ATLASSIAN

T4S partners

IT Asset and Service Configuration Management in Jira Service Management

Contents

Part 2: Navigating *Assets* in Jira Service Management

Part 3: Good Practices for IT Asset and Service Configuration Management

Foreword

IT asset management and service configuration management are both critical components for excellent IT service management (ITSM). Similar to other ITSM practices (e.g., incident, problem, change, etc.), IT asset and service configuration practices are well-established and, at the same time, continuing to evolve. This juncture of traditional concepts and innovative, modern tools produces a new landscape for users to explore and develop.

This handbook will describe Atlassian's approach to IT asset and service configuration management based on ITIL 4 principles and ITSM system implementation learnings using Jira Service Management. The goal here is to guide you through IT asset and service configuration management practices and provide inspiration for applying them to different services to give more value to your organization.

You may want to think of this document as a travel guide. Read the sections that match where you currently are on your journey and skip over the parts you already know or have seen. The document will walk you through the basics of IT asset and service configuration practices, describe the structures of ***Assets*** in Jira Service Management, layout a solution for a common ITSM use case, and provide a high-level checklist to implement your own IT asset and service configuration management strategy.

Wherever you are in your IT asset and service configuration journey, this guide should offer useful information.

01

Atlassian’s approach to IT asset and service configuration management

About IT asset and service configuration management

Both are practices designed to help you understand what key business objects you own and how they're being used, so you can make better decisions, improve the efficiency of various processes, and ultimately, save the business money.

What is IT asset management?

IT asset management (also known as ITAM) is the process of ensuring an organization's IT assets are accounted for, deployed, maintained, upgraded, and disposed of when the time comes. Put simply, it's making sure that the valuable items, tangible and intangible, in your organization are tracked and being used.

An asset is anything that is valuable enough to your business that you want to track it. Common IT assets include:

- Laptops
- Servers
- Phones
- Monitors
- Software
- Network equipment

The same asset management principles can apply to non-IT assets. We often see items like office equipment, buildings, vehicles, contracts, and vendors being stored as assets too.

> ***ITIL 4 definition for IT asset - Any financially valuable component that can contribute to the delivery of an IT product or service.***

What is service configuration management?

Service configuration management ensures that accurate and reliable information about the configuration of services, and the configuration items (CIs) that support them, is available when and where it is needed. This includes information on how CIs are configured and the relationships between them. This high-level view is often called a service map or service model, and forms part of the service architecture.

Examples of IT configuration items include:

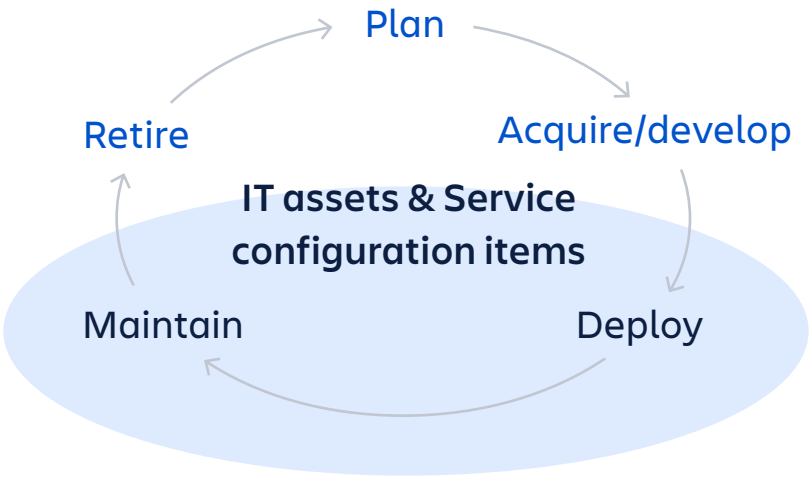
- Laptops
- Servers
- Virtual machines
- Software
- Network adapters
- Databases

Just like with assets, configuration items can expand beyond IT. Examples include employees, procedure documents, vendors, and more.

> ***ITIL 4 definition for IT asset - Any financially valuable component that can contribute to the delivery of an IT product or service.***

IT asset and service configuration management can be thought of as “combined” practices using different perspectives.

- IT asset management is about **content**, understanding what we have and making sure we keep track of our very important stuff, make good purchasing decisions, and other financial considerations.
- Service configuration management is about **context**, understanding the relationships between our very-important stuff and how it all relates together, so we can understand impact.



Customer pain points

What happens if you don't use IT asset and service configuration management?

You can avoid investing in IT asset and service configuration management and not employ the practices at all; however, you will probably pay in other ways:

- Siloed data across different systems and owners, so there is no single source of accurate data.
- Slow response time to customers because staff don’t have quick access to available equipment data.
- Unexpected outages from incorrectly modifying system components, because you couldn't accurately determine which components were impacted by a change.
- Increased costs related to unused equipment and unnecessary licenses/ support.
- Manual effort (months) to determine which system components should change when requirements change.
- Failed implementations because your project's requirements changed, and you didn’t communicate the changes to all parties.

