Informatica Cloud Application Integration

Zendesk Connector

October 2018

Contents

Copyright:

Overview:

Zendesk is one of the world's leading cloud-based customer-service solutions. The Zendesk Connector package enables IICS (IICS) developers to easily incorporate Zendesk functionality into their Cloud Application Integration (CAI) projects.

Prerequisites:

- Zendesk Connector package
- Zendesk credentials (Username, Password, and subdomain)
- Access to Zendesk
- Appropriate role/permissions on Informatica Intelligent Cloud Service to perform an IMPORT operation

Use Cases: Create, Retrieve, and Delete a Zendesk Ticket

Three separate processes are provided, demonstrating how to create a new Zendesk ticket, retrieve an existing Zendesk ticket, and delete an existing Zendesk ticket.

1. Creation of new ticket
    a. Invoke the process with three inputs
        i. Subject

        ii. Description
        iii. Priority
2. Retrieval of an existing ticket
    a. Invoke the process with a Ticket ID
3. Deletion of an existing ticket
    a. Invoke the process with a Ticket ID

Quick Setup (For those familiar with Cloud Application Integration)

1. Download ZendeskConnectorPackage1018.zip from the Informatica CAI GitHub Repository.
2. Import the package into your CAI workspace.
    a. Creates a new project named Zendesk
    b. The Zendesk project contains 5 assets
        i. “Zendesk Core API” Service Connector
        ii. “Zendesk-Core-API-actions” Connection
        iii. “Zendesk Create Ticket” Process
        iv. “Zendesk Verify then Retrieve Ticket” Process
        v. “Zendesk Verify then Delete Ticket” Process
3. Publish the “Zendesk Core API” Service Connector.
4. Edit the “Zendesk-Core-API-actions” Connection.
    a. Insert the required Zendesk information
        i. username
        ii. password
        iii. subdomain
5. Save the Zendesk-Core-API-actions Connection.
6. Publish the “Zendesk-Core-API-actions” Connection.
7. Publish the “Zendesk Create Ticket” Process.
8. Publish the “Zendesk Verify then Retrieve Ticket” Process
9. Publish the “Zendesk Verify then Delete Ticket” Process
10. To test the “Zendesk Create Ticket” process in a REST utility like Postman (https://www.getpostman.com/)
    a. Create a new request in Postman
    b. Set the Service Address to the Service URL of this process
    c. Define a POST operation
    d. Define the Body to “raw” and “JSON (application/json”
    e. Provide input in the body, such as:

```
{
   "Ticket Description": "The smoke is very colorful - ignore",
   "Ticket Priority": "urgent",
   "Ticket Subject": "My printer is on fire!"
}
```

    f. Send
    g. If everything was defined correctly, you should receive output similar to this, although the actual ticket number generated will vary:

```
{
   "output_Ticket ID": "123456"
}
```

11. To test the “Zendesk Verify then Retrieve a Ticket” process in a REST utility like Postman (https://www.getpostman.com/
    a. Create a new request in Postman

b. Set the Service Address to the Service URL of this process
c. Define a POST operation
d. Define the Body to “raw” and “JSON (application/json”
e. Provide input in the body, such as:

```
{
   "input Ticket ID": 123456
}
```

f. Send
g. If everything was defined correctly, you should receive output similar to:

```
{
   "Status": "Ticket successfully retrieved",
   "Output Subject": "My printer is on fire!",
   "Output Priority": "urgent",
   "Output ID": "123456"
}
```

12. To test the “Zendesk Verify then Delete a Ticket” process
    a. Create a new request in Postman
    b. Create a new Set the Service Address to the Service URL of this process
    c. Define a POST operation
    d. Define the Body to “raw” and “JSON (application/json”
    e. Provide input in the body, such as:

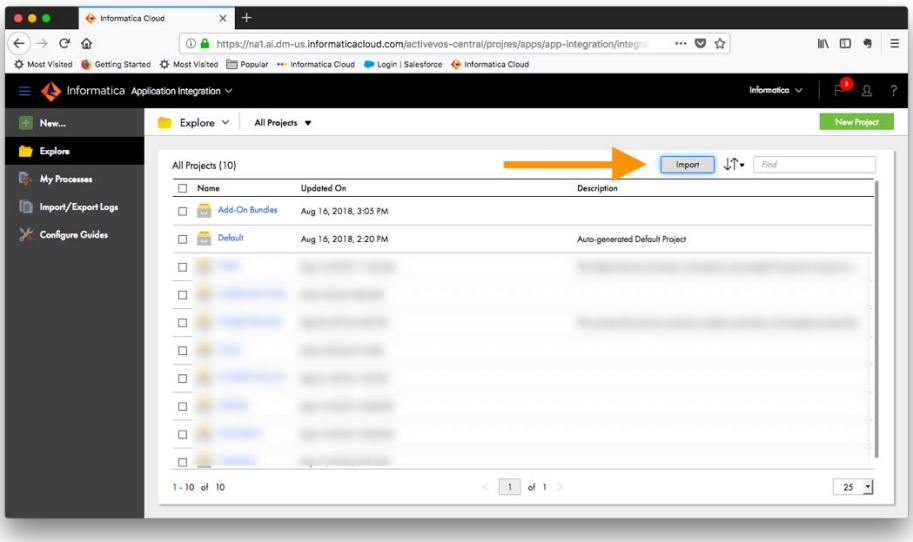
```
{
   "input Ticket ID": 123456
}
```

    f. Send
    g. If everything was defined correctly, you should receive output similar to:

