

**think 2018**

**IBM**



## **Lab Center – Hands-on Lab**

**Session 8424**

**Session Title Building Integrated Apps for IBM Collaboration**

Stefano Pogliani, IBM, stefano.pogliani@fr.ibm.com

Heath McCarthy, IBM, heath\_mccarthy@us.ibm.com

## **Table of Contents**

<b>Disclaimer</b> .....	3
<b>Heading</b> .....	Error! Bookmark not defined.
Subheading Level 1.....	Error! Bookmark not defined.
Subheading Level 2.....	Error! Bookmark not defined.
<b>Heading</b> .....	Error! Bookmark not defined.
<b>We Value Your Feedback!</b> .....	53

## **Disclaimer**

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.

In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

**Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

© 2018 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

**U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**

# Introduction

This workshop was prepared as step by step instruction. It should lead you through the use of the new IBM Connections Cloud extensions and components. All included exercises are going to support you in creating an integration between IBM Connections Cloud and Salesforce.

The way in which the integration works requires both

- Client-side integration  
This part will be experimented during the execution of the labs
- Server-side integration  
This part will be described and shown but you will not have to develop anything server-side during these labs.  
The information provided during the lab, though, should give you enough information to experiment and implement yourself such an integration.

All exercises are fulfilled with screen shots for easier navigation. However, there may be some small differences based on your environment, browser version, language or user account used.

## Icons

The following symbols appear in this document at places where additional guidance is available.

Icon	Purpose	Explanation
	Important!	This symbol calls attention to a particular step or command. For example, it might alert you to type a command carefully because it is case sensitive.
	Information	This symbol indicates information that might not be necessary to complete a step but is helpful or good to know.
	Trouble-shooting	This symbol indicates that you can fix a specific problem by completing the associated troubleshooting information.

## Code, Solutions, Examples

A special Github repository has been created where solutions, examples and the code or the graphics to be used are made available to all the students. We will reference this Github repository throughout this document as “**Reference Repository**”.

This Github repository is available at this address:

<https://github.com/stefanopog/IntegrationThinkLab>

No description, website, or topics provided.

Add topics

6 commits 1 branch 0 releases 1 contributor

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

Stefano Pogliani and Stefano Pogliani ok Latest commit ddd2d71 a minute ago

Lab1 ok a minute ago

README.md Update README.md 13 days ago

README.md

## IntegrationThinkLab

IBM Connections Integration Labs for Think2018

You may want to open a tab of your browser on this address in order to use the material. Feel free to clone this repository for better analyzing the code and the examples.

### Documentation

The text of these exercises, together with other useful documentation, is stored in the Docs folder of the Github repository mentioned above.

## Lab Setup      Useful information

You will all have access to “your own” (for the time being of the lab) IBM Connections Cloud organization.

You will be granted Administrator access to such organization.

Information
URL : <a href="https://apps.na.collabserv.com">https://apps.na.collabserv.com</a>
Username : thinkattendeeX@yopmail.com
 Password : passw0rd
The X is a number from 1 to 20 that will be provided to you by the instructor.

## Lab 1 Extending the OrientMe Screen

This is a very simple exercise, which consists in adding a very simple customization to the OrientMe interface.

What we want to achieve

We are adding:

1. A new tab to the interface

The screenshot shows the OrientMe interface with a new tab labeled 'IBM.com' added to the top navigation bar. A red arrow points to this tab. Below the navigation bar, there is a modal window for IBM with the text 'GDPR enforcement is less than six months away'. The entire modal window is highlighted with a red border.

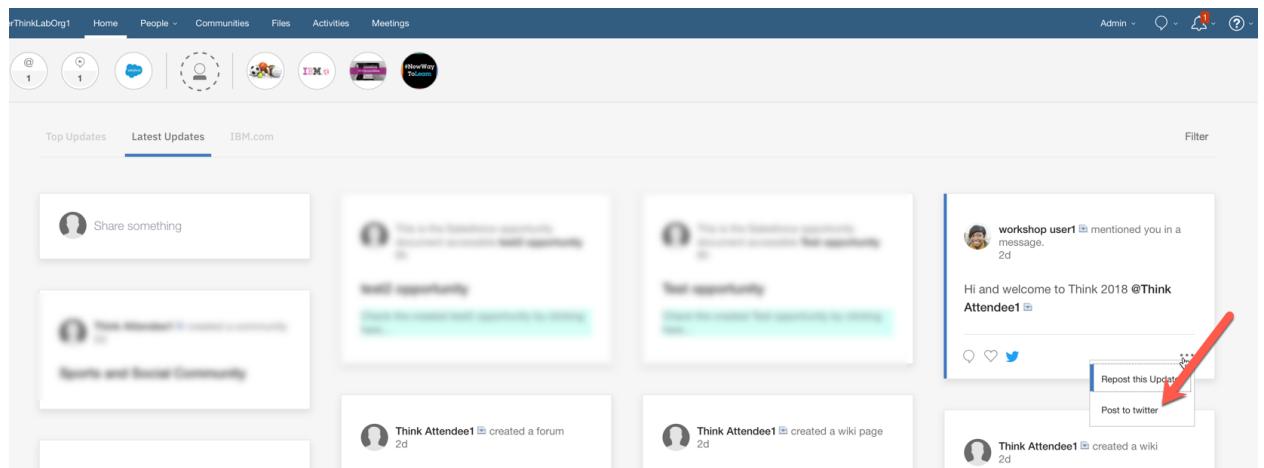
A New tab will give you the possibility to provide users with contextual access to frequently used and important things for doing their business

2. A new Action to the standard tiles

The screenshot shows the OrientMe interface with a new action labeled 'Twitter' added to the standard tiles. A red arrow points to the Twitter icon in the tile actions. The interface includes a navigation bar with tabs like Home, People, Communities, Files, Activities, and Meetings, and a sidebar with various notifications and messages.

The action will provide users a quick shortcut towards frequently used operations. In this case we used “Twitter” (and it is very unlikely that you will promote Tweeting information that may come from internal sources...). But the Action could link the user to some business-related application that would make a great use of the information without forcing the user to copy/paste it or to change the working context.

### 3. A new Menu to the standard tiles:



The Menu item provides the same advantages as the previously described Action.

## Preparation Activities

In order to do those modifications, we need to use an extension to the OrientMe page.

The code of the extension is already prepared and available within the **Reference Repository**, under the **Lab1** folder as shown by the image below:

A screenshot of a GitHub repository page for 'stefanopog / IntegrationThinkLab'. The repository has 0 issues, 0 pull requests, 0 projects, 1 wiki page, and 0 insights. The 'Code' tab is selected. A red arrow points to the 'OrientMe.json' file in the list of files. The repository has 0 stars, 0 forks, and 0 commits. The latest commit was made a minute ago by 'stefanopog' and renamed 'RADME.md' to 'README.md'. Below the repository details, there's a preview of the 'IntegrationThinkLab' page, which includes a title, a subtitle, and a navigation bar at the bottom.

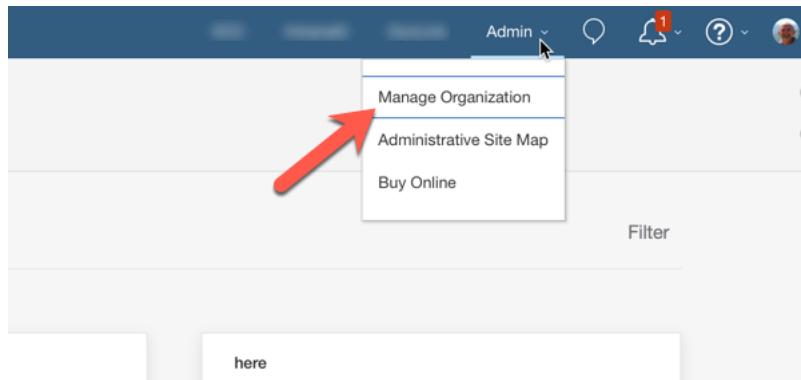
What you need to do is to add the extension definition to the IBM Connections Application Registry.

## Environment SetUp

Open a new window on your browser and log into IBM Connections

<https://apps.na.collabserv.com> providing the credentials the instructors gave you.

Navigate to the Administration screen:



and, once there, go to the “Organization Extensions” panel:

A screenshot of the "User Accounts" page in the IBM Connections administration interface. The page has a dark blue header with various navigation links like Home, Mail, Calendar, etc. Below the header is a "Administration" section. On the left, there is a sidebar with several sections: "Personal" (My Account Settings), "User Accounts" (selected, shown in blue), "System Settings" (Security, Theme, etc.), and "New Apps Manager" (part of the "Organization Extensions" section). A red arrow points to the "Organization Extensions" link in the sidebar. The main content area shows a list of user accounts with columns for Name and Email. The names listed are Amadou Alain, Charles Bounar, Christina Milan, Ed Blanks, Ed El-Amon, Frank Adams, and Gail Chao.

Then move to the **New Apps Manager** as shown below:

Administration

Personal  
My Account Settings

User Accounts  
Organization Account Settings  
Partitions  
Subscriptions  
Announcements  
Internal Apps  
Order History

Organization Extensions  
Connections Mobile App Management  
Chat and Meetings  
Apps

System Settings  
Security  
Theme  
IBM SmartCloud Notes  
Profile Customization

Applications For [REDACTED]

Info: To create Connections Pink applications, view the [new Apps Manager](#)

Add Export Delete Filter by Service: [dropdown]

Preferences

[List of application cards]

Click on the blue “New App” button:

Administration

Personal  
My Account Settings

User Accounts  
Organization Account Settings  
Partitions  
Subscriptions  
Announcements  
Internal Apps  
Order History

Organization Extensions  
Connections Mobile App Management  
Chat and Meetings  
Apps

System Settings  
Security

Info: To create classic Organization Extensions, [click here](#)

Apps All Apps

New App +

[List of application cards]

and then, click on the “Code Editor” button as shown here:

The screenshot shows the IBM Connections Administration interface. On the left, there's a sidebar with several sections: Personal (My Account Settings), Organization Extensions (Connections Mobile App Management, Chat and Meetings, Apps), and System Settings (Security, Theme, IBM SmartCloud Notes). The 'Organization Extensions' section is currently active. In the main content area, there's a 'Basic Information' panel with fields for 'App Name (required)' and 'Short Description (required)'. Below these are 'Cancel' and 'Save' buttons. A red arrow points to the 'Code Editor' tab in the sidebar.

Replace the few lines on the right panel with the content of the “OrientMe.json” file you [previously found](#) on Github and then press the “Save” button:

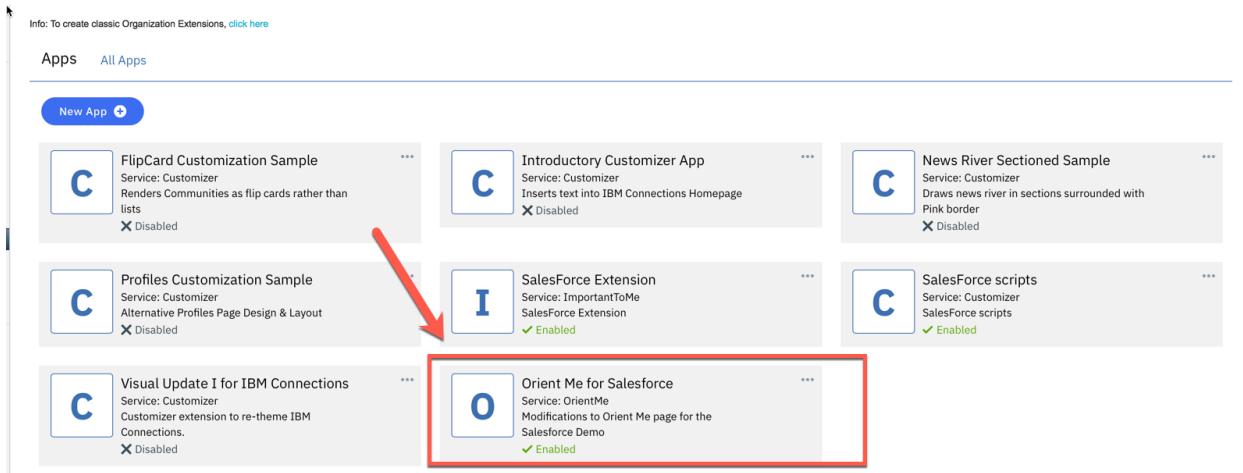
The screenshot shows the same interface as above, but now the 'Code Editor' tab is selected. A warning message in a callout box says: "Warning Import will overwrite editor contents." Below the code editor area, there are 'Cancel' and 'Save' buttons. A red arrow points to the 'Save' button. The code editor contains JSON code:

```

38     "localRefName": "initials_people"
39
40     "actions": [
41       {
42         "type": "create",
43         "visibleFieldId": "bgdiv",
44         "enabled": true,
45         "script": {
46           "source": "https://apps.ce.collabserve.com/files/customizer/orient&lt;br/&gt;_oppy.js",
47           "function": "__createOppy"
48         },
49         "icon": {
50           "type": "png",
51           "data": "data:image/jpeg;base64,/9j/4AAQSkZJRGABAEASABIAAD/7QBkUC
52         },
53         "label": "Create Opportunity"
54       },
55       {
56         "type": "search",
57         "icon": {
58           "type": "png",
59           "data": "data:image/jpeg;base64,/9j/4AAQSkZJRGABAEASABIAAD/7QBkUC
}

```

What does this “Registry Application” do? (you can explore the code you just pasted...)