IT asset and service configuration management are included as key practices used by development and operations teams because they work! These practices keep you from incurring costs preventatively and help IT stop fire fighting. Moreover, teams have learned, through practical experience, that these practices pay for themselves many times over by reducing cybersecurity risk and improving operations. Using IT asset and service configuration management allows teams to focus on innovation rather than fighting chaos.

Real-world examples of why IT asset and service configuration practices matter

Qualys response to ProxyNotShell Microsoft Exchange Server Zero-Day Threat

In September 2022, GTSC, a Vietnamese cybersecurity company, reported active attacks against Microsoft Exchange that include two critical vulnerabilities (now named “ProxyNotShell”) in Microsoft Exchange Server via advisories issued by Zero-day Initiative.

The first flaw is a Server-Side Request Forgery (SSRF) vulnerability and the second flaw allows remote code execution (RCE) when PowerShell is accessible to the attacker. When successfully exploited, this combination of vulnerabilities resulted in an authenticated RCE attack.

Threat actors are chaining these two zero-day vulnerabilities to deploy Chinese Chopper web shells on vulnerable Microsoft Exchange Servers for persistence and data theft.

Because of their IT asset and service configuration management capabilities, Qualys provided its customers with the tools to identify and manage potentially vulnerable assets in their environments within hours of the threat announcement.

Link

Qualys Response to ProxyNotShell Microsoft Exchange Server Zero-Day Threat Using Qualys Cloud Platform

New York Stock Exchange (NYSE) glitch

A software glitch prevented the New York Stock Exchange (NYSE) from processing stock trades for almost 90 minutes.

The financial markets felt the impact even beyond the NYSE trading floor. Since investors couldn’t calculate market indexes without NYSE data, trading also stopped at the American Stock Exchange and some futures and options markets. Trading also slowed on the NASDAQ Stock Market, due to investor reluctance to do business without information on NYSE trading.

A new software installation caused the problem. The NYSE had installed the software on 8 of its 20 trading terminals, and the system was tested the night before go-live. However, on the following morning, a total of 8 installations failed to operate correctly. The NYSE tried to switch back to its old software, but was unable to do so before the opening of the trading session.

Although you might see this as a failure of the NYSE's service configuration management process, in reality, it was a success. Although the problem didn't arise until right before the opening of trading, the NYSE had robust service configuration management processes and tools, which identified and recovered from the problem quickly. Other than some red faces at the NYSE, the damage was minimized. Had the outage continued for longer than 90 minutes, the repercussions would have been much more severe.

Link

Software Glitch Halts Trading on New York Stock Exchange

Why does getting this right matter for the evolution of ITIL, ITSM, ESM, and DevOps in your organization?

IT asset and service configuration management are established ITIL practices that help companies plan/manage IT services and deliver value to their customers. However, the ITIL and ITSM practices continue to evolve to support organizations and their digital transformation -- including DevOps which allows companies to develop and improve products at a faster pace than traditional software development and infrastructure management processes.

ITSM is simply how IT teams manage the end-to-end delivery of IT services to customers. This includes all the processes and activities to design, create, deliver, and support IT services. A team's approach to ITSM can be structured to align with ITIL practices and influenced by DevOps concepts for more efficient service delivery.

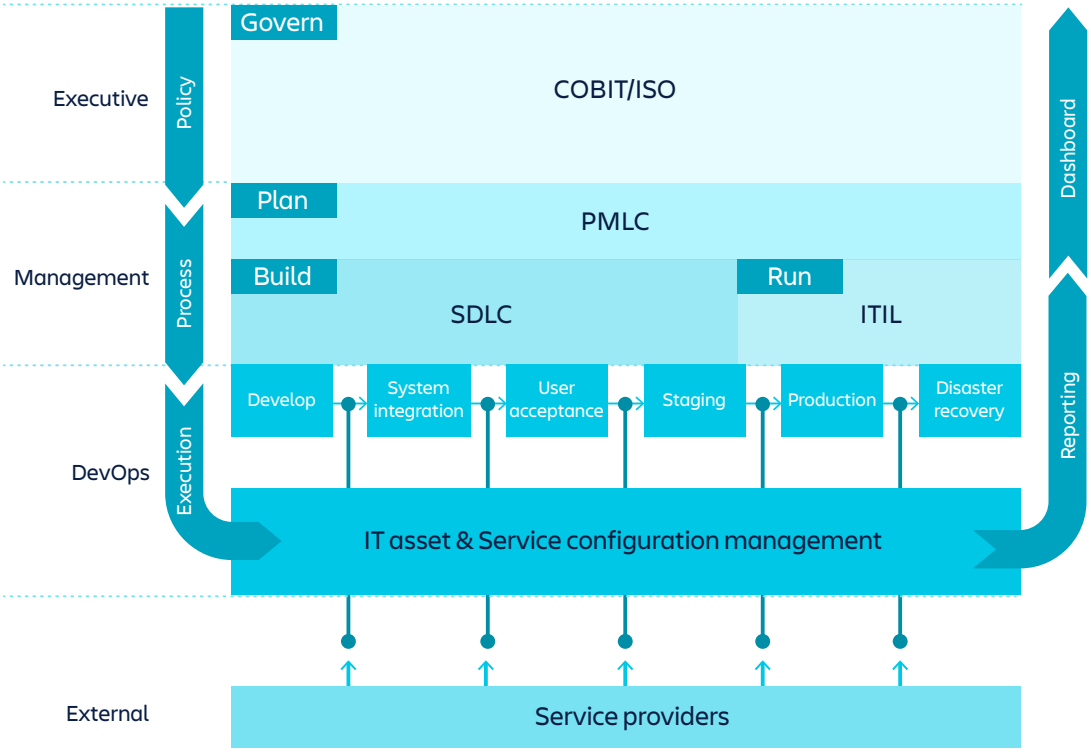
ITSM continues to evolve to meet changing business demands and now comprises flexible and user-friendly service management solutions that support both IT as well as non-IT use cases. Hence, an integrated approach to Enterprise Service Management (ESM) becomes more important than ever. ESM uses ITSM principles and capabilities in business functions to improve their performance, service, and outcomes. ESM improves visibility and access to enterprise services of all forms, accelerates service delivery and of course supports core ITSM processes, such as incident, problem, change, request, and IT asset and service configuration management. The traditional IT service catalog has evolved to focus not just on handling IT tickets, but on providing a holistic, user-centric platform for services such as resource tracking, repair handling and more.

