Atlassian Jira Functions for Resilient

Table of Contents

Release Notes

v2.0.0

- Added App Host support
- Added proxy support
- Added support for https://pypi.org/project/jira/
- Changed config heading from `jira` to `fn_jira`
- Added configs: `timeout`, `auth_method`, `http_proxy` and `https_proxy`
- Added incident field `jira_issue_id`
- Changed column name in `jira_task_references` Data Table from `jira_api_url` to `jira_issue_id_col`

v1.0.2

- Improvements to data table handling
- Bug fixes

v1.0.1

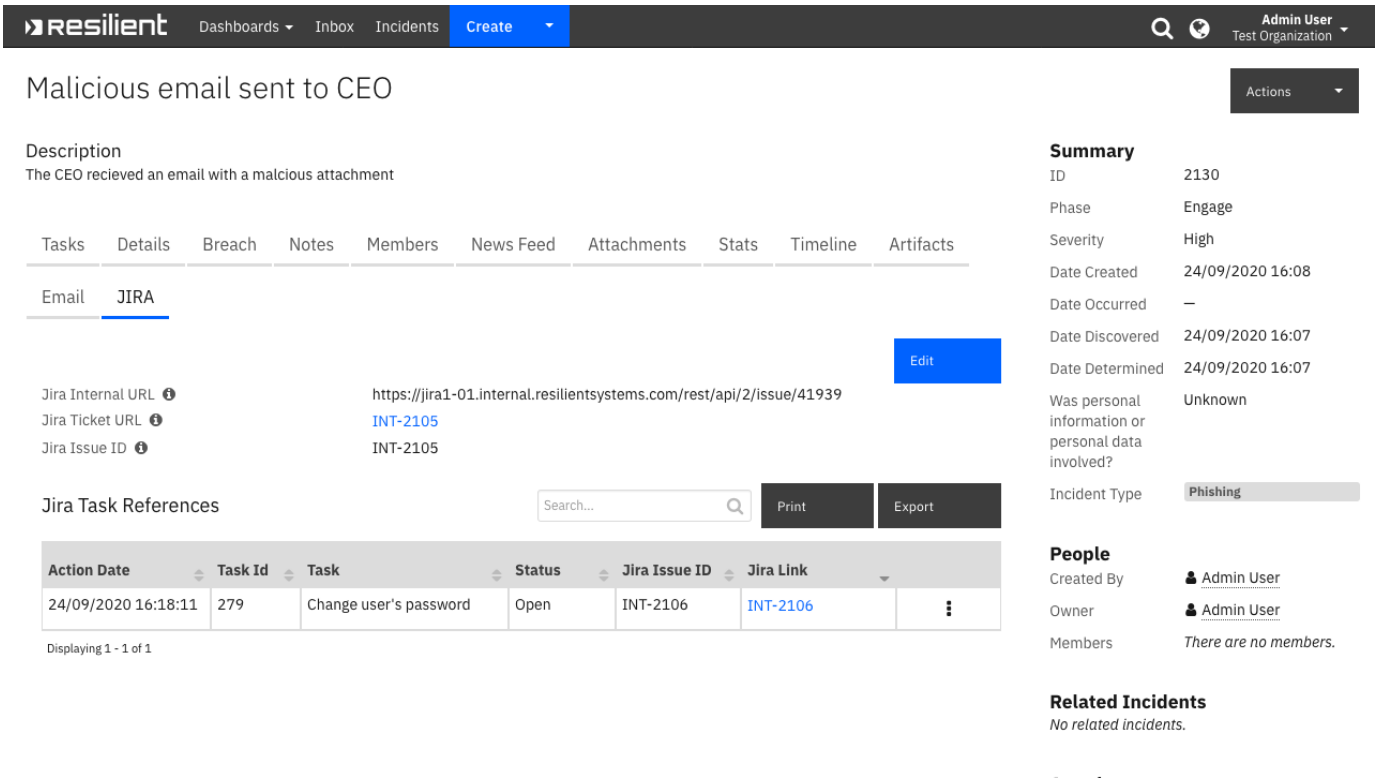
- Support for versions of Resilient 31.0 and beyond

v1.0.0

- Initial Release

Overview

Provides integration with JIRA for Issue Creation, Issue Transition and Comment Creation



This app allows for the tracking of Resilient Incidents and Tasks as Jira Issues. Bidirectional links are saved to allow for easy navigation between the applications.

It also allows for the transitioning of Jira issues when the corresponding incident is closed and adds comments to the Jira issue when a Note is created in Resilient.

Example rules and workflows can used used or modified to meet your business processes.

Key Features

- Issue creation
- Issue transition
- Comment creation

Installation

Requirements

- Resilient platform >= `v35.0.0`
- App Host >= `v1.2.132` (if using App Host)
  - To setup up an App Host see: ibm.biz/res-app-host-setup
- An Integration Server running `resilient_circuits>=32.0.0` (if using an Integration Server)
  - To set up an Integration Server see: ibm.biz/res-int-server-guide
  - If using an API key account, minimum required permissions are:

Name	Permissions
Org Data	Read
Function	Read

- Proxy supported: Yes

Install

- To install or uninstall an App using the App Host see ibm.biz/res-install-app

- To install or uninstall an Integration using the Integration Server see the ibm.biz/res-install-int

