Synthetic Data Generator

This plugin contains a smart service to generate fake data in the given table that can be used for testing purposes.
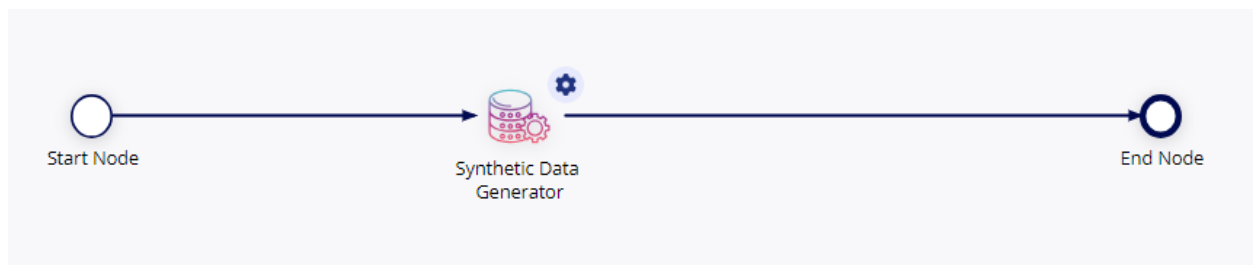
Smart service released as part of this plugin

- Synthetic Data Generator

Smart services

Synthetic Data Generator

Smart service to generate the data in the given table.

When the smart service is deployed, a new data type will be included in the environment.

- SyntheticFieldDetails
  ```
  {
    fieldName: (Text),
    dataType: (Text),
    limit: (Text),
    pattern: (Text)
  }
  ```

Input

- **Data Source** (Text) - Mandatory
  Data source in which the source table is present. The name of the data source can be found in the admin console.

- **Table Name** (Text) - Mandatory
  The name of the table in which the data needs to be generated. The table should be already created.

- **Synthetic field details** (Array of **SyntheticFieldDetails**) - Mandatory
  Specify the columns that need to be populated with data in the source table. We need to specify all the columns in the table except the identifier (primary key) column which is mostly auto-increment in appian.

  fieldName - Specify the name of the column in the table (Case sensitive).
  dataType - Specify the type of data that needs to be populated in this column (Supported types are listed in the table below).
  limit - Specify the limit if any (Supported only for selected data types).
  pattern - Specify the pattern if any (Supported only for selected data types).

  ```
  {
    {
      fieldName: Username",
      dataType: "AppianUsername"
    },
    {
      fieldName: "FullName",
      dataType: "AppianFullName"
    },
    {
       fieldName: "AccountNumber",
       dataType: "AccountNumber",
       pattern: "SBI-???-####"
    }
   }
  }
  ```

- **No Of Rows** (Number - Integer) - Mandatory
  Number of rows that need to be created in the given table.

- **Time Limit** (Number - in milliseconds) - Parameter to set the timeout limit. The maximum time limit that can be set is 50 mins (3000000 ms) and minimum is 1 second (1000 ms)

This table describes the data types that are supported for data generation

Data Type	Limit	Pattern
AppianFullName (generate a appian fullname from the environment)	-	-
AppianUserName (same as AppianFullName)	-	-
AppianFirstName (same as AppianFullName)	-	-
AppianLastName (same as AppianFullName)	-	-
AppianEmail (same as AppianFullName)	-	-
FullName (random name)	-	-
DateTime	minDate-maxDate User can provide min and max date range separated by '/' Date should be in the format **"yyyy-mm-dd"** Example: "2018-12-31/2019-12-31"	-
Date	minDate-maxDate User can provide min and max date range separated by '/' Example: "2018-12-31/2019-12-31"	-
loremIpsum	minLimit-maxLimit Example: **10-15** will create random characters between 10-15	-
AccountNumber	-	If pattern is not given, then a random account number is generated. If pattern is given then,

		# - replaced by random digit, ? - replaced by random characters and all the remaining characters are retained. Example - ###-???-### Will generate 342-ABD-567 SBI-###??? Will generate SBI-453HND
Email	-	If pattern is not given then a random email is generated. If pattern is given then # is replaced by a random number and ? is replaced by a random character. All other characters are retained. If '*' is given then it is replaced by random username Example: *@gmail.com - Replaces '*' with random username. ???###@appian.com - Replaces ? with random character and # with random numbers. **bfs341@appian.com**
FullAddress	-	-
AddressLine1	-	-
AddressLine2	-	-
AddressLine3	-	-
State	-	-
Country	-	-
Zipcode	-	-
RandomNumber	-	If pattern is not given, Random numbers between 0-9 are generated. If pattern is given # are replaced by random digits. ### - will generate a random 3

			digit number.
PhoneNumber		-	If the pattern is not given then a random phone number is generated. If pattern is given, ########## will generate 10 random digit phone numbers. ###-###-### will generate a phone number separated by '-'. In the same way in the given pattern # is replaced by a random digit and all other characters are retained.

Output:

- **TotalRowCount** (Number)
  Returns the number of rows that are generated.
- **errorOccurred** (Boolean)
  Returns true if any error occurred.
- **errorMessage** (Text)
  Returns the error Message.

Supported DataTypes:

MySql, mariaDB.