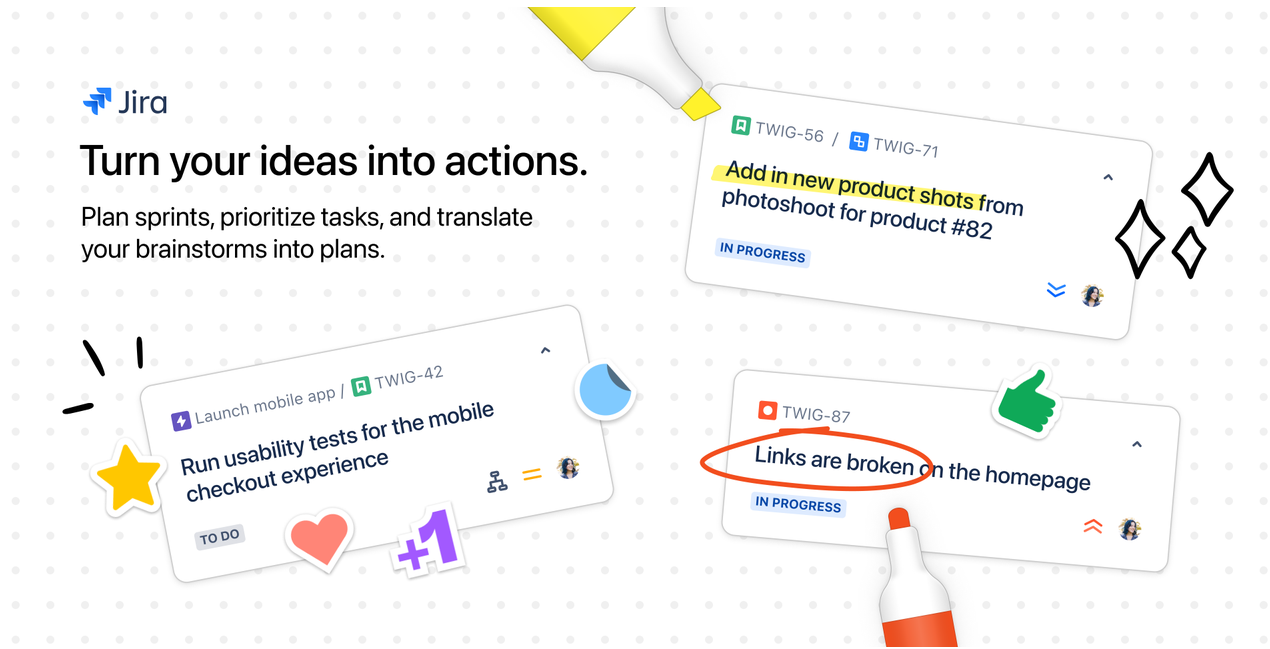


Figma Jira Widget Security Overview



Overview

The [Jira widget](#) is an [OAuth 2.0 App](#) that allows users to view and edit their Jira data in FigJam.

The widget only supports Jira cloud sites. Data read/written with Jira is done via the Jira Cloud REST API. We do not have plans to support on-prem at this time.

Functionality/Behavior

Authentication

For users to query/write data to Jira, they must authenticate with their Jira account. Users need to explicitly grant access to each Jira site they want to use with the widget.

Upon success, the widget running in the browser is given access to the OAuth token. No user credentials are stored server-side. The refresh token expires after 90 days and the OAuth tokens have an absolute expiry time of 365 days.



Once a user authenticates with Jira, they can import either projects (i.e. all issues in the project) or an individual issue from Jira into FigJam. When a user imports a project or issue into FigJam, all issue fields that are displayed on the widget will be visible to anyone who can view the FigJam file.

Imported issues and projects will only reflect details present when they were imported. Users can pull in the latest values from Jira by clicking a refresh button above the widget. Users who have not authenticated with Jira, or do not have the permissions to view an issue in Jira, will be denied from pulling in any updates that have since been made in Jira.

Turning stickies into Jira issues

Authenticated users are able to select sticky notes in FigJam and convert them into Jira issues. Users can select which Jira project they'd like to create these issues under, as well as some field values such as the issue type and reporter. Users are denied from creating issues in projects if they do not have permissions for creating issues in Jira.

Editing existing Jira issues

Users can edit the various metadata on an issue by clicking on the field. Users who are not authenticated will be prompted to do so. Users who do not have permissions to edit the issue in Jira will be denied when attempting to do so with the widget.

