Informatica Cloud Application Integration

Git Service Connector Package

October 2018

Contents

Copyright:

Overview:

Git is one of the leading source code management (SCM) solutions available today. The Git Service Connector package enables Informatica Cloud Application Integration (CAI) developers to easily integrate Git functionality into their CAI projects.

Prerequisites:

- Git access and credentials
- Appropriate account and role/permissions on Informatica Intelligent Cloud Service to perform an IMPORT operation

Use Case #1 – Export IICS Assets into Git

A developer can specify which objects in their IICS workspace are to be exported from IICS using the IICS REST API and imported into Git using the Git Contents API.

To export something from IICS into Git, the sample process “SCM Export to Git” can be invoked with the appropriate IICS credentials, along with information specifying the asset to be loaded into the Git repository. Git access is defined in the Git-Contents-API-actions Connection.

Use Case #2 - Import Git Assets into IICS

A developer can specify which objects in their Git repository are to be exported using the Git Contents API and imported into their CAI workspace using the IICS REST API.

To import something from Git into IICS, the sample process “SCM Import from Git” can be invoked with the appropriate IICS credentials and the filename of the object in your Git repository. Git access is defined in the Git-Contents-API-actions Connection.

Quick Setup

1. Download the GitServiceConnectorPackage1018.zip from the Informatica CAI GitHub Repository.
2. Import the package into your CAI workspace
3. This creates a new project named “IICS REST API and Git” with four assets at the root level of the project.
    a. A folder named “Connections” with two Connections in it
    b. A folder named “Service Connectors” with two Service Connectors in it
    c. A process named “SCM Export to Git”
    d. A process named “SCM Import from Git”
4. Publish the “IICS REST API” Service Connector
5. Publish the “Git Contents API” Service Connector
6. Publish the IICS-REST-API-Actions Connection
7. Edit the Git-Contents-API-Actions, supplying the necessary Git information.
    a. Username
    b. Password
    c. BaseURL
    d. Owner
    e. Repo
    f. Branch
8. Save the Connection.
9. Publish the Connection.
10. Publish the “SCM Export to Git” Process
11. Publish the “SCM Import from Git” Process
12. Test **Import from Git** using Postman
    a. Define a new Request
    b. Configure the Request as a POST.
    c. Set the Body to “raw” and “JSON (application/json)”
    d. Get the Service URL from the Properties Detail of the “SCM Import from Git” process
    e. Set the Request URL in your Postman request to the Service URL obtained previously
    f. Configure your input – importing a project named “DemoProject” that is located at the root level of the “demo” repository in Git. The Git details are defined in the Git-Contents-API-Actions connection.

```
{
  "IICS Hostname": "dm-us.informaticacloud.com",
  "IICS Username": "<your username>",
  "IICS Password": "<your password>",
  "SCM File Path": "DemoProject-18:50:14.989Z.zip"
}
```

    g. Execute the Request
        i. The response should look similar to:

```
{
 "Login Status": "Active",
 "SCM Checkout Status": "DemoProject-18:50:14.989Z.zip",
```

```
 "Import Status": "SUCCESSFUL"
}
```

ii. Check the CAI Project View to see that the DemoProject has been created.

13. Test **Export to Git** using Postman
    a. Define a new Request
    b. Configure the Request as a POST.
    c. Set the Body to “raw” and “JSON (application/json)”.
    d. Get the Service URL from the Properties Detail of the “SCM Export to Git” process.
    e. Set the Request URL in your Postman request to the Service URL obtained previously.
    f. Configure your input – Specifying an existing asset named “FMLab” that is of type “Project” that is to be copied to the root level of the “demo” repository. The Git details are defined in the Git-Contents-API-Actions connection.

```
{
"Enable Source Control": "Yes",
"IICS Hostname": "dm-us.informaticacloud.com",
 "IICS Username": "<your username>",
 "IICS Password": "<your password>",
 "Object Path": "FMLab",
"Object Type": "PROJECT"
}
```

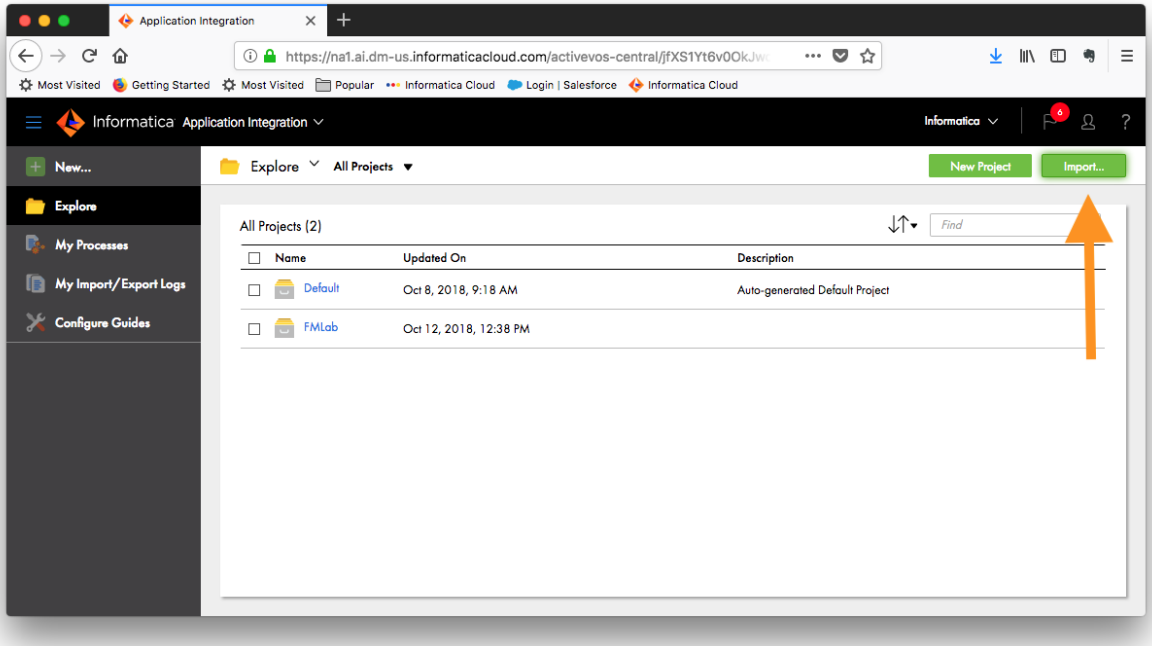
    g. Execute the Request
        i. The response should look like:

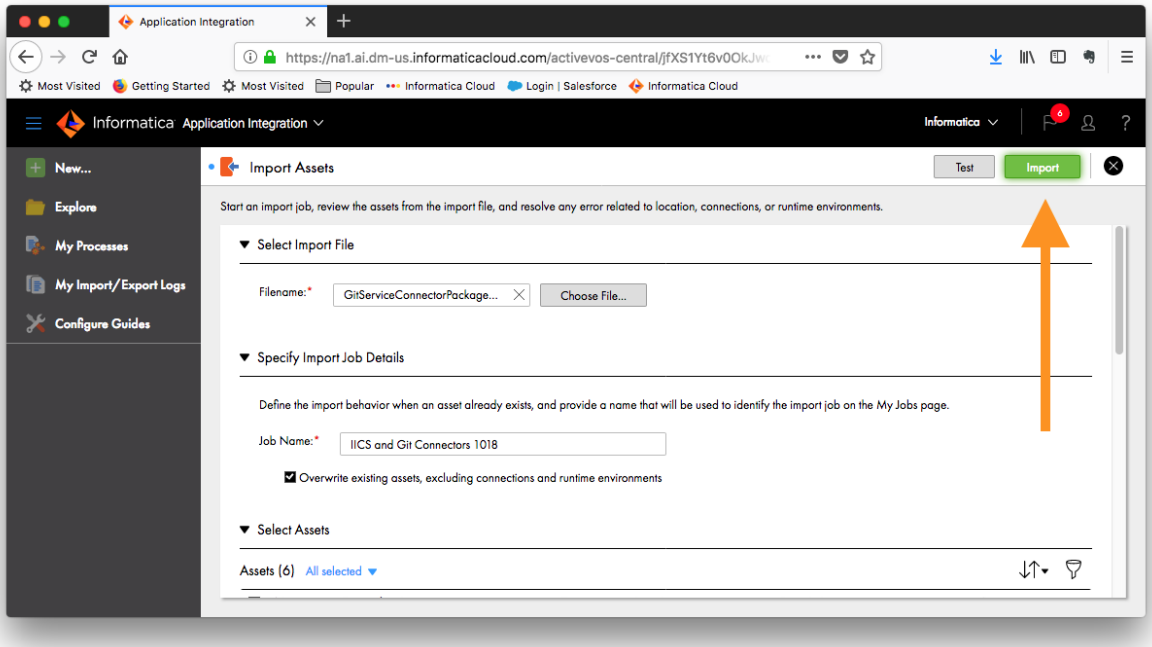
```
{
 "Login Status": "Active",
 "Export Status": "SUCCESSFUL",
 "SCM Checkin Status":
"https://api.github.com/repos/sagarwal1980/demo/git/trees/1
a80939c26805e15d78924ef4447a2d7208ad543"
}
```

        ii. Check your Git repository to see that an export of the specified asset has been created and uploaded.

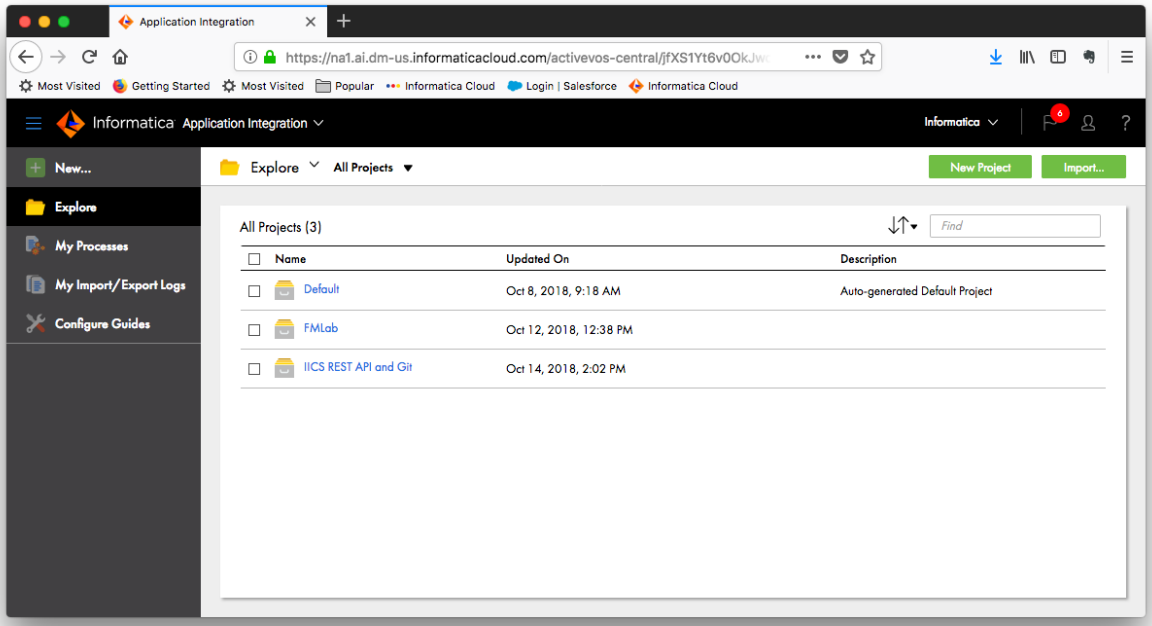
Detailed Setup:

1. Download the GitServiceConnectorPackage1018.zip from the Informatica CAI GitHub Repository.
2. Import the package into your CAI workspace.

