# Lab Guide

## Mobile:

## A New ServiceNow Experience

### Will Lisac & Guerric Sloan

Default Login / Password:

admin / Knowledge16

itil / Knowledge16

employee / Knowledge16

# Lab Goal

This section covers the basics when working with the ServiceNow Native Application for iOS. If you do not have an iPhone, please skip to the next section.

- Install the ServiceNow Native App
- Login to the Lab Instance

## Lab 1.1 Native App Basics

## Install the ServiceNow Native App

1. Open the Apple App Store on your iPhone.
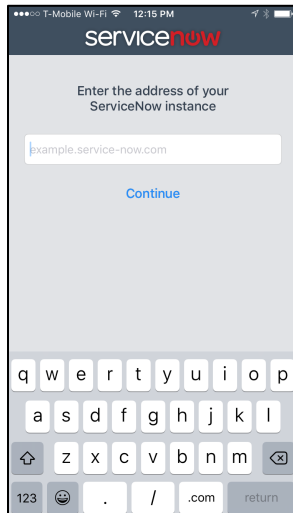
2. Search for **ServiceNow**.
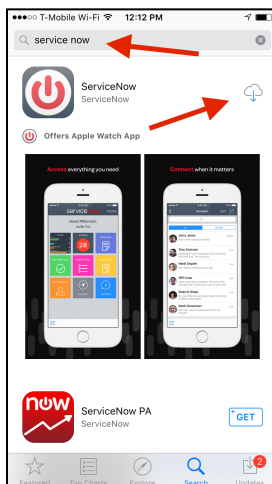


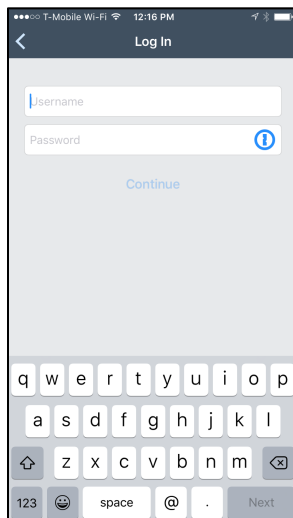3. Install the ServiceNow Application.

## Log in to the Lab Instance

1. Using your iPhone, open the ServiceNow App.

2. If prompted to enable push notifications, tap **OK** to allow them.

   **Note**: You test push notifications later in the lab, which requires them to be enabled.

3. Enter the unique instance URL provided.



4. Log in with the provided administrator credentials.

# Lab Goal

This section covers the basics when working with the ServiceNow Mobile Web UI (User Interface)

- Access the Mobile Web UI on your Phone
- Access the Mobile Web UI on your Desktop

**Lab 1.2
Mobile Web
UI Basics**

## Access the Mobile Web UI on your Phone

1. Using your phone, navigate to the unique instance URL provided.

2. Log in with the provided administrator credentials.
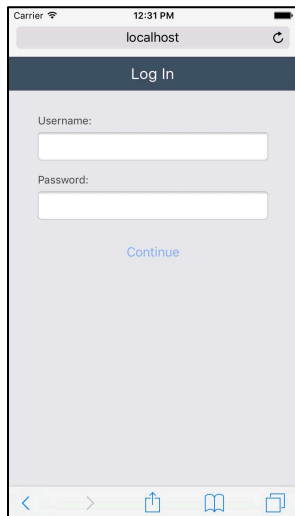
## Access the Mobile Web UI on Your Desktop

1. Using your desktop, go to the unique instance URL provided.

2. Update the URL to navigate directly to the mobile web user interface by appending $m.do. The URL should resemble:

   https://LabPrefix###.lab.servicenow.com/$m.do

   **Note**: To maintain separate desktop and mobile interface sessions, you should open your *Mobile Web UI* session in a private browsing window.

**servicenow**

3. How to start private browsing:

**Google Chrome**:
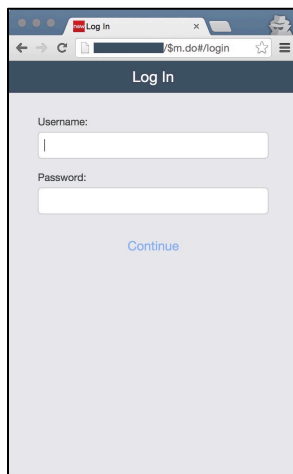https://support.google.com/chrome/answer/95464?hl=en
**Apple Safari**:
https://support.apple.com/kb/PH19216?locale=en_US
**Mozilla Firefox**:
https://support.mozilla.org/en-US/kb/private-browsing-control-what-info-firefox-saves

4. Log in with the provided administrator credentials.

# Lab Goal

This section covers the various system configuration options available to customize the ServiceNow Mobile Experience.
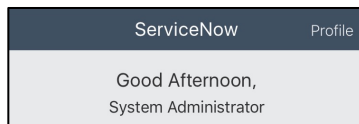
- Access the Lab Instance Desktop UI
- Deactivate and Reactivate the Helsinki Mobile Web UI
- Toggle Native Mobile Application Configurations
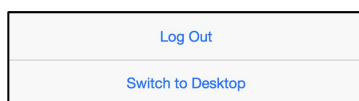
## Lab 2.1 Mobile UI Configuration

## Access the Lab Instance Desktop UI

1. Using your desktop, go to the unique instance URL provided.

2. Log in with the provided administrator credentials.

3. If you are presented with the mobile interface:

   Click **Profile** from the Home screen.

   | ServiceNow | Profile |
   | --- | --- |
   | Good Afternoon, System Administrator | |

   Select **Switch to Desktop**.

   | Log Out |
   | --- |
   | Switch to Desktop |

## Deactivate and Reactivate the Helsinki Mobile Web UI

1. Navigate to **System Properties > Mobile UI Properties**.

   **System Properties**
   Basic Configuration
   Basic Configuration UI16
   My Company
   UI Properties
   **Mobile UI Properties**
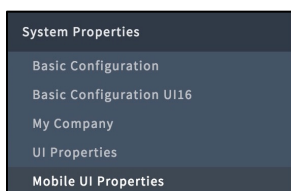
2. Deselect **Enables the updated mobile web experience available since Helsinki**.