---

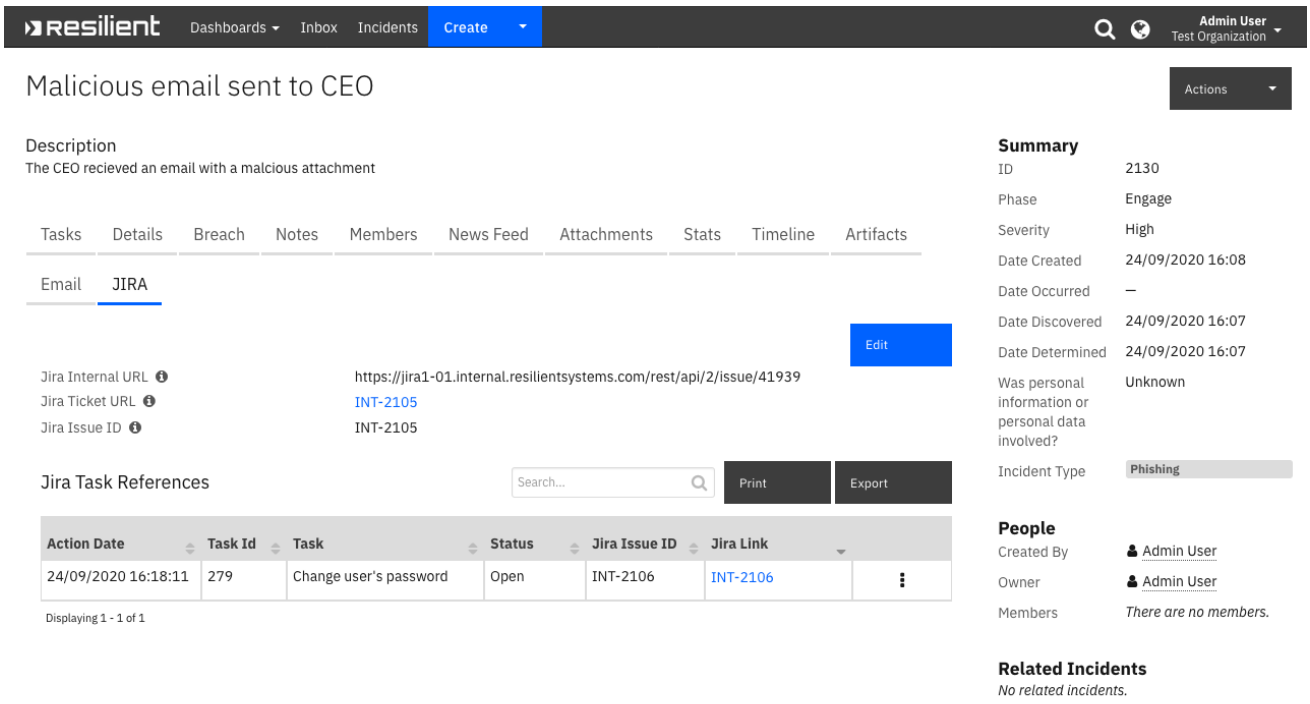
App Configuration

The following table describes the settings you need to configure in the app.config file. If using App Host, see the Resilient System Administrator Guide. If using the integration server, see the Integration Server Guide.

Config	Required	Example	Description
url	Yes	`https://<jira url>`	The URL of your Jira platform
auth_method	Yes	`AUTH`	The method of authentication to use when connecting to your Jira platform. Supported methods are `AUTH` and `BASIC`. For more information on authentication see: https://jira.readthedocs.io/en/latest/examples.html#authentication
user	Yes	`<jira user>`	The username of the Jira account to use with this integration. They must be a user on the Jira platform with the correct permissions
password	Yes	`<jira user password>`	The password for the Jira account to use with this integration
timeout	No	`10`	The number of seconds to timeout after when making a request to the Jira platform
verify_cert	No	`True`	A boolean value. Set to `True` if you want ti verify SSL certificates on each request
http_proxy	No	`http://localhost:3128`	Your HTTP Proxy
https_proxy	No	`https://localhost:3128`	Your HTTPS Proxy