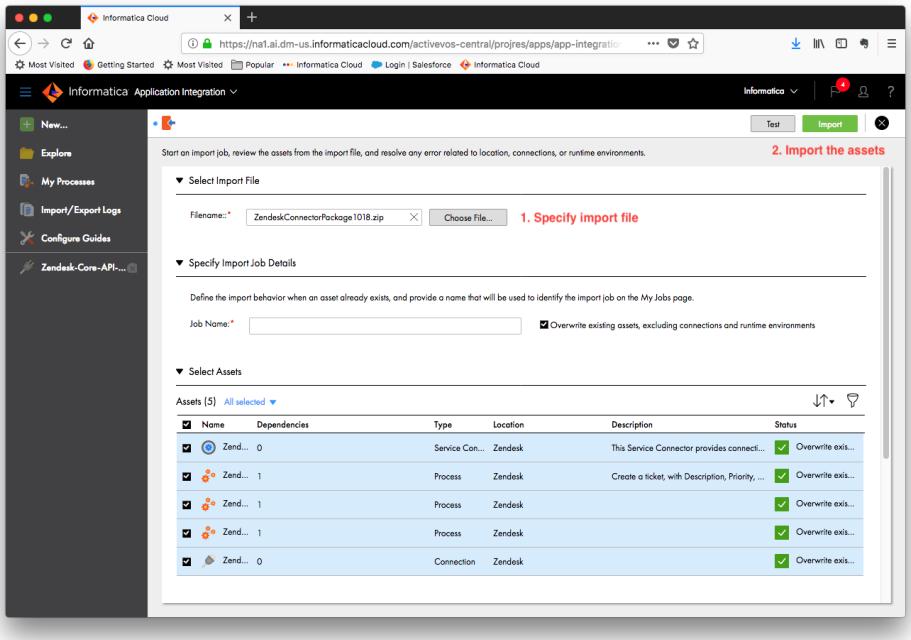
```
{
   "Deletion Status": "Ticket successfully deleted"
}
```

Detailed Setup (For those less familiar with Cloud Application Integration)

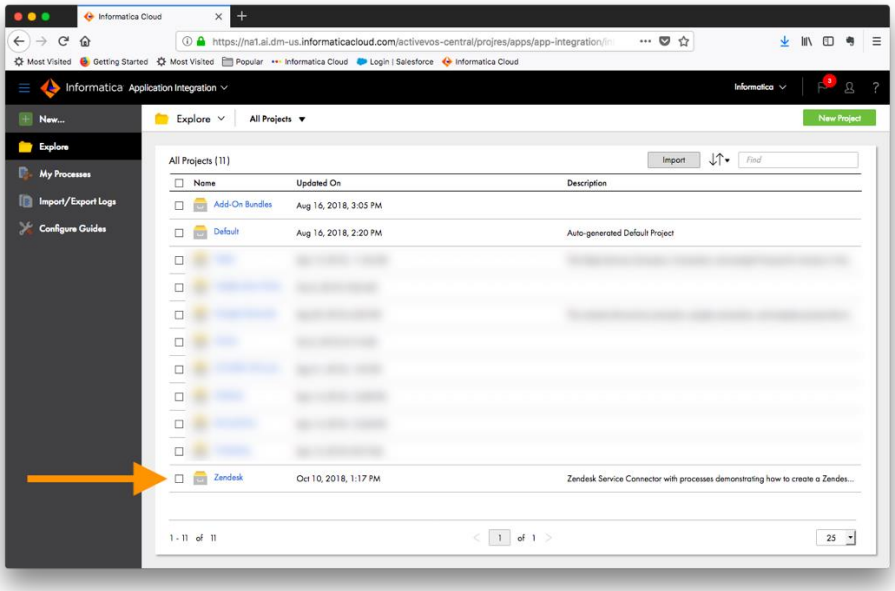
1. Once in Cloud Application Integration, you want to start an “Import” operation by clicking on the Import button in the upper right side of your screen



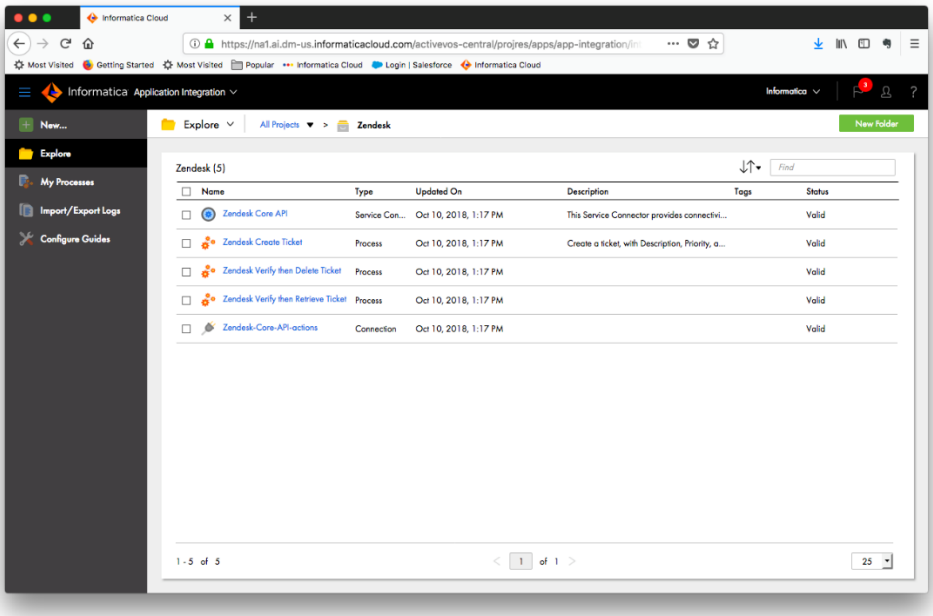
2. Download [ZendeskConnectorPackage1018.zip] from the [Informatica CAI GitHub Repository] and choose it here. The contents of the .zip file are shown in the lower portion of the screen. Then click the green Import button in the upper right corner.



