Osano and Jira – Subject Rights Request Workflow

<u>Overview</u>:

This document walks through the following:

- How to use Osano Webhooks to create Jira Issues once the requestor's email is verified.
- How to update Subject Rights Requests in Osano via the Subject Rights API once the Jira Issue is closed.

<u>Benefits</u>:

Improve subject rights request handling efficiency and time to resolution by:

- Automatically notifying your technical business units about new Subject Rights Requests
- Automatically updating the request within Osano once the Jira ticket is closed.

<u>Requirements</u>:

- In Osano:
  - Access to Webhooks in Osano ("Admin" User Role)
    - https://docs.osano.com/user-roles
    - https://docs.osano.com/integrations/introduction-to-webhooks
  - An Osano API Key
    - Within Osano, click the gear icon in the top-right corner and click "API Keys".
    - Click the purple "+" in the bottom-right corner and follow the prompts to create your API Key.
    - Copy it and save it somewhere safe as we will use it with Jira Webhooks later.
- In Jira:
  - Jira Account
    - With access to Project Settings to create an "Automation"
  - Jira API Token
    - The user creating the token will at least need "Read & Write edits issues & tickets" permissions.
    - The API Token will need to be BASE64 encoded. More information here on how to do that - https://developer.atlassian.com/cloud/jira/platform/basic-auth-for-rest-apis/
  - Jira Project Key (Projects > View All Projects > Key will be in the "Key" column (all caps))
    - More info here on how to find your Project Key - https://marketsplash.com/tutorials/jira/how-to-find-jira-project-key/#link1
  - Custom Fields to receive the Subject Rights values from Osano
    - More info on how to create Custom Fields in Jira is available here - https://support.atlassian.com/jira-cloud-administration/docs/create-a-custom-field/
    - More info on how to find the IDs of Custom Fields in Jira is available here - https://confluence.atlassian.com/jirakb/how-to-find-id-for-custom-field-s-744522503.html

<u>How to Make It</u>:

In Jira:

Step 1: Create an API Key by going to https://id.atlassian.com/manage-profile/security/api-tokens. After logging in, click “Create API token”. Save it for later

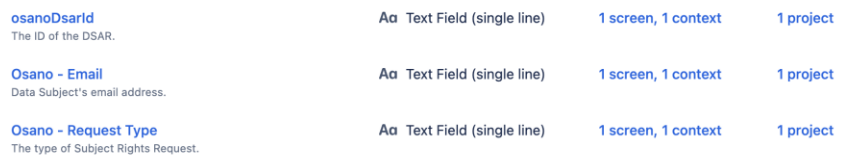
- **Note**: You’ll need to BASE64 your user email AND the API Key you grabbed from Jira. You can find an example of that here - https://developer.atlassian.com/cloud/jira/platform/basic-auth-for-rest-apis/
- You’ll also need to put “Basic” in front of the BASE64 encoded string for the “Authorization” header within the Osano Webhook section. Image of this in the “In Osano” section below.

Step 2: In Jira, click on “Projects” in the top navigation bar, then click “View All Projects” in the dropdown.

- There should be a “KEY” column. Grab the key for the project that you’d like to create “Issues” in and save that key for later.

Step 3: You’ll need to create custom fields in Jira to receive the values from the Osano Webhook and use them in your Jira Workflows.