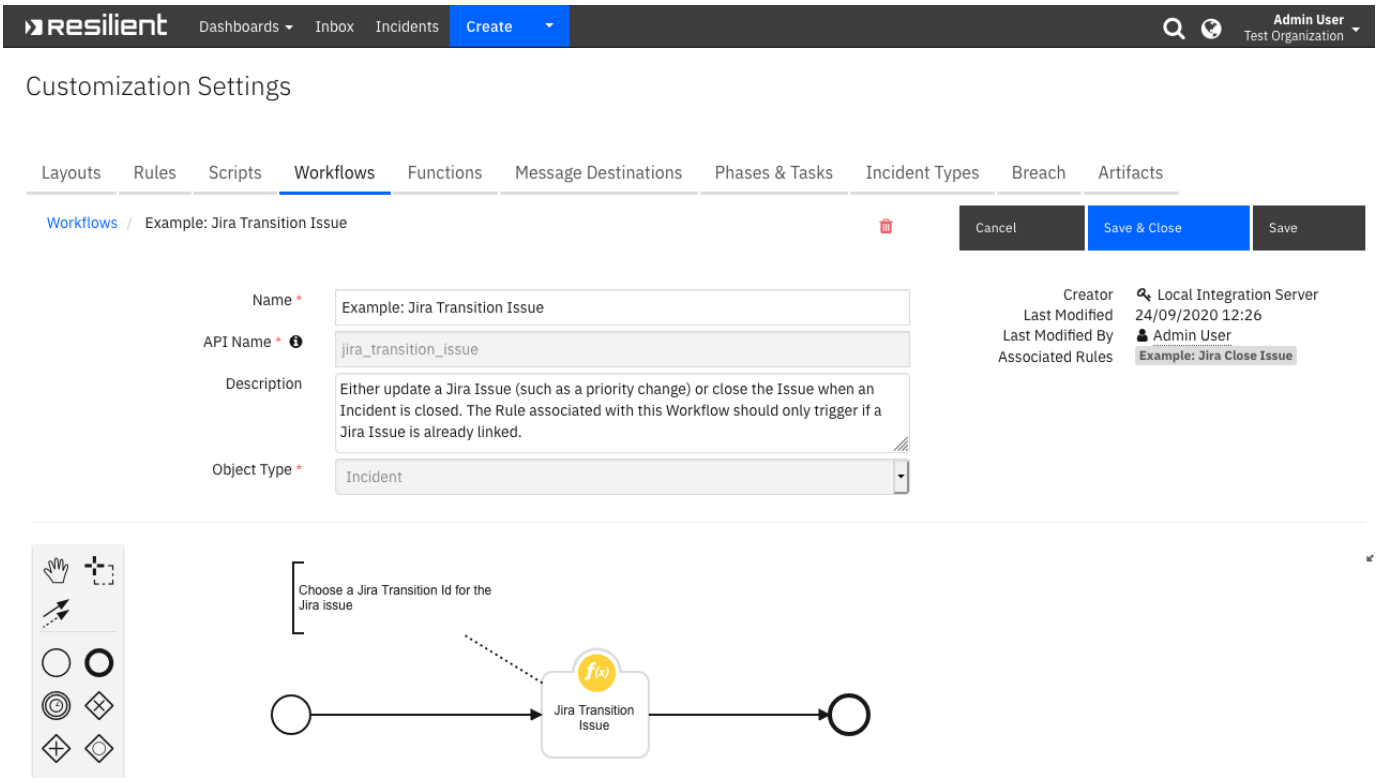
Custom Layouts

- Import the Data Tables and Custom Fields like the screenshot below:



Function - Jira Transition Issue

Transition a Jira issue. This can be used when a Resilient Incident is closed or to change the Jira Issue's workflow state. See example workflow for configuration of function pre-processor and post-processor scripts



▶ Inputs:

Name	Type	Required	Example	Tooltip
jira_comment	text	No	"Updated in IBM Resilient"	The comment to add to the issue in Jira
jira_fields	text	No	-	A JSON String of the fields to set in Jira
jira_issue_id	text	Yes	JRA-1000	The ID of the issue in Jira. Also known as the issue key. E.g: "JRA-1330"
jira_transition_id	text	Yes	11	The ID to transition the Jira issue to. More information can be found in the Jira Documentation on transition_id

▶ Outputs:

```
results = {
    'version': '1.0',
    'success': True,
    'reason': None,
    'content': 'Done',
    'raw': '"Done"',
    'inputs': {
        'jira_issue_id': 'INT-2106',
        'jira_transition_id': 'Close',
        'jira_fields': '{ "resolution":{ "name":"Done" } }',
        'jira_comment': 'Closed in IBM Resilient\n\nResolution: Done\n'
    },
    'metrics': {
        'version': '1.0',
        'package': 'fn-jira',
        'package_version': '2.0.0',
```

```
        'host': 'example',
        'execution_time_ms': 1357,
        'timestamp': '2020-09-24 16:27:09'
    }
}
```

▶ Workflows

▶ Example Pre-Process Script:

```
# Example: Jira Transition Issue pre-processing script

########################################
### Define pre-processing functions ###
########################################
def dict_to_json_str(d):
"""Function that converts a dictionary into a JSON string.
   Supports types: basestring, unicode, bool, int and nested dicts.
   Does not support lists.
   If the value is None, it sets it to False."""

json_entry = u'"{0}":{1}'
json_entry_str = u'"{0}":"{1}"'
entries = []

for entry in d:
  key = entry
  value = d[entry]

  if value is None:
    value = False

  if isinstance(value, list):
    helper.fail('dict_to_json_str does not support Python Lists')

  if isinstance(value, basestring):
    value = value.replace(u'"', u'\\"')
    entries.append(json_entry_str.format(unicode(key), unicode(value)))

  elif isinstance(value, unicode):
    entries.append(json_entry.format(unicode(key), unicode(value)))

  elif isinstance(value, bool):
    value = 'true' if value == True else 'false'
    entries.append(json_entry.format(key, value))

  elif isinstance(value, int):
    entries.append(json_entry.format(unicode(key), value))

  elif isinstance(value, dict):
    entries.append(json_entry.format(key, dict_to_json_str(value)))

  else:
    helper.fail('dict_to_json_str does not support this type: {0}'.format(type(value)))

return u'{0} {1} {2}'.format(u'{', ','.join(entries), u'}')


######################
### Define Inputs ###
######################
```