> Enables the updated mobile web experience available since Helsinki.
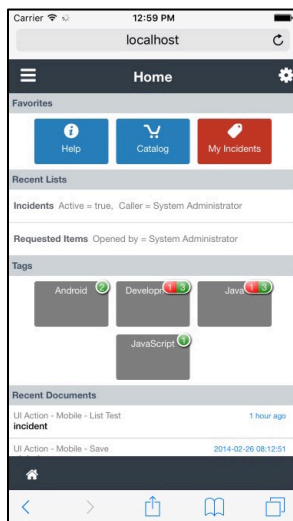> ◯ Yes | No

3. Click [ Save ] .

4. Switch to the **Mobile Web UI**.

5. Refresh using the browser's refresh button [ C ] .

6. Notice the Dublin Mobile Web UI is shown.

   **Note**: This change is not apparent in the ServiceNow Native App. View your instance using Mobile Safari to see the change.



7. Switch to the **Desktop UI**.

8. Check **Enables the updated mobile web experience available since Helsinki**.

> Enables the updated mobile web experience available since Helsinki.
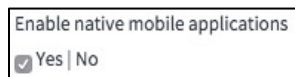> ☑ Yes | No

9. Click [ Save ] .

10. Switch to the **Mobile Web UI**.

11. Refresh using the browser's refresh button .

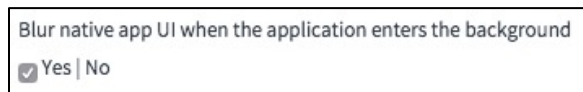12. Notice the Helsinki Mobile Web UI appears again.

## Toggle Native Mobile Application Configurations

1. Switch to the **Desktop UI**.

2. Navigate to **System Properties > Mobile UI Properties**.

3. Deselect **Enable native mobile applications**.

   Enable native mobile applications
   ☑ Yes | No

4. Select **Blur native app UI when the application enters the background**.

   Blur native app UI when the application enters the background
   ☑ Yes | No

5. Click Save.

   **Note**: To complete the following steps, an iPhone running the ServiceNow Native App is required.
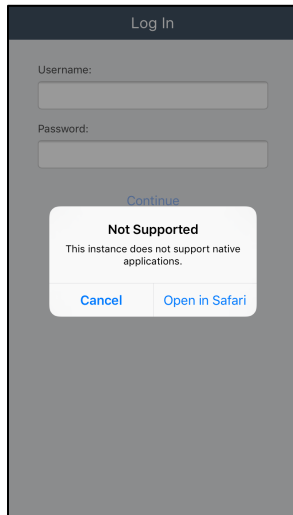
6. Switch to the **Native App**.

7. Tap Profile.

8. Tap Switch Instance.

9. Select your lab instance.

10. Notice that the application is disabled.



11. Switch to the Desktop UI.

12. Navigate to **System Properties > Mobile UI Properties**.

13. Check **Enable native mobile applications**.

14. Click [Save].

15. Switch to the **Native App**.

16. Select your lab instance.
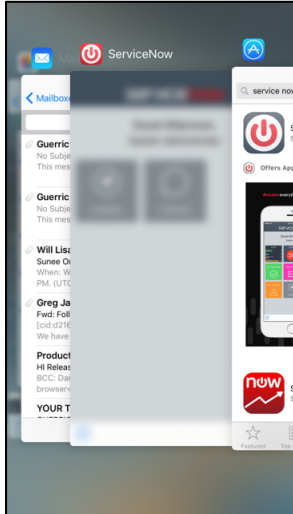
17. Notice that the application is enabled.

18. Double tap the home button on your phone to activate multitasking.



19. Select another application on your phone.

20. Double tap the home button again to activate multitasking.

21. Notice the ServiceNow App content's preview is blurred.

# Lab Goal

This section covers the various system configuration options available to customize the mobile navigator. The mobile navigator configuration controls the ServiceNow applications that are available to users.

- Create a New Application Menu
- Create an Application Module for Incidents
- Create an Application Module for a new Incident Record
- Create an Application Module for Connect Chat
- Create an Application Module for a Map Page
- Refresh the Navigator
- Try out the New Application Modules

## Create a New Application Menu

1. Switch to the **Desktop UI**.

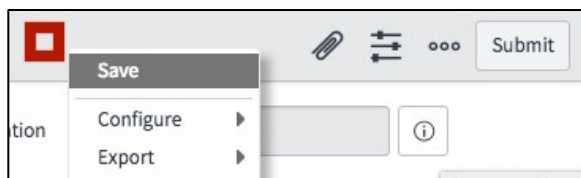2. Navigate to **System Mobile UI > Navigator Apps**.
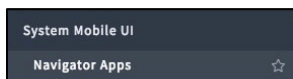
   

3. Click **Application Menus >**  .

4. Input form values:

   Name: **K16 Mobile Lab**
   Order: **42**

5. Right click the form header.

   

