Dev Info Connect Add-on Use Cases

Purpose

The page is intended to show the common use cases from the perspective of a vendor providing a Connect Add-on that utilises our new API to provide "devInfo" data.

This page is intended to be read in conjunction with:

- Integrating with Jira Software Cloud (includes information on the Atlassian Connect framework).

Use Cases

UC-1 Vendor Connect Add-on Installation

Pre-conditions:

- The Vendor Connect Add-on is available in the Marketplace.
- The Vendor Connect Add-on has not already been installed.

Post-conditions:

- The Vendor Connect Add-on has been installed into Jira.
- The Vendor Connect Add-on and Jira have a shared secret for communication.

Triggers

- The Jira Administrator initiates the installation of the Vendor Connect Add-on.

Basic Flow

1. The Jira Administrator goes to the Atlassian Marketplace and initiates installation of the Vendor Connect Add-on.
2. The Jira Administrator approves the installation the Vendor Connect Add-on.
3. The Jira system calls the Vendor Connect Add-on to register the installation, including providing the shared secret, which must be stored by the Vendor Connect Add-on. See Connect Lifecycle for details.
4. The user selects 'configure' from marketplace and is redirected to the Jira Administrator to the Vendor Connect Add-on provided Administration screen. The recommended implementation is as an admin tab panel which is a standard connect iframe. For the configure and getting started page they should use the standard connect configure and post install pages as documented here

Extensions

Installation via an uploaded descriptor

This extension is when a Vendor Connect Add-on is installed by uploaded the descriptor via a custom URL, rather than the Atlassian Marketplace. This would be typically done by Developers during the development of their Connect Add-on.

The changes for the steps are:

- Step 1 is replaced by the Jira Administrator manually installing the Vendor Connect Add-on via an uploaded descriptor.

UC-2 Connecting an account

Pre-conditions:

- The Vendor Connect Add-on is installed

Post-conditions:

- A Vendor Account has been connected to have devInfo data pushed to Jira.

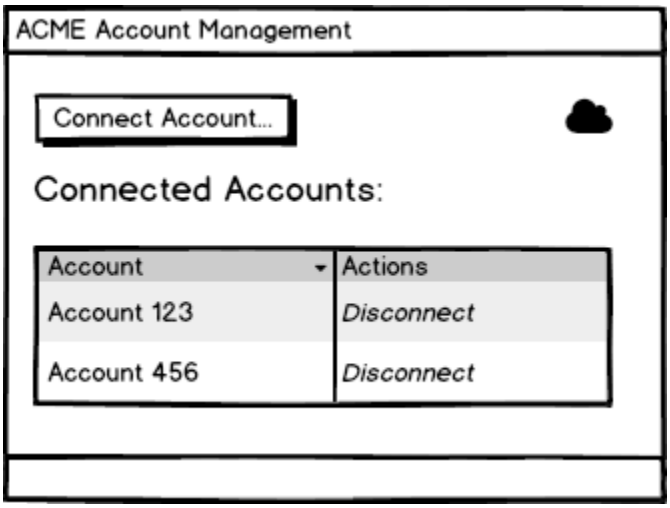
Triggers:

- The Jira Administrator navigates to the Vendor Connect Add-on provided Administration screen. Note, this will have to always be initiated via Jira, it's not possible directly access the Vendor Connect Add-on provided Administration screen.

Basic Flow

1. The Jira Administrator navigates to the Vendor Connect Add-on provided Administration screen. See Administration UI locations for details.
2. The Jira Administrator uses the Vendor Connect Add-on provided Administration screen to connect a Vendor account. Note, it's the complete responsibility of the Vendor Connect Add-on to:
    a. Perform the necessary authentication and authorisation checks.
    b. Record in the the Vendor system that the Vendor account has been connected to have devInfo data pushed to Jira.

The following diagram is a mock up of what a Vendor Connect Add-on provided Administration screen might look like.



UC-3 Providing data

Pre-conditions:

- The Vendor Connect Add-on is installed.
- The Vendor system has recorded that a Vendor account is connected to have devInfo data pushed to Jira.

Post-conditions:

- The Jira has been updated with devInfo data.

Triggers:

- The Vendor system has detected activity that need to be sent to Jira.

Basic Flow

1. The Vendor system detects activity which need to be sent to Jira. These could be related to commits, branches, pull requests or

repositories.
2. The Vendor system calls Jira using the POST method on the REST resource `/devinfo/bulk` to provide the devInfo data.
3. Jira stores the devInfo data, which includes the following structure:the account id and for each repository:
    - properties - arbitrary key/value pairs that are associated with entity
    - repositories
        - branches
        - commits
        - pull requests

UC-4 Deleting a repository

Pre-conditions:

- The Vendor Connect Add-on is installed.
- The Jira Administrator has connected at least one account using the Vendor Connect Add-on Administration screen.

Post-conditions:

- The devInfo data that is stored in Jira that is associated with a repository has been removed.

Triggers:

- The Vendor system has detected that a repository has been removed that has devInfo stored in Jira.