It actually added the following extensions:

- The Twitter Tile Action
- The Twitter Tile Menu
- The new IBM Tab

Feel free to explore the code that you just pasted. The complete documentation for that code is available here: [https://www-10.lotus.com/ldd/appdevwiki.nsf/xpDocViewer.xsp?lookupName=Dev+Guide+topics#action=openDocument&res\\_title=Adding\\_custom\\_entries\\_to\\_Important\\_to\\_Me&content=sdkcontent](https://www-10.lotus.com/ldd/appdevwiki.nsf/xpDocViewer.xsp?lookupName=Dev+Guide+topics#action=openDocument&res_title=Adding_custom_entries_to_Important_to_Me&content=sdkcontent)

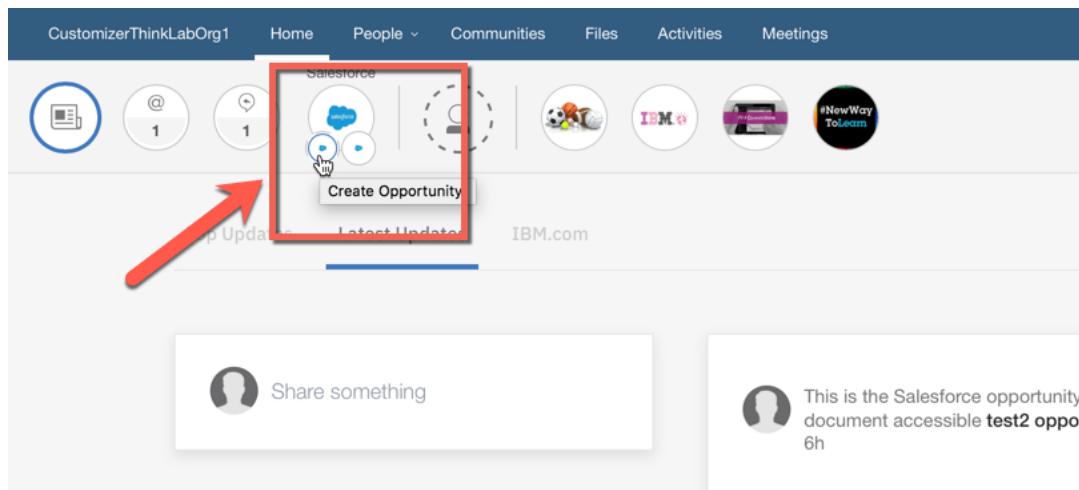
## Lab 2 Creating a new bubble in OrientMe

This is a very simple exercise, which consists in adding a new bubble to the IBM Connections OrientMe interface.

This bubble will give us the possibility to interact with our own Salesforce.com account from within the IBM connections user interface, thus streamlining the daily activities we perform.

What we want to achieve

Once we finish this second lab, our IBM Connections OrientMe page will look like the following:



This bubble will be displayed on the IBM connections OrientMe page of all the users at all the time and provides the following main functionalities:

- Easy access to the customer's Salesforce portal
- Easy creation of an Opportunity
- Easy display of an opportunity
- ....

All of this integrated with the “tiles” shown on the page as we will see during the exercise itself...

### Preparation Activities

In order to do those modifications, we need to add a new extension to the OrientMe page. The code of the extension is already prepared and available within the **Reference Repository** under the **Lab2** folder.

stefanopog / IntegrationThinkLab

Code Issues Pull requests Projects Wiki Insights Settings

Branch: master ▾ IntegrationThinkLab / Lab2 /

Stefano Pogliani and Stefano Pogliani ok Latest commit 8675ef2 11 minutes ago

..

**README.md** ok 11 minutes ago

**SalesforceExtension.json** ok 11 minutes ago

README.md

**IntegrationThinkLab**

IBM Connections Integration Labs for Think2018

**Lab2 : Creating a new OrientMe "bubble"**

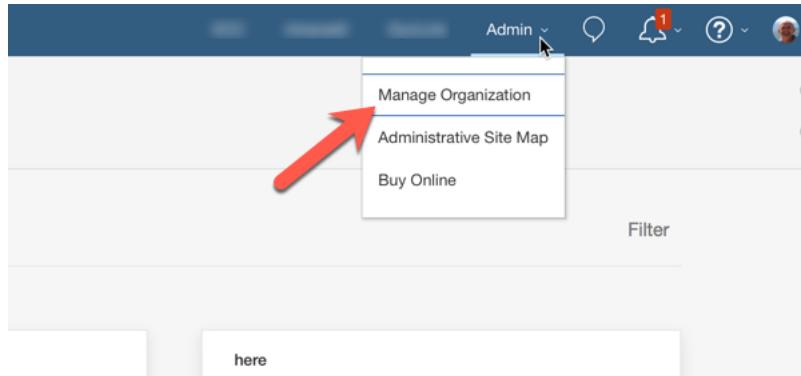
What you need to do is to add the extension definition to the IBM Connections Application Registry.

## Environment SetUp

Open a new window on your browser and log into IBM Connections

<https://apps.na.collabserv.com> providing the credentials the instructors gave you.

Navigate to the Administration screen:



and, once there, go to the “Organization Extensions” panel:

The screenshot shows the IBM Connections Administration interface. The left sidebar has a dark blue header 'User Accounts'. Below it, under 'Organization Extensions', there is a link 'Connections Mobile App Management'. A red arrow points to this link.

**User Accounts**  
Create new user accounts or manage existing accounts. [?](#)

You have 2 of 25 bundled subscriptions available. Overage Available.

[Add User Account](#) [Resend All Expired Invitations](#) [Export User Account List](#)

1 - 10 of 24

Name	Email
Amadou Alain	[REDACTED]
Charles Bounar	[REDACTED]
Christina Milan	[REDACTED]
Ed Blanks	[REDACTED]
Ed El-Amon	[REDACTED]
Frank Adams	[REDACTED]
Gail Chao	[REDACTED]

Then move to the **New Apps Manager** as shown below:

The screenshot shows the IBM Connections Administration interface. The left sidebar has a dark blue header 'Organization Extensions'. Below it, there is a link 'Connections Mobile App Management'. A red arrow points to the text 'Info: To create Connections Pink applications, view the [new Apps Manager](#)'.

**Applications For [REDACTED]**

Info: To create Connections Pink applications, view the [new Apps Manager](#)

[Add](#) [Export](#) [Delete](#) Filter by Service: [\[REDACTED\]](#)

**Preferences**

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

Click on the blue “New App” button:

The screenshot shows the IBM Connections Administration interface. On the left, there is a sidebar with several sections: Personal (My Account Settings), Organization Extensions (Connections Mobile App Management, Chat and Meetings, Apps), and System Settings (Security). The Organization Extensions section is currently selected. In the main content area, there is a heading "Info: To create classic Organization Extensions, click here". Below it, there are two tabs: "Apps" and "All Apps". A prominent blue button labeled "New App" with a plus sign is centered above a list of app cards. A large red arrow points directly at this "New App" button.

and then, click on the “Code Editor” button as shown here:

The screenshot shows the "Basic Information" form for creating a new app. The left sidebar is identical to the previous screenshot, showing the "Organization Extensions" section. The main content area has a header "Basic Information". It includes fields for "App State" (set to "Enabled"), "App Name (required)", and "Short Description (required)". At the bottom are "Cancel" and "Save" buttons. On the left side of the main content area, there is a vertical navigation menu with three items: "Back to Apps", "Basic Information" (which is highlighted with a blue background), and "Code Editor" (which is highlighted with a blue background and has a red arrow pointing to it).

Replace the few lines on the right panel with the content of the “SalesforceExtension.json” file you [previously found](#) on Github and then press the “Save” button:

```

38      "localRefName": "initials_people"
39
40    },
41    "actions": [
42      {
43        "type": "create",
44        "visibleField": "bgdiv",
45        "enabled": true,
46        "script": {
47          "source": "https://apps.ce.collabobserv.com/files/customizer/orient&lt;br/&gt;_createOppty"
48        },
49      },
50      {
51        "icon": {
52          "type": "png",
53          "data": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEASABIAAD/7QBkUC
54        },
55      },
56      {
57        "label": "Create Opportunity"
58      }
59    ]

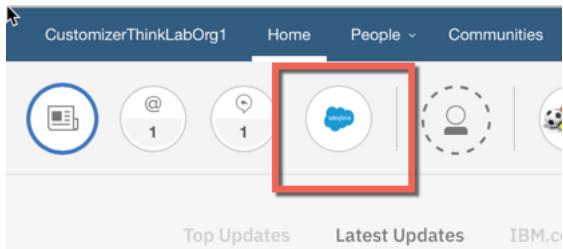
```

What does this “*Registry Application*” do? (you can explore the code you just pasted...)

It actually does several things:

- It declares an extension of type “com.ibm.itm.entry.custom” whose name is **“CustomBubble”**.

This extension represents the Bubble itself as you see it in the picture here below:



It mainly declares the icon shown in the center of the “bubble” and the action that the

interface will play when the user will click on the icon, which is executing the “`__salesforceDemo`” function from the script (**mySalesforce.js**) which is defined by the URL pointed to by the “**source**” attribute

**Information**

Note the use of the “`?repoName=think-samples`” suffix in the URL.

This instructs IBM Connections to look for this file in the specific Files Repository. Each IBMM Connections Cloud organization has its own repository, and other repositories may exist to share code.

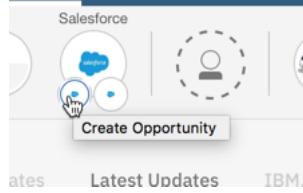


A copy of the **mySalesforce.js** file is available for your convenience under the **Reference Repository** (path **resources/svc\_scripts**).

You will not modify any of the files under resources during this lab, but you are welcome to study them and to ask the instructor explanations.

- It declares **TWO** Actions for the above-mentioned extension:

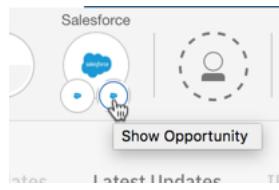
- Create:



This action defines

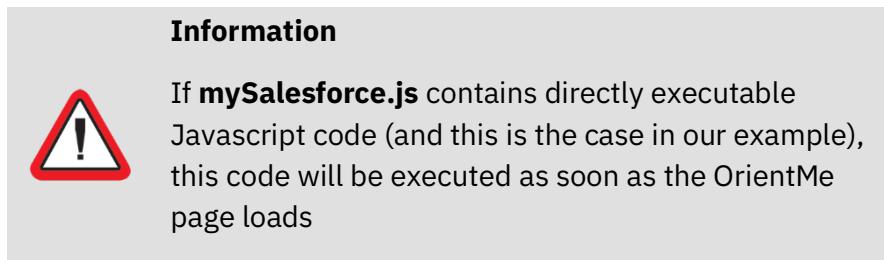
- the icon (in our case we used the same icon for the action and for the “bubble”, but they could be different ....)
- the Label that will be shown to the user hovering with the mouse
- the code that the interface will play when the user will click on the action (in our case, clicking the action will execute the “`__createOppy_stub`” function from the script (**mySalesforce.js**) which is defined by the URL pointed to by the “**source**” attribute

- Show:



This action defines

- the icon (in our case we used the same icon for the action and for the “bubble”, but they could be different ....)
- the Label that will be shown to the user hovering with the mouse
- the code that the interface will play when the user will click on the action (in our case, clicking the action will execute the “`_showOppy_stub`” function from the script (**mySalesforce.js**) which is defined by the URL pointed to by the “**source**” attribute
- It declares another extension which tells IBM Connections to load the **mySalesforce.js** script from the `think-samples repoName` when it loads the page. This is particularly useful in our case because we do want to apply some processing to the OrientMe page independently from the fact that the Javascript file also contains the code used to implement the actions on the “bubble”.