6. Click **Save**.

## Create an Application Module for Incidents

1. Click **Modules >** New .

2. Input form values:

   Name: **Incidents for Hardware**
   Table: **Incident**

3. Create a filter for the **Incident** table:

   Field: **Category**
   Operator: **is**
   Value: **Hardware**

4. Click Submit .

## Create an Application Module for a New Incident Record

1. Click **Modules >** New .

2. Input form values:

   Name: **Create New Incident**
   Path: **/form/incident/-1**

3. Click Submit .

## Create an Application Module for Connect Chat

1. Click **Modules >** New .

2. Input form values:

   Name: **Connect Chat**
   Path: **$chat.do**
   Path Relative To Root: **Checked**

   **Note**: **Path Relative To Root** allows mobile navigator modules that link to desktop pages.

3. Click Submit .

## Create an Application Module for a Map Page

1. Click **Modules >** New .

2. Input form values:

   Name: **All Locations**
   Path: **/map/?sysparm_name=All%20Locations**

   **Note**: Mobile path components and parameters should be correctly URL encoded. *%20* represents a URL encoded space character

3. Click Submit .

## Use the Native App, Refresh the Navigator

**Note**: If you are not using the *Native iOS Application* skip this section

1. Tap .

2. Tap **Home**.

3. Tap Profile .

4. Tap Switch Instance .

5. Select your lab instance from the instance list.

## Use the Mobile Web UI, Refresh the Navigator

**Note**: If you are not using the **Mobile Web UI**, skip this section.

1. Tap .

2. Tap **Home**.

3. Tap your browser's refresh button .

## Try out the New Application Modules

1.  Tap [icon].

2.  Tap **Incidents for Hardware**.

    **Note:** The incidents list should display with the filter you configured.

3.  Tap [icon].

4.  Tap **Create New Incident**.

    **Note**: The incident form should display with empty values to be input.

5.  Tap [icon].

6.  Tap **Connect Chat**.

    **Note**: Connect Chat should be open on your mobile device. On the **Mobile Web UI**, Connect Chat opens in a new tab. Switch back to the **Mobile Web UI** tab to continue the lab.

7.  Tap [icon].

8.  Tap **Map Page**.

    **Note**: The **All Locations** map page should be open.

# Lab Goal

This section covers the shared favorites functionality in the new ServiceNow Mobile Experience. Users can access their favorites on the desktop or their mobile device.

- Use the Desktop UI, Favorite an Application Menu
- Use the Native App, Refresh and View the Favorites
- Use the Mobile Web UI, Refresh and View the Favorites
- Create New Favorites on Mobile
- Edit Favorites on Mobile
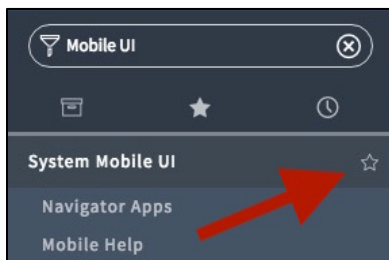- Use the Native App, Create a Favorite Visualization

## Use the Desktop UI, Favorite an Application Menu

1. Switch to the **Desktop UI**.

2. Filter the Navigator by typing **Mobile UI**.



3. Hover over the **System Mobile UI** application menu.

4. Click **Star** to favorite the menu.



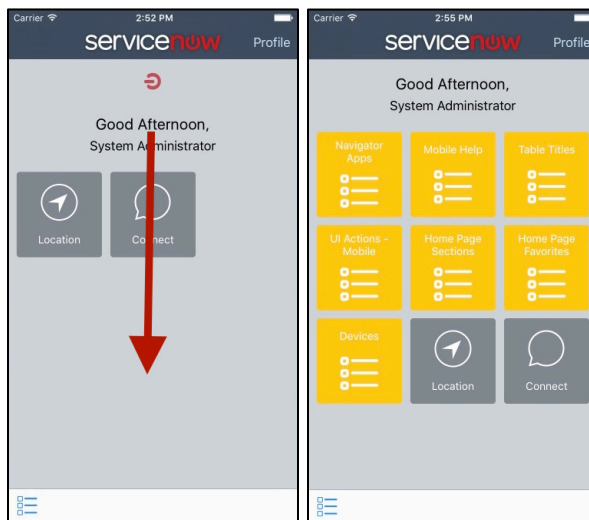## Use the Native App, Refresh and View the Favorites

**Note**: If you are not using the Native iOS Application skip this section.

1. Switch to the **Native App**.

2. Tap .

3. Tap **Home**.

4. Pull down to refresh the favorites.

   **Note**: New favorites should display.

## Use the Mobile Web UI, Refresh and View the Favorites

**Note**: If you are not using the Mobile Web UI, skip this section.

1. Switch to the **Mobile Web UI**.

2. Tap [icon].

3. Tap **Home**.

4. Tap your browser's refresh button [icon].

   **Note:** New favorites should display.

## Create New Favorites on Mobile

**Note**: This section applies to the **Native App** and **Mobile Web UI**.

1. Switch to the **Mobile Web UI** or **Native App**.

2. Tap [icon].

3. Tap **Users**.

4. Tap **Abel Tuter** in the list.
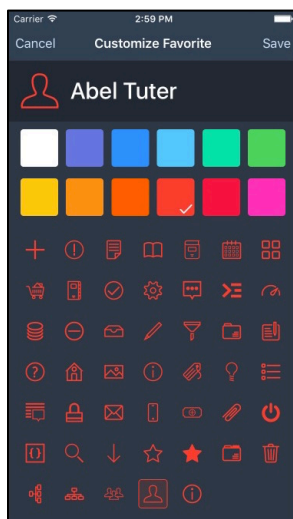


Abel Tuter
abel.tuter1

5. Tap ☆.

6. Type a name for the favorite: **Abel Tuter**.

7. Tap **an icon** for the favorite.

8. Tap a **color**.



9. Tap Save.

10. Tap ⊟.

11. Tap **Home**.

> **Note**: Notice the new favorite on your mobile home screen.