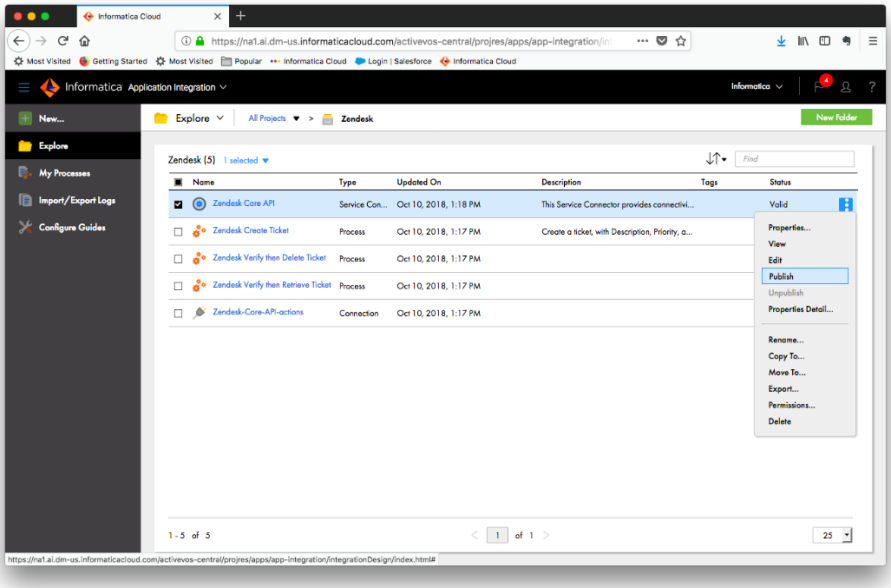
3. There should now be a new project, Zendesk, in your workspace.