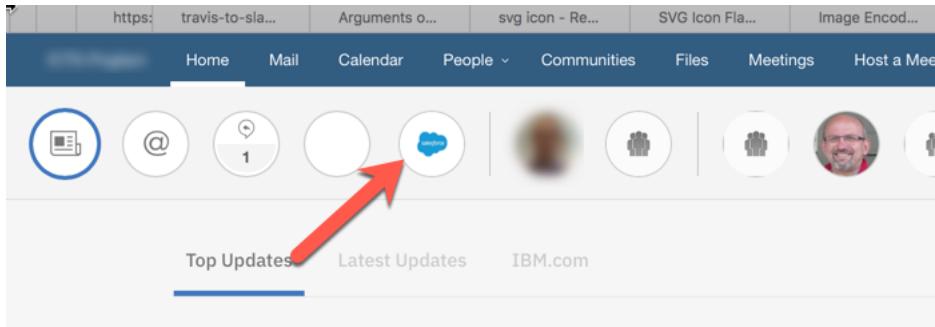
Feel free to explore the code that you just pasted. The complete documentation for that code is available here: [https://www-10.lotus.com/ldd/appdevwiki.nsf/xpDocViewer.xsp?lookupName=Dev+Guide+topics#action=openDocument&res\\_title=Adding\\_custom\\_entries\\_to\\_Important\\_to\\_Me&content=sdkcontent](https://www-10.lotus.com/ldd/appdevwiki.nsf/xpDocViewer.xsp?lookupName=Dev+Guide+topics#action=openDocument&res_title=Adding_custom_entries_to_Important_to_Me&content=sdkcontent)

## Verifying

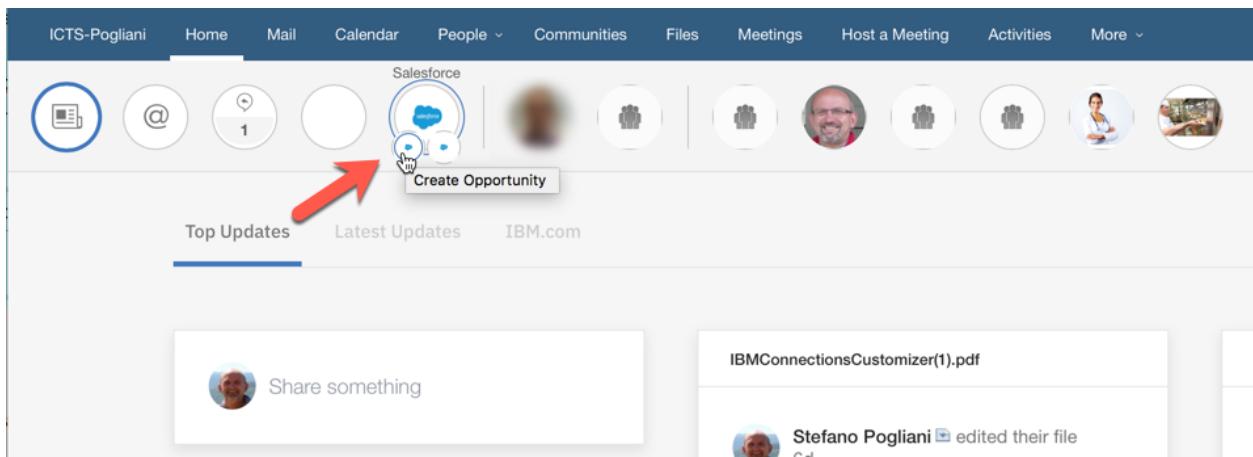
Go back to your OrientMe page by clicking on the **Home** link on the top Menu bar:

The screenshot shows the OrientMe application's main menu bar at the top. The "Home" link is highlighted with a red arrow pointing to it. Below the menu bar, there is a sidebar on the left with various navigation options like Personal, User Accounts, Organization Account Settings, etc. To the right of the sidebar, there is a main content area with sections for Apps, New App, and two preview cards for "Classic Communication Sample" and "Smart Worker".

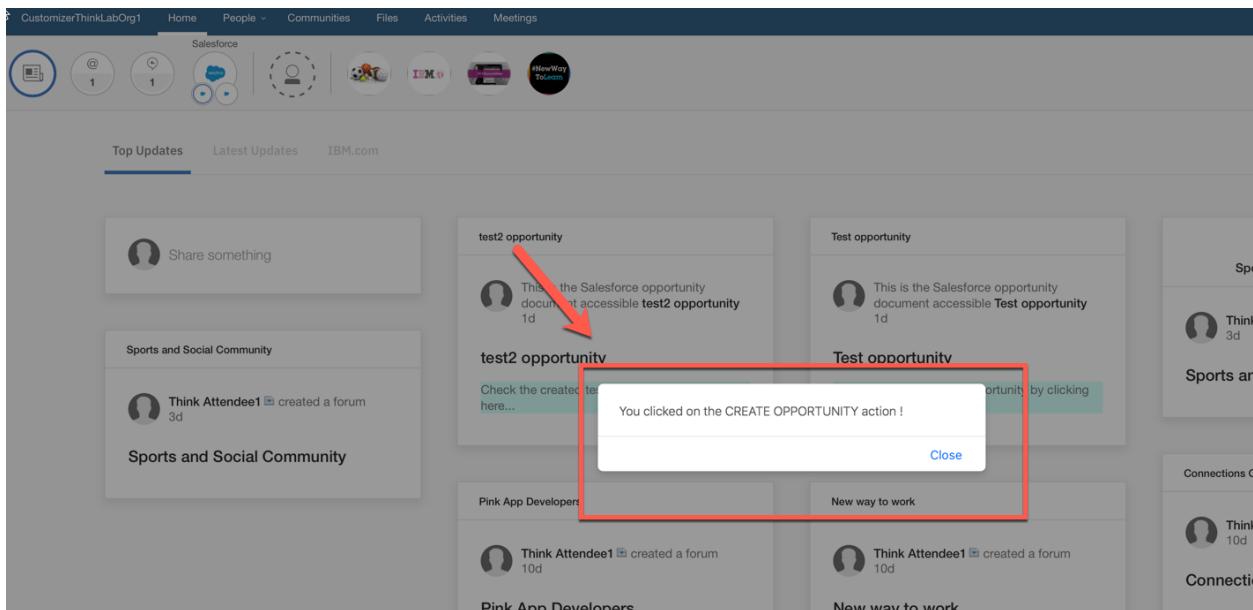
The new “bubble” now appears.



Now, click on the “Create Opportunity” action bubble as shown here:



And you will see a default script taking place:



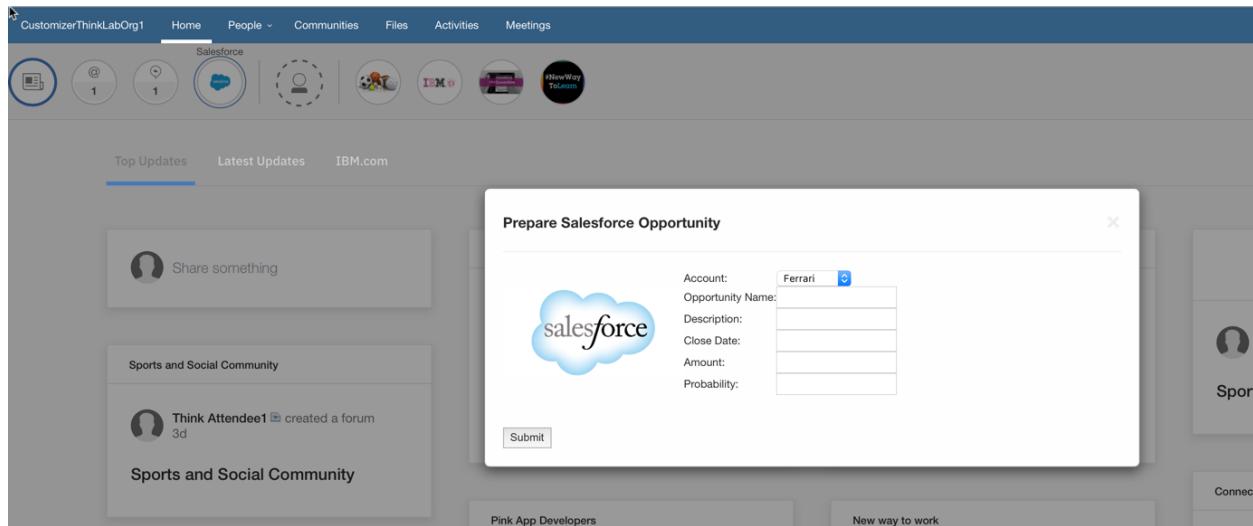
In the next labs, we will create the magic that connects us to the Salesforce APIs.

## Lab 3 Create a Salesforce Oppy from IBM Connections

In this lab we will see how to create a Salesforce Opportunity using the “Create” action we previously added to our “bubble”

### What we want to achieve

Once we finish this third lab, a new popup window will be shown from within the IBM Connections OrientMe page; this popup collects the information that it will use to finally create a new Salesforce opportunity.



### Preparation Activities

In order to do those modifications, we need to add a new extension to the OrientMe page. The code of the extension is already prepared and available within the **Reference Repository** under the **Lab3** folder.

stefanopog / IntegrationThinkLab

Code Issues Pull requests Projects Wiki Insights Settings

Branch: master IntegrationThinkLab / Lab3 /

Stefano Pogliani and Stefano Pogliani ok Latest commit 36a9771 just now

..

**README.md** ok 4 hours ago

**SalesforceScripts.json** ok just now

README.md

## IntegrationThinkLab

IBM Connections Integration Labs for Think2018

**Lab3 : Create a Salesforce Oppy from IBM Connections**

CustomizerThinkLabOrg1 Home People Communities Files Activities Meetings

Top Updates Latest Updates IBM.com

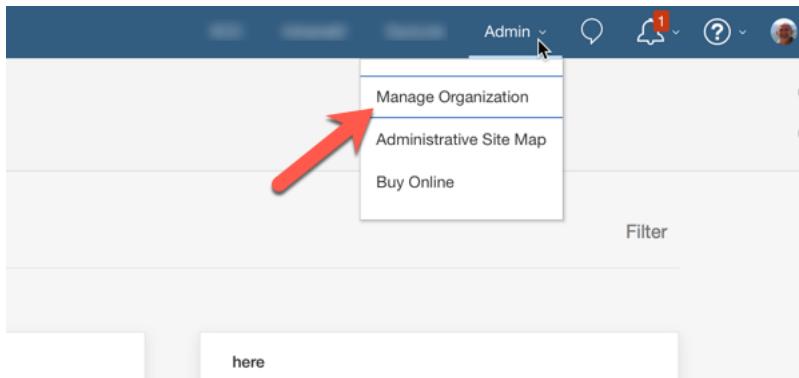
This extension will define the CSS classes that will be used to display the modal dialog shown at the beginning of this chapter. We will use the standard **IBM Connections Customizer** to achieve this step.

### Environment SetUp

Open a new window on your browser and log into IBM Connections

<https://apps.na.collabserv.com> providing the credentials the instructors gave you.

Navigate to the Administration screen:



and, once there, go to the “Organization Extensions” panel:

The screenshot shows the IBM Connections Administration interface. The left sidebar has a dark blue header 'User Accounts'. Below it, under 'Organization Extensions', there is a link 'Connections Mobile App Management'. A red arrow points to this link.

**User Accounts**  
Create new user accounts or manage existing accounts. [?](#)

You have 2 of 25 bundled subscriptions available. Overage Available.

[Add User Account](#) [Resend All Expired Invitations](#) [Export User Account List](#)

1 - 10 of 24

Name	Email
Amadou Alain	[REDACTED]
Charles Bounar	[REDACTED]
Christina Milan	[REDACTED]
Ed Blanks	[REDACTED]
Ed El-Amon	[REDACTED]
Frank Adams	[REDACTED]
Gail Chao	[REDACTED]

Then move to the **New Apps Manager** as shown below:

The screenshot shows the IBM Connections Administration interface. The left sidebar has a dark blue header 'Organization Extensions'. Below it, there is a link 'Connections Mobile App Management'. A red arrow points to the text 'Info: To create Connections Pink applications, view the [new Apps Manager](#)'.

**Applications For [REDACTED]**

Info: To create Connections Pink applications, view the [new Apps Manager](#)

[Add](#) [Export](#) [Delete](#) Filter by Service: [\[REDACTED\]](#)

**Preferences**

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

## The CSS Script

Let's create the **IBM Connections Customizer** extension to inject the CSS file which contains the classes that will be used to display the modal window.

Click on the blue “New App” button:

The screenshot shows the IBM Connections Administration interface. On the left, there is a sidebar with sections for Personal (My Account Settings), Organization Extensions (Connections Mobile App Management, Chat and Meetings, Apps), and System Settings (Security). The main area is titled "Administration" and has a sub-section for "Organization Extensions". A red arrow points to the "New App" button, which is a blue button with a white plus sign and the text "New App". Below the button, there are three blurred preview cards for different apps, each featuring a large blue letter (C, C, I).

and then, click on the “Code Editor” button as shown here:

The screenshot shows the "Basic Information" page for creating a new app. The left sidebar is identical to the previous screenshot. The main area has a "Back to Apps" link and tabs for "Basic Information", "Extensions", and "Code Editor". A red arrow points to the "Code Editor" tab, which is highlighted in blue. To the right, there are fields for "App State" (Enabled), "App Name (required)", "Short Description (required)", and buttons for "Cancel" and "Save".

Replace the few lines on the right panel with the content of the “SalesforceScripts.json” file you [previously found](#) on Github and then press the “Save” button:

```

1 { "id": "76227bb0-2856-11e8-a352-0b5b8ccf086",
2   "name": "SalesforceScripts",
3   "title": "SalesForce scripts",
4   "description": "SalesForce scripts",
5   "services": [
6     "Customizer"
7   ],
8   "extensions": [
9     {
10       "id": "29ac380d-6940-456b-b872-d1d2aa401736",
11       "name": "CSS Stylesheet",
12       "type": "com.ibm.customizer.ui",
13       "payload": {
14         "include-files": [
15           "salesforce/svc_styles/mySalesforce.css"
16         ],
17         "include-repo": {
18           "name": "think-samples"
19         }
20       },
21       "path": "social",
22       "application": "SalesforceScripts"
23     }
24   ]
25 }
26

```

What does this “*Registry Application*” do? (you can explore the code you just pasted...)