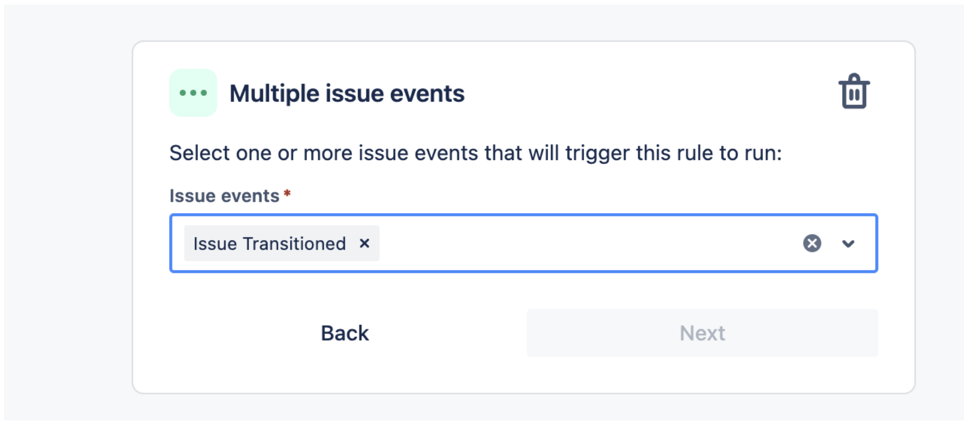
- In the example below, we’ve created three fields. However, you can create custom fields to utilize any of the Osano Webhook Variables found here.
- **Note**: We will use the `osanoDsarId` value in our Jira Webhook URL later. The naming scheme here is your preference, but we will be utilizing the Osano `dsarId` Webhook variable value here and in the Osano API call later in the walkthrough.



osanoDsarId The ID of the DSAR.	Aa Text Field (single line)	1 screen, 1 context	1 project
Osano - Email Data Subject's email address.	Aa Text Field (single line)	1 screen, 1 context	1 project
Osano - Request Type The type of Subject Rights Request.	Aa Text Field (single line)	1 screen, 1 context	1 project

Step 4: Next, you’ll need to create an “Automation” that updates the Subject Rights Request once the issue is closed. To create an Automation from the “Project” view, do the following:

- Click on “Project Settings” (in the left-side navigation)
- Click on “Automation” (in the left-side navigation)
- Click “Create Rule” in the top-right corner
  - For the “Add a Trigger” step, search for and use the “Multiple issue events” trigger.
  - Use “Issue Transitioned” as the value for the “Issue events” input and click “Next”:

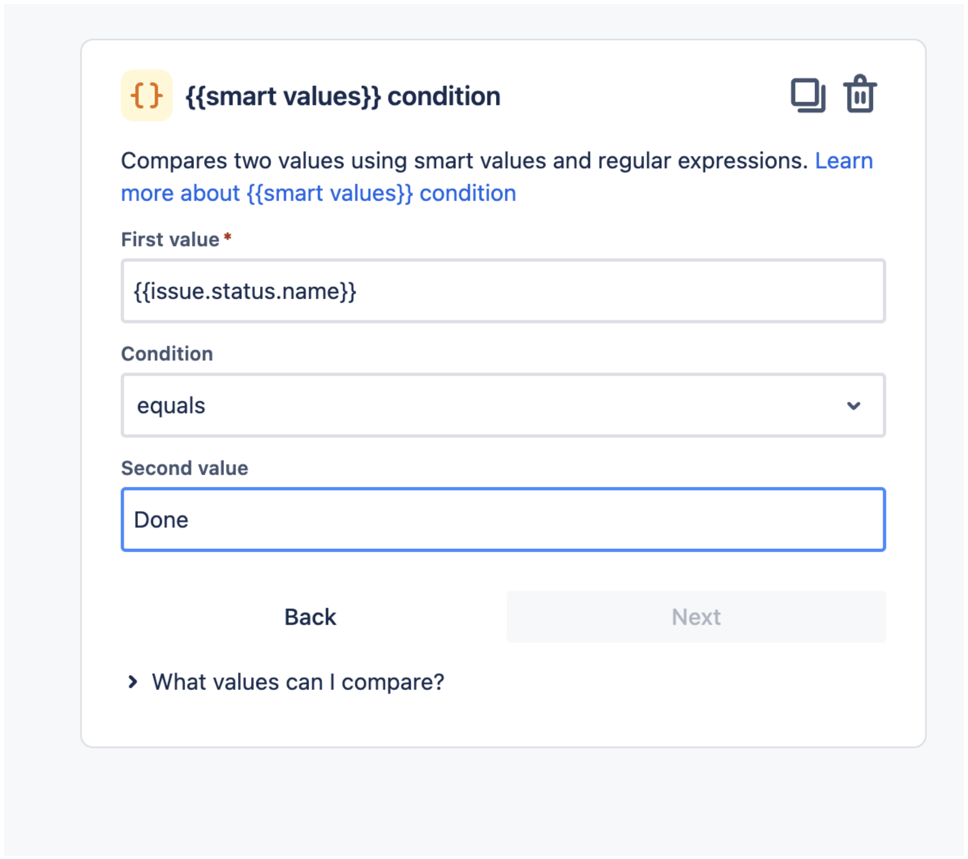

Multiple issue events

Select one or more issue events that will trigger this rule to run:

Issue events *

Issue Transitioned ×

- Next, click “Add Component” and click “IF: Add a condition”
  - Under “All Components” choose “{{smart values}} Condition”. Use the following values and then click “Next”


{{smart values}} condition

Compares two values using smart values and regular expressions. Learn more about {{smart values}} condition

First value *

{{issue.status.name}}

Condition

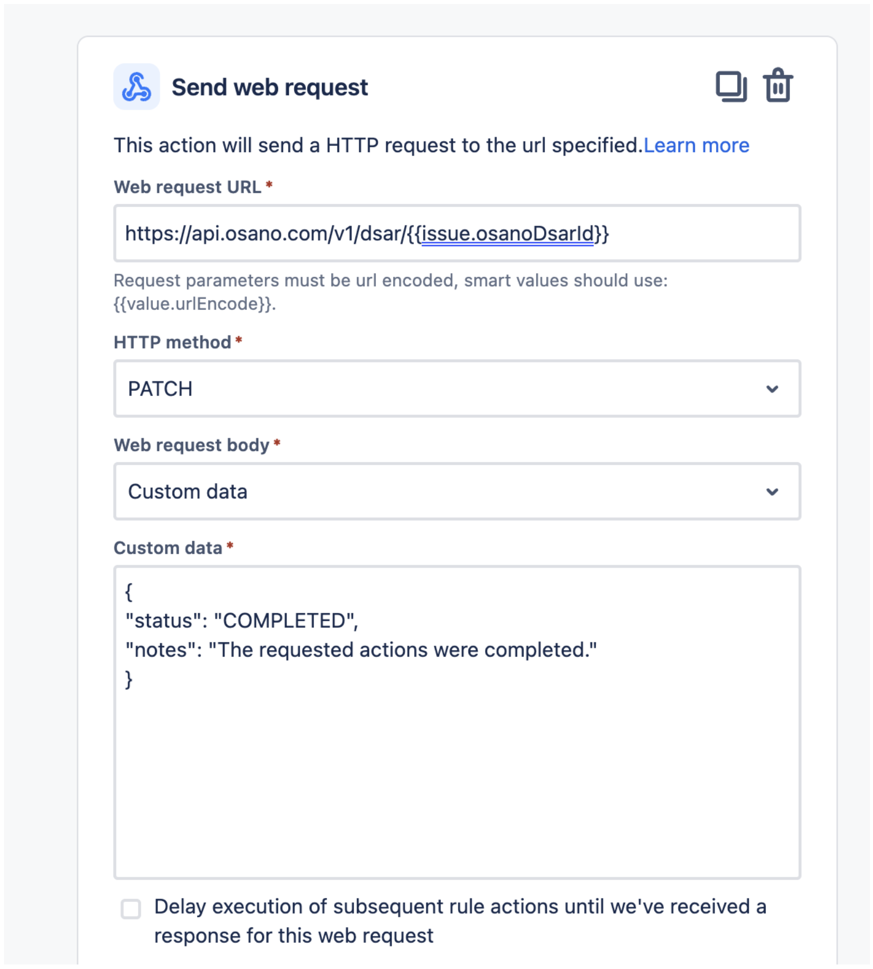
equals

Second value

Done

- Next, click “Add Component” and click “THEN: Add an action”
  - Search for and choose “Send Web Request”. Fill in the fields utilizing the notes below and then click “Next”:
  - **Note**: For the the “Web request URL”, the value in curly braces should be as follows
    - {{issue.JIRA_CUSTOM_FIELD_NAME}}
      - The JIRA_CUSTOM_FIELD_NAME section should be the Jira custom field name that receives the `dsarId` from the Osano Webhook.