## Edit Favorites on Mobile

1.  Long press **Abel Tuter**.

    **Note**: Trigger the long-press gesture by tapping on a favorite and holding down. Release after the long press gesture is triggered.

2.  Tap a **color**.

3.  Tap [Save] .

## Use the Native App, Create a Favorite Visualization

**Note**: If you are not using the Native App skip this section.

1.  Switch to the **Native App**.

2.  Long press **Abel Tuter**.

3.  Tap **Visualizations**.

4.  Tap **User Widget**.

    

5.  Tap [Save] .
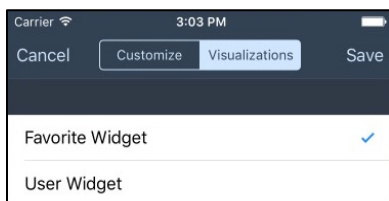
    **Note**: The favorite should now be presented as a user widget visualization.

# Lab Goal

This section covers the various configuration points in the platform to customize the mobile list display and operation.

- Set List Layout Columns
- View List Layout Columns
- Set List Table Titles
- Set List Search Fields
- View List Table Titles
- View List Search Fields
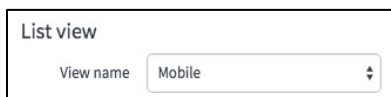- Hide Mobile Filter UI
- View Hidden Mobile Filter UI

## Set List Layout Columns

1. Switch to the **Desktop UI**.

2. Navigate to **Incident > Open**.

3. Right-click the list header to show the list actions menu.
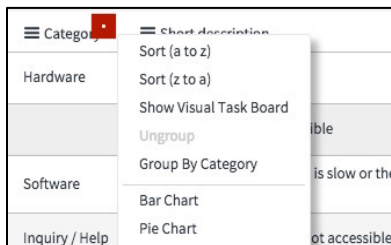


4. Click **Configure > List Layout**.



5. Select **Mobile** for the View name.



6. Find **Priority** in **Available** fields.

7. Click [ > ] to add Priority.

8. Find **Caller** in **Available** fields.

9. Click [ icon ] to dot walk **Caller** fields.

10. Find **First name** in **Available** fields.

11. Click [ > ] to add **Caller.First name**.

12. Click [ **Save** ].

## View List Layout Columns

1. Switch to the **Mobile Web UI** or **Native App**.

1. Navigate to **Incident > Active**.

2. Click [ **Sort** ].

3. Notice the sortable columns display.
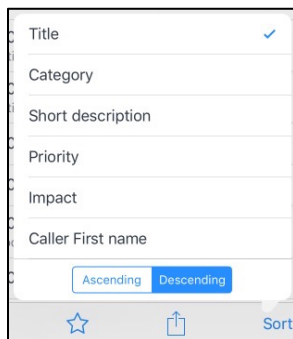


4. Tap **Caller First name**.

5. Notice the actively sorted field displays on each record.

6. Tap anywhere outside the sort popover to close it.

7. Tap any record.

8. Notice the fields display in the card on the record.

| Short description | Sales forecast spreadsheet is READ ONLY | |
|---|---|---|
| Priority | 1 - Critical | > |
| Impact | 1 - High | |
| Caller First name | Taylor | |

## Set List Display Value

1. Switch to the **Desktop UI**.

2. Navigate to **Incident > Open**.

3. Right click the list header to show the list actions menu.
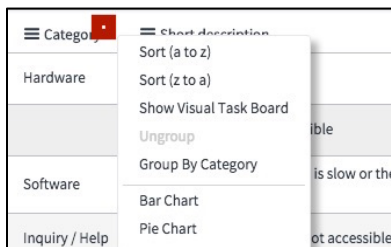


4. Click **Configure > Table**.

5. Find the **Category** column in the list of records.

| ⓘ | Category | String | | 40 | inquiry | false |
|---|---|---|---|---|---|---|

6. Double click the **Display** column to list edit.
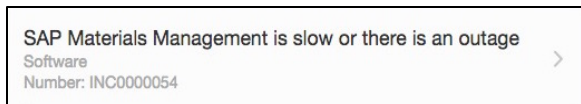
7. Select **true**.



8. Click .

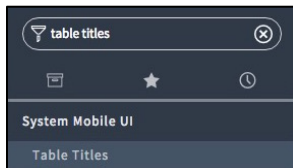9. Click  to save the table changes.

