knowledge15

Everything as a Service

Scripting 201:

Client Scripting for ServiceNow

Matthias Martini
Principal Consultant
ServiceNow

Oliver Schmitt
Principal Consultant
ServiceNow

servicenow

Agenda

Introductions

General Guidelines

Lab Overview

Takeaways

Introductions

Your Presenters

Your Presenters

- Matthias Martini
  - Principal Consultant
  - Professional Services EMEA Central
  - With ServiceNow for 3.5 Years
  - Service Catalog, Commercial, Architecture
- Oliver Schmitt
  - Principal Consultant
  - Professional Services EMEA Central
  - With ServiceNow for 4 Years
  - Custom Apps, Integrations, Architecture

#Know15

knowledge15
April 19–24, 2015 • Mandalay Bay • Las Vegas, NV

General Guidelines

A Brief Look at Best Practices

General Guidelines

- Client Side scripting = JavaScript that runs on browser
- Try using Server Side whenever possible
  - Standardized scalable environment
  - Not exposed to user
- Client Scripting should support completing a form
  - Guide a user through the form, highlight next steps
  - Auto fill fields based on available data
- Client Scripting is NOT suited for Security
  - Way to easy to get around it
  - Client Scripts and UI Policies only run on forms

General Guidelines

- Reduce server lookups
  - Preload information whenever possible using Display Business Rules
  - https://wiki.servicenow.com/index.php?title=Business_Rules#Display_Business_Rules
  - Get all information with one server call
- Avoid synchronous server calls
  - Complex queries may block the browser for several second → Bad user experience
- Avoid DOM manipulation
  - No usage of gel() or getElementById()
  - Use g_form methods instead

Lab Overview

Client Scripting Done the Right Way

Lab Environment

- Instance URL will be: https://scripting201-<###>.lab.service-now.com
- Please raise your hand if you have difficulties logging into your instance
- Submit feedback via the application in your instance

Lab Overview

- Multiple Client Side APIs used in labs
  - GlideAjax
  - GlideList2
  - GlideDialogForm
  - Additional APIs documented in the wiki: http://wiki.servicenow.com/index.php?title=Client_API_Reference

- Build your own solution based on pseudo code for a more challenging exercise
- Work on the lab challenges when finished with an exercise.

Lab 1 Auto Watch List for Priority 1 Incidents

- How to structure a client script
  - Check for isLoading and newValue
  - Prevent unnecessary execution
  - https://wiki.servicenow.com/index.php?title=Client_Script_Best_Practices#Run_Only_Necessary_Scripts
- How to execute server side code and retrieve additional information
  - Don't use GlideRecord or getReference
  - Use asynch GlideAjax calls instead
  - https://wiki.servicenow.com/index.php?title=GlideAjax

Lab 1 Auto Watch List for Priority 1 Incidents

Challenge:

Script Include

```
return user.manager.sys_id + '|' + user.manager.name;
```

Client Script

```
else{
    var array = managerID.split('|');
    g_form.setValue('watch_list', array[0], array[1]);
    g_form.addInfoMessage('Callers manager added to watchlist');
}
```

Lab 2 “Open in new Tab” context menu

- Context menus are less known field for Client scripting
- Great way of enhancing user’s interface

- GlideList2 enables Client scripts (UI Actions & UI Context Menus) to interact with Lists and Related Lists - e.g.:
  - Setting filters
  - Trigger refresh
  - Get name of table list belongs to
  - Complete reference in wiki: https://wiki.servicenow.com/index.php?title=GlideList2_(g_list)

Lab 3 Create Missing CIs from Incident

- UI Actions area able to execute client side code as well
- Some scenarios may require action on different table / record
- Possible solution: GlideDialogForm
  - Shows a specific form (view) for a specific table
  - Incorporates callback function to handle result of form processing
  - Closes after submission
- Always make sure to define proper conditions for UI Actions

Lab 3 Create Missing CIs from Incident

Challenge

Condition: current.cmdb_ci.nil() && current.canWrite() && gs.hasRole('itil’)

Lab 4 Focus on Most Important Change

- UI Policies have some advantages over Client scripts
  - Defined order
  - Easy to use condition builder
  - Easy to use Policy Actions
  - → use UI Policies whenever you may benefit from one of those

- Execute if true / Execute if false scripts enable execution of client side code

Lab 5 Debugging Client Scripting

- Know your Debugging tools
- ServiceNow JavaScript Debugger
  - Detailed analysis of client and server side code being executed
- Instance Debug settings
  - Application ‘System Diagnostics’ → Section ‘Session Debug’
  - E.g.: ‘Debug Log’, ‘Debug UI Policies’
- Log statements and related System Logs
  - Application ‘System Logs’
- Browser integrated tools (Developer Tools, Console) or Extensions (Firebug)

Top Takeaways

1 Respect Best Practices listed in wiki – they'll make your life easier.

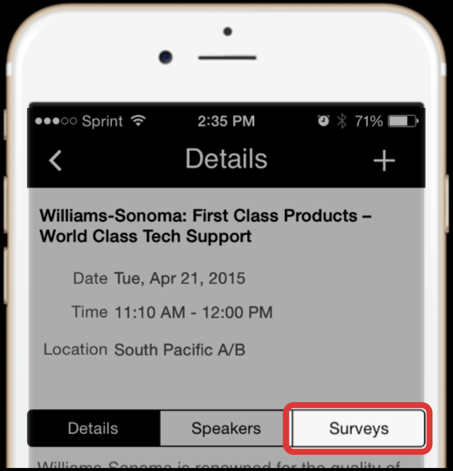
2 Know all your options – consider alternatives to Client Scripts.

3 Your scripting should support the user – security has to be server side.

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

Matthias Martini

Principal Consultant

ServiceNow

matthias.martini@servicenow.com

Oliver Schmitt

Principal Consultant

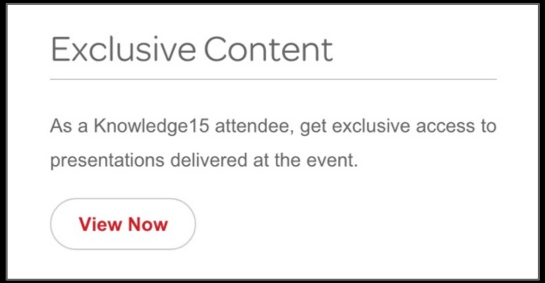
ServiceNow

oliver.schmitt@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