App Name	Description	Status
FlipCard Customization Sample	Service: Customizer Renders Communities as flip cards rather than lists	Disabled
Introductory Customizer App	Service: Customizer Inserts text into IBM Connections Homepage	Disabled
News River Sectioned Sample	Service: Customizer Draws news river in sections surrounded with Pink border	Disabled
Profiles Customization Sample	Service: Customizer Alternative Profiles Page Design & Layout	Disabled
SalesForce Extension	Service: ImportantToMe SalesForce Extension	Enabled
Visual Update I for IBM Connections	Service: Customizer Customizer extension to re-theme IBM Connections.	Disabled
Orient Me for Salesforce	Service: OrientMe Modifications to Orient Me page for the Salesforce Demo	Enabled
<b>SalesForce scripts</b>	Service: Customizer SalesForce scripts	<b>Enabled</b>

Well, it simply tells IBM Connections:

- To load the **mySalesforce.css** script file for the [@think-samples repository](#) (we have seen what this means [during this step of a previous exercise](#))
- To load this CSS script only for the “**social**” service (actually, the OrientMe set of pages)

## The Modal Dialog

This time we will not create a new “*Registry Application*”, but we will modify an existing one. Click on the “**Salesforce Extension**” we created in the previous exercise.

The screenshot shows the IBM Connections Admin interface under the 'Apps' section. A red arrow points to the 'SalesForce Extension' app, which is highlighted with a red border. The app details are as follows:

<b>SalesForce Extension</b>	Service: ImportantToMe	SalesForce Extension

Other apps listed include: FlipCard Customization Sample, Introductory Customizer App, News River Sectioned Sample, Profiles Customization Sample, SalesForce scripts, Visual Update I for IBM Connections, and Orient Me for Salesforce.

This opens the configuration for the “**Salesforce Extension**” application. Click on the “**Code Editor**” button as shown here:

The screenshot shows the 'App Editor' screen for the 'SalesForce Extension' app. On the left, there is a sidebar with a 'Code Editor' button highlighted by a red arrow. The right panel displays the JSON definition of the Registry Application, with line numbers from 1 to 25 visible on the left side of the code area.

```
1 {  
2   "id": "6e44a0d0-2856-11e8-8f95-1fa2738fa741",  
3   "name": "SalesforceExtension",  
4   "title": "SalesForce Extension",  
5   "description": "SalesForce Extension",  
6   "services": [  
7     "ImportantToMe"  
8   ],  
9   "extensions": [  
10    {"  
11      "id": "5d0b088b-60dd-42cc-84bc-7053804b9122",  
12      "name": "CustomBubble",  
13      "type": "com.ibm.itm.entry.custom",  
14      "payload": {  
15        "label": "Salesforce.com",  
16        "name": "Salesforce",  
17        "expires": "2018-12-12",  
18        "show": true,  
19        "showHighlight": false,  
20        "script": {  
21          "source": "https://apps.na.collab.observ.c",  
22          "function": "__salesforceDemo"  
23        },  
24        "icon": {  
25          "url": "data:image/jpeg;base64,/9j/4AAQ"...,  
26        }  
27      }  
28    }  
29  ]  
30}  
31}
```

Now, on the right panel you will see the JSON definition of the Registry Application. Search for the “Create” Action section (should be around line 39).

Modify the value for the **function attribute** of the action from `__createOppy_stub` to `__createOppy`.

Info: To create classic Organization Extensions, [click here](#)

Apps All Apps

< Back to Apps App Editor

Basic Information Check out the help and examples section for working with apps, if you need some guidance.

Extensions

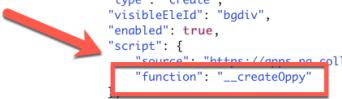
**Code Editor**

Delete App

Warning Import will overwrite editor contents. X

```

27
28     "initialsImg": {
29         "type": "jpg",
30         "localRefName": "initials_people"
31     },
32     "actions": [
33         {
34             "type": "create",
35             "visibleElId": "bgdiv",
36             "enabled": true,
37             "script": {
38                 "source": "https://apns-na.collabobserv.com/files/customizer/sale",
39                 "function": "__createOppy"
40             }
41             "icon": {
42                 "type": "png",
43                 "data": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEASABIAAD/7QBi
44             },
45             "label": "Create Opportunity"
46         },
47         {
48             "type": "show",
49             "icon": {
50                 "type": "png",
51                 "data": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEASABIAAD/7QBi
52             }
53         }
54     ]
55 }
```



Then press the “Save” button:

Home People Communities Files Activities Meetings Admin

Info: To create classic Organization Extensions, [click here](#)

Apps All Apps

< Back to Apps App Editor

Basic Information Check out the help and examples section for working with apps, if you need some guidance.

Extensions

**Code Editor**

Delete App

Discard Changes **Save Changes**

Warning Import will overwrite editor contents. X

```

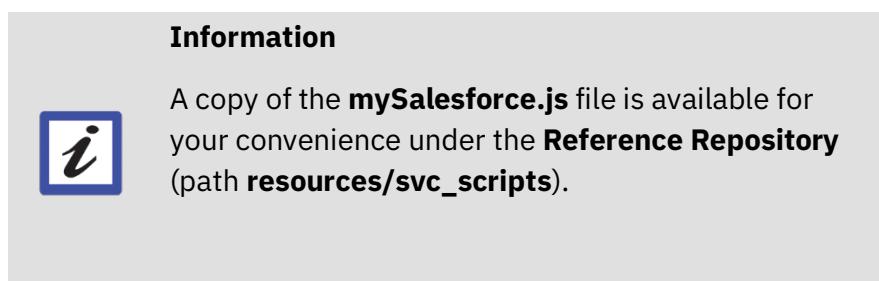
27
28     "initialsImg": {
29         "type": "jpg",
30         "localRefName": "initials_people"
31     },
32     "actions": [
33         {
34             "type": "create",
35             "visibleElId": "bgdiv",
36             "enabled": true,
37             "script": {
38                 "source": "https://apns-na.collabobserv.com/files/customizer/sale",
39                 "function": "__createOppy"
40             }
41             "icon": {
42                 "type": "png",
43                 "data": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEASABIAAD/7QBi
44             },
45             "label": "Create Opportunity"
46         },
47         {
48             "type": "show",
49             "icon": {
50                 "type": "png",
51                 "data": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEASABIAAD/7QBi
52             }
53         }
54     ]
55 }
```



What have we done here?

The javascript code inside the **mySalesforce.js** script defines several functions.

One of these functions is `__createOppy`. Look at its code:



```
function __createOppy() {
    _getSFDCAccounts();
}
```

Well, it simply invokes the `_getSFDCAccounts` function, right?

Let's now look at the code for the `_getSFDCAccounts` function.

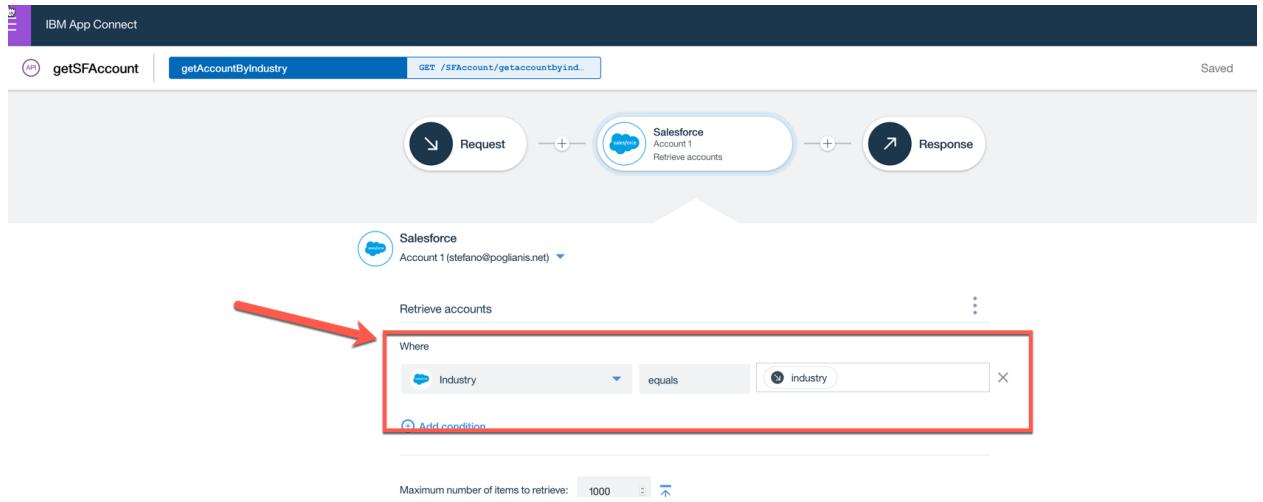
```
function _getSFDCAccounts() {
    var xmlhttpReq = false;
    if (window.XMLHttpRequest) {
        // Mozilla/Safari
        xmlhttpReq = new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        // IE
        xmlhttpReq = new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttpReq.open('GET',
        'https://service.us.apiconnect.ibmcloud.com/gws/apigateway/api/543aba51314bff5bc7db871ec21796e77ad958439786c0a57a',
        true);
    xmlhttpReq.setRequestHeader('accept', 'application/json');
    //xmlHttpReq.setRequestHeader('Content-Type', 'application/json');
    xmlhttpReq.setRequestHeader('x-ibm-client-id', '954ca1f5-b101-402e-877b-82e3a6ab264f');
    xmlhttpReq.onreadystatechange = function () {
        if (xmlhttpReq.readyState == 4) {
            if ((xmlhttpReq.status == 200)) {
                //alert(xmlhttpReq.responseText);
                __buildCreateOppyScreen(xmlhttpReq.responseText);
            } else {
                alert('Unfortunately there was an ERROR ' + xmlhttpReq.status + ' / ' + xmlhttpReq.statusText);
            }
        }
    };
    xmlhttpReq.send();
}
function __buildCreateOppyScreen(responseText) {
```

In addition to some vanilla code which defines an `XMLHttpRequest` object, configures it and executes it, there are two important things to note:

1. The URL for the GET operation.

It points to an '**IBM App Connect**' flow which retrieves the list of the Accounts within the "Technology" domain for a specific instance of Salesforce.

The instructor will show live this part. As a documentation you can refer to the following image:



- Once '**IBM App Connect**' successfully returns the list, it passes the result to another function, `__buildCreateOppyScreen`, that we will analyze immediately.

The code in the `__buildCreateOppyScreen` function dynamically builds a form that:

- Displays a dropdown associated to the Accounts retrieved by `__getSFDCAccounts` function
- Collects (via standard HTML `<input>` elements) the information required to create a Salesforce opportunity associated to one of the Accounts
  - Name of the opportunity
  - Description
  - Close Date
  - Amount
  - Probability
- Associates the **Click event** for the **Submit button** to the execution another '**IBM App Connect**' flow (via the function `__sendCreateOppy`)

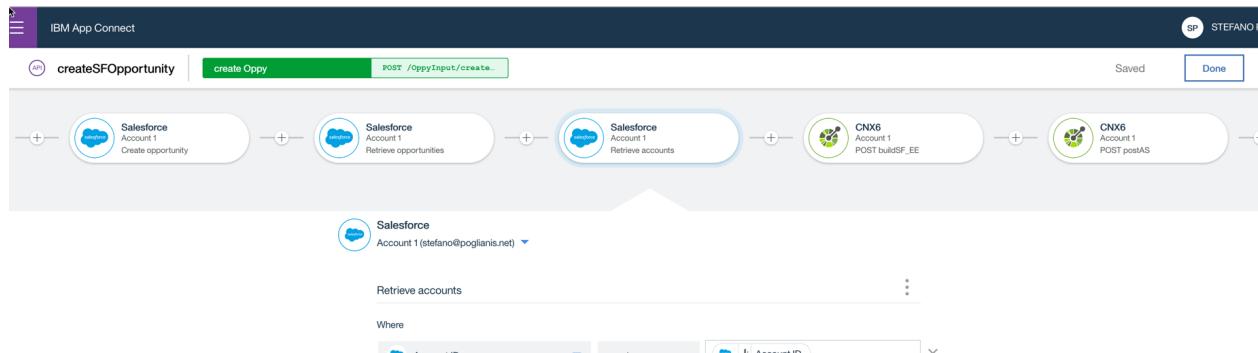
```

var _button1 = document.createElement("input");
_button1.setAttribute('type', 'button');
_button1.setAttribute('name', '_btn1');
_button1.setAttribute('value', 'Submit');
_button1.addEventListener("click", function(){
    // Build the Message
    var theMsg = {};
    // Opportunity INFOS
    theMsg.accountID = _select1.value; //"001r00001laoCN";
    theMsg.name = document.getElementById('_input2').value;
    theMsg.description = document.getElementById('_input3').value;
    theMsg.closeDate = document.getElementById('_input4').value; //"2021-02-01T01:12:41.460Z";
    theMsg.amount = document.getElementById('_input5').value;
    theMsg.probability = document.getElementById('_input6').value;
    // user
    theMsg.userid = window.user.id;
    // Go with HTTP
    __sendCreateOppy('https://service.us.apiconnect.ibmcloud.com/gws/apigateway/api/543aba51314bff5bc7db871ec21796e77ad958439786c0a57a776b6943de6aac/FZS70E/OppyInpu
});
```

1

3. The `__sendCreateOppy` function will simply build an XMLHttpRequest object, invokes a POST operation on the ‘**IBM App Connect**’ flow and displays a result to the user.