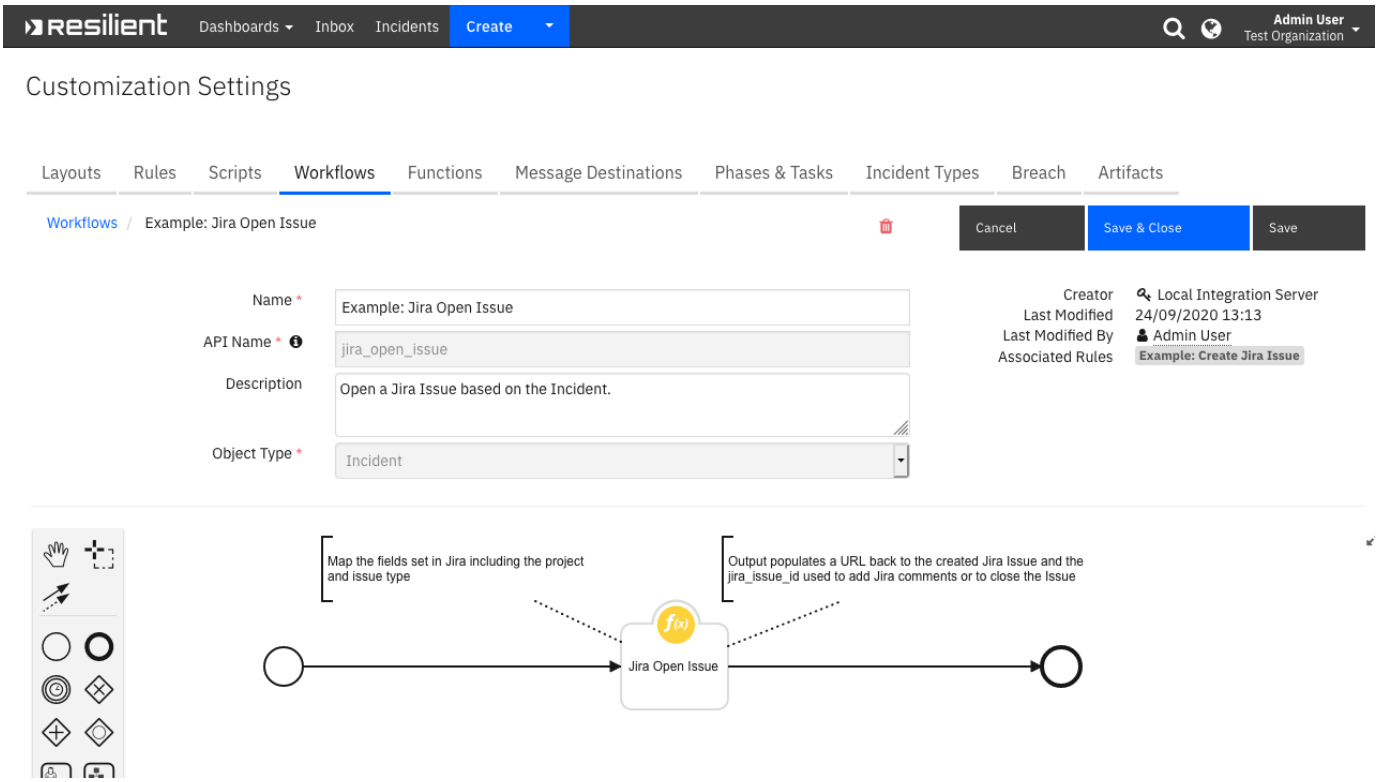
```
inputs.jira_issue_id = incident.properties.jira_issue_id
inputs.jira_transition_id = "Close"
inputs.jira_comment = u"Closed in IBM Resilient\n\nResolution:
{0}\n{1}".format(incident.resolution_id, incident.resolution_summary.content)

resolution_map = { "unresolved": "Obsolete", "duplicate": "Duplicate", "not an issue":
"Won't Do", "resolved": "Done" }

# Define JIRA fields here
inputs.jira_fields = dict_to_json_str({
"resolution": { "name": resolution_map.get(str(incident.resolution_id).lower(), "Done")
}
})
```

▶ Example Post-Process Script:

```
from java.util import Date
time_now = Date().time

if results.success:
row.date = time_now
row.status = "Closed"
```

Function - Jira Open Issue

Create a jira issue. To be used when a Resilient Incident is created. See example workflow for configuration of function pre-processor and post-processor scripts



NOTE: Some fields in Jira are required to open a new Issue. Ensure you add those fields in the pre-process script to `jira_fields` when creating a new Jira Issue

▶ Inputs:

Name	Type	Required	Example	Tooltip
incident_id	number	Yes	-	-
jira_fields	text	No	-	A JSON String of the fields to set in Jira
task_id	number	No	-	-

