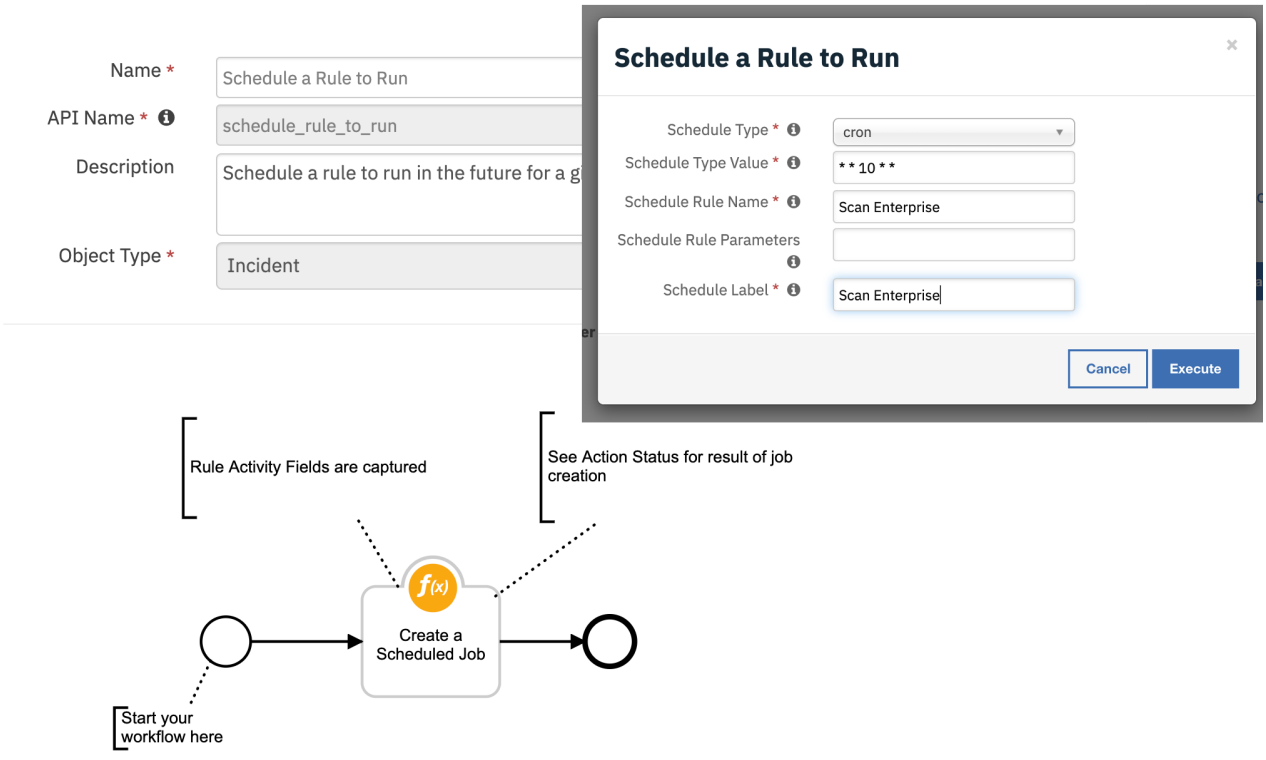
fn-scheduler Functions for IBM Resilient

Release Notes

Version	Date	Notes
1.0.3	Oct. 2020	Conditional PostgreSQL dependency
1.0.2	Sept. 2020	PostgreSQL support
1.0.1	May 2020	App Host support
1.0.0	Nov. 2019	Initial Release

Overview

Resilient Circuits Components for fn_scheduler



This package of functions allows an enterprise to schedule a rule to run in the future associated with a incident, task, artifact, and datatable. Times to run can be specified in the following ways:

1. cron (ex. * 0 * * * for every night at midnight)
2. interval (ex. 5h for every 5 hours)
3. date (ex. 2019/10/23 12:00:00)
4. delta (ex. 1h for one hour in the future)

Schedule rules using `cron` and `interval` are reocurring whereas `date` and `delta` are single event schedules. Scheduled rules are persisted so that restarts of resilient-circuits will resume already scheduled rules.