## View List Display Value

1. Switch to the **Mobile Web UI** or **Native App**.

2. Navigate to **Incident > Active**.

3. Notice the **Category** displays below the incident title.



SAP Materials Management is slow or there is an outage
Software
Number: INC0000054

## Set List Table Titles

1. Switch to the **Desktop UI**.

2. Navigate to **System Mobile UI > Table Titles**.



3. Find the **Incident** table title configuration row.



2013-06-24 20:47:50     Incident [incident]     short_description

4. Click .

5. Click  to edit **Fields**.

6. Click **Short description.**

7. Click  to remove **Short description**.

8. Find **Number** in **Available** fields.

9. Click  to add **Number**.
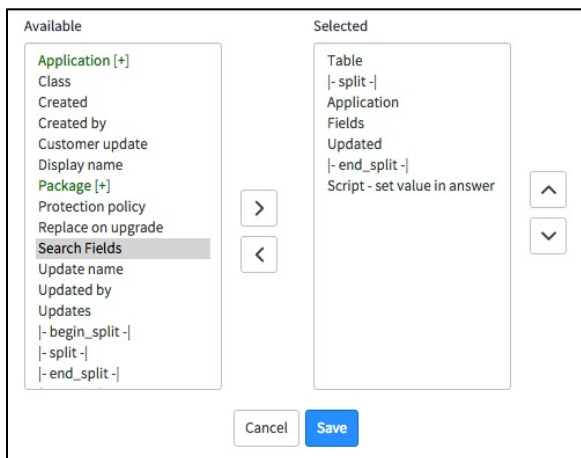
10. Find **State** in **Available** fields.

11. Click [ > ] to add **State**.

12. Click [ 🔒 ] to set **Fields**.

## Set List Search Fields

1. Click [ 🔒 ] to edit **Search fields**.

   **Note**: **Search fields** might not be on the form. Add Search fields to the form by editing the form layout and adding **Search fields** to the current layout.



2. Find **Number** in **Available** fields.

3. Click [ > ] to add **Number**.

4. Find **State** in **Available** fields.

5. Click [ > ] to add **State**.

6. Click [ 🔒 ] to set *Search fields*

7. Click [ Update ].

## View List Table Titles

1. Switch to the **Mobile Web UI** or **Native App**.

2. Navigate to **Incident > Active**.

3. Notice each record displays the **Number** and **State**.

## View List Search Fields

1. Tap ⬚ to search.

2. Type **INC0000027**.

3. Notice a single record is shown.

4. Type **Hardware**.

5. Notice Hardware records are shown.

## Hide Mobile Filter UI

1. Switch to the **Desktop UI**.

2. Navigate to **Incident > Open**.

3. Right click the list header to show the list actions menu.
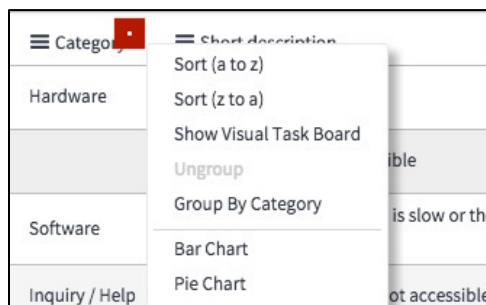
4. Click **Configure > List control**.
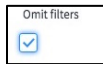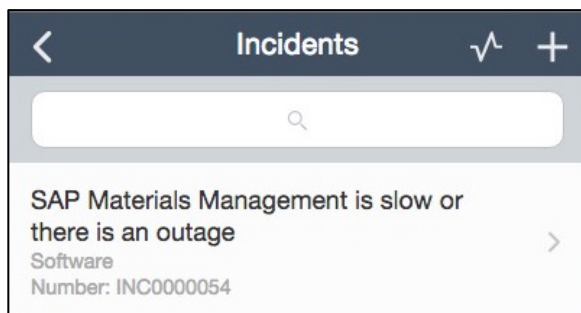
5. Check **Omit filters**.



6. Click .

## View Hidden Mobile Filter UI

1. Switch to the **Mobile Web UI** or **Native App**.

2. Navigate to **Incident > Active**.

3. Notice the option to filter is hidden.

# Lab Goal

This section covers the basic configuration required to create new push notifications for the ServiceNow Native App.

- Create a Push Notification
- Create a Push Message
- Set Push Notification Message
- Use the Native App, Test Push Notifications
- Use the Desktop, Test Push Notifications

## Create a Push Notification

1. Switch to the **Desktop UI**.

2. Navigate to **System Notification > Push > Create Push Notification**.
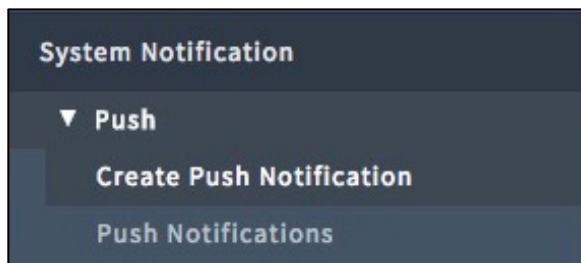


3. Input form values:

   Name**: New Incident While In Lab**
   Table: **Incident**
   When to send:
   Inserted: **Check**

4. Click the **Who will receive** tab.

| When to send | Who will receive | What to send |
|---|---|---|

5. Click [icon] button to add yourself.

6. Click [Submit] .

## Create a Push Message

1. Navigate to **System Notification > Push > Push Messages**.

2. Click [New] .

3. Input form values:

   Name: **New Incident Lab Message**
   Push App: **ServiceNow Mobile application**
   Push Message Content: **Generic Record Payload**
   Message:
   **New Incident**
   **${number}: ${short_description}**

4. Click [Submit] .

## Set Push Notification Message

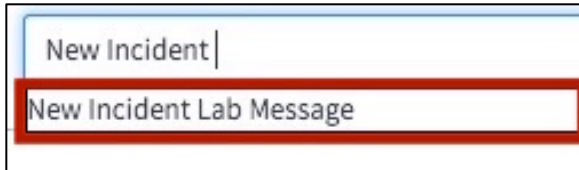1. Navigate to **System Notification > Push > Push Notifications**.