ITIL is the most widely accepted approach to ITSM. ITIL focuses on practices for aligning IT services with business needs. ITIL can help organizations adjust to ongoing digital transformation and scale. ITIL 4, the recent update to ITIL standards, represents a paradigm shift for IT teams. It guides teams to a business- and customer-value frame of reference, and encourages a more adaptable, high-velocity approach based on how your team works. The ITIL 4 Guiding Principles promote collaboration, simplicity, and feedback.

DevOps emphasizes accelerated IT service delivery enabled by updated agile and lean practices. DevOps also focuses on improved collaboration between development and IT operations teams, so organizations can build, test, and release software faster and more reliably. The promised benefits include increased trust, faster software releases, an ability to solve critical issues quickly, and better management of unplanned work.

As you can see from the diagram below, it's critical to think through how you plan, build, run and govern processes and handoffs. When ITIL, ITSM, and DevOps approaches are combined, teams expand their focus from development and delivery of new features to also include the ongoing performance of that functionality when in live operation and the customer value the capability provides.

And all these practices depend on a reliable, accurate source of asset and service configuration data to accomplish their objectives.



IT asset management

Too often, IT assets are tracked in many different places, by many different people. Naturally, chaos and inaccuracy follow, and IT teams can't make informed decisions. As IT evolves, teams become more reliant on SaaS (Software as a Service) vendors for critical services, and it's necessary to track the consumption of "on-demand services" in dynamic cloud environments. IT asset management must adapt from spreadsheets to more effective, modern practices. With increased control, visibility, and assigned responsibility, teams can reduce excess consumption, including over-provisioning and idle instances, to avoid unnecessary costs. A recent ITAM Review report indicated that computer hardware is still the top IT spending category, accounting for 30% of overall IT budgets, which is why IT asset management is crucial to master.

Service configuration management

In an era of cloud computing and anything as a service, IT teams are now managing a very different type of IT environment. While they may rely on a Configuration Management Database (CMDB), many IT organizations struggle to find value from their CMDB implementations and have even experienced failed CMDB projects. They're not alone. According to a Gartner report, 75% of CMDB initiatives fail.[1] The reason stems from starting a CMDB deployment with too wide of a scope. As a result, teams attempt to collect large amounts of information (valuable or not) upfront and struggle to maintain and keep it current. The deployment ultimately shows little value for the organization and, instead, results in lengthy projects and wasted resources.

According to *Foundation ITIL 4 Edition* (Axelos Limited, 2019), "It is important that the effort needed to collect and maintain configuration information is balanced with the value that the information creates. Maintaining large amounts of detailed information about every component, and its relationships to other components, can be costly, and may deliver very little value. The requirements for service configuration management must be based on an understanding of the organization's goals, and how service configuration management contributes to value creation."

Decision making requires data…effective decision making requires reliable data.

Accurate information regarding system assets and configurations improves request fulfillment, service delivery, audit processes, as well as software development and debugging. A Forrester report emphasized the benefits of

1 Gartner, Inc. "Break the CMDB Failure Cycle With a Service Asset and Configuration Management Program." Published 5 May 2020.

a CMDB in providing high-quality services and support, and the economic benefits this yields for a business.

Benefits of IT asset and service configuration management include:

- Reduced risk of outages and security breaches through visibility and tracking of the changes to your systems.
- Cost reduction by having detailed knowledge of all the elements of your configuration, avoiding wasteful duplication of your technology assets.
- Improved experience for your customers and internal staff by rapidly detecting and correcting improper configurations that could negatively impact performance.
- Greater agility and faster problem resolution, enabling you to provide a higher quality of service and reduce software engineering costs.
- Efficient change management by knowing your baseline configuration, and having the visibility to design changes that avoid problems.
- Quicker restoration of service. In an outage, you'll be able to recover faster because your configuration is documented and automated.
- Better release management and clear status accounting.