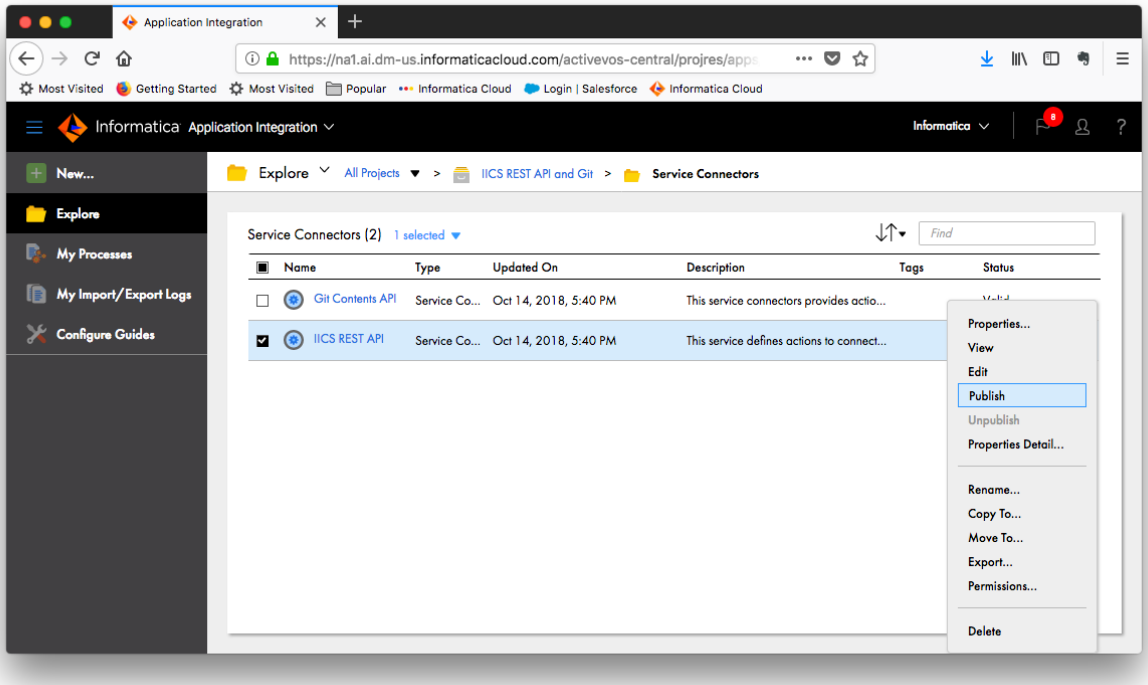




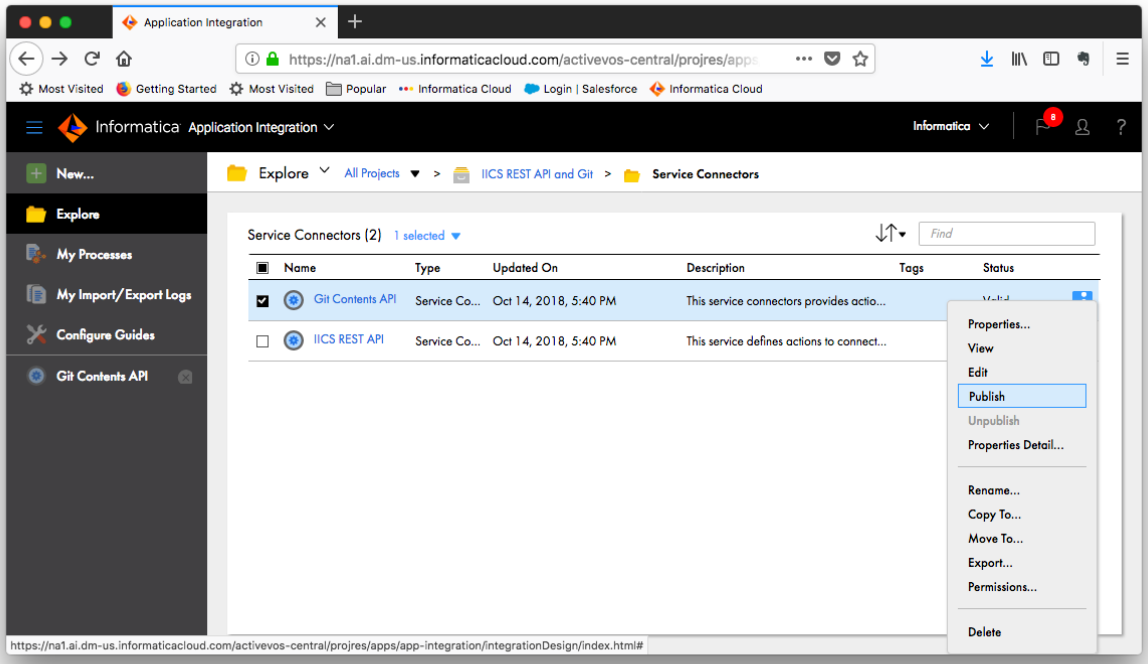
3. This will create a new project called “IICS REST API and Git” at the root level of your CAI environment.



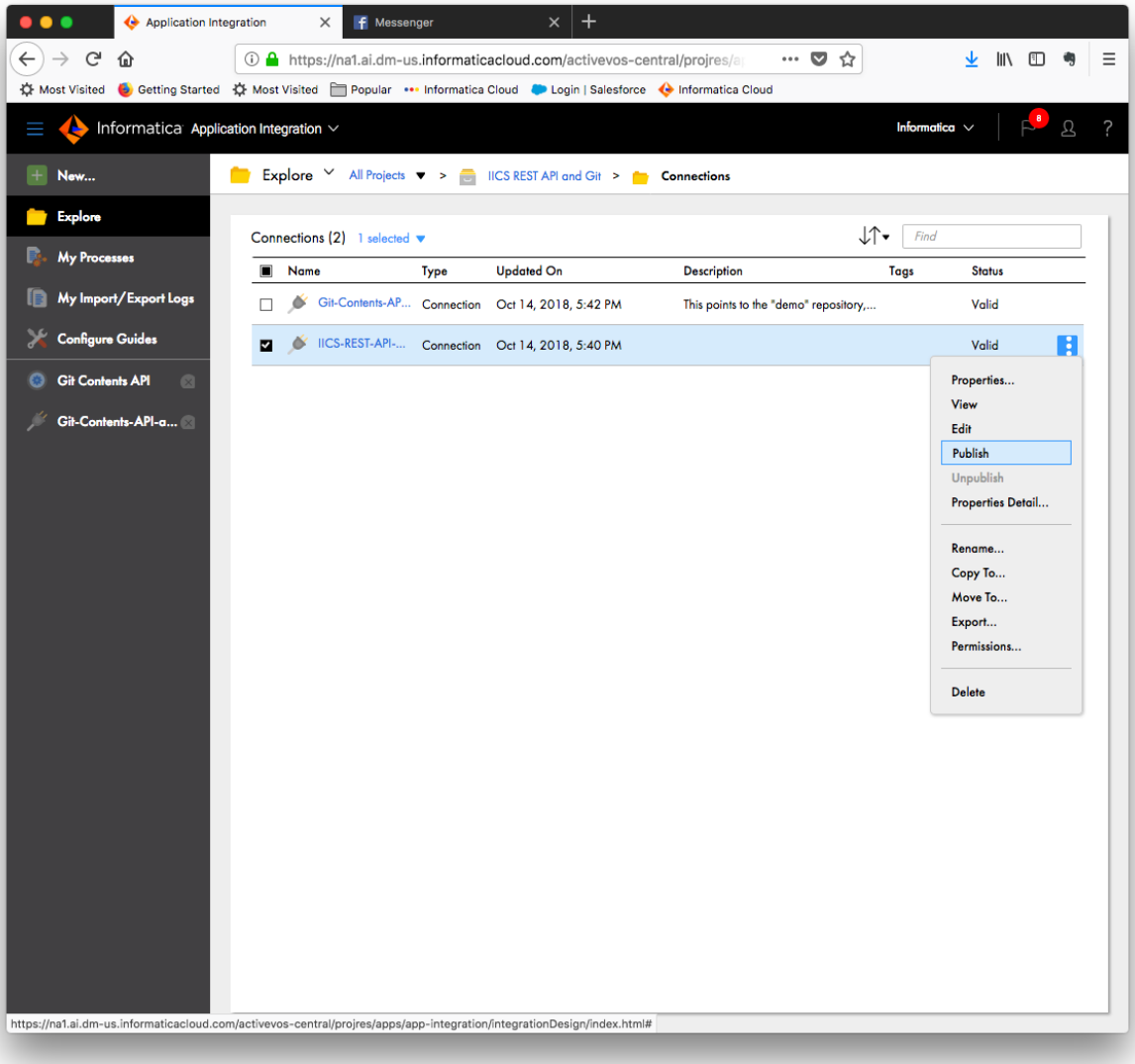
4. Publish the “IICS REST API” Service Connector



