Add New Incident Type

-This project lets you run a script that adds new functionality to the Resilient UI.

- This integration

* A component for a “resilient circuits” framework
* A fragment of a configuration file

# Project Overview

The project runs as a standalone Python application. When this application starts, it connects to the Resilient platform and starts listening for actions. If the add\_new\_incident action is invoked then the application locates the script and runs it. When the script is run a new functionality is added to the Resilient UI. After running the script, the user will be able to create a new incident type and have it be saved to the incident from the New Incident page, rather then Customization Settings.

## Installation

Install the project onto a Windows or Unix machine. This machine requires

* Python version 2.7.6 or later
* Network access to the Resilient appliance via ports 443 and 65001
* Resilient Systems application version 24 or later, with Action Module.

## Python Libraries

You must install several Python libraries that are required by the application.

First, install the “co3” module, which is the Resilient Systems REST API client library for Python; and the “resilient\_circuits” module, which is the Resilient action module application framework for Python. You must also install the Add\_Inc library.

Follow the installation instructions here:

<https://github.com/Co3Systems/co3-api/tree/master/python>

## Basic Configuration

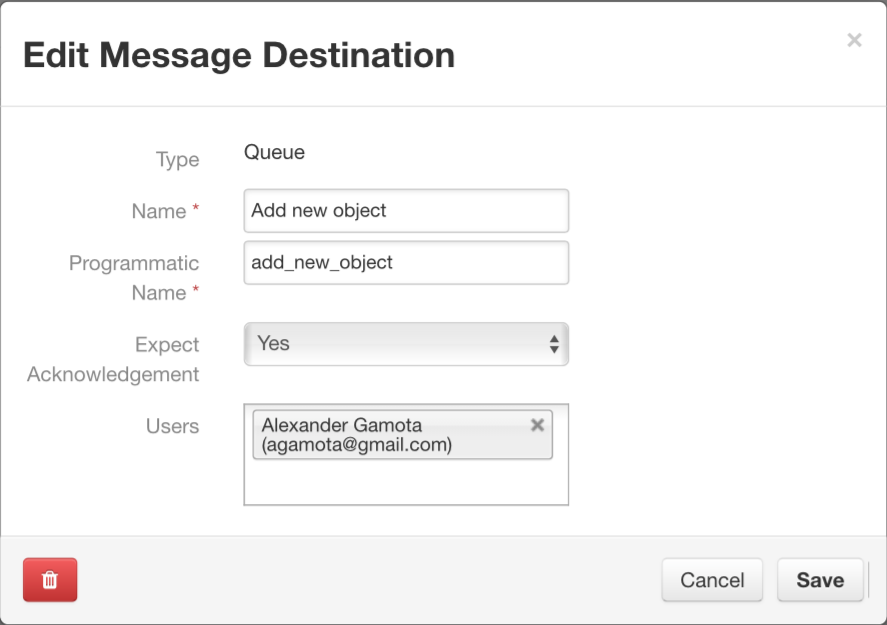
First, edit the “app.config” file and edit the values needed to connect to your Resilient server. You only need to edit: hostname, email, password, & org values.

### UI Configuration

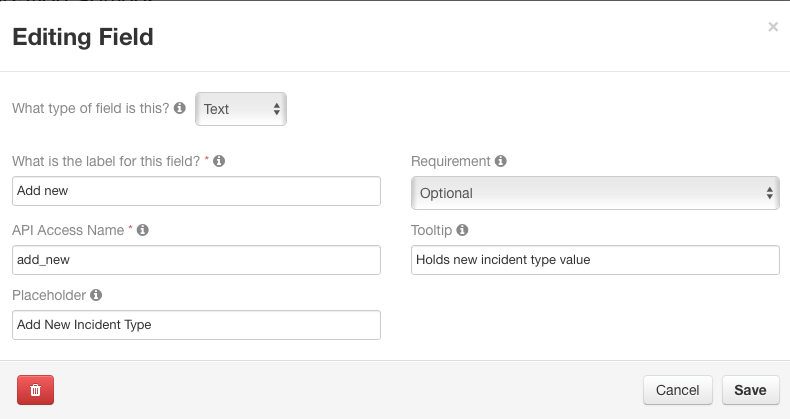
To run the auto\_add\_new\_inc\_types script to Resilient you must configure the UI as follows:

In Resilient, open Customization Settings 🡪 Actions 🡪 Message Destinations.

