



# Snowflake Procedure Integration with Python/FastAPI

Sno	Date	Modification	Author	Verified By
1	2019/09/07	Initial Document	Nishtha Vijay	Sumit Goyal

## Table of Contents

Snowflake Procedure integration with python/fastApi .....	3
Business Requirement.....	4
Solutions:.....	4
Steps to use python snowflake for Integration .....	<b>Error! Bookmark not defined.</b>
Code to fetch data fromSnowflake cloud .....	<b>Error! Bookmark not defined.</b>
Testing:.....	6
Steps to test Integartion.....	6
Final Result. ....	<b>Error! Bookmark not defined.</b>

# Snowflake Procedure Integraion with Python/fastApi

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Mostly used in Data-science and machine-learning.

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints.

## Based on open standards

- **OpenAPI** for API creation, including declarations of path operations, parameters, body requests, security, etc.
- Automatic data model documentation with **JSON Schema** (as OpenAPI itself is based on JSON Schema).
- Designed around these standards, after a meticulous study. Instead of an afterthought layer on top.
- This also allows using automatic **client code generation** in many languages.

## Snowflake Stored Procedure:

Stored procedures allow you to extend Snowflake SQL by combining it with JavaScript so that you can include standard programming constructs such as branching and looping. Stored procedures also make code easier to maintain and re-use.

Snowflake stored procedures are implemented through JavaScript and, optionally, SQL:

- JavaScript provides the control structures (branching and looping).
- SQL is executed within the JavaScript by calling functions in an API. SQL is not required in a stored procedure, but is typically included.



# Business Requirement

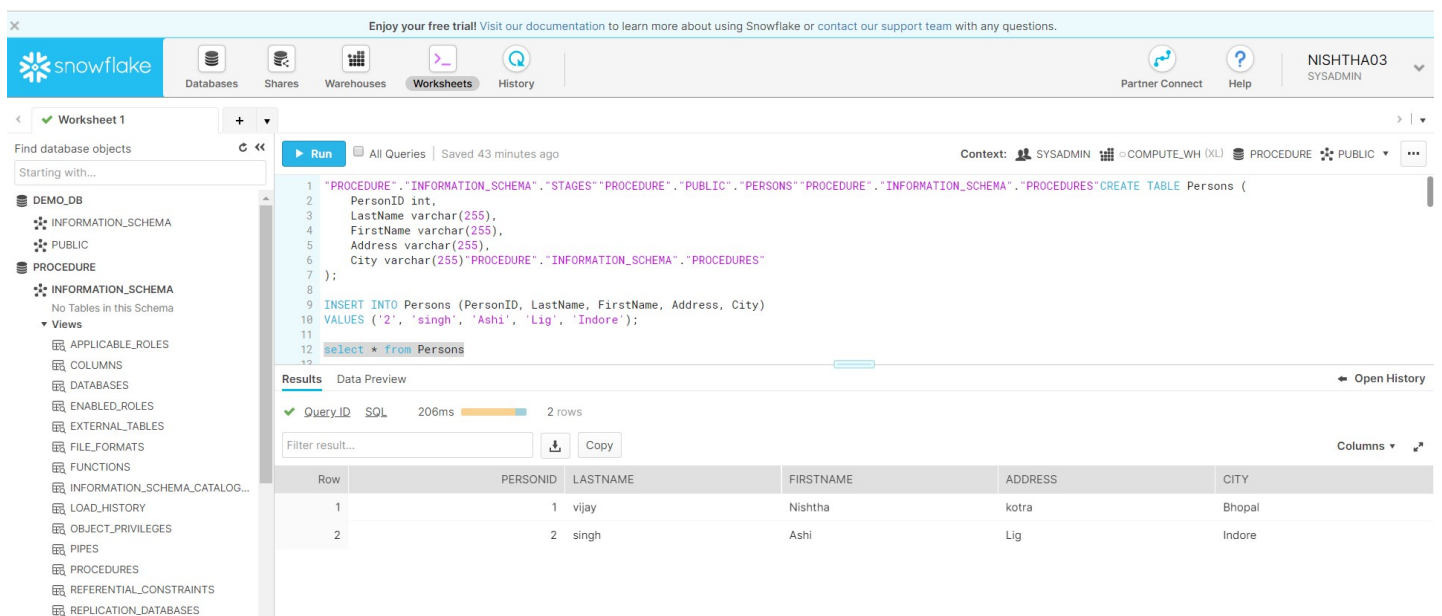
The main objective of this project is that store data in a place where there is no chance to lose. That's why we need to store data in a cloud or with the help of that we can reuse modules.