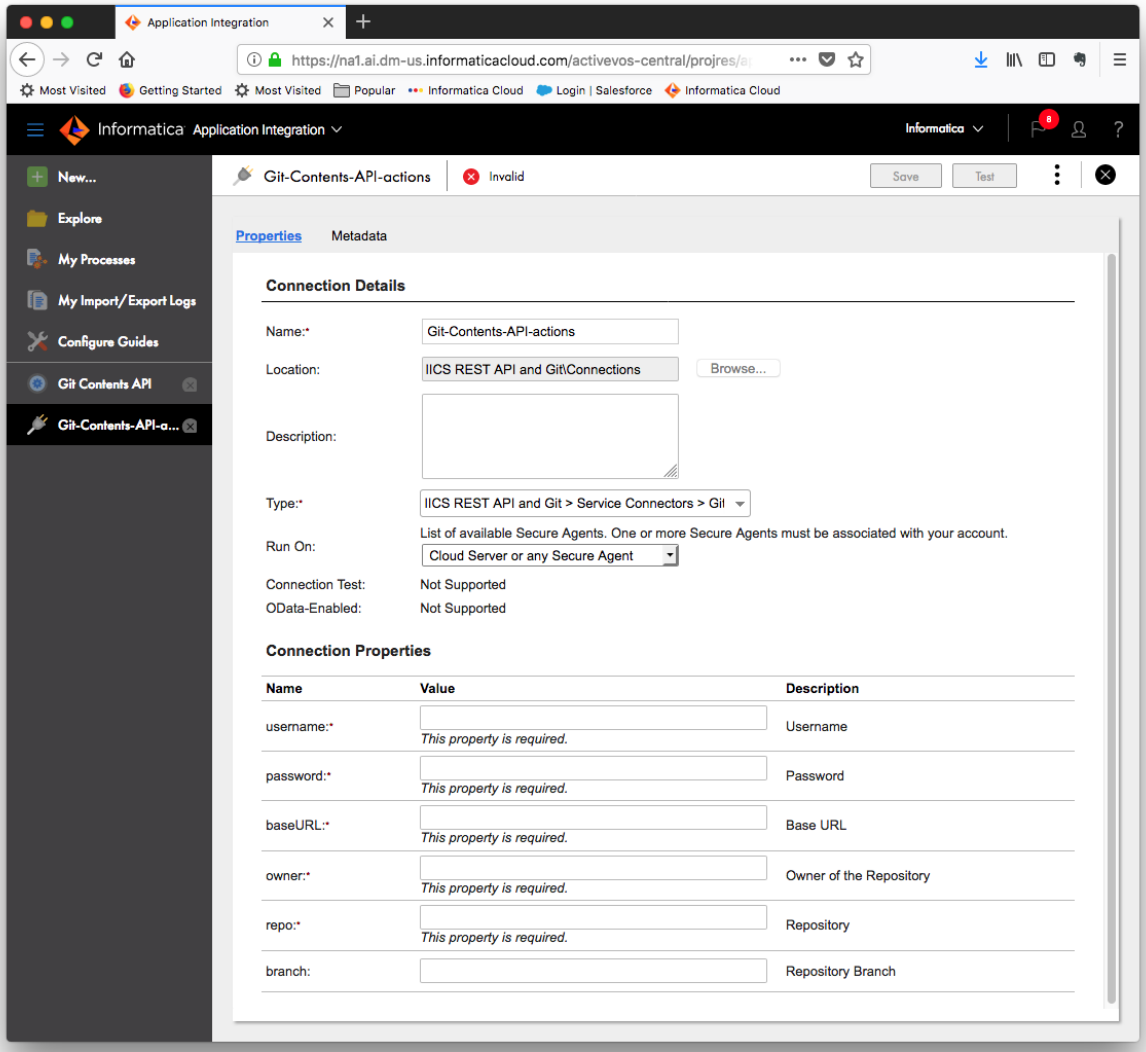
5. Publish the “Git Contents API” Service Connector



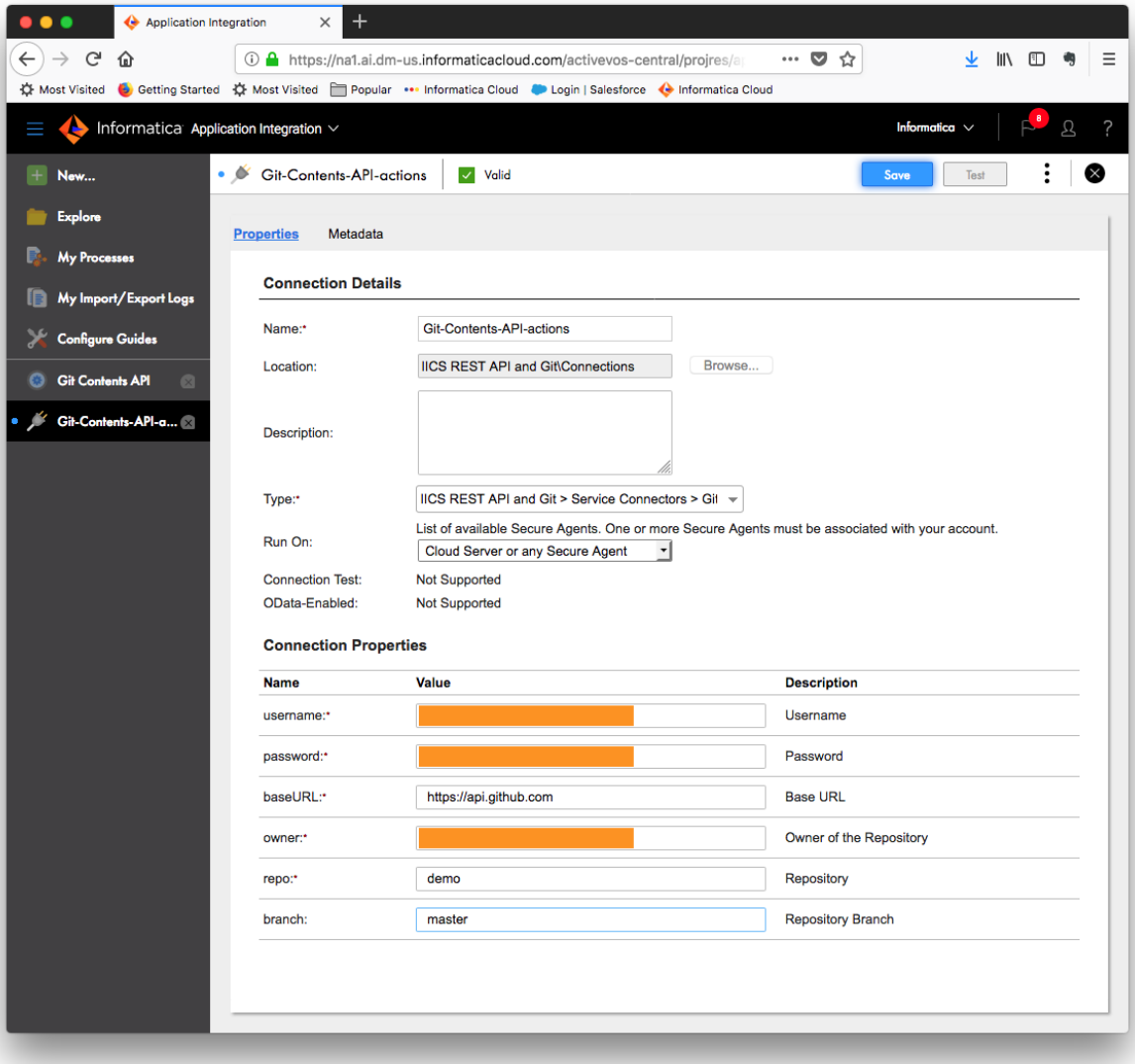
6. Publish the IICS-REST-API-Actions Connection