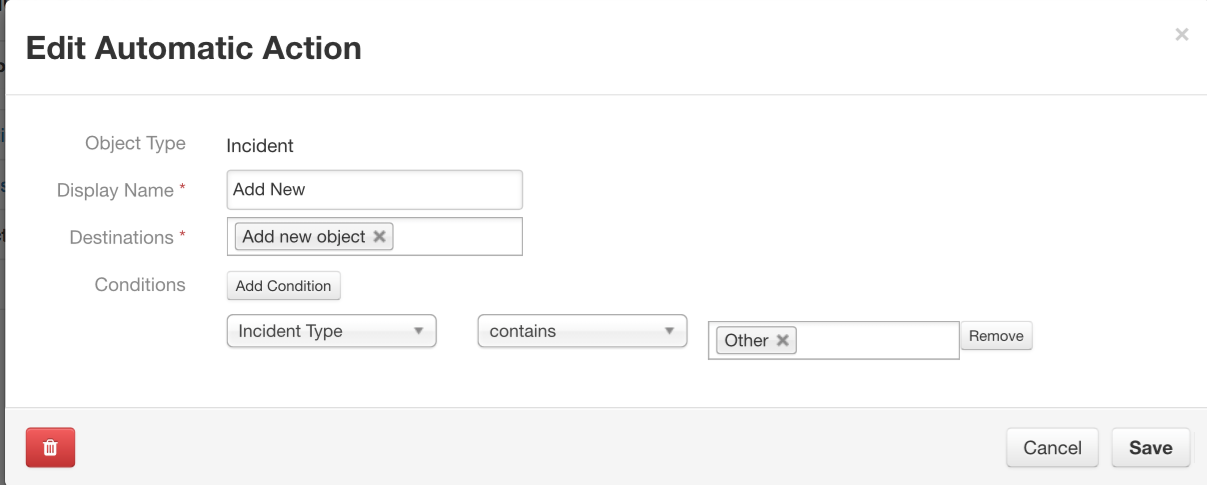
Click on Message Destinations and copy the fields below except Users. Enter your API user account to the Users list for this message destination:



Next open 🡪 Action Fields. Enter the information as below. Tooltip and Placeholder don’t have to be the same as below.



Next open 🡪 Automatic Actions. Enter the information as listed below.



## Running the Application

Follow the configuration instructions from Basic Configuration to set appropriate values into the “app.config” file.

Add the auto\_add\_new\_inc\_type.py file to the components folder.

Access the shell runner directory.

Then append the app.config.fragment to the app.config file by using:

cat app.config.fragment >> app.config

Then, start the application, using:

python run.py

You might be asked to enter your email and server:

python run.py --email [example@site.com](mailto:example@site.com) --host hostname

If ran successfully you should expect the following code:

INFO app.py Configuration file is app.config

INFO connectionpool.py Starting new HTTPS connection (1): resilient

INFO app.py App Started

INFO actions\_component.py STOMP connected

INFO actions\_component.py Component <Shell/actions.shell 51251:MainThread (queued=0) [S]> registered to actions.shell

INFO actions\_component.py Subscribe to 'shell'

INFO component\_loader.py Loaded component 'auto\_add\_new\_inc\_type.py'

INFO app.py Components loaded

On Windows, the application can be configured to run as a service; instructions are not included here, but are available on request.

# Configuration

The default behavior above only requires that you

* Configure the action, using the administration UI. (Listed under UI configuration)
* Deploy the auto\_add\_new\_inc\_type.py file to the components file of the repository.