Today's enterprises rely on increasingly complex technology environments, with IT assets ranging from software to purchase orders to laptops or servers. With IT asset and service configuration management software, you can better track IT assets and service configurations in your inventory, minimizing delay and human error. When a new device configuration is discovered or when an IT asset's contract is close to expiration, you can receive actionable alerts designed to provide a real-time understanding of your IT asset inventory.

Return on investment for IT asset and service configuration management

Some of the financial benefits contributing to positive ROI results found in leveraging IT asset and service configuration management include:

- IT cost reduction. Optimization of IT operations reduces costs in multiple areas, including infrastructure, outsourced services and management software.
- Service quality improvement. Ensuring that existing services are available

at any time and new/enhanced services can be released quickly.

- Risk reduction. Reduction of downtime caused by system outages, cyber attacks, security intrusions, and change and configuration activities.
- IT staff productivity increases. Optimization of IT staff activities through automation reduced IT staff time spent "keeping the lights on", freeing up valuable staff resources for business-related initiatives.

There are plenty of ROI calculations that you can apply for your business. The metric is cost avoidance in areas such as:

- The number of devices tracked and monitored by an IT asset and service configuration management system.
- The effort and cost for a system/network engineer to handle IT asset and service configuration management processes manually.
- When (not if) a system outage occurs with no backup configuration.
- A bulk configuration update to many systems, or a new required rollout.
- When your business has to comply with an IT asset or service configuration audit request, or pass a technology risk assessment.

The benefits of IT asset and service configuration management flow into all these activities. These activities take time, and time is money.

Why IT leadership values IT asset and service configuration management

Adoption of IT asset and service configuration management practices provide necessary visibility into an organization's technology landscape. IT asset management can be thought of as the "universe" of technologies, and service configuration management provides in-depth transparency into each asset therein.

These processes enable organizations to not only respond to security threats, but also run IT Operations effectively. When a cybersecurity event, like BlueKeep, occurs, the first question we ask ourselves is ... "what is impacted?". Not having a quick, definitive, concise answer spins up another unwarranted crisis - a desperate search for the "right answer". At that time, rightfully so, everyone volunteers ... causing more chaos, because everyone brings forward a different version of the "truth". At the end, when all the crises have been overcome, we reflect to realize that the price-tag - the cost of business disruption, overtime, extra work, vendor fees, etc. - is enormous and unaffordable.

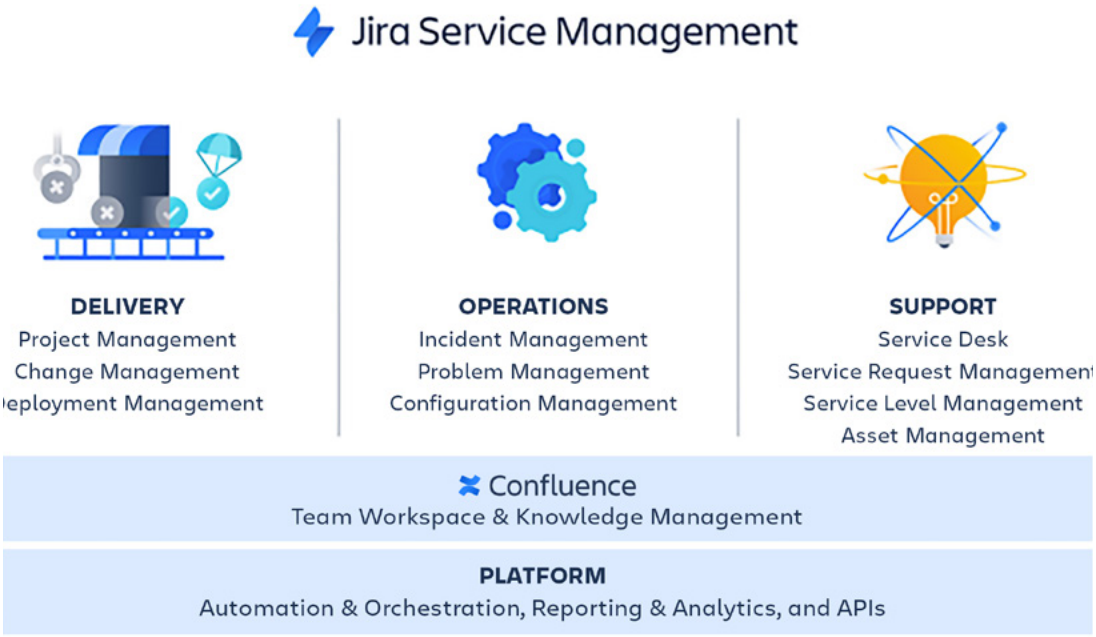
Because the technology landscape is constantly evolving with adoption of new tools, our IT asset and service configuration management capabilities need to continuously improve and adapt to changing operational/business needs. And given the dependency on organizational collaboration, agile capabilities need to be supported by an effective framework to drive expected outcomes and continuous improvement.