User Credentials

The `accessToken` & `refreshToken` are stored in locally inside the browser and are not persisted on Figma's servers. Client storage is specific to the current user ID and widget ID. Other widgets run by the same user and other users running the same widget will not be able to read this value. More details for how OAuth is handled for widgets and plugins can be found [here](#).

Jira Data

Once the user has granted access to a Jira site, they will be able to use the widget to read/write the following resources provided the user has permissions to do so in Jira:

- read project fields (eg. description, avatar)
- read user properties (name, avatar, email) - Users can configure which fields are accessible to OAuth applications in their account settings (example: if Alice &

Bob are coworkers and Alice chooses to hide their email, then Alice's email will be `null` when Bob is using the Jira widget and the Jira widget will not have access to Alice's email)

- read the following resources global to the site (statuses, issue types, priorities)
- read/write to issue fields (eg. priority, assignee, etc)

A full list of OAuth permission scopes can be found further down in the Appendix.

The widget only persists the minimum subset required to support widget functionality. This data is stored within the state of the widget. Other widgets in FigJam do not have access to this data. FigJam serializes data contained in widgets with Figma's servers and clients who have access to the FigJam file. Figma encrypts file data at rest and in transit and stores keys in a CloudHSM that is Fips 140-2 level 3 compliant. Figma's servers live within Figma's own VPCs to prevent unauthorized network requests. Figma services are hosted in US-based AWS facilities

The following Jira resources are persisted:

- **Issue**
 - id
 - summary
 - key
 - url
 - issue type
 - assignee - if applicable
 - name - if said assignee has their visibility set to anyone
 - avatar - if said assignee has their visibility set to anyone
 - priority
 - status
 - epic
 - parent
 - subtasks
 - flag
 - due date
 - project id
- **Project**
 - id
 - key

- description
- url
- projectTypeKey - software, service_desk, or business
- icon
- open issue count
- closed issue count
- subtask count
- issue ids

Jira Widget Updates

When new versions of the widget are published, users are required to manually update the widget to pull the latest changes. Users can revert the widget back to its previous version by undoing the update action or restoring a previous version of the FigJam file.

We have a rigorous code review and QA process for widget updates. If we find security vulnerabilities during code review we fix them before we ship. Changes are thoroughly tested before being deployed. Error messages returned to end-users will not contain coding information/sensitive information.

FAQ

Does the widget store any sensitive information?

The widget uses minimal non-sensitive PII including email and name for rendering assignee and reporter metadata for a Jira issue. Jira users can configure what information is accessible to OAuth applications in their account settings

Example: If Alice & Bob are coworkers and Alice chooses to hide their email, then Alice's email will be `null` when Bob is using the Jira widget and the Jira widget will not have access to Alice's email

In addition, the widget abides by Figma's privacy policy

<https://www.figma.com/privacy/>

Who has access to data in the widget?

In addition to the user using the Jira widget, users given permissions to view the FigJam file will be able to see issue and project metadata rendered in the widget. Only authorized users with permissions to view the Jira data are allowed to refresh the widget with changes made in Jira.

Employee access is limited to select and authorized members of the engineering team. Access is limited to the production environment where customer data is located. All personnel are required to sign Non Disclosure Agreement (NDA) or Confidentiality Agreements (CA) as a condition of employment to protect customer information.

Where geographically is the widget data stored?

User credentials are stored locally on the user's computer. Jira data pulled in for the widget is stored in the FigJam file and serialized to Figma's servers. Figma services are hosted in US-based AWS facilities. Figma encrypts file data at rest and in transit & Figma's servers live within Figma's own VPCs to prevent unauthorized network requests.

Appendix

List of OAuth permission scopes

Jira OAuth apps define a list of permission scopes that define what read/write operations can be performed. [Link to Jira's OAuth Permission Scopes Documentation](#)

We use the following OAuth scopes:

- `offline_access`
 - Required for:
 - refresh token for re-authenticating the user after the access token has expired
- `read:application-role:jira`
 - Required for:
 - get project

- get project paginated
 - get recent projects
 - get current user
 - find users assignable to issues
- `read:audit-log:jira`
 - Required for:
 - search for issues using JQL
- `read:avatar:jira`
 - Required for:
 - get issue
 - get create issue metadata
 - search issues using JQL
 - get project
 - get project paginated
 - get recent projects
 - get issue types for project
 - get current user
 - find users assignable to issues
- `read:field-configuration:jira`
 - Required for:
 - get issue
 - get create issue metadata
 - search issues using JQL
 - get transitions
- `read:group:jira`
 - Required for:
 - get project
 - get project paginated
 - get recent projects
 - get current user
 - find users assignable to issues
- `read:issue-details:jira`
 - Required for:
 - search issues using JQL
- `read:issue-link:jira`
 - not needed today, may use in the future
- `read:issue-meta:jira`