▶ Outputs:

```
results = {
    'version': '1.0',
    'success': True,
    'reason': None,
    'content': {
        'issue_url': 'https://jira1-01.example.com/browse/INT-2105',
        'issue_url_internal': 'https://jira1-01.example.com/rest/api/2/issue/41939',
        'issue_key': 'INT-2105',
        'issue': {
            'expand':
'renderedFields,names,schema,operations,editmeta,changelog,versionedRepresentations',
            'id': '41939',
            'self': 'https://jira1-01.example.com/rest/api/2/issue/41939',
            'key': 'INT-2105',
            'fields': {
                'issuetype': {
                    'self': 'https://jira1-01.example.com/rest/api/2/issuetype/10001',
                    'id': '10001',
                    'description': 'Created by Jira Software - do not edit or delete.
Issue type for a user story.',
                    'iconUrl': 'https://jira1-
01.example.com/images/icons/issuetypes/story.svg',
                    'name': 'Story',
                    'subtask': False
                },
                'timespent': None,
                'project': {
                    'self': 'https://jira1-01.example.com/rest/api/2/project/10101',
                    'id': '10101',
                    'key': 'INT',
                    'name': 'Example',
                    'projectTypeKey': 'software',
                    'avatarUrls': {
                        '48x48': 'https://jira1-01.example.com/secure/projectavatar?
avatarId=10324',
                        '24x24': 'https://jira1-01.example.com/secure/projectavatar?
size=small&avatarId=10324',
                        '16x16': 'https://jira1-01.example.com/secure/projectavatar?
size=xsmall&avatarId=10324',
                        '32x32': 'https://jira1-01.example.com/secure/projectavatar?
size=medium&avatarId=10324'
                    }
                },
                'fixVersions': [],
                'aggregatetimespent': None,
                'resolution': None,
                'customfield_10901': {
                    'self': 'https://jira1-
01.example.com/rest/api/2/customFieldOption/10807',
                    'value': 'No',
```

```
                    'id': '10807'
                },
                'resolutiondate': None,
                'lastViewed': None,
                'watches': {
                    'self': 'https://jira1-01.example.com/rest/api/2/issue/INT-
2105/watchers',
                    'watchCount': 1,
                    'isWatching': True
                },
                'created': '2020-09-24T15:10:06.296+0000',
                'customfield_10220': None,
                'priority': {
                    'self': 'https://jira1-01.example.com/rest/api/2/priority/2',
                    'iconUrl': 'https://jira1-
01.example.com/secure/attachment/14751/high.svg',
                    'name': 'High',
                    'id': '2'
                },
                'labels': [],
                'timeestimate': None,
                'aggregatetimeoriginalestimate': None,
                'versions': [],
                'customfield_10219': None,
                'issuelinks': [],
                'assignee': None,
                'updated': '2020-09-24T15:10:06.296+0000',
                'status': {
                    'self': 'https://jira1-01.example.com/rest/api/2/status/1',
                    'description': 'The issue is open and ready for the assignee to
start work on it.',
                    'iconUrl': 'https://jira1-
01.example.com/images/icons/statuses/open.png',
                    'name': 'Open',
                    'id': '1',
                    'statusCategory': {
                        'self': 'https://jira1-
01.example.com/rest/api/2/statuscategory/2',
                        'id': 2,
                        'key': 'new',
                        'colorName': 'blue-gray',
                        'name': 'To Do'
                    }
                },
                'components': [],
                'timeoriginalestimate': None,
                'description': 'IBM Resilient Link:
https://example.ibm.com:443/#incidents/2130\n\nThe CEO recieved an email with a malcious
attachment',
                'timetracking': {},
                'attachment': [],
                'aggregatetimeestimate': None,
                'summary': 'IBM Resilient: Malicious email sent to CEO',
                'creator': {
                    'self': 'https://jira1-01.example.com/rest/api/2/user?
username=example',
                    'name': 'example',
                    'key': 'example',
                    'emailAddress': 'example@ibm.com',
                    'avatarUrls': {
                        '48x48': 'https://jira1-01.example.com/secure/useravatar?
ownerId=example&avatarId=10713',
```

```
                        '24x24': 'https://jira1-01.example.com/secure/useravatar?
size=small&ownerId=example&avatarId=10713',
                        '16x16': 'https://jira1-01.example.com/secure/useravatar?
size=xsmall&ownerId=example&avatarId=10713',
                        '32x32': 'https://jira1-01.example.com/secure/useravatar?
size=medium&ownerId=example&avatarId=10713'
                    },
                    'displayName': 'example',
                    'active': True,
                    'timeZone': 'UTC'
                },
                'subtasks': [],
                'reporter': {
                    'self': 'https://jira1-01.example.com/rest/api/2/user?
username=example',
                    'name': 'example',
                    'key': 'example',
                    'emailAddress': 'example@ibm.com',
                    'avatarUrls': {
                        '48x48': 'https://jira1-01.example.com/secure/useravatar?
ownerId=example&avatarId=10713',
                        '24x24': 'https://jira1-01.example.com/secure/useravatar?
size=small&ownerId=example&avatarId=10713',
                        '16x16': 'https://jira1-01.example.com/secure/useravatar?
size=xsmall&ownerId=example&avatarId=10713',
                        '32x32': 'https://jira1-01.example.com/secure/useravatar?
size=medium&ownerId=example&avatarId=10713'
                    },
                    'displayName': 'example',
                    'active': True,
                    'timeZone': 'UTC'
                },
                'aggregateprogress': {
                    'progress': 0,
                    'total': 0
                },
                'environment': None,
                'duedate': None,
                'progress': {
                    'progress': 0,
                    'total': 0
                },
                'comment': {
                    'comments': [],
                    'maxResults': 0,
                    'total': 0,
                    'startAt': 0
                },
                'votes': {
                    'self': 'https://jira1-01.example.com/rest/api/2/issue/INT-
2105/votes',
                    'votes': 0,
                    'hasVoted': False
                },
                'worklog': {
                    'startAt': 0,
                    'maxResults': 20,
                    'total': 0,
                    'worklogs': []
                }
            }
        }
```

```
    },
    'raw': '{"issue_url": "https://jira1-01.example.com/browse/INT-2105",
"issue_url_internal": "https://jira1-01.example.com/rest/api/2/issue/41939",
"issue_key": "INT-2105", "issue": ...
    'inputs': {
        'incident_id': 2130,
        'jira_fields': '{ "summary":"IBM Resilient: Malicious email sent to
CEO","issuetype":"Story","project":"INT","description":"<div class=\\"rte\\"><div>The
CEO recieved an email with a malcious attachment</div></div>","priority":{ "name":"High"
} }'
    },
    'metrics': {
        'version': '1.0',
        'package': 'fn-jira',
        'package_version': '2.0.0',
        'host': 'example',
        'execution_time_ms': 2016,
        'timestamp': '2020-09-24 16:09:51'
    }
}
```