Basic Flow

1. Vendor system has detects that a repository has been removed that has devInfo stored in Jira.
2. The Vendor system calls Jira using DELETE method on the REST resource `/devinfo/repository/{repositoryid}` to delete all devInfo data associated with the specified repository.
3. Jira removes all devInfo data associated with the repository.

UC-5 Deleting a branch

Pre-conditions:

- The Vendor Connect Add-on is installed.
- The Jira Administrator has connected at least one account using the Vendor Connect Add-on Administration screen.

Post-conditions:

- The branch devInfo data that is stored in Jira that is associated with a repository has been removed.

Triggers:

- The Vendor system has detected that a branch has been deleted that has devInfo stored in Jira.

Basic Flow

1. Vendor system detects that a branch has been removed that has devInfo stored in Jira.
2. The Vendor system calls Jira using DELETE method on the REST resource `/devinfo/repository/{repositoryid}/branch/{branchid}` to delete the devInfo data associated with the specified repository and branch.
3. Jira removes the devInfo data associated with the repository and branch.

UC-6 Deleting a pull request

Pre-conditions:

- The Vendor Connect Add-on is installed.
- The Jira Administrator has connected at least one account using the Vendor Connect Add-on Administration screen.

Post-conditions:

- The pull request devInfo data that is stored in Jira that is associated with a repository has been removed.

Triggers:

- The Vendor system has detected that a pull request has been deleted that has devInfo stored in Jira.

Basic Flow

1. Vendor system detects that a pull request has been deleted that has devInfo stored in Jira.
2. The Vendor system calls Jira using DELETE method on the REST resource `/devinfo/repository/{repositoryid}/pullrequest/{pullrequestid}` to delete the devInfo data associated with the specified repository and pull request.
3. Jira removes the devInfo data associated with the repository and pull request.

UC-7 Deleting a commit

Pre-conditions:

- The Vendor Connect Add-on is installed.
- The Jira Administrator has connected at least one account using the Vendor Connect Add-on Administration screen.

Post-conditions:

- The commit devInfo data that is stored in Jira that is associated with a repository has been removed.

Triggers:

- The Vendor system has detected that a pull request has been deleted that has devInfo stored in Jira.

Basic Flow

1. Vendor system detects that a commit has been deleted that has devInfo stored in Jira.
2. The Vendor system calls Jira using DELETE method on the REST resource `/devinfo/repository/{repositoryid}/commit/{commitid}` to delete the devInfo data associated with the specified repository and commit.
3. Jira removes the devInfo data associated with the repository and commit.

UC-8 Disconnecting an account

Pre-conditions:

- The Vendor Connect Add-on is installed.
- The Vendor system has recorded that a Vendor account is connected to have devInfo data pushed to Jira.
- The Vendor system has recorded the Vendor account identifier for all devInfo data pushed to Jira by using the properties capability.

Post-conditions:

- The Vendor system has recorded that a Vendor account is no longer connected to have devInfo data pushed to Jira.
- The devInfo data stored in Jira for an account has been removed.

Triggers:

- The Jira Administrator navigates to the Vendor Connect Add-on provided Administration screen. See Administration UI locations for details. Note, this will have to always be initiated via Jira, it's not possible directly access the Vendor Connect Add-on provided Administration screen.

Basic Flow

1. The Jira Administrator navigates to the Vendor Connect Add-on provided Administration screen.

2. The Jira Administrator uses the Vendor Connect Add-on provided Administration screen to disconnect a Vendor account. Note, it's the complete responsibility of the Vendor Connect Add-on to:
    a. Record in the the Vendor system that the Vendor account has been disconnected from having devInfo data pushed to Jira.
    b. Call Jira using the DELETE method on the REST resource `/devinfo/bulkByProperties?accountid={accountid}` to delete all devInfo data associated with the specified account id. Note: this is assuming that the property `accountid` was used to associated devInfo data to an account (the Vendor is free to use other properties).
3. Jira removes all devInfo data associated with the account (via properties).

UC-9 Vendor Connect Add-on Uninstallation

Pre-conditions:

- The Vendor Connect Add-on is installed

Post-conditions:

- The Vendor Connect Add-on is uninstalled.
- The devInfo data stored in Jira for the Vendor Connect Add-on has been removed.
- All Vendor Accounts has been disconnected to have devInfo data pushed to Jira.

Triggers:

- The Jira Administrator initiates the uninstallation of the Vendor Connect Add-on.

Basic Flow

1. The Jira Administrator goes to the Connect Add-on Administration screen and initiates the uninstallation of the Vendor Connect Add-on.
2. The Jira Administrator approves the uninstallation the Vendor Connect Add-on.
3. The Jira system removes all devInfo data stored for the Vendor Connect Add-on.
4. The Jira system calls the Vendor Connect Add-on to unregister the installation. See Connect Lifecycle for details.
5. The Vendor Connect Add-on disconnects all registered Vendor accounts so that devInfo data is no longer sent to Jira.