- Required for:
 - get issue
 - get create issue metadata
 - search issues using JQL
- `read:issue-security-level:jira`
 - Required for:
 - get issue
- `read:issue-status:jira`
 - Required for:
 - get all statuses for project
- `read:issue-type-hierarchy:jira`
 - Required for:
 - get project
 - get project paginated
 - get recent projects
- `read:issue-type:jira`
 - Required for:
 - get project
 - get project paginated
 - get recent projects
 - get all statuses for project
 - get issue types for project
- `read:issue-type.property:jira`
 - not needed today, may use in the future
- `read:issue:jira-software`
 - not needed
- `read:issue:jira`
 - Required for:
 - get issue
 - find users for picker
 - bulk create issue
- `read:issue.changelog:jira`
 - Required for:
 - get issue
- `read:issue.property:jira`
 - not needed today, may use in the future
- `read:issue.time-tracking:jira`

- not needed
- `read:issue.transition:jira`
 - Required for:
 - get transitions
- `read:issue.vote:jira`
 - Required for:
 - get issue
- `read:issue.votes:jira`
 - not needed today, may use in the future
- `read:priority:jira`
 - Required for:
 - get priority
- `read:project-category:jira`
 - Required for:
 - get project
 - get project paginated
 - get recent projects
 - get issue types for project
- `read:project-version:jira`
 - Required for:
 - get project
 - get project paginated
 - get recent projects
- `read:project:jira`
 - Required for:
 - get project
 - get project paginated
 - get recent projects
 - get issue types for project
 - find users assignable to issues
- `read:project.avatar:jira`
 - not needed today, may use in the future
- `read:project.component:jira`
 - Required for:
 - get project
 - get project paginated
 - get recent projects

- `read:project.property:jira`
 - Required for:
 - get project
 - get project paginated
 - get recent projects
- `read:status:jira`
 - Required for:
 - get issue
 - get transitions
 - get all statuses for project
- `read:user:jira`
 - Required for:
 - get issue
 - get project
 - get project paginated
 - get recent projects
 - get current user
 - find users assignable to issues
 - find users for picker
- `write:attachment:jira`
 - Required for:
 - bulk create issue
- `write:comment:jira`
 - Required for:
 - bulk create issue
- `write:comment.property:jira`
 - Required for:
 - bulk create issue
- `write:issue:jira`
 - Required for:
 - bulk create issue
 - edit issue
 - transition issue
- `write:issue.property:jira`
 - Required for:
 - transition issue

List of Jira API routes used

Get issue

Returns the details for an issue.

- **Required oauth scopes**

read:issue-meta:jira, read:issue-security-level:jira,
read:issue.vote:jira, read:issue.changelog:jira, read:avatar:jira,
read:issue:jira, read:status:jira, read:user:jira, read:field-
configuration:jira

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issues/#api-rest-api-3-issue-issueidorkey-get>

Get create issue metadata

Returns details of projects, issue types within projects, and, when requested, the create screen fields for each issue type for the user. Use the information to populate the requests in [Create issue](#) and [Create issues](#).

- **Required oauth scopes**

read:issue-meta:jira, read:avatar:jira, read:field-configuration:jira

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issues/#api-rest-api-3-issue-createmeta-get>

Search for issues using JQL

Searches for issues using [JQL](#).

- **Required oauth scopes**

read:issue-details:jira, read:audit-log:jira, read:avatar:jira,
read:field-configuration:jira, read:issue-meta:jira

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issue-search/#api-rest-api-3-search-get>

Get priority

Returns an issue priority.

- **Required oauth scopes**

read:priority:jira

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issue-priorities/#api-rest-api-3-priority-id-get>

Get transitions

Returns either all transitions or a transition that can be performed by the user on an issue, based on the issue's status.

- **Required oauth scopes**

`read:issue.transition:jira`, `read:status:jira`, `read:field-configuration:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issues/#api-rest-api-3-issue-issueidorkey-transitions-get>

Get project

Returns the [project details](#) for a project.