The Atlassian approach

Atlassian's approach is to balance autonomy with alignment. We want teams to have the flexibility to run fast and operate with autonomy, while ensuring IT feels confident that work is aligned and doesn't introduce risk to the business.

Atlassian understands that every organization is different. Maybe you need to map complex dependencies across an enterprise. Or you want to keep a record of intangible assets like licenses and compliance documents to reduce risk. Or perhaps your requirements are more straightforward and involve tracking an inventory of computers.

Assets is built on the Jira Service Management platform and provides scalable IT asset and service configuration tracking functionality to meet your high-velocity team's needs. ***Assets*** combines asset repository and CMDB capabilities required to effectively manage asset and CI data. Whether you're looking for a lightweight asset tracker or an enterprise-grade system, ***Assets*** in Jira Service Management empowers you to define your assets how you like, work with them in whatever way suits you and your business best, and provide a platform for extending system monitoring and maintenance through automation.



Atlassian's approach allows teams to unite on one platform: Bringing delivery, operations and support into one collaborative experience. ***Assets*** in Jira Service Management provides IT, development, and business teams with visibility across critical business systems and enables collaboration regarding priorities and resource allocation.

With Jira Service Management on the same platform as Jira Software, all assets and related issues are stored in one place and teams can easily understand how assets relate to their workloads. You'll know the reason for acquiring the hardware, who it's assigned to, and its past history. Whether it's a trouble ticket, new hire requisition, purchase order, Jira Service Management and Jira Software enable seamless communication, visibility, and reduced friction between dev and IT teams.

Teams can:

- Better respond to service requests by gaining greater context of issues
- Minimize IT risk by understanding the downstream impact of changes
- Troubleshoot and resolve major incidents and problems faster
- Track IT resources and gain visibility into the relationships between critical applications, services, and the underlying infrastructure
- Discover and track assets which aids with planning, audits, and compliance
- Manage assets outside of IT, including in HR, sales, legal, facilities, and other functions

The best performing IT teams typically use the following practices.

EMBRACE A TEAM-CENTRIC APPROACH

Open teams work better together. Many IT teams believe they're using the "right" tools and following the "right" processes, but still fail to achieve results. In fact, these tools and processes can actually create inefficiencies, for example, between various IT Ops and Dev teams due to silos and lack of knowledge sharing. Atlassian found that establishing a culture around collaboration and transparency is the foundation to a successful IT asset and service configuration management implementation. By using the Atlassian suite, you're already one step closer to leading your organization toward open knowledge sharing. Open and collaborative culture is infused in the Atlassian toolset.

STEP BACK, AND START WHERE YOU ARE

As you define your organization's culture and practices, ITIL 4's Guiding Principles are a great place to begin. One of these principles is, "Start where you are." With 34 ITIL 4 management practices to consider, this can feel overwhelming. Instead of building from scratch, take a moment to observe and analyze the services, methodologies, people, and tools you already have. Then use these insights to identify where to start and what to continue, change, or build upon.

TAKE A TOP-DOWN APPROACH STARTING WITH THE SERVICE LAYER

When beginning an ITSM deployment, the idea of fully defining your service model down to the infrastructure can be paralyzing. Instead of diving into infrastructure and microservices out of the gate, focus on the top services most critical to your business (such as an e-commerce platform if you're a retailer). To identify these services, review tickets from the past few months to understand which services are most utilized.

ACHIEVE QUICK WINS WITH A MINIMAL VIABLE PRODUCT

For many organizations, getting employees to embrace change can be difficult. Maximize your chances for success by taking an agile approach to deploying your ITSM solution. Instead of rolling a full-blown solution at once,

identify your organization's biggest pain points, and focus on the practice, service, or use case that will be most impactful. By starting with a minimal viable product (MVP) and iterating on the solution over time, you'll help your organization overcome the fear of change while satisfying a significant portion of your stakeholders.

MATCH YOUR SOFTWARE STACK TO YOUR MATURITY AND NEEDS

In their 2022 Buyer's Guide for ITSM Platforms, Gartner predicted that "I&O leaders will overspend by $2 billion on buying unused features of ITSM platforms in 2026, up from $1 billion in 2021." [2] Instead of committing upfront to a costly ITSM platform with complex features you'll never use, take an adaptive approach to build your solution. The needs of your business are constantly changing – so buy only what you need. Atlassian's ITSM solution offers out-of-the-box ITIL practices with the flexibility to scale as you grow. And, our broad ecosystem of Marketplace apps allows you to customize and extend your capabilities, without the need for specialized consultants.