▶ Workflows

▶ Example Pre-Process Script:

```
# Example: Jira Open Issue [Incident] pre-processing script

########################################
### Define pre-processing functions ###
########################################
def list_to_json_str(l):
"""
Function that converts a list into a JSON string.
Supports types: basestring, unicode, bool, int, list and dicts.
If the value is None, it sets it to False.
"""
list_as_str = ''
json_entry = u'{0},'
json_entry_str = u'"{0}",'

for value in l:

  if value is None:
    value = False

  if isinstance(value, list):
    list_as_str += json_entry.format(list_to_json_str(value))

  elif isinstance(value, dict):
    list_as_str += json_entry.format(dict_to_json_str(value))

  elif isinstance(value, basestring):
    value = value.replace(u'"', u'\\"')
    value = value.replace("\n", "\\n")
    list_as_str += json_entry_str.format(unicode(value))

  elif isinstance(value, unicode):
    list_as_str += json_entry.format(unicode(value))

  elif isinstance(value, bool):
    value = 'true' if value is True else 'false'
```

```
      list_as_str += json_entry.format(value)

    elif isinstance(value, int):
      list_as_str += json_entry.format(value)

    else:
      helper.fail('list_to_json_str does not support this type: {0}'.format(type(value)))

  return u'{0} {1} {2}'.format(u'[', list_as_str[:-1], u']')

  def dict_to_json_str(d):
  """
  Function that converts a dictionary into a JSON string.
  Supports types: basestring, unicode, bool, int, list and nested dicts.
  If the value is None, it sets it to False.
  """

  json_entry = u'"{0}":{1}'
  json_entry_str = u'"{0}":"{1}"'
  entries = []

  for entry in d:
    key = entry
    value = d[entry]

    if value is None:
      value = False

    if isinstance(value, list):
      entries.append(json_entry.format(unicode(key), list_to_json_str(value)))

    elif isinstance(value, dict):
      entries.append(json_entry.format(key, dict_to_json_str(value)))

    elif isinstance(value, basestring):
      value = value.replace(u'"', u'\\"')
      value = value.replace("\n", "\\n")
      entries.append(json_entry_str.format(unicode(key), unicode(value)))

    elif isinstance(value, unicode):
      entries.append(json_entry.format(unicode(key), unicode(value)))

    elif isinstance(value, bool):
      value = 'true' if value is True else 'false'
      entries.append(json_entry.format(key, value))

    elif isinstance(value, int):
      entries.append(json_entry.format(unicode(key), value))

    else:
      helper.fail('dict_to_json_str does not support this type: {0}'.format(type(value)))

  return u'{0} {1} {2}'.format(u'{', ','.join(entries), u'}')

  ######################
  ### Define Inputs ###
  ######################

  # ID of this incident
  inputs.incident_id = incident.id

  # A map for JIRA priorities
```

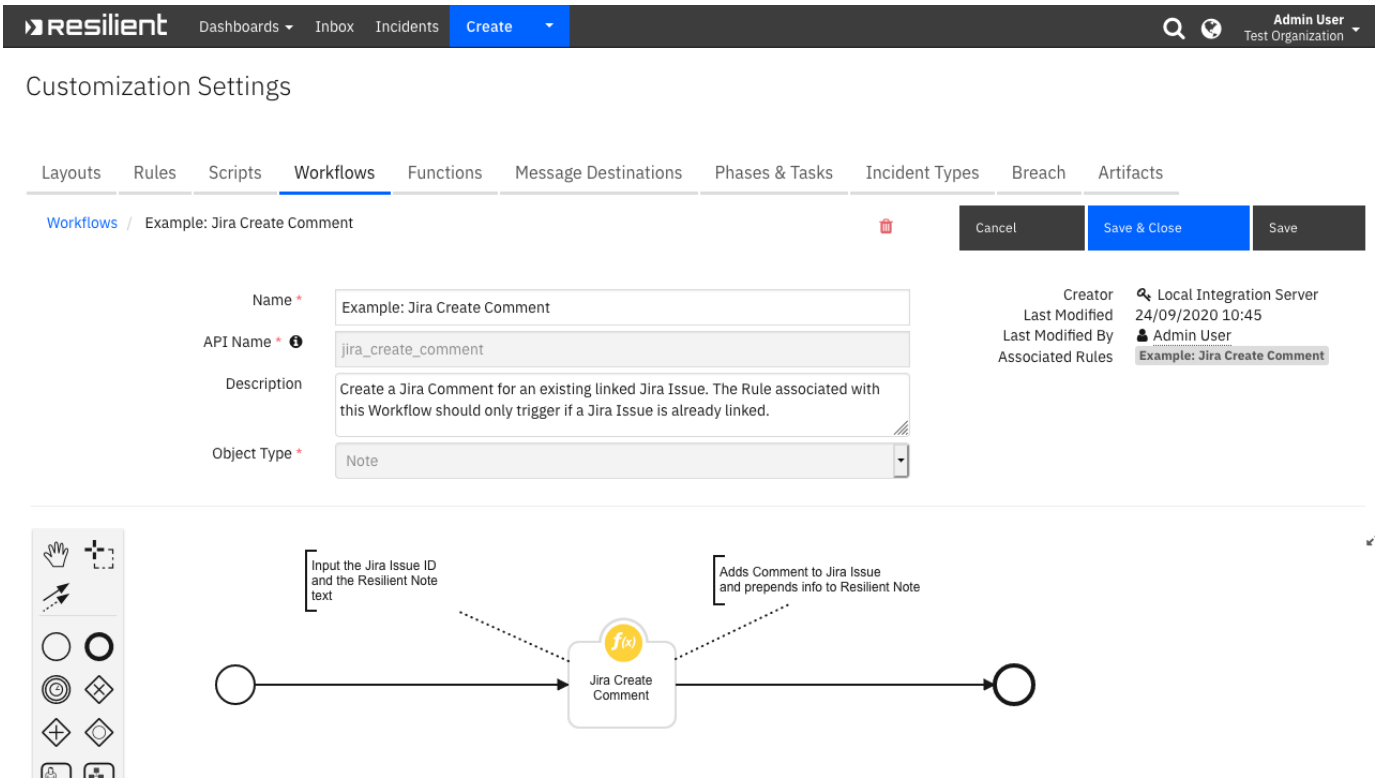
```
priority_map = { "Low": {"name": "Low"}, "Medium": {"name": "Medium"}, "High": {"name":
"High"} }
jira_priority = priority_map.get(incident.severity_code, {"name": "Low"})

# Define JIRA fields here
inputs.jira_fields = dict_to_json_str({
"project": "INT",
"issuetype": rule.properties.jira_issue_type,
"priority": jira_priority,
"summary": u"IBM Resilient: {0}".format(incident.name),
"description": incident.description.content if incident.get("description") else "Created
in IBM Resilient"
})
```