- **Required oauth scopes**

`read:issue-type:jira`, `read:project:jira`, `read:project.property:jira`, `read:user:jira`, `read:application-role:jira`, `read:avatar:jira`, `read:group:jira`, `read:issue-type-hierarchy:jira`, `read:project-category:jira`, `read:project-version:jira`, `read:project.component:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-projects/#api-rest-api-3-project-projectidorkey-get>

Get projects paginated

Returns a [paginated](#) list of projects visible to the user.

- **Required oauth scopes**

`read:issue-type:jira`, `read:project:jira`, `read:project.property:jira`, `read:user:jira`, `read:application-role:jira`, `read:avatar:jira`, `read:group:jira`, `read:issue-type-hierarchy:jira`, `read:project-category:jira`, `read:project-version:jira`, `read:project.component:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-projects/#api-rest-api-3-project-search-get>

Get recent projects

Returns a list of up to 20 projects recently viewed by the user that are still visible to the user

- **Required oauth scopes**

`read:issue-type:jira`, `read:project:jira`, `read:project.property:jira`,
`read:user:jira`, `read:application-role:jira`, `read:avatar:jira`,
`read:group:jira`, `read:issue-type-hierarchy:jira`, `read:project-`
`category:jira`, `read:project-version:jira`, `read:project.component:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-projects/#api-rest-api-3-project-recent-get>

Get all statuses for project

Returns the valid statuses for a project. The statuses are grouped by issue type, as each project has a set of valid issue types and each issue type has a set of valid statuses.

- **Required oauth scopes**

`read:issue-status:jira`, `read:issue-type:jira`, `read:status:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-projects/#api-rest-api-3-project-projectidorkey-statuses-get>

Get issue types for project

Returns issue types for a project.

- **Required oauth scopes**

`read:issue-type:jira`, `read:avatar:jira`, `read:project-category:jira`,
`read:project:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issue-types/#api-rest-api-3-issuetype-project-get>

Get current user

Returns details for the current user.

- **Required oauth scopes**

`read:application-role:jira`, `read:group:jira`, `read:user:jira`,
`read:avatar:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-myself/#api-rest-api-3-myself-get>

Find users assignable to issues

Returns a list of users that can be assigned to an issue. Use this operation to find the list of users who can be assigned to:

- a new issue, by providing the `projectKeyOrId`.
- an updated issue, by providing the `issueKey`.
- to an issue during a transition (workflow action), by providing the `issueKey` and the transition id in `actionDescriptorId`. You can obtain the IDs of an issue's valid transitions using the `transitions` option in the `expand` parameter of [Get issue](#).

- **Required oauth scopes**

`read:issue:jira`, `read:project:jira`, `read:user:jira`, `read:application-role:jira`, `read:avatar:jira`, `read:group:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-user-search/#api-rest-api-3-user-assignable-search-get>

Find users for picker

Returns a list of users whose attributes match the query term. The returned object includes the `html` field where the matched query term is highlighted with the HTML strong tag. A list of account IDs can be provided to exclude users from the results. This operation takes the users in the range defined by `maxResults`, up to the thousandth user, and then returns only the users from that range that match the query term. This means the operation usually returns fewer users than specified in `maxResults`. To get all the users who match the query term, use [Get all users](#) and filter the records in your code.

Privacy controls are applied to the response based on the users' preferences. This could mean, for example, that the user's email address is hidden. See the [Profile visibility overview](#) for more details.

- **Required oauth scopes**

`read:user:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-user-search/#api-rest-api-3-user-picker-get>

Bulk create issue

Creates up to **50** issues and, where the option to create subtasks is enabled in Jira, subtasks. Transitions may be applied, to move the issues or subtasks to a workflow step other than the default start step, and issue properties set.

- **Required oauth scopes**

`write:issue:jira`, `write:comment:jira`, `write:comment.property:jira`,
`write:attachment:jira`, `read:issue:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issues/#api-rest-api-3-issue-bulk-post>

Edit issue

Edits an issue. A transition may be applied and issue properties updated as part of the edit.

The edits to the issue's fields are defined using `update` and `fields`. The fields that can be edited are determined using [Get edit issue metadata](#).

- **Required oauth scopes**

`write:issue:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issues/#api-rest-api-3-issue-issueidorkey-put>

Transition issue

Performs an issue transition and, if the transition has a screen, updates the fields from the transition screen.

- **Required oauth scopes**

`write:issue:jira`, `write:issue.property:jira`

- **Api docs**

<https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issues/#api-rest-api-3-issue-issueidorkey-transitions-post>

For more information about Figma's security, please see

<https://www.figma.com/security/>.

Information collected as described herein is subject to our privacy notice, available at

<https://www.figma.com/privacy/>.