**Python Utility for ‘Service Search Text’ Synchronization**

**Purpose**

This document outlines the requirements for a Python utility that updates a JSON file with the latest service categories and tags from a PostgreSQL database. This utility ensures the JSON file always contains the most recent data by checking for changes and resetting update flags to avoid processing of the same records again.

**Description**

This python utility checks PostgreSQL tables 'tbl\_service\_category' and 'tbl\_service\_tag' for any records with the 'updated' flag set to True.

**Scope**

This utility does the following:

* Checks if there are any updated records in the service category and service tag tables.
* Retrieves all records from these tables if updates are found.
* Creates a JSON file (‘service\_search\_text.json’) with the updated data.
* Replaces the old JSON file with the new one.
* Resets the update flags in the database tables to prevent reprocessing the same records.

**Requirements**

(Database Tables)

1. ‘tbl\_service\_category’

* ‘id’- Unique identifier for each category.
* ‘name’- Name of the category.
* ‘updated’- Flag indicating if the record has been updated.

1. ‘tbl\_service\_tag’

* ‘id’- Unique identifier for each tag.
* ‘name’- Name of the tag.
* ‘categories’- Array of category IDs associated with the tag.
* ‘updated’- Flag indicating if the record has been updated.

**Attributes**

* ‘service\_tag\_id’ - ID of the service tag (nullable).
* ‘service\_categories’ – Array of category IDs.
* ‘service\_search\_text’ – Text for searching the result.

**Generation Rules**

1. From ‘tbl\_service\_category’:

* Create an object for each record.
* ‘service\_tag\_id’ is set to ‘null’.
* ‘service\_categories’ contains the record’s ‘id’.
* ‘service\_search\_text’ is the record’s ‘name’.

2. From ‘tbl\_service\_tag’:

* Create an object for each record.
* ‘service\_tag\_id’ is set to the record’s ‘id’.
* ‘service\_categories’ is the record’s ‘categories’.
* ‘service\_search\_text’ is the record’s ‘name’.

**Process**

1. Initial Check

Look for records with ‘updated’ flag set to ‘True’ in both tables.

2. Data Retrieval

If updates are found, get all records from both tables.

3. JSON File Generation

* Create objects based on the records using the predefined rules.
* Generate a ‘service\_search\_text.json’ file with these objects.

4. File Replacement

Replace the old JSON file with the new one.

5. Reset Update Flags

Setting the ‘updated’ flag to ‘False’ for all processed records to avoid reprocessing them.

**Post-Execution**

* The JSON file is updated with the latest data.
* All records in the database tables have their ‘updated’ flag reset to ‘False’ to prevent duplication on future runs.

**Example**

* For a service category:

{

"service\_tag\_id": null,

"service\_categories": [1],

"service\_search\_text": "Category Name"

}

* For a service tag:

{

"service\_tag\_id": 1,

"service\_categories": [1, 2],

"service\_search\_text": "Tag Name"

}

**Execution**

1. The utility finds updated records from the service category table.

2. Retrieves all records from both tables.

3. Generates ‘service\_search\_text.json’ with objects from the records.

4. Replaces the old JSON file.

5. Resets the ‘updated’ flags to ‘False’.