4. Open the Zendesk project, which should have 5 assets.

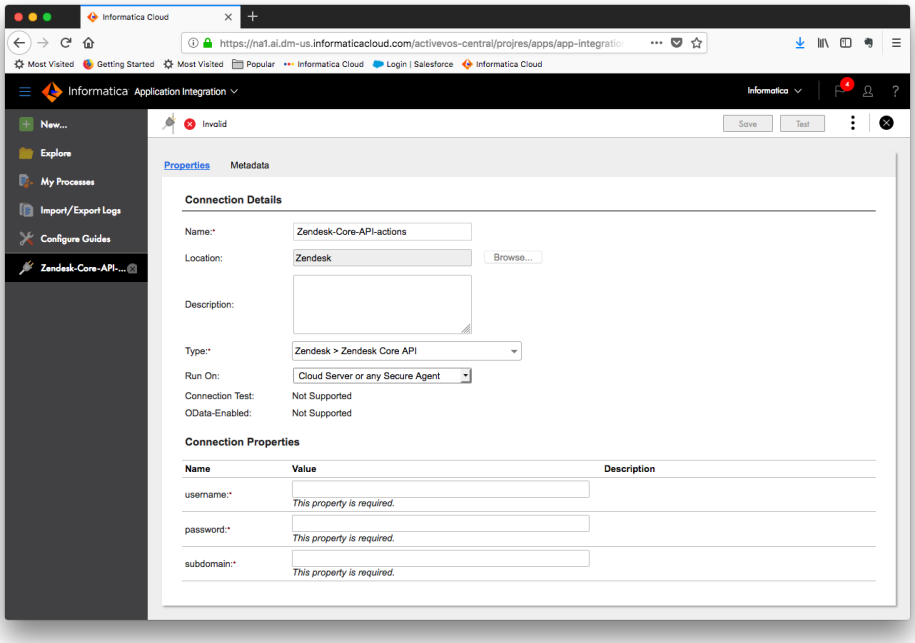


5. Publish the “Zendesk Core API” Service Connector

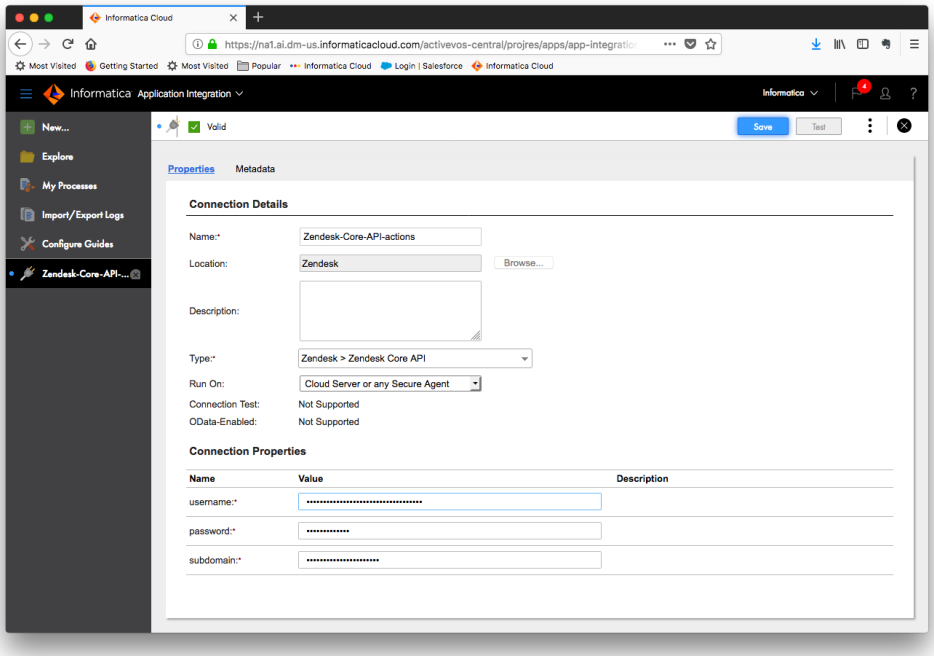


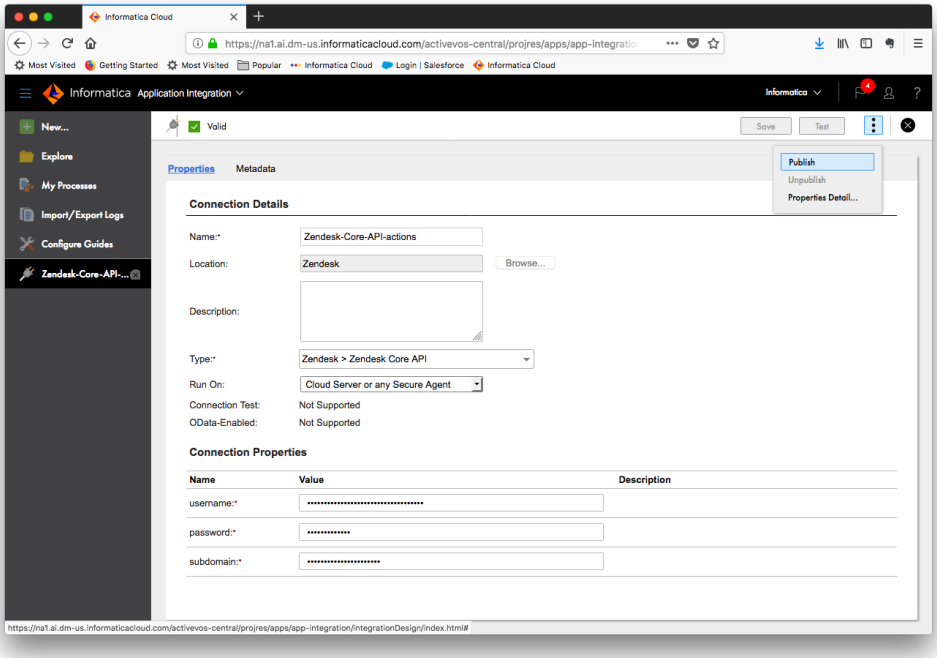
6. Open the “Zendesk-Core-API-actions” Connection

7. Enter the required Zendesk information.

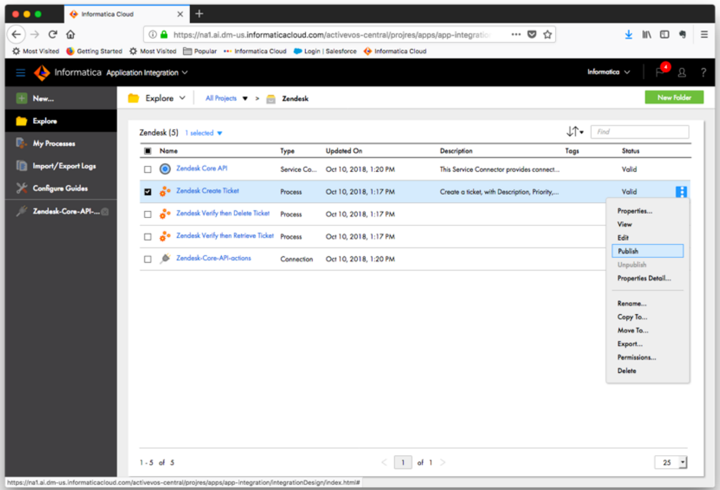


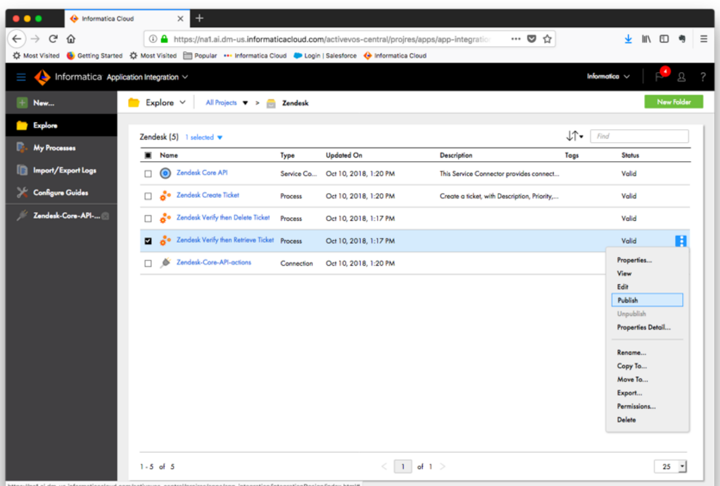
8. Save, then Publish the “Zendesk-Core-API-actions” Connection