SCALE YOUR SOLUTION AND CELEBRATE SUCCESS

As you continue on your journey, communication is critical to increasing adoption. Once a service or practice is up and running, shout it from the rooftops. Offer hands-on training, pass out stickers, and incentivize usage through contests. Customers have found that after adopting Jira Service Management for IT, business teams, from HR to Legal, begin to realize the value and request service desks of their own. To manage and scale this growth, treat each request as an endeavor of shared objectives. Seek first to understand the problem each team is facing, and solve it in a consultative manner. Finally, don't forget to celebrate each milestone with your team!

2 Gartner, Inc. "A Buyer's Guide to ITSM Platforms." Published 4 August 2022.

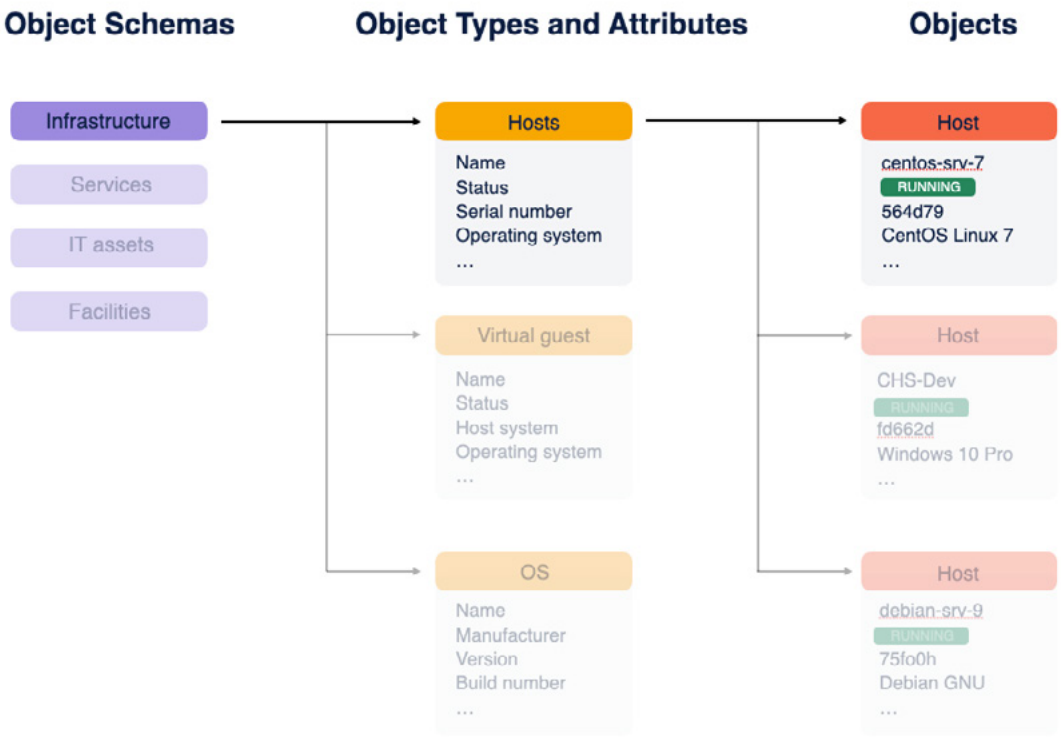
02

Navigating ***Assets*** in Jira Service Management

Key terms

Assets in Jira Service Management is a versatile tool that records and maps the relationships and dependencies between your assets, CIs and services. ***Assets*** functionality clears away the unneeded boundaries between asset and service configuration management. It's a database of objects – digital representations of your assets, ranging from hardware and software to employees or various CIs. You can make your objects whatever you need them to be, and have them displayed in and be affected by issues in Jira Service Management, and even Jira Software.

Assets in Jira Service Management is comprised of a few basic entities: object schema, object types and attributes, objects and their relationships.



Object schema

An **object schema** is a collection of information used to track assets, CIs, and resources, and to understand and visualize the critical relationships between them.

Each object schema holds unique information in the form of object types, objects, attributes, icons, references, and statuses. An object schema also has its own set of permissions and automation, which allows you to hide or show different information and perform various actions for different users or groups.

Object schemas work like maps that hold all of your assets, CIs, and resources together. You can have many object schemas, and refer to objects inside them from your issues and requests.

Note: The Services object schema is a special case – it contains services that your site uses across multiple projects. The Services object schema will be covered in a later section.

PRO TIP

When creating object schemas, you should consider the following:

- Which groups will access, own, and maintain the data? For example, if the IT team updates server data and the HR team updates employee info, you should create different object schemas for these data types.
- How is the data acquired? For example, if phone data is based on a data feed from an external vendor, then the data should be tracked in a separate object schema.
- How is the data used? If object schema data is used similarly across service projects, the data should be stored in a single schema object. For example, if multiple projects reference departments, then the data should be stored in a single object schema.