2. Find **New Incident While In Lab**.

3. Click [i] .

4. Click **What to send** tab.
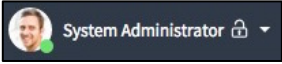
| When to send | Who will receive | What to send |
|---|---|---|

5. Click [🔒].

6. Type **New Incident Lab Message**.

7. Click the first autocomplete result to add the message.



8. Click [Update].
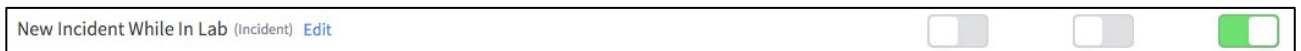
## Subscribe to the Push Notification

1. In the navpage header, click .

2. Click **Profile**.

3. Click **Notification Preferences** under *Related Links*.

4. Type **new** in the search input.



5. Find the **New Incident While In Lab** notification

6. In the **ServiceNow Mobile Application** column, click [⬜].

7. Notice the notification is enabled.



**Note:** Notifications can be automatically enabled for users if **Push Default Registrations** are configured. For more information, go to the *Push Notifications Lab* during Creator Con.

## Use the Native App, Test Push Notifications

**Note**: If you are not using the Native iOS Application you cannot test push notifications.

1. Switch to the **Native App**.

2. Tap your phone **Home** button to background the ServiceNow App.

## Use the Desktop, Test Push Notifications

1. Switch to the **Desktop UI**.

2. Navigate to **Incident > Open**.

3. Click [New].

4. Input form values:

   Caller: **Abel Tuter**
   Short description: **Testing out push notifications for the lab**

5. Click [Submit].

6. After a short delay, you should receive a push notification on your phone for the newly created incident.

# Lab Goal

This section covers the configuration of input helpers for the ServiceNow Native App. Input helpers are only available on the ServiceNow Native App.
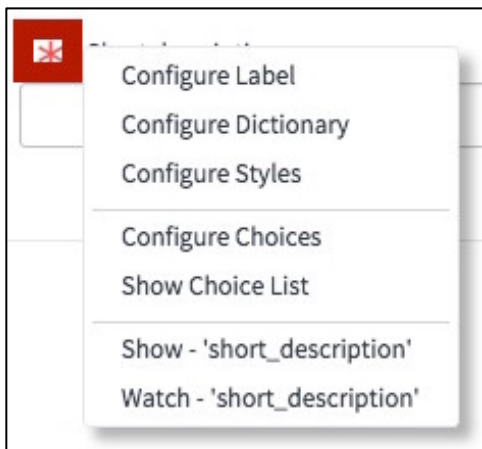
- Enable the Barcode Scanner Input Helper
- Enable the Current Location Input Helper
- Use the Native App, Test Input Helpers

## Enable the Barcode Scanner Input Helper

1. Switch to the **Desktop UI**.
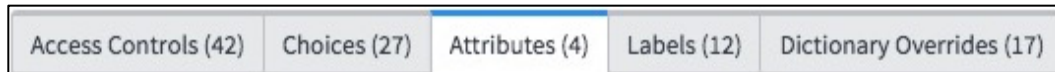
2. Navigate to **incident.do**.



3. Right click **Short description**.



4. Click **Configure dictionary**.

5. Click the **Attributes** tab.

| Access Controls (42) | Choices (27) | Attributes (4) | Labels (12) | Dictionary Overrides (17) |

6. Click New on the **Attributes** related list.

7. Click Q.

8. Search for **barcode**.

9. Click **Enable barcode scanner** to select the attribute.

10. Input form values:

   Value: **true**

   **Note:** The value must be lowercased **true** exactly.

11. Click Submit.

## Enable the Current Location Input Helper

1. Click New on the **Attributes** related list.

2. Click Q.

3. Search for **location**.

4. Click **Enable current location input helper** to select the attribute.

5. Input form values:

   Value: **true**

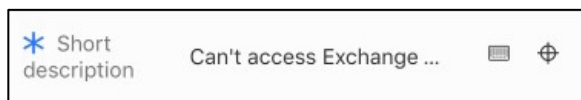   **Note:** The value must be lowercased **true** exactly.

6. Click Submit.

7. Click  .

## Use the Native App, Test Input Helpers

**Note**: If you are not using the Native iOS Application you cannot test input helpers.

1. Switch to the **Native App**.

2. Navigate to **Incident > Active**.

3. Tap any incident.

4. Tap the  to view the incident form.

5. Notice the input helpers next to the **Short description** field.



6. Tap  .

7. Scan this barcode:
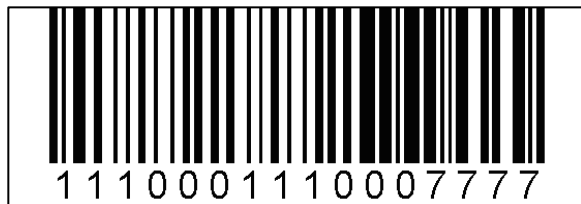


8. Notice the **Short description** value set to **1110001110007777**.

9. Tap  .

   **Note:** If prompted to enable location sharing, click **Allow**.

10. Notice the **Short description** value set to your current location.

# Lab Goal

This section covers Client Scripting fundamentals when working with the ServiceNow Mobile Experience. We explore asynchronous form validation as well as basic Client Script debugging techniques.