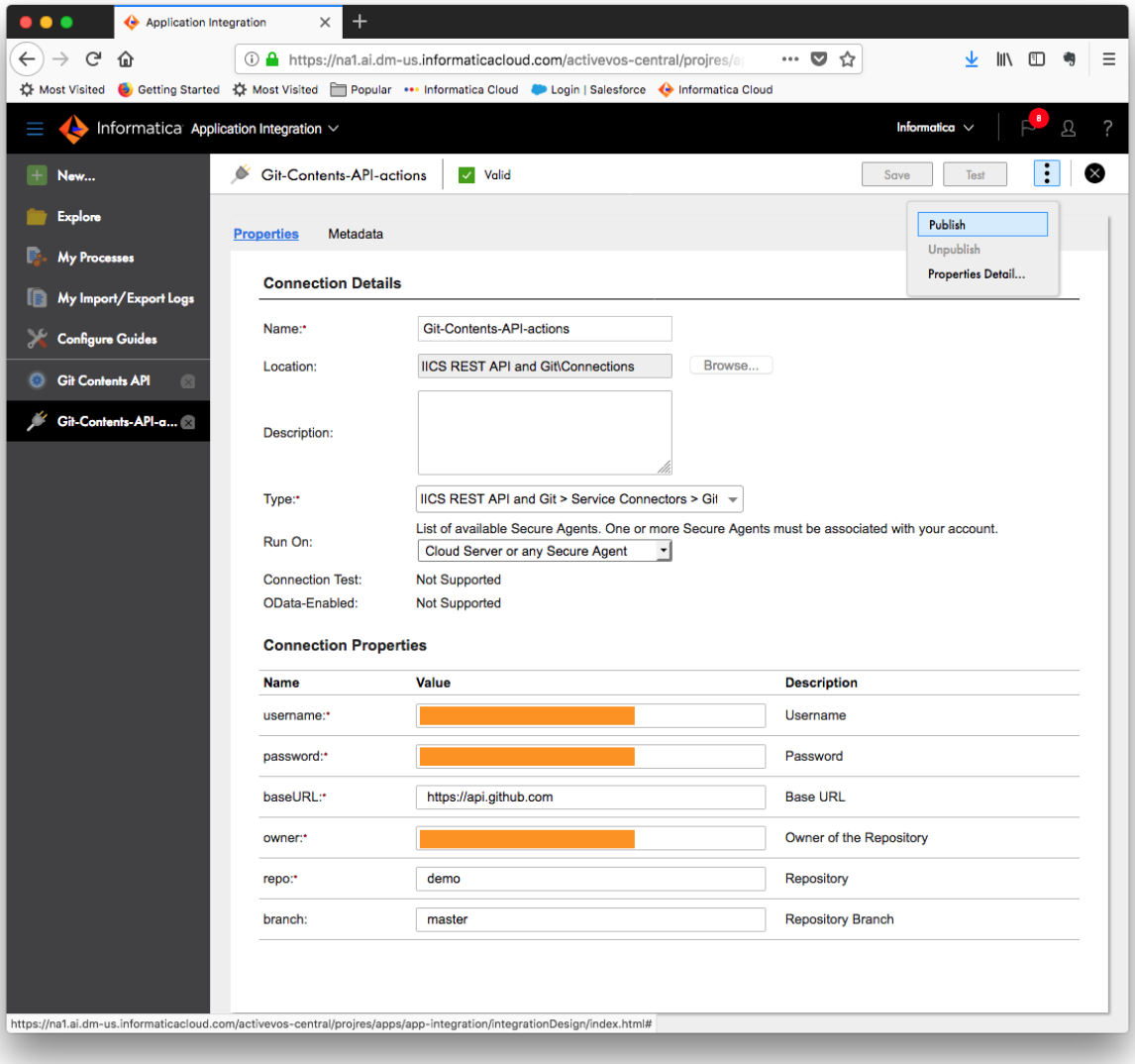
7. Supply the necessary values in the Git-Contents-API-actions Connection that correspond to your Git environment.