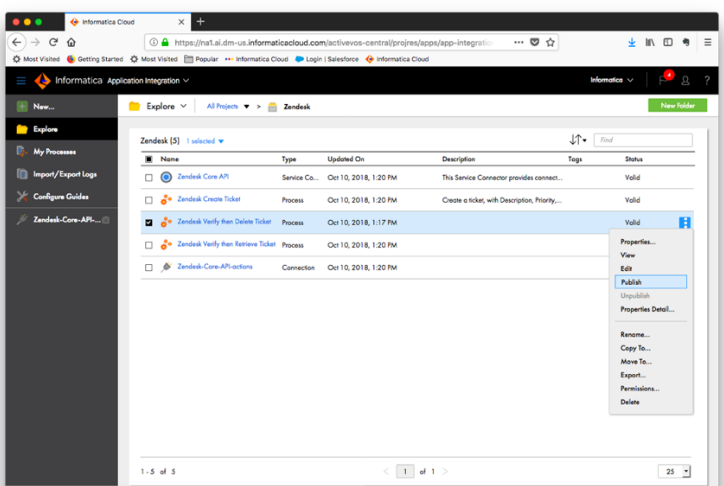




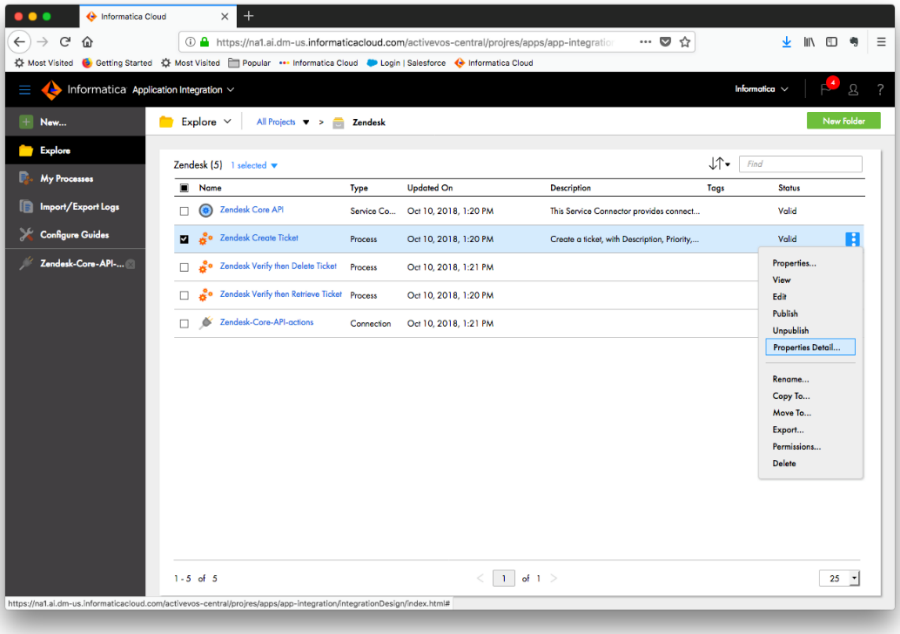
9. Publish the three Processes.

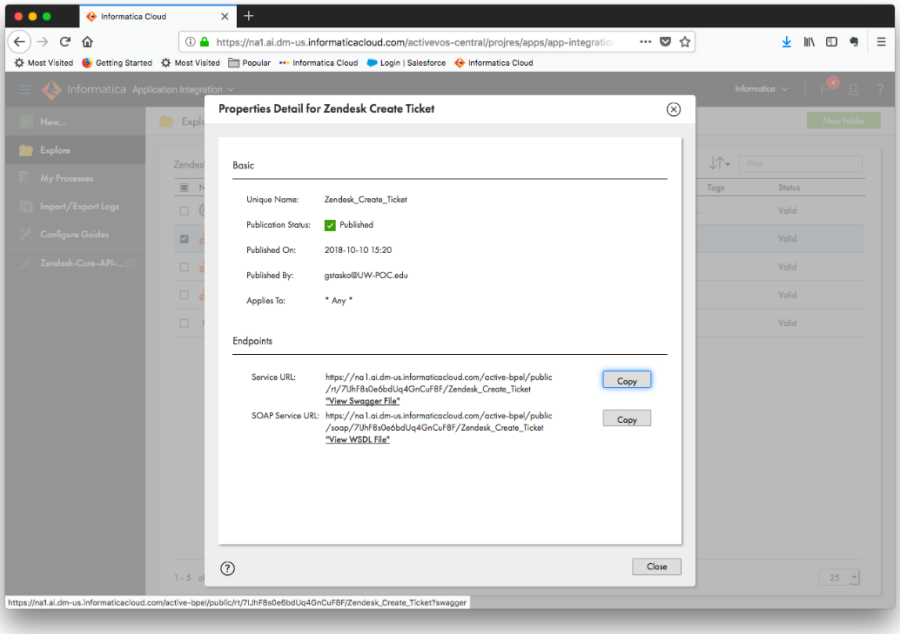






10. Testing the “Zendesk Create Ticket” process in a REST utility like Postman (https://www.getpostman.com/)
    a. Copy the Service URL from the Properties Detail of the process

