:ui_form>
    <input type="hidden" name="poll_id"
      value="${jvar_poll_id}" />
    <input type="hidden" name="user_id"
      value="${jvar_userid}" />

    <j:while test="${ans.next()}">
      <div class="poll_answer">
        <g:ui_radio name="poll_answer"
          value="${ans.sys_id}" text="${ans.u_text}" />
      </div>
    </j:while>
    <g:ui_button action="save">
      ${gs.getMessage('poll_vote_button')}</g:ui_button>

  </g:ui_form>
</j:if>
```

Lab 10: The Processing Script

```
var pr = new GlideRecord('u_poll_response');

if (!JSUtil.nil(poll_id) && !JSUtil.nil(user_id)) {
    pr.initialize();
    pr.u_poll = poll_id;
    pr.u_user = user_id;
    pr.u_answer = poll_answer;
    pr.update();
}
```

Best Practices

- Document your work
- Avoid dynamic JEXL inside evaluate tags
- Use descriptive variable names
- Use appropriate indentation

Top Takeaways

1

Most of what you need can be done in Phase 1.

2

Understand Jelly vs. JavaScript variables.

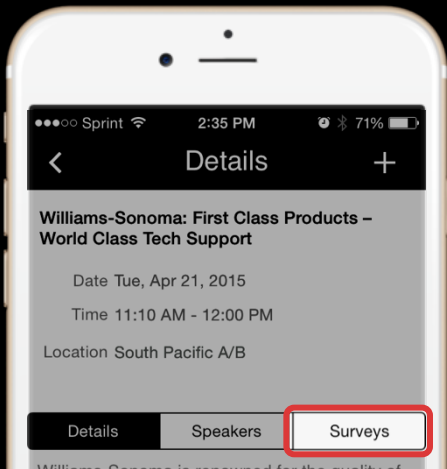
3

Use best practices (no dynamic JEXL in evaluate tags).

How Did We Do?

Your feedback on this session helps us deliver great content.

Please take a moment to complete a session survey in the Knowledge15 app or use the survey forms at the back of the room.



Everything as a Service

Thank You

Tyler Jones

*Engineering Manager,
General UI*

ServiceNow

tyler.jones@servicenow.com

Will Leingang

*Senior Developer,
General UI*

ServiceNow

will.leingang@servicenow.com

Get Presentations

As a Knowledge15 attendee, you have exclusive access to breakout and lab session content from the event.

1. Go to knowledge.servicenow.com
2. Log into the community
3. Click on **View Now** button

Exclusive Content

As a Knowledge15 attendee, get exclusive access to presentations delivered at the event.

[View Now](#)

knowledge.servicenow.com

@jylertones | @wleingang | #Know15

© 2015 ServiceNow All Rights Reserved

knowledge15

April 19–24, 2015 • Mandalay Bay • Las Vegas, NV