

# Appvibe Asset Tracker

- [Overview of Features](#)
  - [Key Features](#)
- [Getting Started](#)
  - [Installation](#)
- [Configuration](#)
  - [Setup and Customization](#)
- [Working With Records](#)
  - [Adding and Viewing Records](#)
- [FAQ](#)
  - [FAQ](#)
- [Support](#)

# Overview of Features

# Key Features

Appvibe Asset Tracker is a powerful application designed to manage IT assets, including hardware, software licenses, and peripherals, directly within Jira projects. By seamlessly integrating asset management into Jira, teams can streamline their IT processes and enhance collaboration across departments. The tool's robust functionality is tailored to fit the dynamic needs of businesses looking to optimize asset utilization and maintain control over inventory tracking.

Incorporating customer feedback is essential to IT asset management as it helps organizations adapt and refine their asset tracking strategies. Customer insights can reveal bottlenecks or inefficiencies in current processes, leading to more informed decision-making and innovation. By leveraging feedback, teams can ensure their asset management practices align with user needs and industry standards, ultimately improving operational efficiency and customer satisfaction.

- **Feedback Submission:** Capture feedback seamlessly through Jira issues and dedicated forms.
- **Organized Views:** Use tags, filters, and saved views to sort and prioritize feedback efficiently.
- **Direct Jira Integration:** Link customer feedback to Jira issues, ensuring actionable follow-ups.
- **Customizable Layout:** Add, remove, or rearrange columns to suit your team's feedback tracking needs.
- **Activity History:** Maintain a clear log of actions and updates made to each feedback record.

In conclusion, Appvibe Asset Tracker not only streamlines IT asset management within Jira but also empowers teams to harness customer feedback effectively. The combination of these features equips organizations to maintain precise control over their assets while fostering a feedback-driven culture that encourages continual improvement.

# Getting Started

# Installation

Asset Tracker is a Jira app designed to streamline asset management within your projects. Follow these steps to install and enable Asset Tracker via the Atlassian Marketplace.

## **Step 1:**

Log in to Jira with Admin privileges to ensure you can perform installation and configuration tasks.

## **Step 2:**

Open the Atlassian Marketplace by clicking on Apps in the top menu, then select Explore more apps.

## **Step 3:**

Use the search bar in the Marketplace to find Appvibe Asset Tracker.

## **Step 4:**

Install the App by selecting Get app or Try it free. Confirm the installation and grant the required permissions when prompted.

## **Step 5:**

Once installed, navigate to Manage Apps in the Admin section to verify that Asset Tracker is enabled.

After successful installation, a link to Asset Tracker will appear under the Apps section in the menu on the left side of your Jira screen, providing quick access to its features.

# Configuration

# Setup and Customization

Navigate to the configuration and setup of the Appvibe Asset Tracker app by going to **Manage Apps** in Jira, locating **Appvibe Asset Tracker**, and selecting the **Configuration** button. This will direct you to the configuration page where you can manage Roles and Permissions, view System Fields, set up Custom Fields, and configure Conditional Formatting.

## Roles and Permissions

The Appvibe Asset Tracker app supports three user roles:

1. **Admin:** This super role grants complete access to all features, including configuration and setup.
2. **Editor:** Users with the Editor role can update and view Asset Tracker data fully.
3. **Viewer:** Viewers can only view Asset Tracker data and cannot make updates.

Administrators can assign these roles to any Jira groups. By default, Jira Administrators have access to the Asset Tracker configuration screens.

## System Fields

Asset Tracker includes predefined fields essential for asset management:

- Asset Type
- Serial Number
- Location
- Assigned User
- Purchase Date
- Warranty Status
- Tags

These represent the core fields necessary for managing your assets effectively.

## Custom Fields

To define additional fields, use the **Add Field** button. You can create new fields with these parameters:

- **Field Key:** A unique key identifier
- **Field Label:** A user-friendly label for the field
- **Field Type:** Choose from Text, Number, Date, Select, or Multi-Select
- **Required:** Specify if the field is mandatory
- **Sortable:** Indicate if the field can be sorted
- **Filterable:** Determine if the field can be filtered

## Conditional Formatting

Conditional Formatting lets you apply formatting rules based on field values. Upon meeting specified conditions, you can alter labels and highlight colors to visually emphasize key data points.

## Summary

The Appvibe Asset Tracker offers flexibility and control through its robust configuration settings, enabling tailored asset management to fit your organization's needs. By customizing roles, fields, and data formatting, you can align asset tracking with your specific requirements.

# Working With Records

# Adding and Viewing Records

Adding Records to Assets in the Appvibe Asset Tracker is straightforward and can be performed at a system, project, or Jira Issue level. This flexibility allows developers and project managers to tailor asset management to the needs of their organization, ensuring that records are accurately maintained and accessible.

When adding a record, you can input data for several predefined fields: Asset Type, Serial Number, Location, Assigned User, Purchase Date, and Warranty Status, as well as any custom fields you've defined. To add these records, navigate to the Assets dashboard page and click the Create button. Input the necessary information in fields such as Asset Type, Serial Number, Location, Assigned User, Purchase Date, Warranty Status, tags, and any custom fields. Once the record is created, it is easily accessible for review within the Assets dashboard.

## 1. **Create and Manage Multiple Views**

Users can create and manage multiple views within the dashboard by selecting the View drop-down at the top. This feature allows for creating new views, managing existing ones, or clearing views. Note that only users with admin roles can make changes to a view.

## 2. **Filter by a Column Value**

Records can be filtered by column values by clicking the column title icon and applying data filters. Be aware that filtering is available only on columns where this feature is enabled.

## 3. **Sort by a Column Value**

Records can be sorted by double-clicking a column title. This action organizes records in either ascending or descending order based on the selected column.

## 4. **View Activity History**

All modifications to data are tracked within the Activity History. Users can access this information by clicking the three dots on the top right of the dashboard and selecting Activity History.

In summary, Appvibe Asset Tracker offers a user-friendly interface for adding and managing asset records at various organizational levels. The dashboard provides essential functionalities such as creating/customizing views, filtering, sorting, and viewing activity history, ensuring comprehensive asset management within Jira.

# FAQ

# FAQ

## **What is the object and purpose of this app?**

The purpose of the app is to manage IT assets such as hardware, software licenses, and peripherals within Jira projects.

## **What types of feedback can the app track?**

The app supports tracking Assets, Asset Type, Serial Number, Location, Assigned User, Purchase Date, and Warranty Status. Additionally, you can add custom fields to track other related data as needed.

## **How do I install the app?**

You can install the app from the Atlassian Marketplace. Navigate to the Apps section in Jira, search for Asset Tracker, click Install, and follow the prompts.

## **How do I get support for the app?**

Support is available via our support page at <https://appvibe.com/support>.

# Support