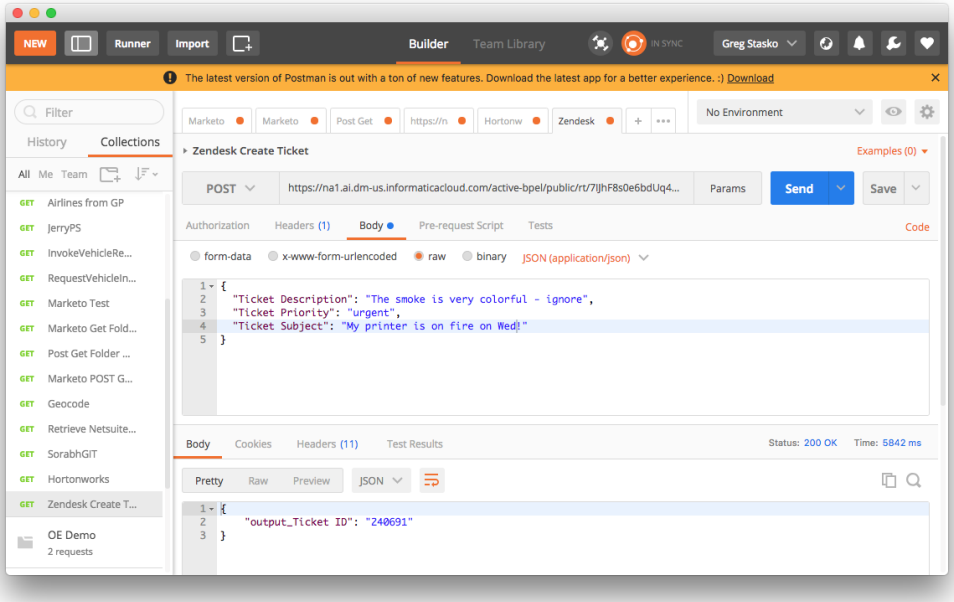




    b. Define a new request in Postman
        i. Set the address of the request to the Service URL obtained previously
        ii. Type is POST
        iii. Body is “raw” and set to “JSON (Application/json)”
        iv. Sample input:

```
{
```

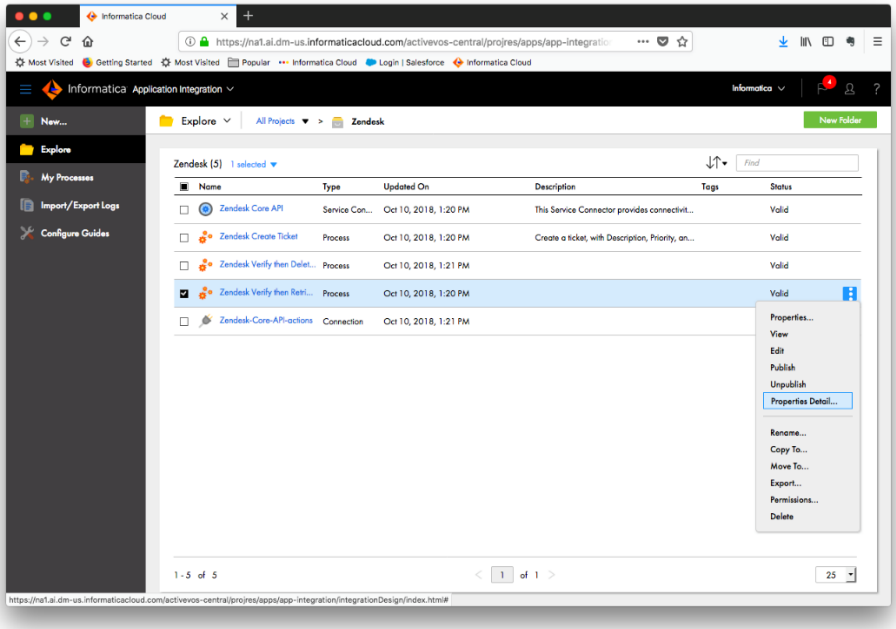
```
   "Ticket Description": "The smoke is very colorful -
ignore",
   "Ticket Priority": "urgent",
   "Ticket Subject": "My printer is on fire on Wed!"
}
```