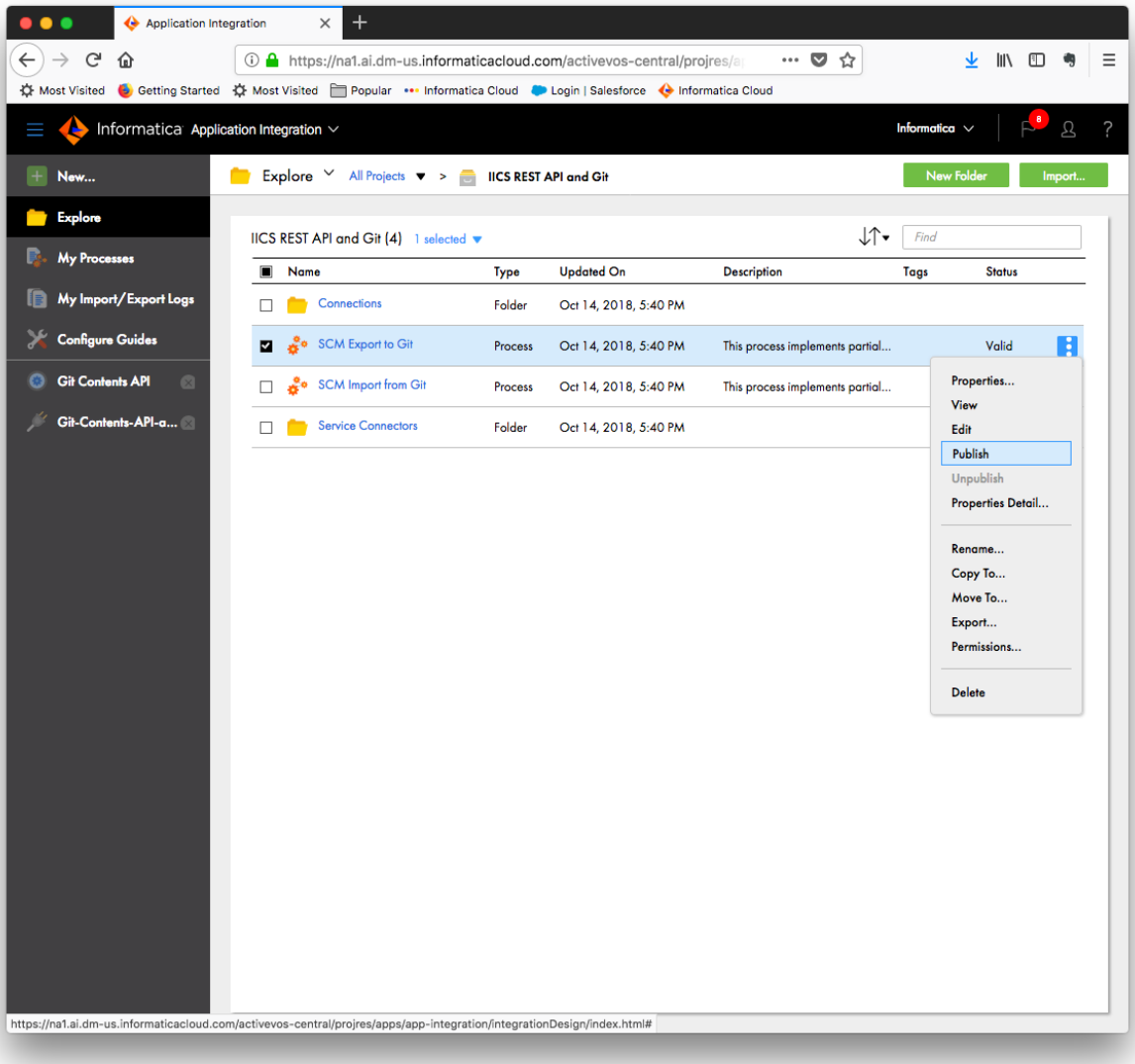
8. Save the Connection



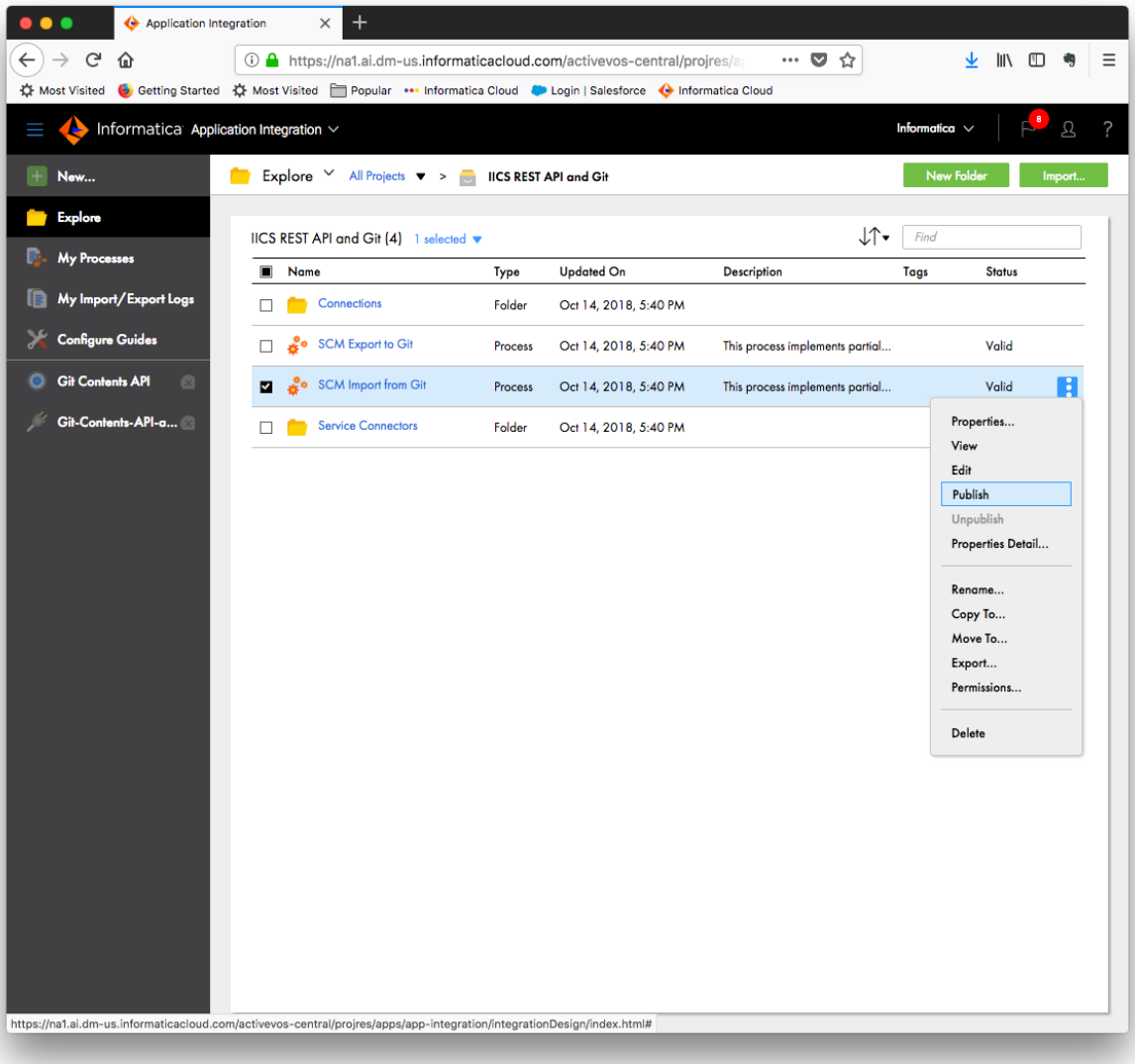
9. Publish the Connection



10. Publish the “SCM Export to Git” Process

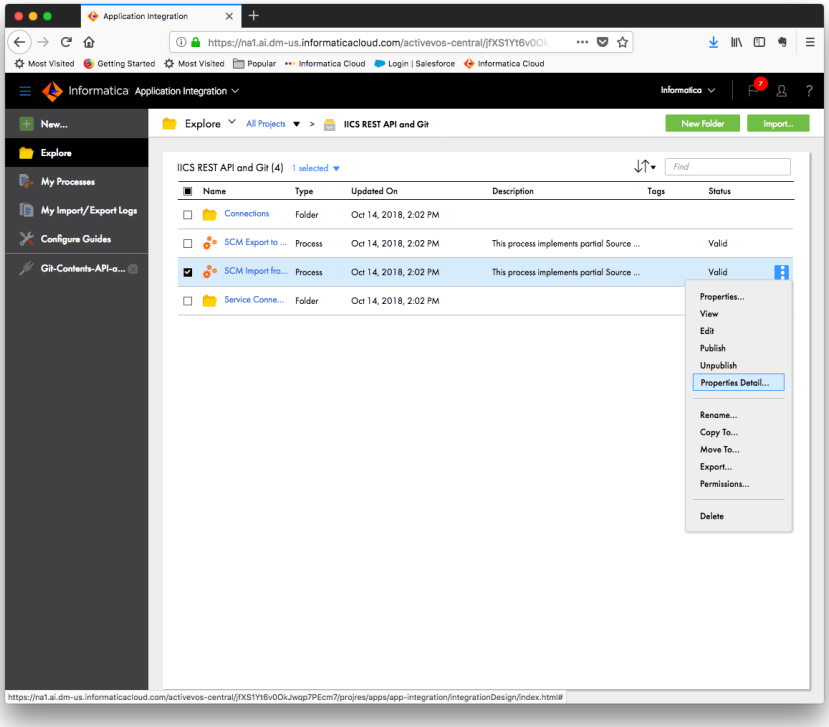


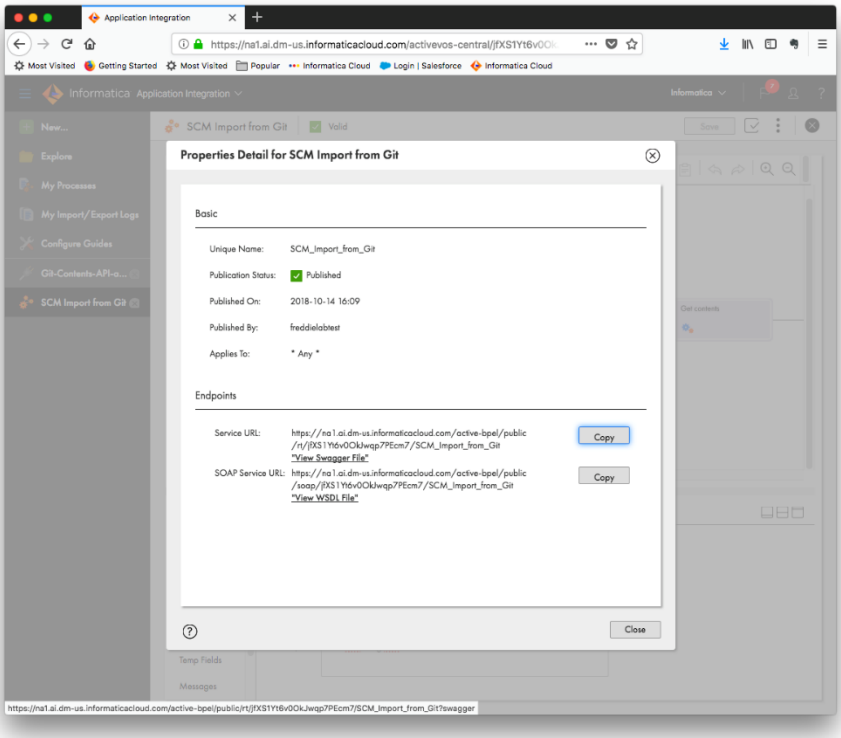
11. Publish the “SCM Import from Git” Process