## Solutions:

**Note:**In this document we explain step by step Integration between Python/FastAPI and Snowflake (To Fetch data from stored procedure) using python connector.

## Steps :

- 1) Login with your snowflake account
- 2) Now create a table named Persons with parameters as PersonID, LastName, FirstName, Address, City  
>>>create table Persons(PersonID int, LastName varchar(20), FirstName varchar(20), Address varchar(30), City varchar(15));
- 3) Insert some records in the table by entering the following command in your worksheet.  
>>>INSERT INTO Persons (PersonID, LastName, FirstName, Address, City)  
VALUES ('1', 'sons', 'john', 'Lig', 'London');
- 4) To check whether we are doing right or not just enter the command in you worksheet  
>>> select \* from persons;
- 5) If you are getting the following screen then you are in the right way.



The screenshot displays the Snowflake web interface. The top navigation bar includes 'snowflake', 'Databases', 'Shares', 'Warehouses', 'Worksheets', and 'History'. The user is logged in as 'NISHTHA03 SYSADMIN'. The main workspace shows a worksheet with the following SQL code:

```
1 "PROCEDURE"."INFORMATION_SCHEMA"."STAGES"."PROCEDURE"."PUBLIC"."PERSONS"."PROCEDURE"."INFORMATION_SCHEMA"."PROCEDURES"."CREATE TABLE Persons (  
2 PersonID int,  
3 LastName varchar(255),  
4 FirstName varchar(255),  
5 Address varchar(255),  
6 City varchar(255)"PROCEDURE"."INFORMATION_SCHEMA"."PROCEDURES"  
7 );  
8  
9 INSERT INTO Persons (PersonID, LastName, FirstName, Address, City)  
10 VALUES ('2', 'singh', 'Ashi', 'Lig', 'Indore');  
11  
12 select * from Persons
```

The 'Results' section shows a data preview with 2 rows:

Row	PERSONID	LASTNAME	FIRSTNAME	ADDRESS	CITY
1	1	vijay	Nishtha	kotra	Bhopal
2	2	singh	Ashi	Lig	Indore

6) Create a procedure by entering the following command.

```
>>>create or replace procedure read_person_proc()
  returns String not null
  language javascript
  as
  $$
  var my_sql_command = "select * from Persons";
  var statement1 = snowflake.createStatement( {sqlText: my_sql_command} );
  var result_set1 = statement1.execute();
  // Loop through the results, processing one row at a time...
  while (result_set1.next()) {
    var column1 = result_set1.getColumnValue(1);
    var column2 = result_set1.getColumnValue(2);
    var column3 = result_set1.getColumnValue(3);
    var column4 = result_set1.getColumnValue(4);
    var column5 = result_set1.getColumnValue(5);
    var column = column1+' '+column2+' '+column3+' '+column4+' '+column5
  }
  return column;
  $$
;
```

7) To call the procedure just enter the command

```
>>>call read_person_proc();
```

## **Now our requirement is we need to call that procedure in python using FastAPI**

Below code will help you to fetch/call procedure form Snowflake into FastAPI.

```
import snowflake.connector as sf
username='your username'
password='your password'
account='your account'
warehouse='your ware house'
database='your data base name'
ctx=sf.connect(user=username,password=password,account=account)
@app.get('/fetchdata')
async def fetchdata():
    cursor = ctx.cursor()
```

```
cursor.execute("use warehouse your_warehouse_name")

cursor.execute("alter session set timestamp_type_mapping = 'TIMESTAMP_NTZ'")
#
cursor.execute("alter session set timezone = 'Europe/Berlin'")
#
cursor.execute("alter session set TIME_OUTPUT_FORMAT = 'HH24:MI:SS.FF'")
#
cursor.execute("use database database_name")
#
cursor.execute("use schema schema_name")
sql = cursor.execute("call read_person_proc()")

for data in sql:
return data
```

**You need to follow each step otherwise you may face some issues**

## **Testing:**

### **Steps to test Integartion**

Step1: Start fastApi Server by entering the following command

>>>uvicornmain:app--reload

Step2: Go to the browser and hit the following url

>>><http://127.0.0.1:8000/fetchdata>

If you get something like this. Congratulations you are successfully connected.

.....Thanks for reading.....