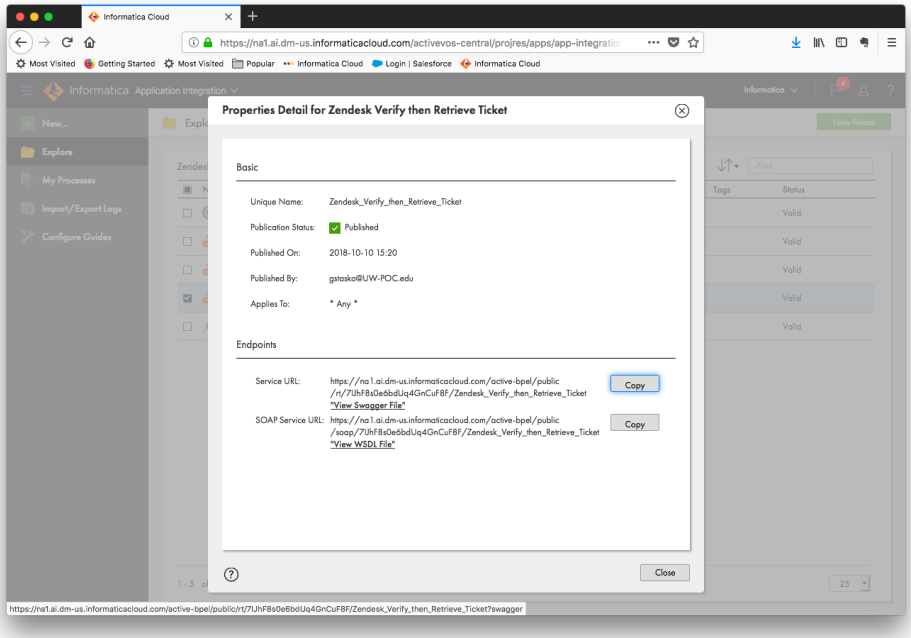


If everything was defined correctly, you should receive output similar to:

```
{
   "output_Ticket ID": "123456"
}
```

11. Testing the “Zendesk Verify then Retrieve a Ticket” in a REST utility like Postman (https://www.getpostman.com/)
    a. Copy the Service URL from the Properties Detail of this process



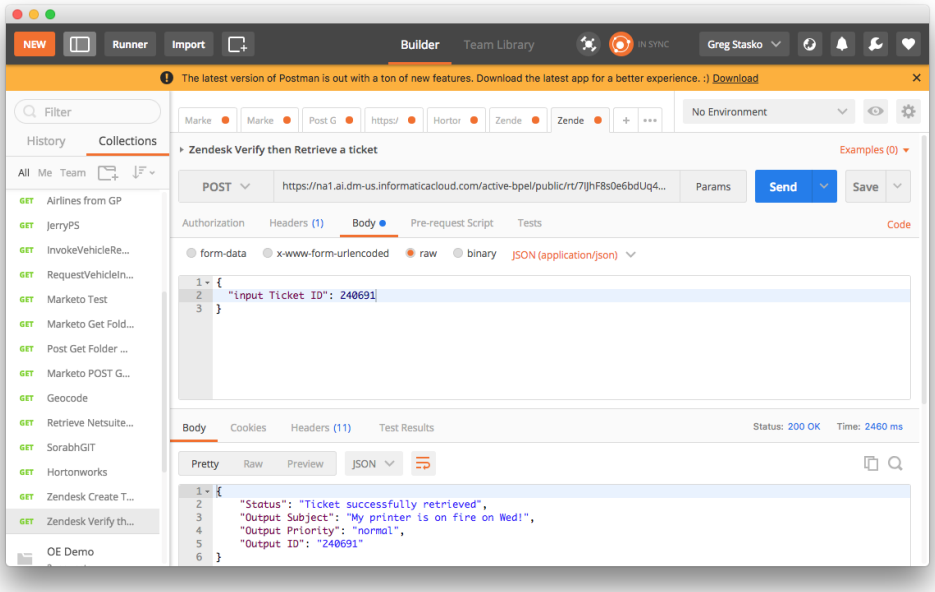


    b. Define a new request in Postman
        i. Set the address of the request to the Service URL obtained previously
        ii. Type is POST
        iii. Body is “raw” and set to “JSON (Application/json)”