- Create an Asynchronous Client Script
- Test an Asynchronous Client Script
- Create a Broken Client Script
- Debug a Broken Client Script

## Create an Asynchronous Client Script

1. Switch to the **Desktop UI**.

2. Navigate to **incident.do**.

3. Click ▤.

4. Click **Configure > Client Scripts**.

5. Click New.

6. Input form values:

    Name: **Async scripting in lab**
    UI Type: **Both**
    Type: **onSubmit**

7.  Copy **Script**:

```
function onSubmit() {

  // store a custom value to determine
  // if we passed validation
  if (g_form.isCustomValidated === true) {
    console.log('Custom validation for the form passed!');
    return true;
  }
  var gr = new GlideRecord('incident');
  gr.addQuery('active', true);

  // save the action name
  var actionName = g_form.getActionName();
  gr.query(function() {
    console.log('GlideRecord rows: ' + gr.getRowCount());

    // we have GlideRecord results here
    // but we'll resubmit with our
    // custom validation flag set
    g_form.isCustomValidated = true;

    // resubmit with the previous action name
    g_form.submit(actionName);
  });
  console.log('Halting submit -- loading data...');
  return false;
}
```

8.  Click `Update`.

## Test an Asynchronous Client Script

1.  Switch to the **Mobile Web UI** using your desktop.

    **Note**: To facilitate debugging Client Scripts, we test on a desktop browser. Refer to **Section 2.1: Access the Mobile Web UI on your Desktop**.

2.  Navigate to **Incident > Active**.

3.  Tap any incident.

4. Tap [ > ] to view the incident form.

5. Open your browser's console to view log messages:
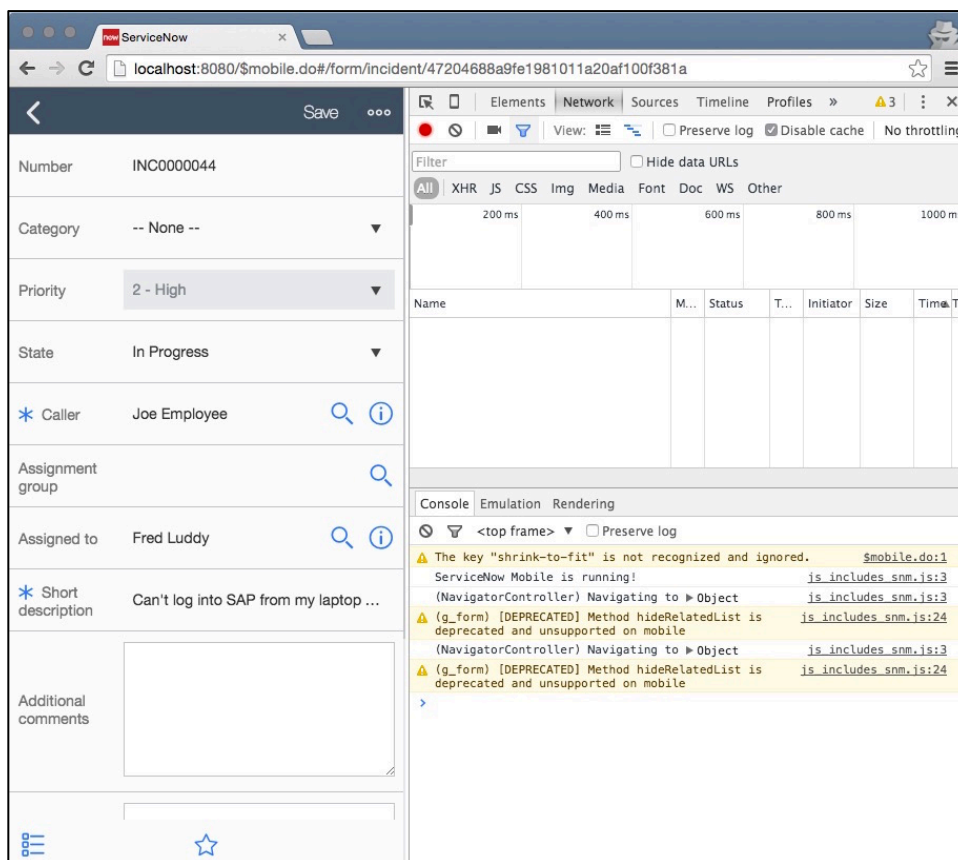
    Chrome Mac: **CMD + OPTION + I**
    Chrome Windows: **CTRL + SHIFT + I**
    Firefox Mac: **CMD + OPTION + K**
    Firefox Windows: **CTRL + SHIFT + K**
    Safari Mac: **CMD + OPTION + I**



6. Tap [ Save ].

7. Notice the console messages are output from the Client Script.

## Create a Broken Client Script

1. Switch to the **Desktop UI**.

2. Navigate to **incident.do**.

3. Click ☰.

4. Click **Configure > Client Scripts**.

5. Click New.

6. Input form values:

   Name: **Broken script in lab**
   UI Type: **Both**
   Type: o**nSubmit**

7. Copy **Script**:

```
function onSubmit() {

  // we did some validation?
  if (isValidated) {
    console.log('Form checks out!');
  }
  return true;
}
```

8. Click Submit.

## Debug a Broken Client Script

1. Switch to the **Mobile Web UI** using your desktop.

2. Navigate to **Incident > Active**.

3. Tap any incident.

4. Tap [ > ] to view the incident form.

5. Open your browser's console to view log messages.

6. Notice the error message in the console.

7. Switch to the **Desktop UI**.

8. Find **Broken script in lab** in the list of Client Scripts.

9. Click ⓘ to edit the Client Script.

10. Copy **Script**:

```
function onSubmit() {

  // we did some validation!
  var isValidated = true;
  if (isValidated) {
    console.log('Form checks out!');
  }
  return true;
}
```

11. Click Update .

12. Switch to the **Mobile Web UI**.

13. Refresh the current incident using your browser's refresh button ⟳ .

14. Retest the Client Script by tapping Save .

15. Notice no errors in the browser's console.

# Lab Goal

This section covers UI Action fundamentals when working with the ServiceNow Mobile Experience. This lab explores different configurations available for Mobile UI Actions.