The instructor will show live this part. As a documentation you can refer to the following image:

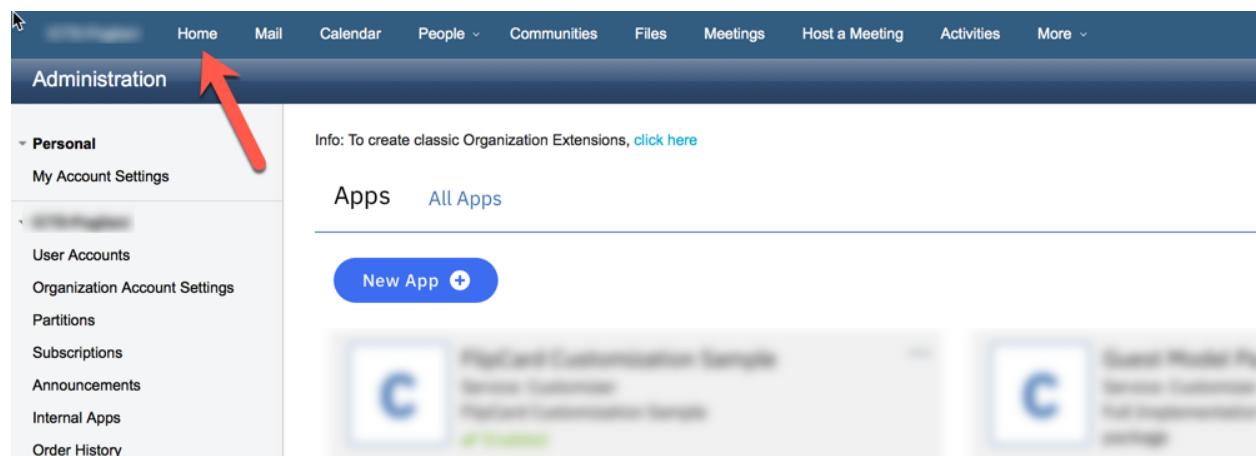


- a. The first node creates the Salesforce opportunity using the information we provided in **IBM Connections Cloud**
- b. The second node retrieves the details of the newly created opportunity
- c. The third node retrieves the information about the Account for which the opportunity has been created
- d. The fourth node prepares the information to be sent to **IBM Connections Cloud**.
- e. The fifth node actually send the information to **IBM Connections Cloud** (we will see this in the next exercise)

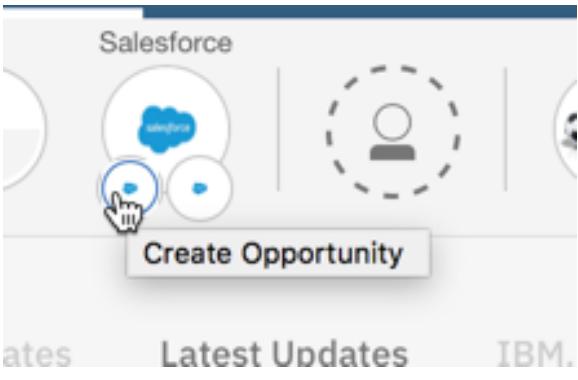
## Verifying

Now it is time to try yourself!

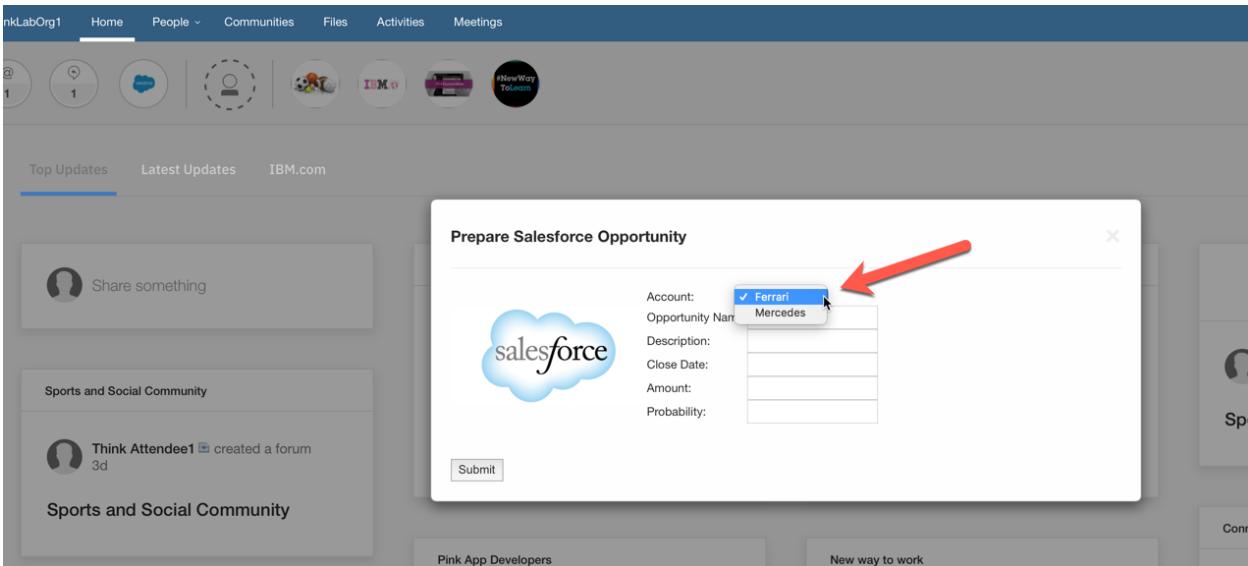
Go back to your OrientMe page by clicking on the **Home** link on the top Menu bar:



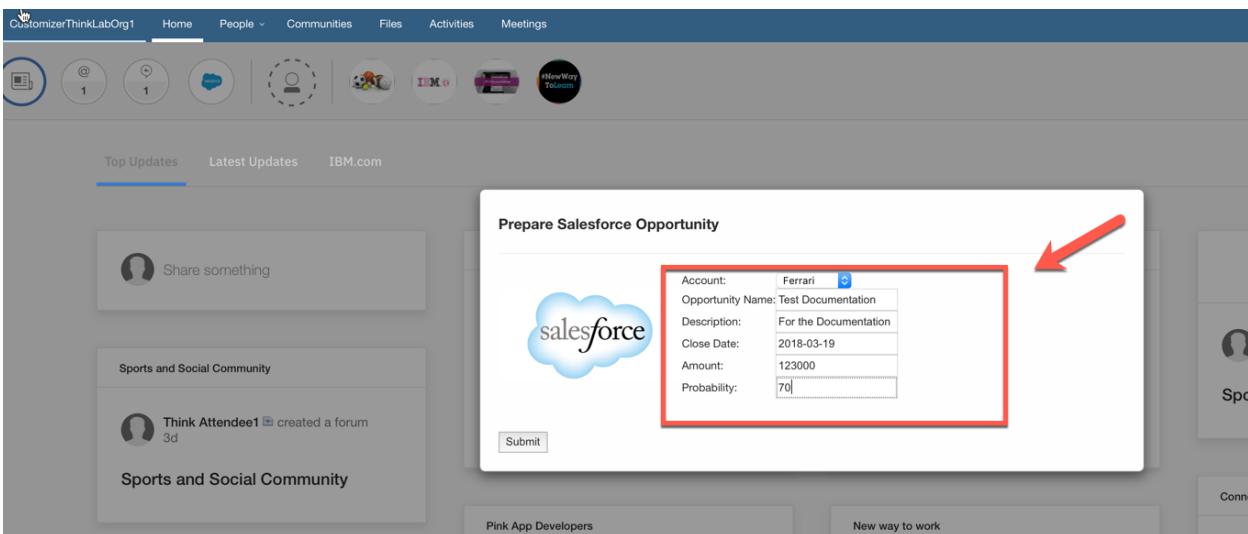
Click on the “Create Opportunity” Action as shown here:



You will see the new form appearing. Note that the dropdown is already filled with the two accounts (Ferrari and Mercedes) available in the Test Salesforce instance.



Choose one of the Accounts and then fill the details for the opportunity as shown here:



## Important

The code which manages the <input> fields in the form has not been developed to be production ready 😊



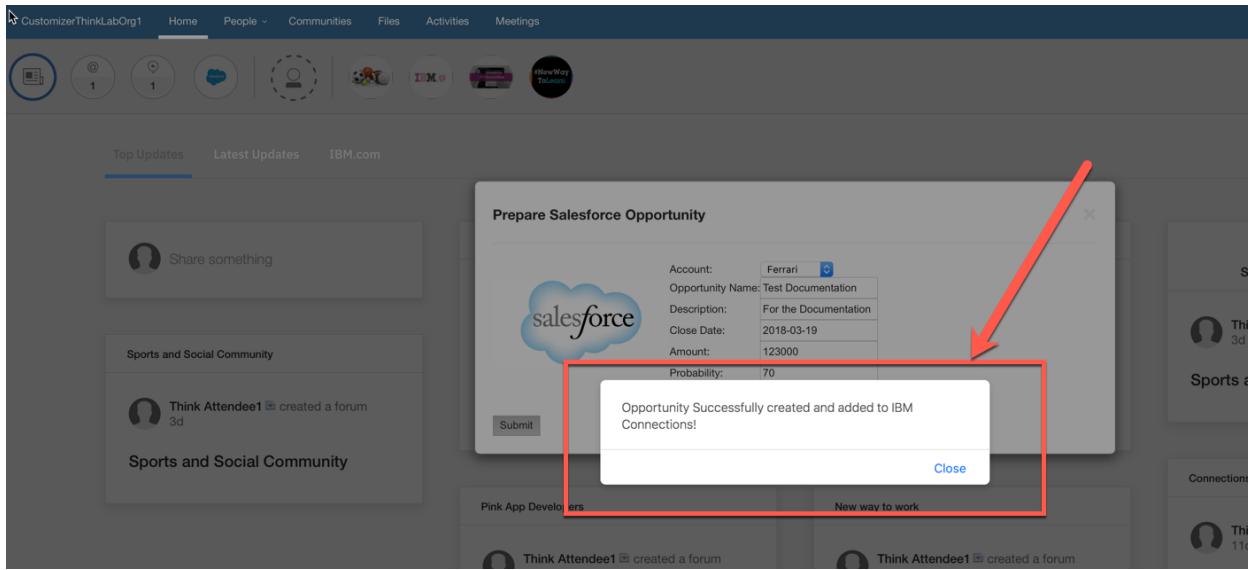
So, be careful entering the information. We really ask that you enter:

- The **Close Date** in the format yyyy-mm-dd
- The **Amount** as a numeric value
- The **Probability** as a positive integer < 100

Click the **Submit button** and wait a couple of seconds:

A screenshot of the IBM App Connect interface. At the top, there's a navigation bar with links like Home, People, Communities, Files, Activities, and Meetings. Below the navigation bar is a toolbar with various icons. The main area shows a feed of updates, including one from 'Think Attendee1' and another from 'Sports and Social Community'. A modal dialog box titled 'Prepare Salesforce Opportunity' is open in the center. It contains a 'salesforce' logo and fields for Account (set to 'Ferrari'), Opportunity Name ('Test Documentation'), Description ('For the Documentation'), Close Date ('2018-03-19'), Amount ('123000'), and Probability ('70'). A red arrow points to the 'Submit' button at the bottom left of the dialog. The background shows some blurred text and icons related to app development and connectivity.

to get the response from '**IBM App Connect**' as shown here:



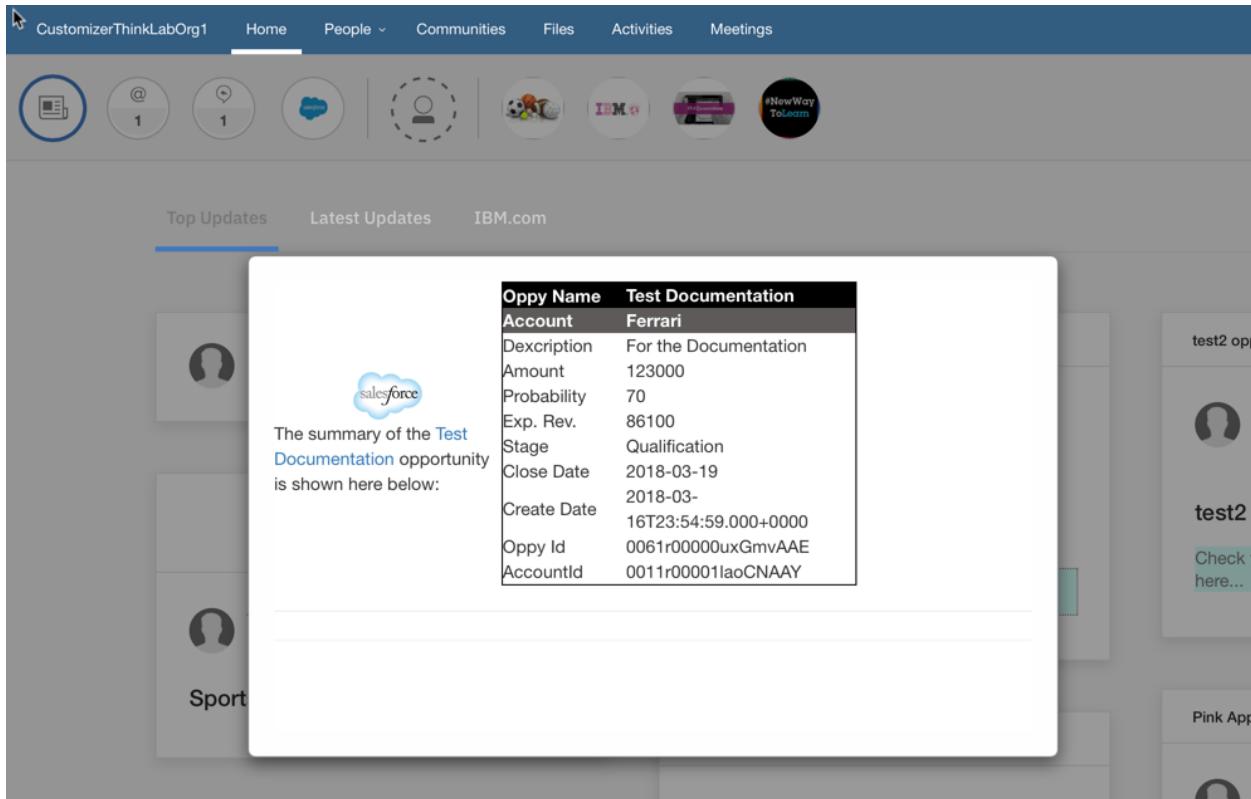
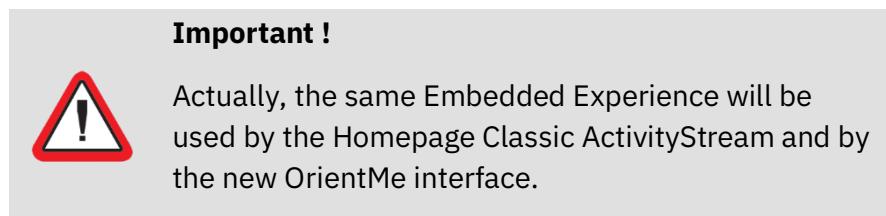
## Lab 4 View the Salesforce Oppy from IBM Connections

In the previous lab we have seen how we created a new Salesforce opportunity from the **IBM Connections Cloud** interface. In this lab, we will show how you can visualize a newly created Salesforce opportunity from within the **IBM Connections Cloud** interface.

### What we want to achieve

Once we finish this fourth lab, we will have enriched the OrientMe page with the ability to display information related to a Salesforce opportunity that was created using **IBM Connections Cloud**.

For the people who are familiar with the concept of an “**Embedded Experience**” in the Homepage Classic ActivityStream, what we achieve in this lab is to provide an “**Embedded Experience**” for OrientMe.

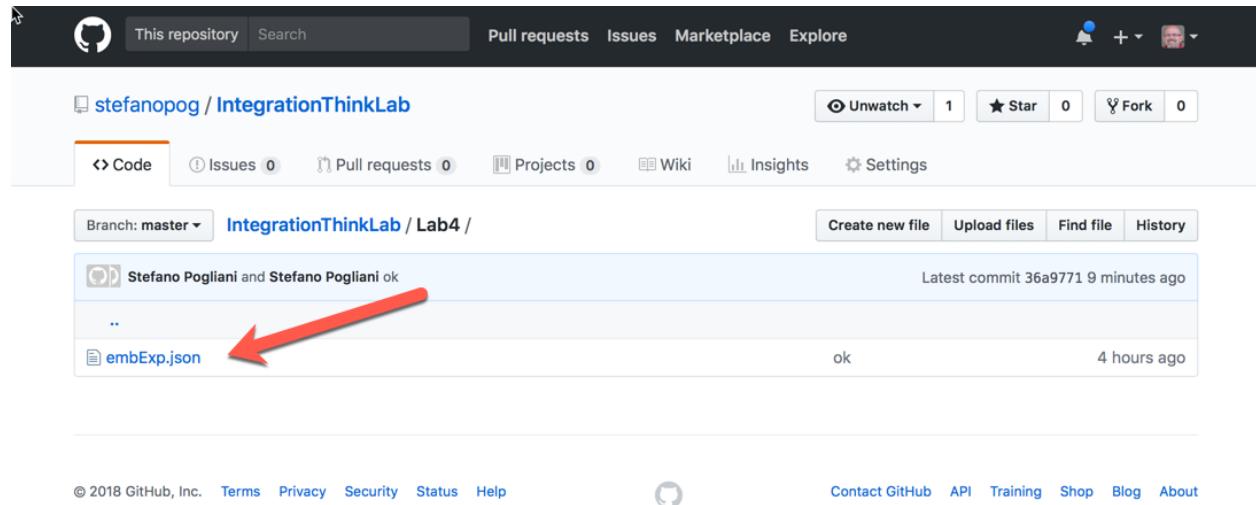


The screenshot shows the IBM Connections homepage. At the top, there is a navigation bar with links for Home, People, Communities, Files, Activities, and Meetings. Below the navigation bar is a toolbar with various icons. The main content area features tabs for Top Updates and Latest Updates, with the Latest Updates tab currently selected. A sidebar on the left lists items like "Salesforce", "Sport", and "Pink App". A central modal window displays a summary of a Salesforce opportunity named "Test Documentation" associated with the account "Ferrari". The modal includes a table with the following data:

Oppy Name	Test Documentation
Account	Ferrari
Description	For the Documentation
Amount	123000
Probability	70
Exp. Rev.	86100
Stage	Qualification
Close Date	2018-03-19
Create Date	2018-03-19T23:54:59.000+0000
Oppy Id	0061r00000uxGmvAAE
AccountId	0011r00001laoCNAAY

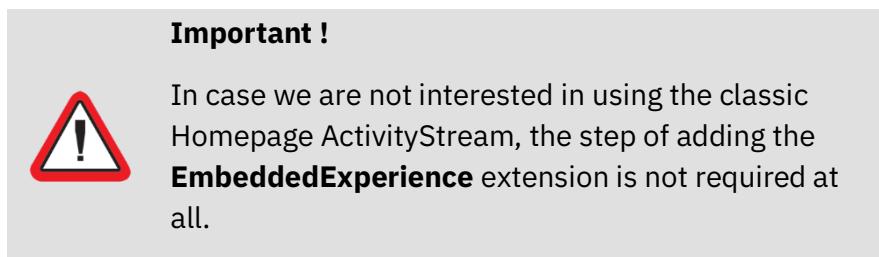
## Preparation Activities

In order to achieve this result, we have to create a new **IBM Connections Cloud** extension. The code of the extension is already prepared and available within the **Reference Repository** under the **Lab4** folder.



The screenshot shows a GitHub repository page for 'stefanopog / IntegrationThinkLab'. The 'Code' tab is selected. A red arrow points to the 'embExp.json' file in the list of files. The file details show it was committed by Stefano Pogliani and Stefano Pogliani at 36a9771 9 minutes ago. The status is 'ok' and it was last updated 4 hours ago. The page includes standard GitHub navigation and footer links.

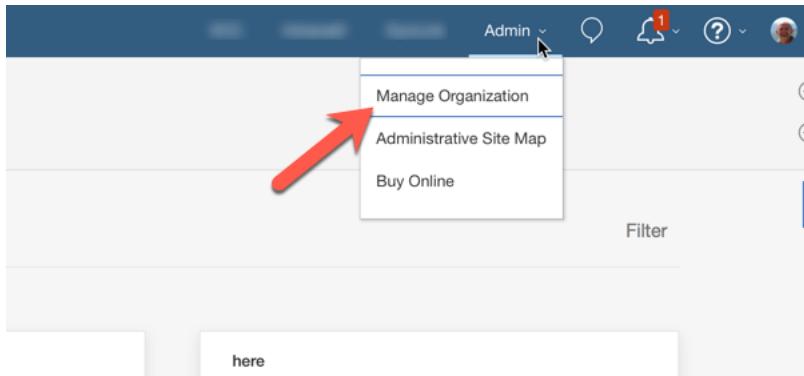
This is the standard way to declare an Embedded Experience in **IBM Connections Cloud**: we tell the system to safely associate an HTML file and an icon file to a given topic.



## Environment SetUp

Open a new window on your browser and log into IBM Connections <https://apps.na.collabserv.com> providing the credentials the instructors gave you.

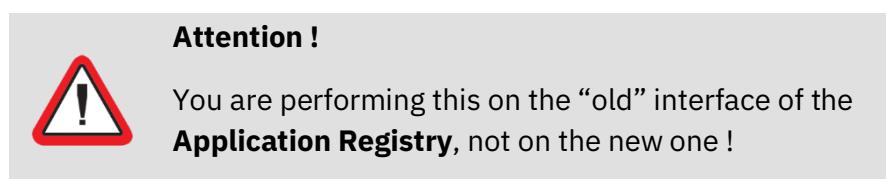
Navigate to the Administration screen:



and, once there, go to the “Organization Extensions” panel:

A screenshot of the 'User Accounts' page under the 'Administration' section. The left sidebar has a 'User Accounts' section with a blue header, containing links like 'Organization Account Settings', 'Partitions', 'Subscriptions', etc. A red arrow points to the 'Organization Extensions' link. The main content area shows a table of user accounts with columns for Name and Email. The names listed are Amadou Alain, Charles Bounar, Christina Milan, Ed Blanks, Ed El-Amon, Frank Adams, and Gail Chao. Each name has a dropdown arrow next to it.

Click on the blue “New App” button:



The screenshot shows the Salesforce Administration interface. On the left, there's a sidebar with sections like Personal, CustomizerThinkLabOrg1 (which is expanded), and System Settings. The main area is titled 'Applications For CustomizerThinkLabOrg1' and contains an 'Info' message about the new Apps Manager. Below it are buttons for Add, Export, Delete, and Filter by Service (set to All). A red arrow points to the 'Add' button. The word 'Preferences' is also visible below the toolbar.

and then, check the “Manually install an extension” checkbox as shown here and fill the fields with the following values:

- Service : choose “Activity Stream” from the dropdown
- Extension Point: choose “Activity Stream Event” from the dropdown
- Extension Point : “SFDC”
- Description: “Salesforce Demo”
- Icon URL:  
<https://apps.na.collabserv.com/files/customizer/salesforce/salesforce.png?repoName=think-samples>
- URL :  
<https://apps.na.collabserv.com/files/customizer/salesforce/salesforce.html?repoName=think-samples>

#### Important !



Note, once again, the use of the  
?repoName=think-samples suffix for both the URL  
and the Icon URL parameters.

CustomizerThinkLabOrg1 Home People Communities Files Activities Meetings

Administration

Personal My Account Settings

CustomizerThinkLabOrg1 User Accounts Organization Account Settings Partitions Subscriptions Announcements Internal Apps Order History

**Organization Extensions**

Connections Mobile App Management Chat and Meetings Apps

System Settings Security Theme

Extensions For CustomizerThinkLabOrg1. Edit Extension for CustomizerThinkLabOrg1

**Edit Extension for CustomizerThinkLabOrg1**

**Extension Details**

Upload an extension from a JSON file  no file selected

Manually install an extension

Service:

Extension Point:

Name:

Description:

Icon URL:

URL:

Open in a new Window

The new “*Registry Application*” is now declared!

CustomizerThinkLabOrg1 Home People Communities Files Activities Meetings

Administration

Personal My Account Settings

CustomizerThinkLabOrg1 User Accounts Organization Account Settings Partitions Subscriptions Announcements Internal Apps Order History

**Organization Extensions**

Connections Mobile App Management Chat and Meetings Apps

System Settings Security Theme

Applications For CustomizerThinkLabOrg1

Info: To create Connections Pink applications, view the [new Apps Manager](#)

Filter by Service: All

**Preferences**

	<b>SFDC</b>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Service:	<b>Activity Stream</b>	
Salesforce Demo		
Type:	<b>Extension</b>	
URL:	<a href="https://apps.na.collabServ.com/files...">https://apps.na.collabServ.com/files...</a>	
Icon URL:	<a href="https://apps.na.collabServ.com/files...">https://apps.na.collabServ.com/files...</a>	

## Verifying

Do you remember that, when we explained the "[Lab 3: Create a Salesforce Oppy from IBM Connections](#)", we stated that a feedback about the newly created opportunity is sent back to **IBM Connections Cloud**?