Functions available include:

1. Scheduling a rule
2. Listing scheduled rules
3. Removing a scheduled rule

Requirements

- Resilient platform >= `v33.0.5087`
- An Integration Server running `resilient_circuits>=30.0.0`
    - To set up an Integration Server see: https://ibm.biz/res-int-server-guide

---

Installation

App Host

All the components for running this integration in a container already exist when using the App Host app.

To install,

- Navigate to Administrative Settings and then the Apps tab.
- Click the Install button and select the downloaded file: app-fn_scheduler-x.x.x.zip.
- Go to the Configuration tab and edit the app.config file, editing the settings for Scheduler.

Config	Required	Example	Description
timezone	Yes	`utc`	*Specify the timezone (ex. America/New_York) which scheduled rules should follow.*
thread_max	Yes	`20`	*Number of threads which can run at the same. Typically, triggered rules run for a very short time to kick off a Resilient rule.*
datastore_dir	No	`/path/to/sqlite_folder`	*Specify a data path for the sqlite persistent datafile (ex. /path/to/scheduler.sqlite)*
db_url	No	postgresql+pypostgresql://res_test:res_test@192.168.1.215:5432/res_test	*Specify a PostgreSQL db to retain the schedules. Uncomment and remove the setting datastore_dir.*

Integration Server

- Download the `app-fn_scheduler.zip`.
- Copy the `.zip` to your Integration Server and SSH into it.
- **Unzip** the package:

```
$ unzip fn_scheduler-x.x.x.zip
```

- **Install** the package:

```
$ pip install fn_scheduler-x.x.x.tar.gz
```

- Import the **configurations** into your app.config file:

```
$ resilient-circuits config -u
```

- Import the fn_scheduler **customizations** into the Resilient platform:

```
$ resilient-circuits customize -y -l fn-scheduler
```

- Open the config file, scroll to the bottom and edit your fn_scheduler configurations:

```
$ nano ~/.resilient/app.config
```

Config	Required	Example	Description
timezone	Yes	`utc`	*Specify the timezone (ex. America/New_York) which scheduled rules should follow*
thread_max	Yes	`20`	*Number of threads which can run at the same. Typically, triggered rules run for a very short time to kick off a Resilient rule.*
datastore_dir	No	`/path/to/sqlite_folder`	*Specify a data path for the sqlite persistent datafile (ex. /path/to/scheduler.sqlite)*
db_url	No	postgresql+pypostgresql://res_test:res_test@192.168.1.215:5432/res_test	*Specify a postgres db to retain the schedules. Uncomment and remove the setting datastore_dir.**

- **Save** and **Close** the app.config file.
- [Optional]: Run selftest to test the Integration you configured:

```
$ resilient-circuits selftest -l fn-scheduler
```

- **Run** resilient-circuits or restart the Service on Windows/Linux:

```
$ resilient-circuits run
```

Upgrades to v1.0.3

If upgrading to v1.0.3, add the following comments and settings to your app.config [fn_scheduler] section:

```
# db url if using PostgreSQL DB. Use this with AppHost
#db_url=postgresql+pypostgresql://username:password@host:port/database
```

Use these settings to connect to a PostgreSQL db, rather than a SQLite db.

Uninstall

- SSH into your Integration Server.
- **Uninstall** the package:

```
$ pip uninstall fn-scheduler
```

- Open the config file, scroll to the [fn_scheduler] section and remove the section or prefix `#` to comment out the section.
- **Save** and **Close** the app.config file.

Troubleshooting

There are several ways to verify the successful operation of a function.

Resilient Action Status

- When viewing an incident, use the Actions menu to view **Action Status**.
- By default, pending and errors are displayed.
- Modify the filter for actions to also show Completed actions.
- Clicking on an action displays additional information on the progress made or what error occurred.

Resilient Scripting Log

- A separate log file is available to review scripting errors.
- This is useful when issues occur in the pre-processing or post-processing scripts.
- The default location for this log file is: `/var/log/resilient-scripting/resilient-scripting.log`.

Resilient Logs

- By default, Resilient logs are retained at `/usr/share/co3/logs`.
- The `client.log` may contain additional information regarding the execution of functions.

Resilient-Circuits

- The log is controlled in the `.resilient/app.config` file under the section [resilient] and the property `logdir`.
- The default file name is `app.log`.
- Each function will create progress information.
- Failures will show up as errors and may contain python trace statements.

Support

This is an IBM Supported app. Please search IBM Support for assistance. Also reference the Resilient Community for any discussion between customers and IBM.