Object type

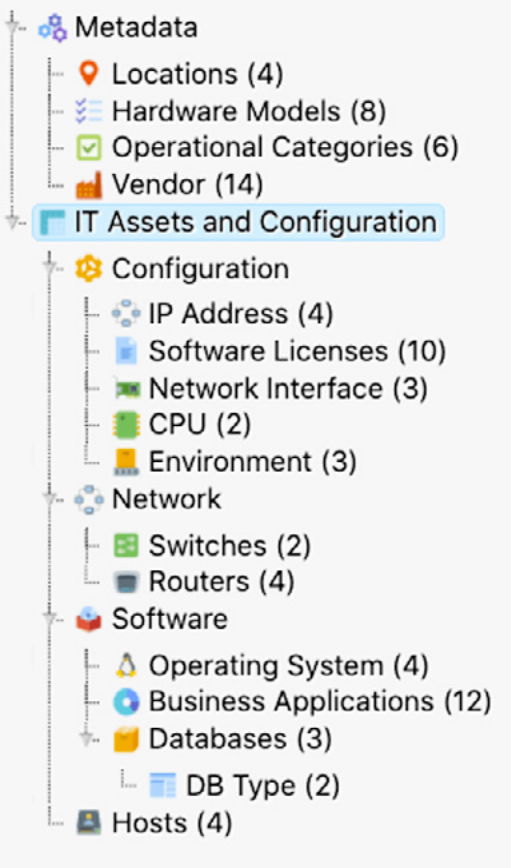
An object type groups objects that use the same kind of information, conveyed through their common attributes. Rather than a single PC, your object types would be Computers, Hardware, Software, Employees, and so on. You can create as many different object types as you like, and then group your various objects within. Object types can be whatever you want them to be because ***Assets*** is very open and flexible.

Common object types include:

- Business services
- Hosts
- Laptops
- Software

But they don't have to be IT assets. For example, many people add other useful information, such as:

- Vendors
- Locations
- Employees
- Medical equipment

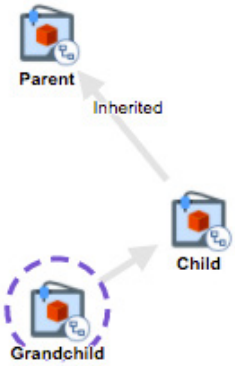


PRO TIP

When creating object types, you should:

- Use unique object type names within an object schema.
- Use less than 30 object types in an object schema when possible. This will simplify the screen and make maintaining the data easier.

Additionally, you can configure object types to inherit attributes from their parent object types. This is useful if you need to create multiple object types, nested in one another, and want them to keep the same structure. Each object type has its own fields, but also “inherit” the fields from its parent.



A good example for inheritance is an object type Host, with child objects Linux Host and Windows Host, which inherit common attributes, such as IP Address and Host name.

Generally, when an object type is a child or sub-type, it inherits all of the attributes of the parent type and then adds a few attributes of its own.

Additionally, you can set some object types as abstract, meaning that they can’t contain any objects of their own but can pass their attributes to their children, who can themselves contain objects. Inheritance and abstract object types can be used to create object schemas that are both simple and powerful.

PRO TIP

Through simple drag-and-drop, you can organize object types into a tree hierarchy in a way that makes sense for your organization. This tree is mainly for navigation and readability.

Attribute

An **attribute** represents a specific piece of information that is attached to an object, such as a description of that object, its model number, another associated object, or a user assigned as the object's owner. Every object includes four default attributes: an attribute set as the object's label, the object's key, the date and time the object was created, and the date and time the object was last modified.

A label is the title of an object that appears wherever an object is referenced. The label of an object type is marked with the label icon in the **Attributes** view of the object type. The default label is the attribute "Name."

Attributes can hold many different types of information – text values, numerical values, or even references to other objects. The list below includes attribute types:

Default – Represents text, boolean, integer, float, date, datetime, URL, email, text area, select, IP address, etc.

Object – Enables reference to another object

User – Enables selecting a user from a Jira group and connecting objects to users

Group – Enables selecting a Jira group and connect objects with user(s) in specified Groups

Project – References a Jira Project to your objects.

Status – Defines the statuses that should be allowed, and left empty means all statuses allowed.

They can also be customized to hold very specific information, such as a postal code, a certain pattern of strings, an object of a particular type, or a mandatory value. Additional attribute configurations include:

Unique - Validate attribute values to be unique within the object type.

Cardinality - Specify a minimum and maximum number of attributes values that can be associated to the attribute. This is common when you need to set an attribute multiple choices or required / mandatory.

Validation - Validate attributes of default type "Text", "Email" & "URL" with regular expressions. This can be handy if you want to validate specific information, like an IP address, a domain name, a phone number, or anything else that may require validation.

Options - You may add multiple Options to a "Select" Type Attribute by adding them as options.

Suffix - For default type "Integer" and "Float" you can set a suffix for the attribute. Example is "$" for an attribute "Salary"

Show sum - For default type "Integer" and "Float" you can choose to add the values and display the sum of the attribute values

Filter objects - For attribute of type "Object" you may filter objects to be selected by AQL (Assets Query Language). By using this filter it is possible to create dependencies to other fields when creating/editing objects.