        - The required "Headers" are:
            - Content-Type: Application/JSON
            - x-osano-api-key: {{YOUR_OSANO_API_KEY}}
        -
    - Example below of what the completed "Send Web Request" component should look like:



This action will send a HTTP request to the url specified.Learn more

Web request URL *

https://api.osano.com/v1/dsar/{{issue.osanoDsarId}}

Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.

HTTP method *

PATCH

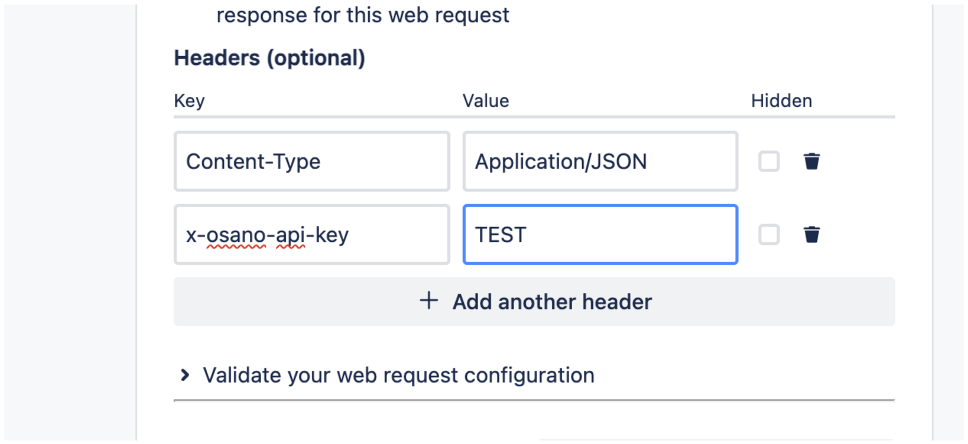
Web request body *

Custom data

Custom data *

```
{
"status": "COMPLETED",
"notes": "The requested actions were completed."
}
```

☐ Delay execution of subsequent rule actions until we've received a response for this web request



response for this web request

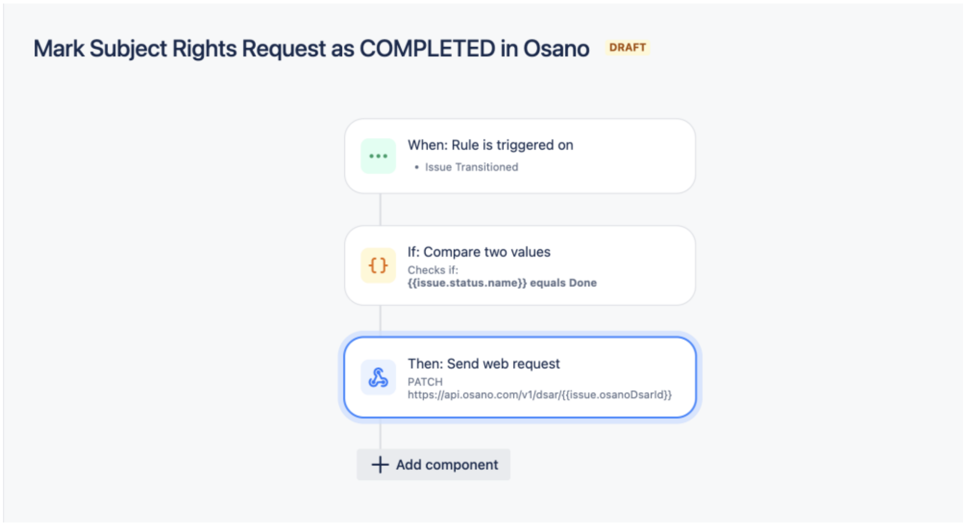
Headers (optional)

Key	Value	Hidden
Content-Type	Application/JSON	☐
x-osano-api-key	TEST	☐

- In the top-right corner, click "Turn on rule".