If you refresh the OrientMe page, you will likely see something similar to this picture:

The screenshot shows the OrientMe application interface. At the top, there is a navigation bar with links for 'CustomizerThinkLabOrg1', 'Home', 'People', 'Communities', 'Files', 'Activities', and 'Meetings'. Below the navigation bar is a toolbar with various icons: a blue circle with a document icon, a white circle with '@ 1', a white circle with a downward arrow icon, a white circle with a cloud icon, a dashed circle with a person icon, a white circle with a soccer ball icon, a white circle with an IBM logo, and a black circle with '#NewWay ToLearn'. Underneath the toolbar, there are three tabs: 'Top Updates' (which is selected), 'Latest Updates', and 'IBM.com'. The main content area displays a feed of updates. One update is highlighted with a red border and a red arrow points to it. The highlighted update is titled 'Test Documentation opportunity' and contains the text: 'This is the Salesforce opportunity document accessible Test Documentation opportunity 4h'. Below this, another update is visible with the title 'Test Documentation opportunity' and a summary: 'Check the created Test Documentation opportunity by clicking here...'. Other updates in the feed include: 'Sports and Social Community' (1 / 3), 'Think Attendee1 created a wiki 3d', 'Sports and Social Community', 'Connections Customizer Playbacks', and 'Think Attendee1 created a forum 3d'.

An event related to the Opportunity we created in the previous lab has been delivered to **IBM Connections Cloud** and it appears on the OrientMe page.

In addition, the part of the tile that contains the "**summary of the event**", has been highlighted in light blue color.

If you click now on the light-blue highlighted summary, you will see this:

The screenshot shows the Salesforce homepage with a modal window open. The modal contains a summary of an opportunity named "Test Documentation" with account "Ferrari". The details listed are:

Oppy Name	Test Documentation
Account	Ferrari
Description	For the Documentation
Amount	123000
Probability	70
Exp. Rev.	86100
Stage	Qualification
Close Date	2018-03-19
Create Date	2018-03-16T23:54:59.000+0000
Oppy Id	0061r00000uxGmvAAE
AccountId	0011r00001laoCNAAY

The background of the homepage is greyed out, and there are other cards visible like "Sport" and "Pink App".

Now, click on the top part of the grey background for the modal window in order to make it disappear and, then, select to see the Classical ActivityStream-based Homepage by selecting the “Use Classic Home View” button as shown here:

The screenshot shows the Salesforce user profile menu. A red arrow points to the "Use Classic Home View" option, which is highlighted with a red box. Other options in the menu include "My Profile", "Notification Settings", "Account Settings", "Downloads and Setup", "Invite Guest", and "Log Out".

You will find your ActivityStream which looks like this:

Share Something: update your status or upload a file.

I'm Following | Status Updates | Discover

This is the Salesforce opportunity document accessible [Test Documentation opportunity](#)  
Please Review it  
Check the created Test Documentation opportunity by clicking here...

Save this • Stop Following 4h ago

This is the Salesforce opportunity document accessible [test2 opportunity](#)  
Please Review it  
Check the created test2 opportunity by clicking here...

Save this • Stop Following Yesterday

Meetings

Enter meeting ID

Events

You are not following any upcoming events.

Add to Personal Calendar

Action Required

Saved

Mentions 1

My Notifications 1

CustomizerThinkLabOrg1 Home People Communities Files Activities Meetings

Clicking anywhere on the first item, you will see the EmbeddedExperience associated with the event appearing:

Share Something: update your status or upload a file.

I'm Following | Status Updates | Discover

This is the Salesforce opportunity document accessible [Test Documentation opportunity](#)  
Please Review it  
Check the created Test Documentation opportunity by clicking here...

Save this • Stop Following 4h ago

This is the Salesforce opportunity document accessible [test2 opportunity](#)  
Please Review it  
Check the created test2 opportunity by clicking here...

Save this • Stop Following Yesterday

The summary of the **Test Documentation** opportunity is shown here below:

Copy Name	Test Documentation
Account	Ferrari
Description	For the Documentation
Amount	123000
Probability	70
Exp. Rev.	86100
Stage	Qualification
Close Date	2018-03-19
Create Date	2018-03-19T23:54:59.000+0000
Copy Id	0061r000000uGmvaAE
AccountId	0011r00001la0CNAAY

Meetings

Enter meeting ID

Events

You are not following or attending any upcoming events.

Add to Personal Calendar

Action Required

Saved

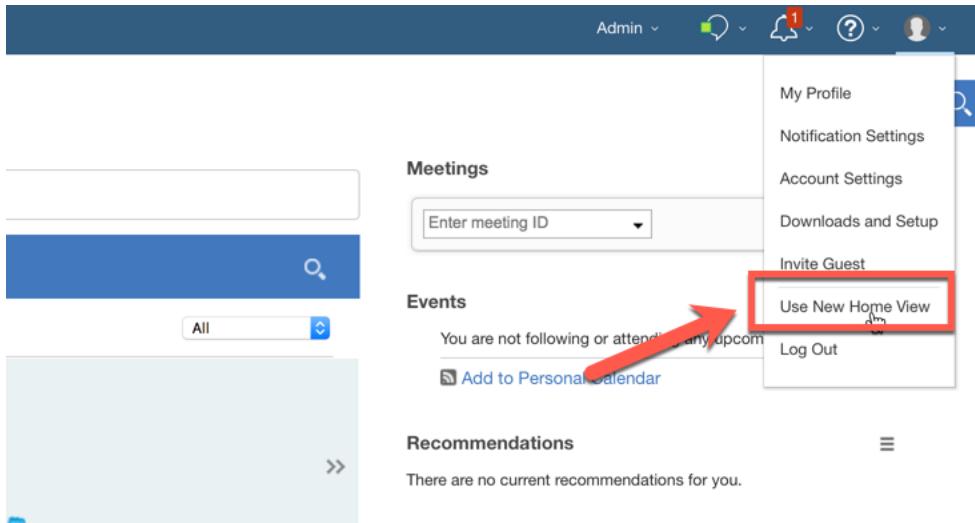
Mentions 1

My Notifications 1

CustomizerThinkLabOrg1 Home People Communities Files Activities Meetings

And, wow, the EmbeddedExperience is the same as the one shown on the OrientMe page!

Go back to the OrientMe page, now, clicking here:



And let's see how this was made.

1. An event was sent back to IBM Connections Cloud after the opportunity creation.  
This event contains several information, including:
  - a. The information about the created opportunity.  
This information is stored in the “**context**” attribute of the event.
  - b. The link to the HTML file which displays the EmebeddedExperience (in our case this is :  
<https://apps.na.collabserv.com/files/customizer/salesforce.html?repoName=think-samples>

**Important !**

You can see the code form that file under the resources folder of the Reference Repository

```

stefanopog / IntegrationThinkLab
Branch: master - IntegrationThinkLab / resources /
Stefano Poglian and Stefano Poglian ok
...
svc_scripts
svc_styles
salesforce.html
salesforce.png
salesforce_big.png

```

- c. Did you notice, once again, the “?repoName=think-samples” suffix ?

- Now that we have all the information we want to show and the HTML file which displays them, we need to retrieve them.

When we click on the “light-blue summary” in the tile, we programmatically get the identifier of the event underlying the tile:

```

1 function _captureEvent (target) {
2   if (target.id.startsWith('ic-tile:urn:lsid:lconn.ibm.com:activitystreams.story:')) {
3     //
4     // Find the summary and highlight it
5     //
6     var myChild = dojo.query('.ic-summary.ic-summary-article', target)[0];
7     myChild.style['background-color'] = 'rgb(211, 255, 248)';
8     //
9     // Add eventlistener to the Child to simulate EmbeddedExperience
10    //
11    myChild.firstChild.addEventListener('click', function (e) {
12      //
13      // Stop ballooning of events
14      //
15      e.stopPropagation();
16      //
17      // Retrieve the story
18      //
19      var __tmp = target.id.split(':');
20      var _story = __tmp[__tmp.length-1];
21      //
22      // Now retrieve the story from the ActivityStream
23      //
24      var xmlhttpReq = false;
25      if (window.XMLHttpRequest) {
26        // Mozilla/Safari
27        xmlhttpReq = new XMLHttpRequest();
28      } else if (window.ActiveXObject) {
29        // IE
30        xmlhttpReq = new ActiveXObject("Microsoft.XMLHTTP");
31      }
32      xmlhttpReq.open('GET', 'https://apps.na.collaberv.com/connections/opensocial/basic/rest/activitystreams@me@all@all/' + _story, true);
33      xmlhttpReq.setRequestHeader('Content-Type', 'application/json');
34      xmlhttpReq.onreadystatechange = function() {
35        if (xmlhttpReq.readyState == 4) {
36          if ((xmlhttpReq.status == 200)) {
37            //
38            // We got the answer
39          }
40        }
41      }
42    })
43  }
44}

```



Using this identifier, we perform an API call against IBM Connections Cloud to retrieve the content of that event; the content includes the “context” and the URL for the EmbeddedExperience !

```

1  function _captureEvent (target) {
2    if (target.id.startsWith('ic-title:urn:lsid:lconn.ibm.com:activitystreams.story')) {
3      //
4      // Find the summary and highlight it
5      //
6      var myChild = dojo.query('.ic-summary.ic-summary-article', target)[0];
7      myChild.style['background-color'] = 'rgb(211, 255, 248)';
8      //
9      // Add eventListener to the Child to simulate EmbeddedExperience
10     //
11     myChild.firstChild.addEventListener('click', function (e) {
12       //
13       // Stop ballooning of events
14       //
15       e.stopImmediatePropagation();
16       //
17       // Retrieve the story
18       //
19       var __tmp = target.id.split(':');
20       var _story = __tmp[__tmp.length-1];
21       //
22       // Now retrieve the story from the ActivityStream
23       //
24       var xmlhttpReq = false;
25       if (window.XMLHttpRequest) {
26         // Mozilla/Safari
27         xmlhttpReq = new XMLHttpRequest();
28       } else if (window.ActiveXObject) {
29         // IE
30         xmlhttpReq = new ActiveXObject("Microsoft.XMLHTTP");
31       }
32       xmlhttpReq.open('GET', 'https://apps.na.collabser.com/connections/opensocial/basic/rest/activitystreams/@me/all@all/' + _story, true);
33       xmlhttpReq.setRequestHeader('Content-Type', 'application/json');
34       xmlhttpReq.onreadystatechange = function() {
35         if (xmlhttpReq.readyState == 4) {
36           if ((xmlhttpReq.status == 200)) {
37             //
38             // We got the answer
39           }
40         }
41       }
42     });
43   }
44 }

```



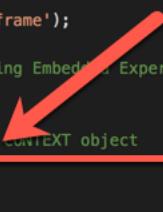
- Once we get the URL for the EmebeddedExperience, we display it as a modal iFrame :

```

1  var __tmp2 = JSON.parse(xmlhttpReq.responseText);
2  var _context = __tmp2.entry.openSocial.embed.context;
3  //
4  // create the iFrame to load the salesforce.html page
5  //
6  var _iframe = document.createElement('iframe');
7  _iframe.classList.add('modal-content');
8  _iframe.src = _context.url; // Retrieving Embedded Experience from CONTEXT Object
9  _iframe.style.height="400px";
10 //
11 // When the iFrame will load, pass the CONTEXT object
12 //
13 _iframe.onload = function () {
14   let newCtx = {};
15   newCtx.source = {};
16   newCtx.source.resourceContext = _context;
17   _iframe.contentWindow.postMessage(JSON.stringify(newCtx), '*');
18 };
19 //
20 // ...
21 
```



and, then, we deliver to the loaded modal iFrame the previously retrieved “context” (which contains all the details for the newly created opportunity) in the same way as **IBM Connections Cloud** does with the classic EmbeddedExperience



```

    var __tmp2 = JSON.parse(xmlHttpReq.responseText);
    var _context = __tmp2.entry.openSocial.embed.context;
    //
    // create the iFrame to load the salesforce.html page
    //
    var _iframe = document.createElement('iframe');
    _iframe.classList.add('modal-content');
    _iframe.src = _context.url; // Retrieving Embed's Experience from CONTEXT Object
    _iframe.style.height="400px";
    //
    // When the iFrame will load, pass the CONTEXT object
    //
    _iframe.onload = function () {
        let newCtx = {};
        newCtx.source = {};
        newCtx.resourceContext = _context;
        _iframe.contentWindow.postMessage(JSON.stringify(newCtx), '*');
    };
    //

```

- The only thing we are missing is how did we highlight the summary of the tile in light-blue and how did we make it clickable.