12. Test **Import from Git** using Postman
    a. Define a new Request
    b. Configure the Request as a POST.
    c. Set the Body to “raw” and “JSON (application/json)”

d. Get the Service URL from the Properties Detail of the “SCM Import from Git” process

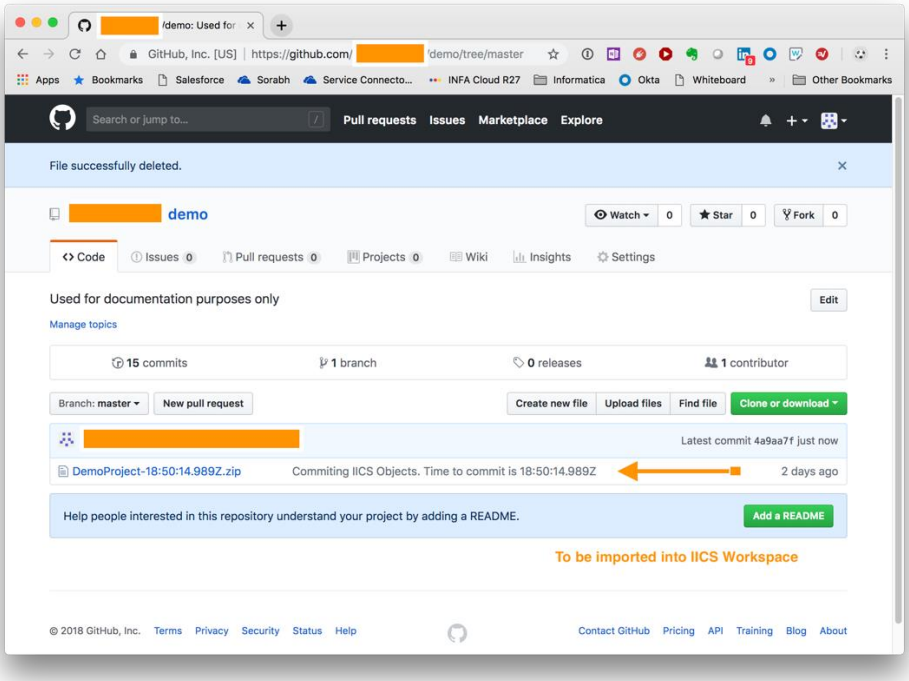




e. Set the Request URL in your Postman request to the Service URL obtained above

f. Configure your input – importing a project named “DemoProject” from the root level of the “demo” repository into our IICS workspace. The Git details are defined in the Git-

Contents-API-Actions connection.



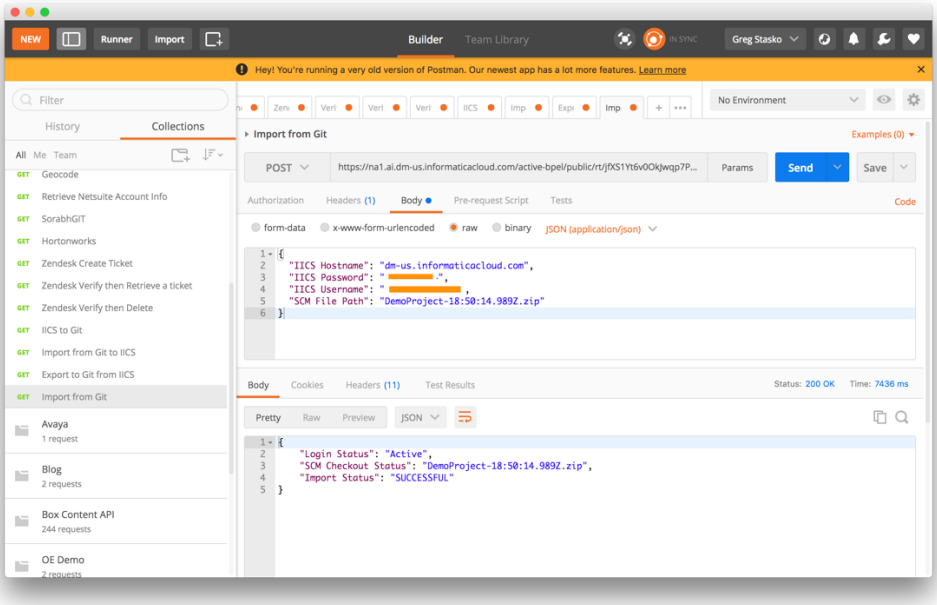
i.
```
{
  "IICS Hostname": "dm-us.informaticacloud.com",
  "IICS Username": "<YourIICSUserName>",
  "IICS Password": "<YourIICSPassword>",
  "SCM File Path": "DemoProject-18:50:14.989Z.zip"
}
```

g. Execute the Request

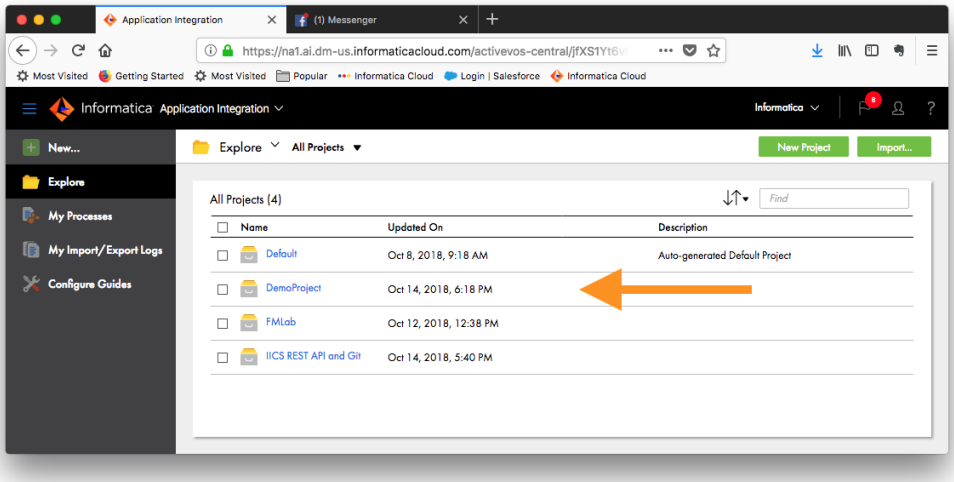
i. The response should look similar to:

```
{
 "Login Status": "Active",
 "SCM Checkout Status": "DemoProject-18:50:14.989Z.zip",
 "Import Status": "SUCCESSFUL"
```