  - Fill in the “Rule name” and “Who can edit this rule?” sections and then click on “Turn on rule”.

- An overview of what the Automation flow should look like:

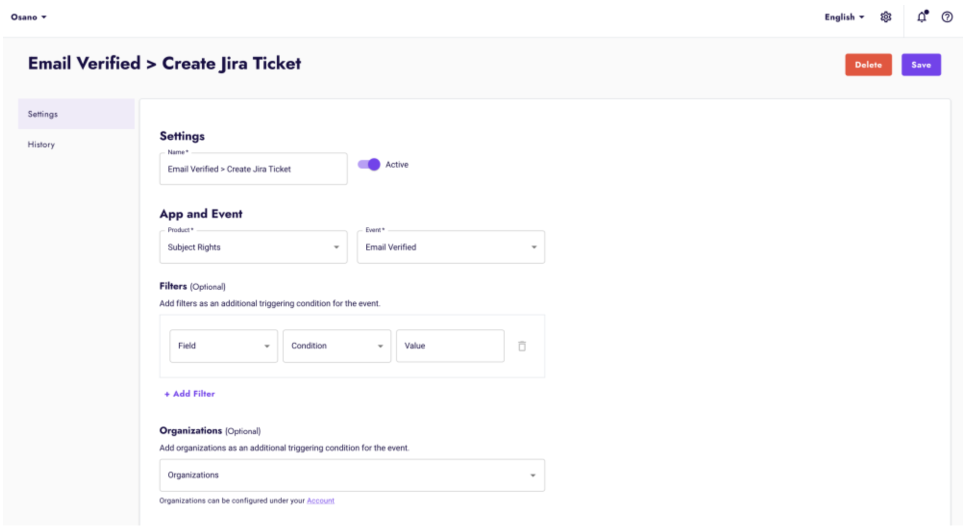

Mark Subject Rights Request as COMPLETED in Osano DRAFT

When: Rule is triggered on
- Issue Transitioned

If: Compare two values
Checks if:
{{issue.status.name}} equals Done

Then: Send web request
PATCH
https://api.osano.com/v1/dsar/{{issue.osanoDsarId}}

In Osano:



Email Verified > Create Jira Ticket

Settings

Name*: Email Verified > Create Jira Ticket
Active

App and Event

Product*: Subject Rights
Event*: Email Verified

Filters (Optional)
Add filters as an additional triggering condition for the event.

Field | Condition | Value

+ Add Filter

Organizations (Optional)
Add organizations as an additional triggering condition for the event.

Organizations

Organizations can be configured under your Account

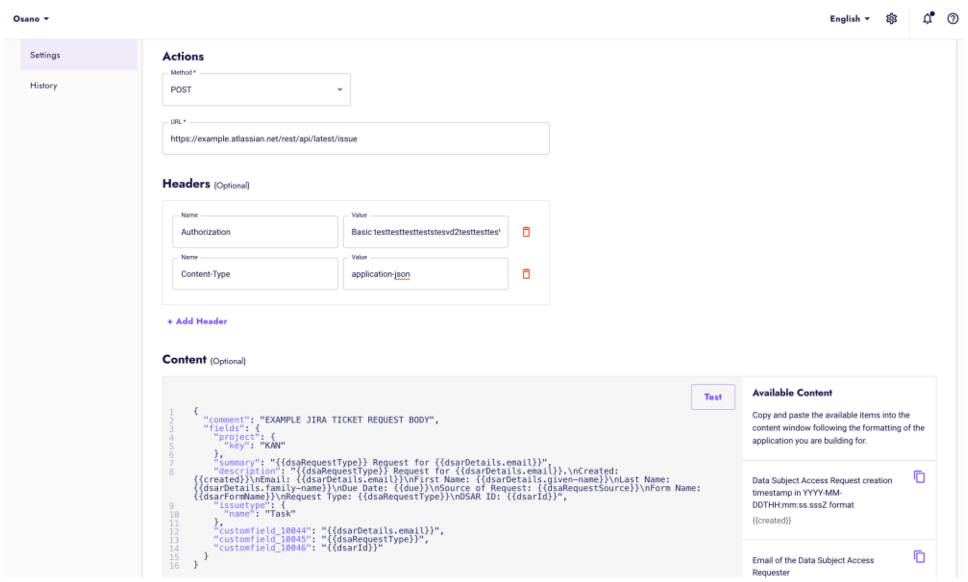
Step 1: Navigate to Webhook section in Osano by clicking the gear icon in the top-right corner, then clicking “Webhooks” in the drop-down menu.