The “[framework\_section]” section of the config file has:

# ---------------------------------------------------------------------------

# The 'framework\_section' action component

# ---------------------------------------------------------------------------

[framework\_section]

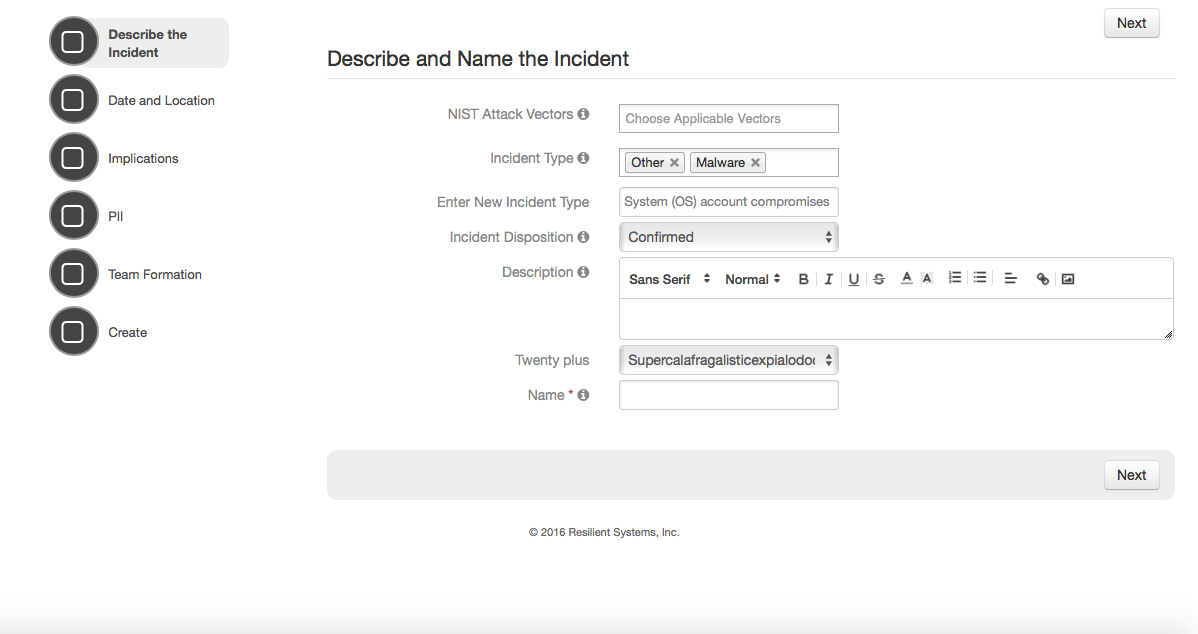
queue=add\_new\_object

This creates the default behavior described above:

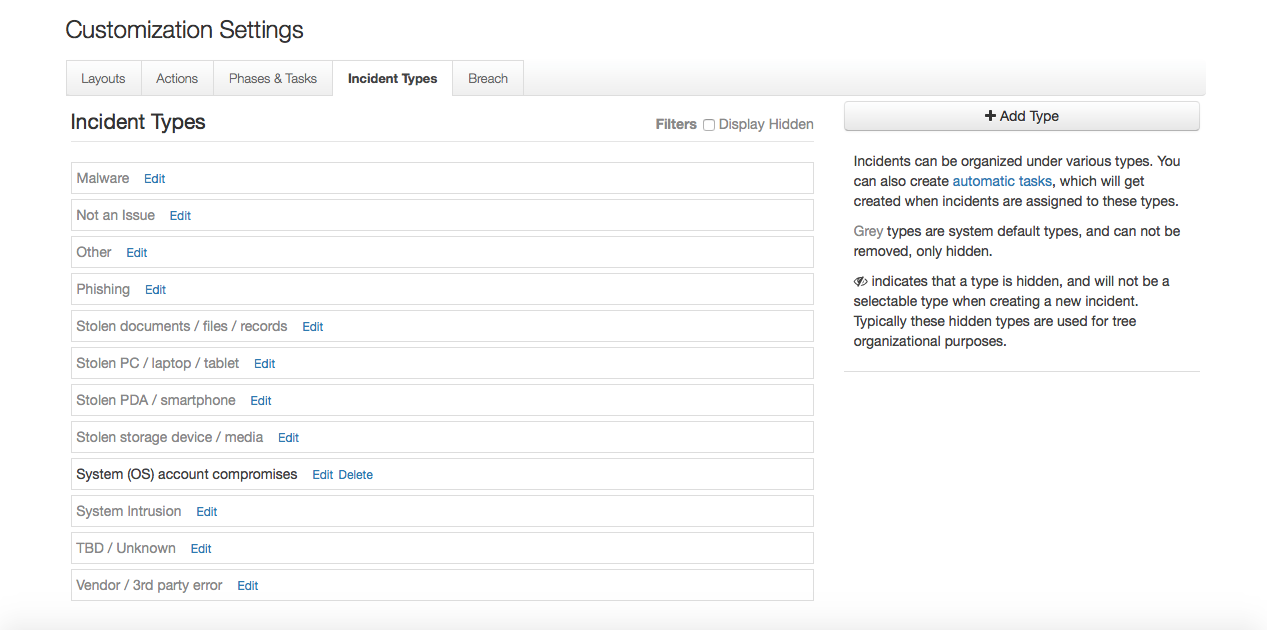
* Listens to actions on the message destination (queue) named “add\_new\_object”
* For that action, the script “auto\_add\_new\_inc\_type” is executed, passing the artifact value as a command- line parameter