If you look at the end of the **mySalesforce.js** script, you will find the following code:



```

if (typeof (dojo) != "undefined") {
    require(["dojo/domReady!"], function () {
        //console.log('sono qui 2');
        var aaa = dojo.query('article.ic-stack-tile.ic-tile-sfdc_oppy.ic-is-post');
        for (let _i = 0; _i < aaa.length; _i++) {
            _captureEvent(aaa[_i]);
        }
        require(
            ['dojo/on', 'dojo/_base/window', 'dojo/query'],
            //
            // We want to capture when the container of the user records (for Likes, Comments, Downloads...)
            // actually gets filled with the HTML elements that contain the information for the users
            // who are liking, commenting, downloading...
            //
            function (on, win) {
                //console.log('sono qui 3');
                on(win.doc,
                    'article.ic-tile.ic-tile-sfdc_oppy.ic-is-post:DOMNodeInserted',
                    function(evt) {_captureEvent(evt.target);})
            });
    */
}

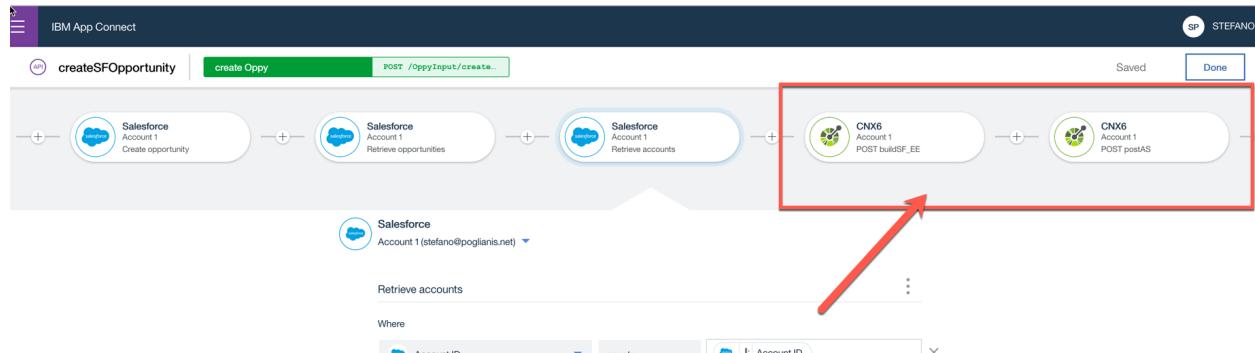
```

This code is executed as soon as the **mySalesforce.js** script is loaded by the OrientMe page. It retrieves the tiles on the first tab of the OrientMe page and it subscribes for the creation of the relevant tiles in the second tab.

This closes the circle 😊

## Appendix Using IBM Cloud Functions

During the explanation of “[Lab3 Create a Salesforce Oppy from IBM Connections](#)” we have seen an IBM App Connect flow that performs processing on Salesforce and also on IBM Connections Cloud.



How did we implement those actions?

### The importance of “serverless”

The instructor will walk through the use of **IBM Cloud Functions** (see <https://www.ibm.com/cloud/functions>) and of the new **IBM Cloud Functions Shell** (<https://www.ibm.com/blogs/bluemix/2017/10/serverless-composition-ibm-cloud-functions/>).

Roughly, we developed a set of “functions” that wrap the standard IBM Connections Cloud APIs. This allows us to simplify the use of those APIs.

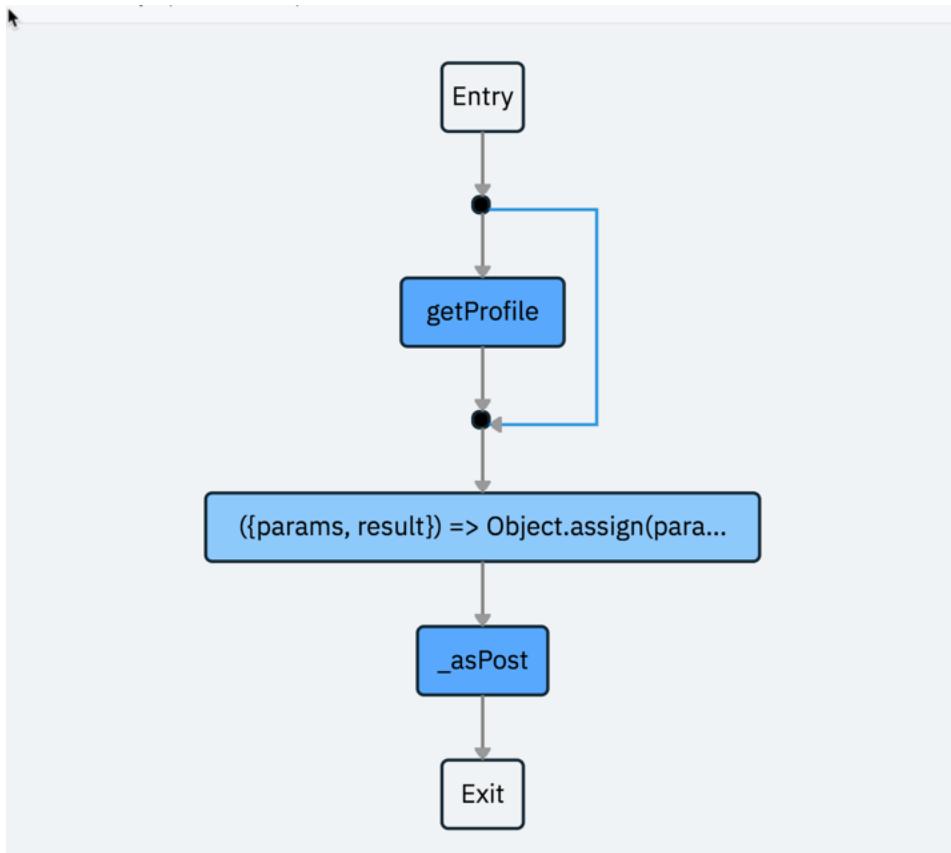
The added value of using **IBM Cloud Functions** (which is an implementation of the **Apache OpenWhisk** project, see <https://openwhisk.apache.org/>) is the scalability and manageability of the “functions” we created.

In a nutshell:

- We created “atomic functions” using NodeJS. An example of such an “atomic function” are the **\_getProfile** or the **\_asPost** functions:

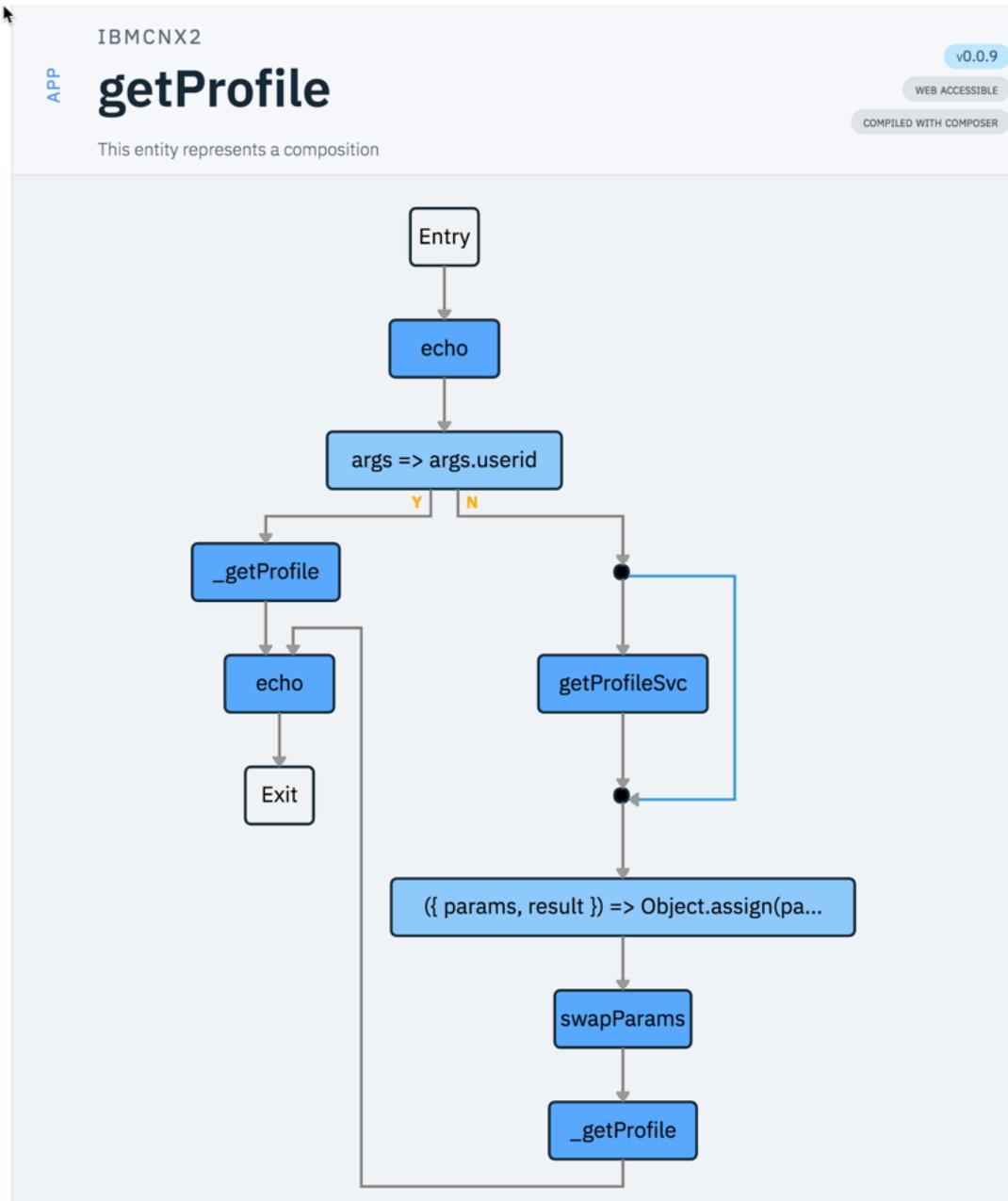
```
return new Promise(function (resolve, reject) {
    request(options, function (error, response, body) {
        if (error) {
            console.log('_getProfile: error during request : -> ' + JSON.stringify(error, ' ', 2));
            reject(error);
        } else {
            if (response.statusCode != 200) {
                console.log('_getProfile: error in response : -> ' + JSON.stringify(response, ' ', 2));
                reject({
                    code: response.statusCode,
                    msg: response.statusMessage
                });
            } else {
                //
                // Find the info
                //
                console.log('_getProfile: parsing...');
                parser.parseString(body, function (err, result) {
                    if (err) {
                        console.log("_getProfile : Parser Error : " + JSON.stringify(err, ' ', 2));
                        reject(err);
                    }
                    if (result.feed.entry) {
                        //
                        // We set the "displayName" property of the credentials
                        //
                        //resolve({msg: 'ok'});
                        try {
                            var p = _getUserDetail(result.feed.entry[0]);
                            console.log('_getProfile : succesfull return : -> ' + JSON.stringify(p, ' ', 2));
                            resolve(p);
                        } catch (err) {
                            reject(err);
                        }
                    }
                });
            }
        }
    });
});
```

- We, then, created higher-level functions using the IBM Cloud Functions Shell. Here is an example of the **asPost** composition:



Since Posting to the IBM Connections ActivityStream requires the ID of the user (which is not always easy to get), we defined this composition which retrieves, first, the ID of the user from her mail address (the “**getProfile**” step).

Likewise the **getProfile** function is a composition:



In In

this composition, if the input is missing it is assumed that the user is asking information about herself, so the **getProfileSvc** and **\_getProfile** actions are called in sequence.

- We are, then, exposing some of the “atomic functions” and the “compositions” as a **SWAGGER** document:

## IBM Connections Cloud APIs

topolino

Version v1

[Find out more about Swagger](#)

### Security

✓ basicAuth (HTTP Basic Authentication)

HTTP Basic Authentication. Works over [HTTP](#) and [HTTPS](#)

### Paths

/getProfile

[GET /getProfile](#)

/getMyProfile

[GET /getMyProfile](#)

/getProfileSvc

[GET /getProfileSvc](#)

/buildSF\_EE

[POST /buildSF\\_EE](#)

/getAS

[GET /getAS](#)

/postAS

[POST /postAS](#)

which, incidentally, allows easy testing of those APIs.

- Finally, we are importing the SWAGGER Document into **IBM App Connect** (see <https://www.ibm.com/cloud/app-connect>) where we develop higher level compositions.

#### Note



The Swagger document is available as `IBMConnectionsCloud.yaml` under the **Resources** folder of the **Reference Repository**

- You can try yourself the SWAGGER document using the free Swagger Inspector tool available here <https://inspector.swagger.io/builder> (better with Chrome)

The screenshot shows the Swagger Inspector interface. At the top, there's a navigation bar with links like 'Welcome to the Glob...', 'Travel @IBM', 'Learning', etc. Below it is the 'SWAGGER inspector' logo from SMARTBEAR. A search bar contains the URL 'http://service.us.ibmcloud.com/gws/apigateway/api/78ee16ab2e0832faf42a8464fc5a4facb91e87a33f3acddd5d9c74fc4a448c0/cnx'. To the right are buttons for 'SEND' and 'USE DEFINITION'. The main area is divided into 'Request' and 'Response' sections. Under 'Request', there's a 'Parameters (1)' tab with a 'userid' entry and a 'value' field. An 'ADD NEW PARAMETER' button is below it. Under 'Response', there's a 'PRETTY' button and a placeholder text 'Send a request to see the response'. On the right side, a sidebar lists several API endpoints:

Method	Endpoint
POST	/buildSF_EE
GET	/getAS
GET	/getMyProfile
GET	/getProfile
GET	/getProfileSvc
POST	/postAS

At the bottom right, there are two small checkboxes with dates: '31 Jan 2018 02:57' and '31 Jan 2018 02:56'.

- The IBM Cloud Functions code will soon be made available at this address:  
<https://github.com/stefanopog/IBMCnxCF>

## We Value Your Feedback!

- Don't forget to submit your Think 2018 session and speaker feedback! Your feedback is very important to us – we use it to continually improve the conference.
- Access the Think 2018 agenda tool to quickly submit your surveys from your smartphone, laptop or conference kiosk.