Step 2: Under **Settings,** add a **Name** for the Webhook and toggle it to **Active**.

Step 3: Under **App and Event** select the following:

- “Subject Rights” for **Product**
- “Email Verified” for **Event**

Step 4: (optional) Add filters and Organizations if applicable.



Actions

Method *: POST

URL *: https://example.atlassian.net/rest/api/latest/issue

Headers (Optional)

Name	Value
Authorization	Basic testtesttestteststesvd2testtesttes'
Content-Type	application-json

Content (Optional)

```
{
  "comment": "EXAMPLE JIRA TICKET REQUEST BODY",
  "fields": {
    "project": {
      "key": "KAN"
    },
    "summary": "{{dsaRequestType}} Request for {{dsarDetails.email}}",
    "description": "{{dsaRequestType}} Request for {{dsarDetails.email}}.\nCreated:
{{created}}\nEmail: {{dsarDetails.email}}\nFirst Name: {{dsarDetails.given-name}}\nLast Name:
{{dsarDetails.family-name}}\nDue Date: {{due}}\nSource of Request: {{dsaRequestSource}}\nForm Name:
{{dsarFormName}}\nRequest Type: {{dsaRequestType}}\nDSAR ID: {{dsarId}}",
    "issuetype": {
      "name": "Task"
    },
    "customfield_10044": "{{dsarDetails.email}}",
    "customfield_10045": "{{dsaRequestType}}",
    "customfield_10046": "{{dsarId}}"
  }
}
```

Available Content

Copy and paste the available items into the content window following the formatting of the application you are building for.

Data Subject Access Request creation timestamp in YYYY-MM-DDTHH:mm:ss.sssZ format

{{created}}

Email of the Data Subject Access Requester

Step 5: For your **Headers** add the following:

- Authorization
  - {Your Base64 encoded string from Jira}
- Content-Type
  - application-json

Step 6: Fill in the **Content** window with the desired Jira Issue Issue. Example JSON is below. You can reference the Osano “Webhook Substitutable Variables” in the side-panel and here.

- **Note**: Be sure to include your Jira custom fields so that you can reference them in the Jira webhook.

```
{
 "comment": "EXAMPLE JIRA TICKET REQUEST BODY",
 "fields": {
  "project": {
   "key": "KAN"
  },
  "summary": "{{dsaRequestType}} Request for {{dsarDetails.email}}",
  "description": "{{dsaRequestType}} Request for {{dsarDetails.email}}.\nCreated: {{created}}\nEmail: {{dsarDetails.email}}\nFirst Name: {{dsarDetails.given-name}}\nLast Name:{{dsarDetails.family-name}}\nDue Date: {{due}}\nSource of Request: {{dsaRequestSource}}\nForm Name: {{dsarFormName}}\nRequest Type: {{dsaRequestType}}\nDSAR ID: {{dsarId}}",

  "issuetype": {

   "name": "Task"

  },

  "customfield_10044": "{{dsarDetails.email}}",

  "customfield_10045": "{{dsaRequestType}}",

  "customfield_10046": "{{dsarId}}"

 }

}
```

Step 7: Click “Save” in the top-right corner to save the Webhook.

You should now be able to handle Subject Rights Requests more efficiently by:

- Automatically creating Jira Issues when a Subject Rights request is received
- Automatically updating the request within Osano once the Jira ticket is closed