Examples:

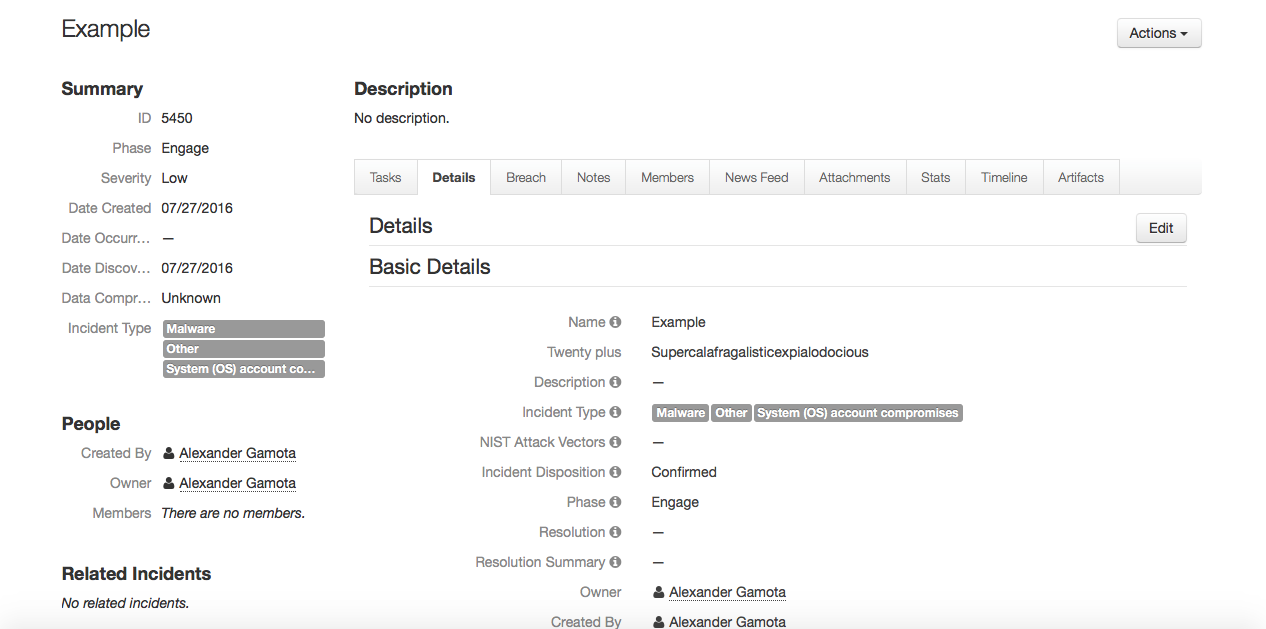
If configured successfully, you will get functionality that resembles the below images:



\*When other is selected the Enter New Incident Type Window opens prompting entry.



\*After the incident is created, the ‘System (OS) account compromises’ type is saved under Incident Types



\*The ‘System (OS) account compromises’ type is saved under the incidents Incident Type field.