- Create a Hidden UI Action
- Create an Update Style UI Action
- Create a Redirecting UI Action
- Create a List UI Action
- Test a Hidden UI Action
- Test a Redirecting UI Action
- Test an Update Style UI Action
- Test a List UI Action

## Create a Hidden UI Action

1. Switch to the **Desktop UI**.

2. Navigate to **System Mobile UI > UI Actions – Mobile**.

3. Click [New].

4. Input form values:

   Name: **Hidden Test**
   Action name: **hidden_test**
   Table: **Incident**
   Form button: **Check**
   Condition: **current.category == 'hardware'**
   Hide When Disabled: **Check**

5. Click [Submit].

## Create an Update Style UI Action

1. Click [New].

2. Input form values:

   Name: **Back Test**
   Action name: **back_test**
   Table: **Incident**
   Form more item: **Check**
   Navigate back on completion: **Check**

3. Click Submit .

## Create a Redirecting UI Action

1. Click New .

2. Input form values:

   Name: **Redirect Test**
   Action name: **redirect_test**
   Table: **Incident**
   Form more item: **Check**

3. Copy **Script**:

```
var gr = new GlideRecord('incident');
gr.addQuery('category', 'hardware');

// create a desktop URL
var url = 'incident_list.do?sysparm_query=';
url += gr.getEncodedQuery();
action.setRedirectURL(url);
```

4. Click Submit .

## Create a List UI Action

1. Click New .

2. Input form values:

    Name: **List Test**
    Action name: **list_test**
    Table: **Incident**
    List button: **Check**
    Color: **#FF77CC**
    Condition: **current.state == 1**

3. Copy **Script**:

```
current.setValue('state', 2);
current.update();
```

4. Click Submit .

# Test a Hidden UI Action

1. Switch to the **Mobile Web UI** or **Native App**.

2. Navigate to **Incident > Active**.

3. Tap an incident record.

4. Tap ❯ to show the form.

5. Change the **Category** to **Hardware**.
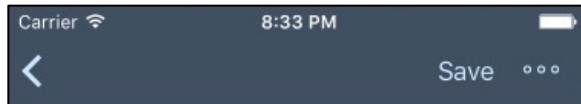
6. Tap Save .

7. Notice the **Hidden Test** UI action is visible.



8. Change the **Category** to **Software**.

9. Tap Save .

10. Notice the **Hidden Test** UI action is hidden.

## Test a Redirecting UI Action

1.  Tap .

2.  Tap **Redirect Test**.

3.  Notice the form redirects to the incident list with the specified filter.

## Test an Update Style UI Action

1.  If you are not on an incident list, navigate to **Incident > Active**.

2.  Tap an incident record.

3.  Tap  to show the form.

4.  Tap .

5.  Tap **Back Test**.

6.  Notice the form navigates back to the list.

## Use the Native App, Test a List UI Action

**Note**: If you are not using the *Native App*, skip this section.

1.  Switch to the *Native App*.

2.  Navigate to **Incident > Active**.

3. Swipe left on an incident with state **New** to view UI actions.
   **Note**: We've configured the UI action to only show on records with state == 'New'. If you don't see the state reflected on the incidents, search for "trouble" and try swiping the resulting record
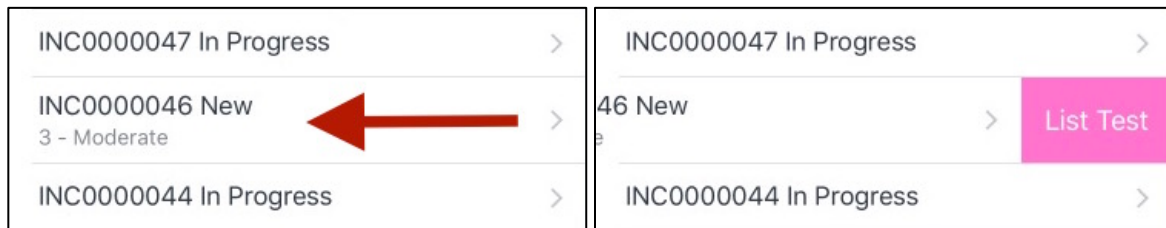


4. Tap **List Test**.

5. Notice the UI action is triggered.

## Use the Mobile Web UI, Test a List UI Action

**Note**: If you are not using the *Mobile Web UI*, skip this section.

1. Switch to the *Mobile Web UI*.

2. Navigate to **Incident > Active**.

3. Tap **List Test**.

4. Notice the UI action is triggered.

# Complete

Congratulations on completing this lab. Review all the functionality we covered:

- Mobile UI Configuration
- Mobile Navigator Apps
- Favorites
- Lists and List Configuration
- Push Notifications
- Mobile Form Input Helpers
- Mobile Form Client Scripts
- Mobile UI Actions

**Lab 10.1 Complete**

## Questions?

Now that you completed the lab, we want your feedback. Stick around after the lab to ask the mobile team your mobility-related questions.

servicenow