▶ Example Post-Process Script:

```
if results.success:
results_content = results.get("content", {})
incident.properties.jira_url = "<a href='{}' target='blank'>{}
</a>".format(results_content.get("issue_url"), results_content.get("issue_key"))
incident.properties.jira_internal_url = results_content.get("issue_url_internal")
incident.properties.jira_issue_id = results_content.get("issue_key")
```

Function - Jira Create Comment

Create a Jira comment. To be used when a Resilient Note is created. See example workflow for configuration of function pre-processor and post-processor scripts



▶ Inputs:

Name	Type	Required	Example	Tooltip

Name	Type	Required	Example	Tooltip
jira_comment	text	No	"Updated in IBM Resilient"	The comment to add to the issue in Jira
jira_issue_id	text	Yes	JRA-1000	The ID of the issue in Jira. Also known as the issue key. E.g: "JRA-1330"

▶ Outputs:


```
results = {
    'version': '1.0',
    'success': True,
    'reason': None,
    'content': {
        'self': 'https://jira1-01.example.com/rest/api/2/issue/41939/comment/53123',
        'id': '53123',
        'author': {
            'self': 'https://jira1-01.example.com/rest/api/2/user?username=.example',
            'name': 'example',
            'key': 'example',
            'emailAddress': 'example@ibm.com',
            'avatarUrls': {
                '48x48': 'https://jira1-01.example.com/secure/useravatar?
ownerId=.example&avatarId=10713',
                '24x24': 'https://jira1-01.example.com/secure/useravatar?
size=small&ownerId=.example&avatarId=10713',
                '16x16': 'https://jira1-01.example.com/secure/useravatar?
size=xsmall&ownerId=.example&avatarId=10713',
                '32x32': 'https://jira1-01.example.com/secure/useravatar?
size=medium&ownerId=.example&avatarId=10713'
            },
            'displayName': 'example',
            'active': True,
            'timeZone': 'UTC'
        },
        'body': 'Please note that the CEO is travelling',
        'updateAuthor': {
            'self': 'https://jira1-01.example.com/rest/api/2/user?username=.example',
            'name': 'example',
            'key': 'example',
            'emailAddress': 'example@ibm.com',
            'avatarUrls': {
                '48x48': 'https://jira1-01.example.com/secure/useravatar?
ownerId=.example&avatarId=10713',
                '24x24': 'https://jira1-01.example.com/secure/useravatar?
size=small&ownerId=.example&avatarId=10713',
                '16x16': 'https://jira1-01.example.com/secure/useravatar?
size=xsmall&ownerId=.example&avatarId=10713',
                '32x32': 'https://jira1-01.example.com/secure/useravatar?
size=medium&ownerId=.example&avatarId=10713'
            },
            'displayName': 'example',
            'active': True,
            'timeZone': 'UTC'
        },
        'created': '2020-09-24T15:23:12.870+0000',
        'updated': '2020-09-24T15:23:12.870+0000'
    },
    'raw': '{"self": "https://jira1-
01.example.com/rest/api/2/issue/41939/comment/53123", "id": "53123", "author": ...
```

```
    'inputs': {
        'jira_issue_id': 'INT-2105',
        'jira_comment': '<div class="rte"><div>Please note that the CEO is
travelling</div></div>'
    },
    'metrics': {
        'version': '1.0',
        'package': 'fn-jira',
        'package_version': '2.0.0',
        'host': 'example',
        'execution_time_ms': 1150,
        'timestamp': '2020-09-24 16:22:57'
    }
}
```

