



Opsgenie to Salesforce Case Automation

Overview

This Python script automates the creation, tagging, and closing of Salesforce cases based on open alerts in Opsgenie. It enables seamless integration between your on-call alerting system and Salesforce CRM by:

- Checking for new Opsgenie alerts
 - Creating a Salesforce Case if no existing Case is found
 - Tagging the Opsgenie alert with the Salesforce Case Number
 - Adding links to Salesforce and documentation back into Opsgenie
 - Closing the Salesforce case when the alert is resolved in Opsgenie
-



Features

- Connects to Opsgenie and Salesforce via respective APIs
 - Identifies alerts needing action (no case tag, or alert still open)
 - Matches Opsgenie alert recipient to Salesforce user via email
 - Assigns Salesforce Case to the on-call recipient
 - Adds relevant notes and documentation links
 - Automatically closes Salesforce Cases if the Opsgenie alert is resolved
-



Project Structure

```
bash
CopyEdit
```

```
.
├── script.py           # Main script file (this code)
├── config.json         # Configuration file with credentials and API keys
└── README.md          # This file
```



Requirements

- Python 3.7+
- Required Python packages:


```
bash
CopyEdit
```

```
pip install pandas simple-salesforce opsgenie-sdk requests
```

Configuration

Create a `config.json` file in the same directory as the script:

```
json
CopyEdit
{
  "sfops": {
    "opsgenie_api_url": "https://api.opsgenie.com/v2/alerts",
    "opsgenie_api_url_notes":
"https://api.opsgenie.com/v2/alerts/{alertId}/notes",
    "opsgenie_api_key": "YOUR_OPSGENIE_API_KEY",
    "sf_username": "your_salesforce_username",
    "sf_password": "your_salesforce_password",
    "sf_token": "your_salesforce_security_token"
  }
}
```

 **Note:** Never commit `config.json` with real credentials to source control.

How to Run

```
bash
CopyEdit
python script.py
```

This will:

1. Fetch current alerts from Opsgenie.
 2. Create Salesforce cases for any alerts without an existing case.
 3. Tag the alert with the new Salesforce Case Number.
 4. Add notes with direct links to the Salesforce case and relevant documentation.
 5. Close Salesforce cases for resolved alerts.
-

Example Workflow

1. An alert is triggered in Opsgenie.
2. This script identifies that the alert is open and has no Salesforce case number tag.
3. A new Salesforce case is created and assigned to the alert recipient.
4. The Salesforce Case Number is tagged back in Opsgenie.

5. Links to the case and related documentation are added as alert notes.
 6. Once the alert is resolved, the associated Salesforce case is automatically closed.
-

Logging & Error Handling

- Basic `try/except` error handling and logging is implemented using the `logging` and `traceback` modules.
 - Errors during Salesforce updates or case creation will be logged to the console.
-

Documentation Links

The script adds predefined documentation notes based on the alert type:

- **Default:** [On-Call Alert Guide](#)
 - **Mass ENS Alerts:** [Mass ENS Related Alerts](#)
-

Notes

- Script assumes the Salesforce user can query and create cases and users.
 - Alert recipient must have a matching email in Salesforce to assign case ownership.
 - Only one tag (Salesforce Case Number) is used to determine whether a case already exists.
-

Future Improvements

- Environment variable support for credentials
 - Unit testing for core logic
 - Retry/backoff for API failures
 - Enhanced logging with timestamps and log levels
-

Contact

For questions, reach out to Nikole Kerns at 15nakerns@gmail.com