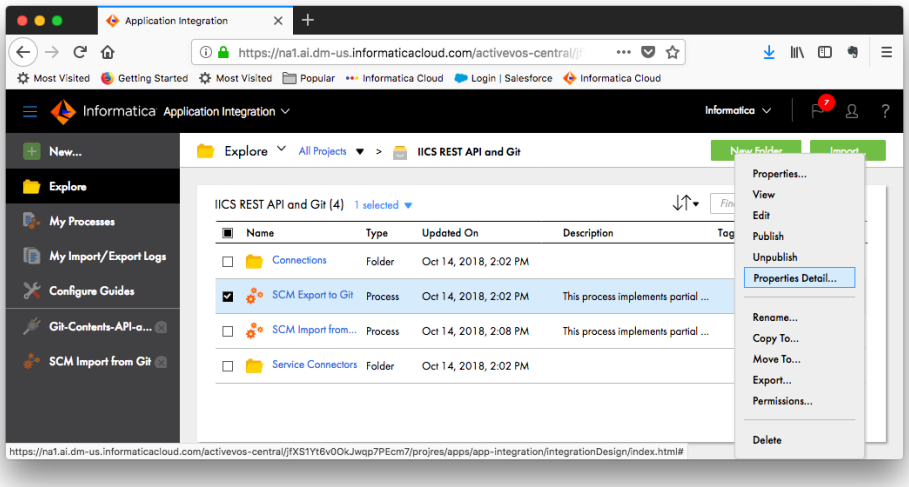
}

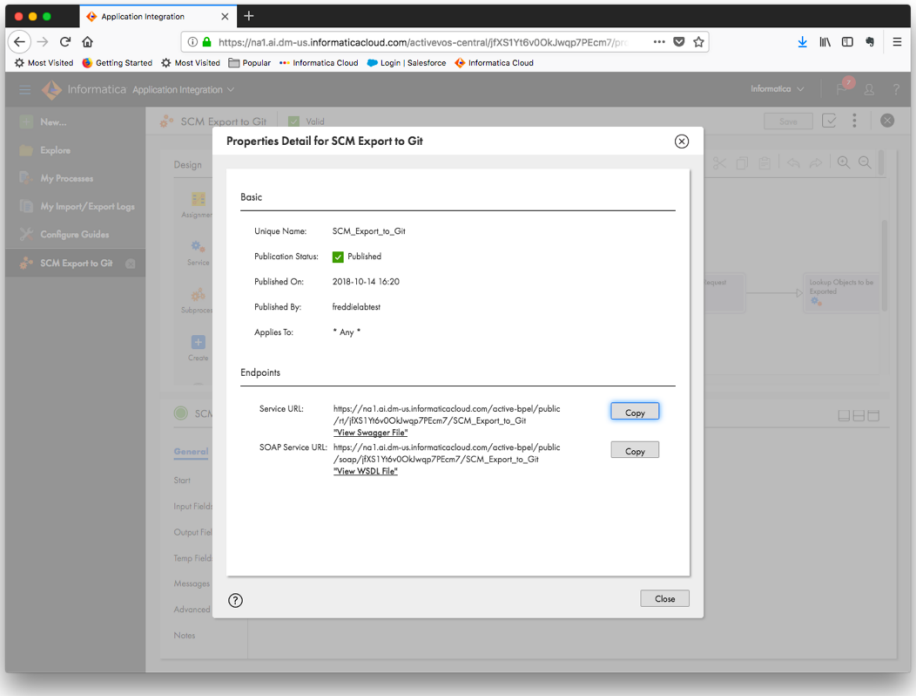


ii. Check your CAI Project View to see that a new project named “DemoProject” has been created.

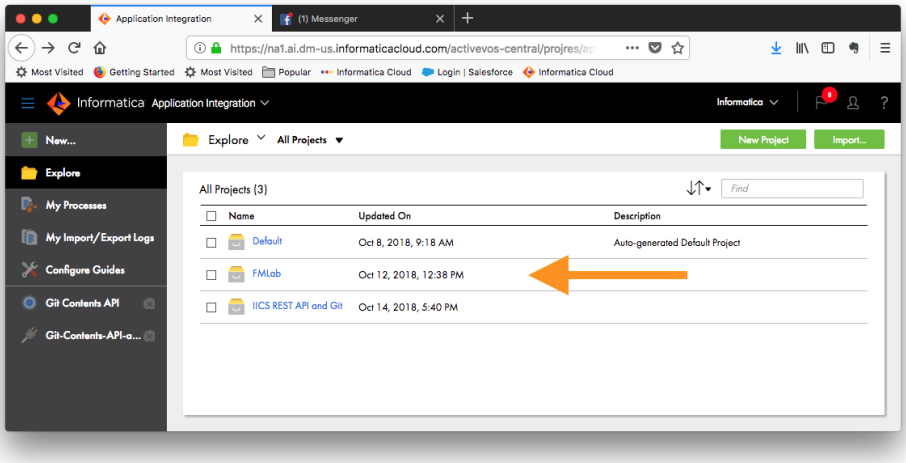


13. Test **Export to Git** using Postman
    a. Define a new Request
    b. Configure the Request as a POST.
    c. Set the Body to “raw” and “JSON (application/json)”
    d. Get the Service URL from the Properties Detail of the “SCM Export to Git” process





    e. Set the Request URL in your Postman request to the Service URL obtained above
    f. Configure your input – Specifying an existing asset named “FMLab” that is of type “Project” that is to be copied to the root level of the “demo” Git repository. The Git details are defined in the Git-Contents-API-Actions connection.



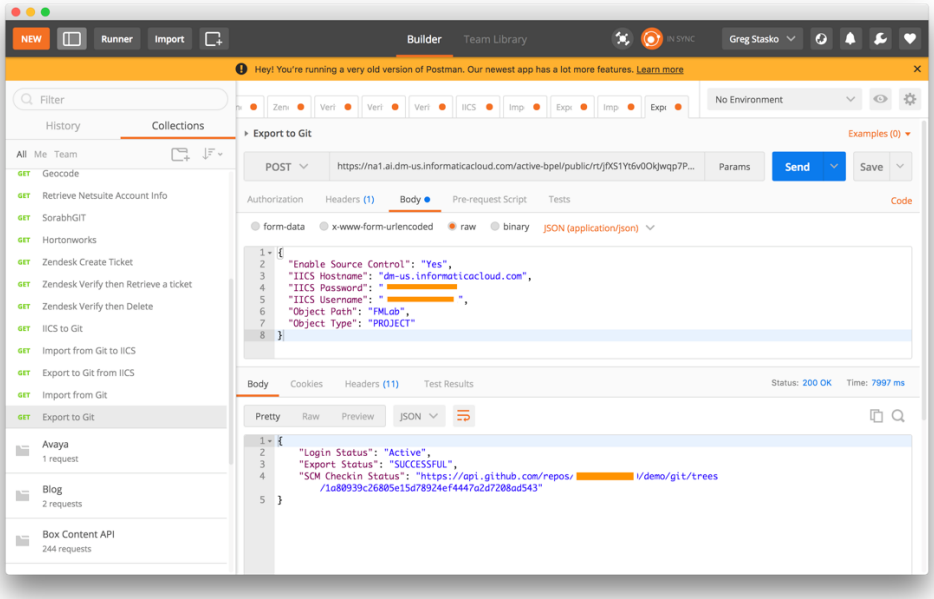
i.
```
{
"Enable Source Control": "Yes",
"IICS Hostname": "dm-us.informaticacloud.com",
 "IICS Username": "<your username>",
 "IICS Password": "<your password>",
 "Object Path": "FMLab",
"Object Type": "PROJECT"
}
```

g. Execute the Request

i. The response should look like:

```
{
 "Login Status": "Active",
 "Export Status": "SUCCESSFUL",
 "SCM Checkin Status":
"https://api.github.com/repos/<yourGitLogin>/demo/git/trees
/1a80939c26805e15d78924ef4447a2d7208ad543"
```

}



ii. Check your Git repository to see that an export of “FMLab” has been created, timestamped, and uploaded.