▶ Workflows

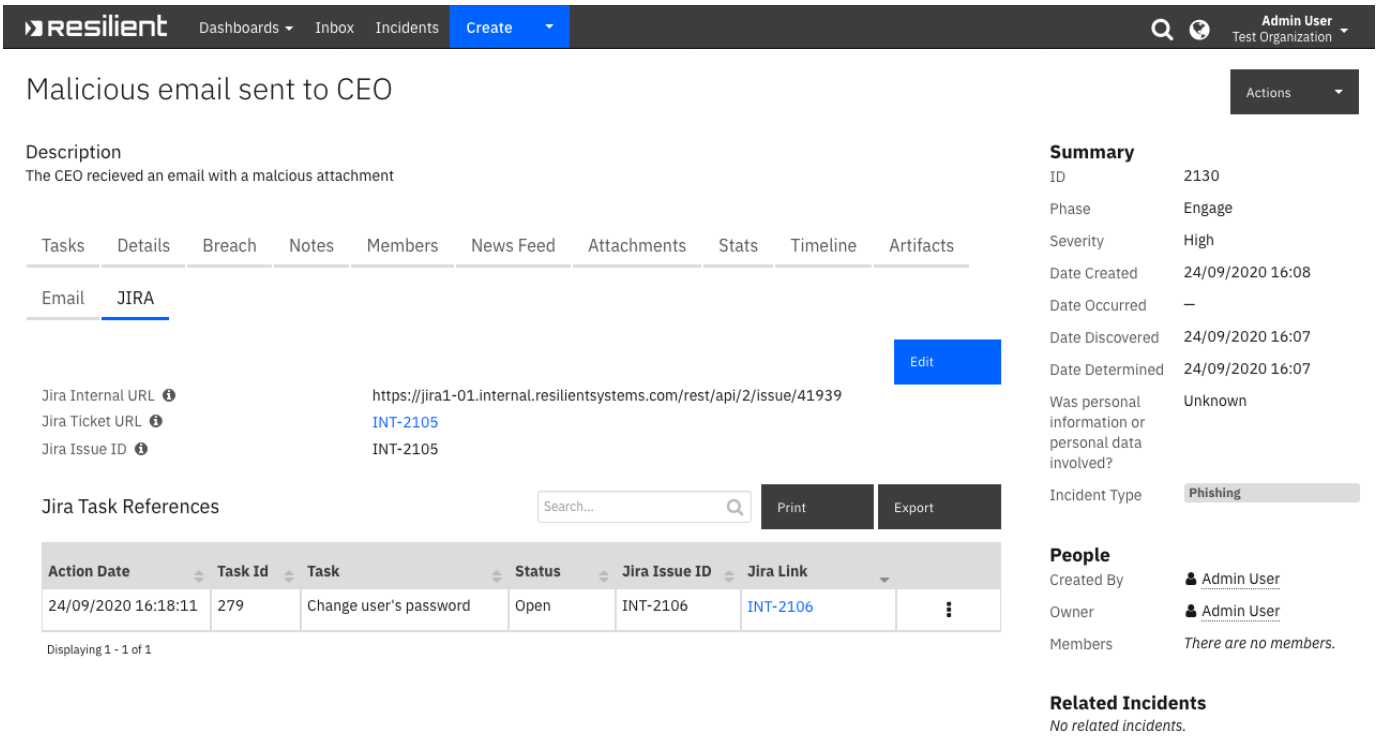
▶ Example Pre-Process Script:

```
# Example: Jira Create Comment pre-processing script

inputs.jira_issue_id = incident.properties.jira_issue_id
inputs.jira_comment = note.text.content
```

▶ Example Post-Process Script:

```
# Example: Jira Create Comment post-process script

# Import Date
from java.util import Date

# Get the current time
dt_now = Date()

issue_id = results.get("inputs", {}).get("jira_issue_id")

# Prepend message and time to the note
note.text = u"<b>Sent to the Jira issue {0} at {1}</b><br>{2}".format(issue_id, dt_now,
unicode(note.text.content))
```

Data Table - Jira Task References



API Name:

jira_task_references

Columns:

Column Name	API Access Name	Type	Tooltip
Action Date	`date`	`datetimepicker`	-
Jira Issue ID	`jira_issue_id_col`	`text`	-
Jira Link	`jira_link`	`textarea`	-
Status	`status`	`text`	-
Task	`task`	`textarea`	-
Task Id	`task_id`	`text`	-

Custom Fields

Label	API Access Name	Type	Prefix	Placeholder	Tooltip
Jira Ticket URL	`jira_url`	`textarea`	`properties`	-	Contains URL back to the Jira issue created via the UI
Jira Internal URL	`jira_internal_url`	`text`	`properties`	-	The REST API URL
Jira Issue ID	`jira_issue_id`	`text`	`properties`	JRA-1000	The ID of the issue in Jira. E.g. JRA-1000

Rules

Rule Name	Object	Workflow Triggered

Rule Name	Object	Workflow Triggered
Example: Create Jira Issue (Task)	task	`example_jira_open_issue_task`
Example: Jira Close Issue (Task)	jira_task_references	`jira_transition_issue_task`
Example: Create Jira Issue	incident	`jira_open_issue`
Example: Jira Close Issue	incident	`jira_transition_issue`
Example: Jira Create Comment	note	`jira_create_comment`

Troubleshooting & Support

If using the app with an App Host, see the Resilient System Administrator Guide and the App Host Deployment Guide for troubleshooting procedures. You can find these guides on the IBM Knowledge Center, where you can select which version of the Resilient platform you are using.

If using the app with an integration server, see the Integration Server Guide

For Support

This is an IBM Supported app. Please search https://ibm.com/mysupport for assistance.