iv. Sample input:

```
{
   "input Ticket ID": 123456
}
```

v. Send

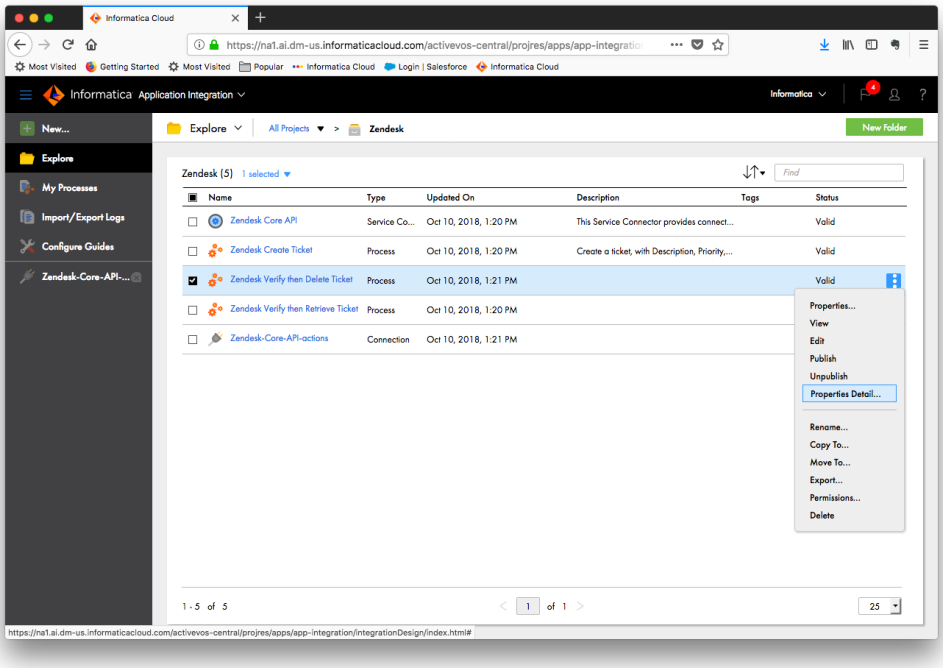


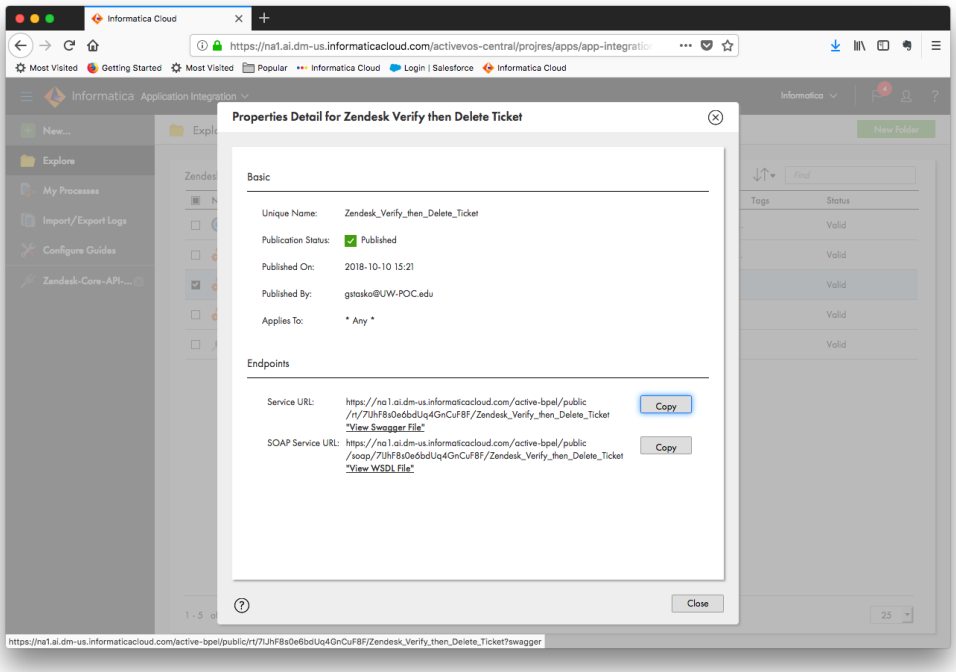
vi. If everything was defined correctly, you should receive output similar to:

```
{
   "Status": "Ticket successfully retrieved",
   "Output Subject": "My printer is on fire!",
   "Output Priority": "urgent",
   "Output ID": "123456"
}
```

12. Testing the “Zendesk Verify then Delete Ticket” process in a REST utility like Postman (https://www.getpostman.com/)

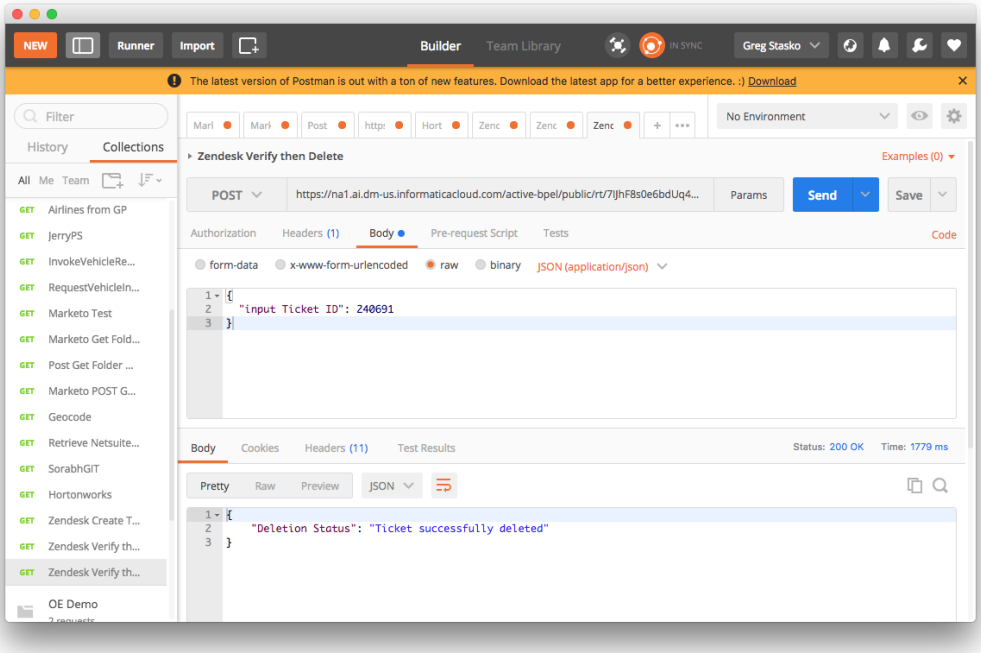
    a. Copy the Service URL from the Properties Detail of this process





b. Define a new request in Postman
   i. Set the address of the request to the Service URL obtained previously
   ii. Type is POST
   iii. Body is “raw” and set to “JSON (Application/json)”
   iv. Sample input:

```
{
    "input Ticket ID": 123456
}
```



If everything was defined correctly, you should receive output similar to:

```
{
    "Deletion Status": "Ticket successfully deleted"
}
```