Include children - For attribute of type "Object" you may include child objects in the reference

PRO TIP

If an attribute is used in many places and has the same repeated values, it may make more sense to create a separate object type. For example, you may have an attribute for Vendor in the object types for Laptop, Phone, Printer, Monitor, etc.; for each object, you will type (or import) the vendor name for that particular laptop or phone. While this method works fine, it's more efficient to have an object type called "Vendor" and set each vendor as an object for a number of reasons:

- You may want to track additional information for vendors, such as a support contact number or links to contracts. Rather than repeating this data for every laptop or phone, you can simply link to the vendor object. This also helps if you want to perform elements of vendor management within Jira Service Management.
- The Vendor will be standardized this way, meaning reports are easier to run. If you wanted to report on the number of support requests per Vendor, you can be confident you're not missing something because someone wrote Micrsoft or Aple somewhere.
- If the Vendor rebrands or needs to be changed in some way, then you only need to update it in one place.

Vendor is just one example but others include business importance levels, deployment environments, departments, and locations.

Object

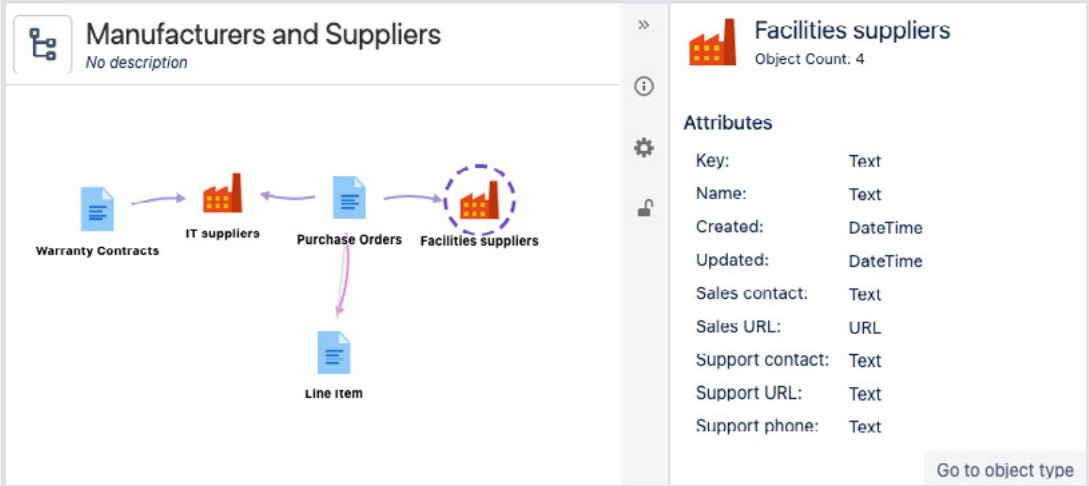
Objects are your actual assets or CIs. Every object is a digital representation of anything that you're mapping, be it a specific computer, employee, office they work at, or even a license for your software. You can create as many objects as you like, and group them within object types that represent their characteristics (an employee wouldn't be the best fit for the hardware object type).

Using object actions, you can keep your physical assets in check by printing labels and QR codes, view their dependencies with other objects, and see which Jira issues they're on.

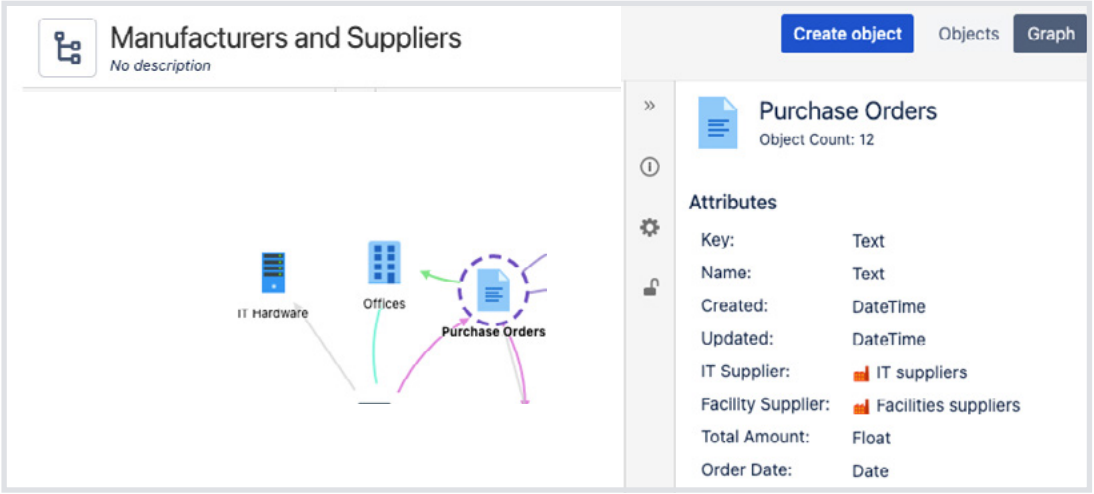
Reference

A **reference** is a connection between two different objects. Each object can be related to many other objects and dependencies defined resulting in a dynamic and powerful network of assets and CIs. References have a color and a name for better identification. This capability helps users to have meaningful graphs during impact analysis or dependency mapping, etc.

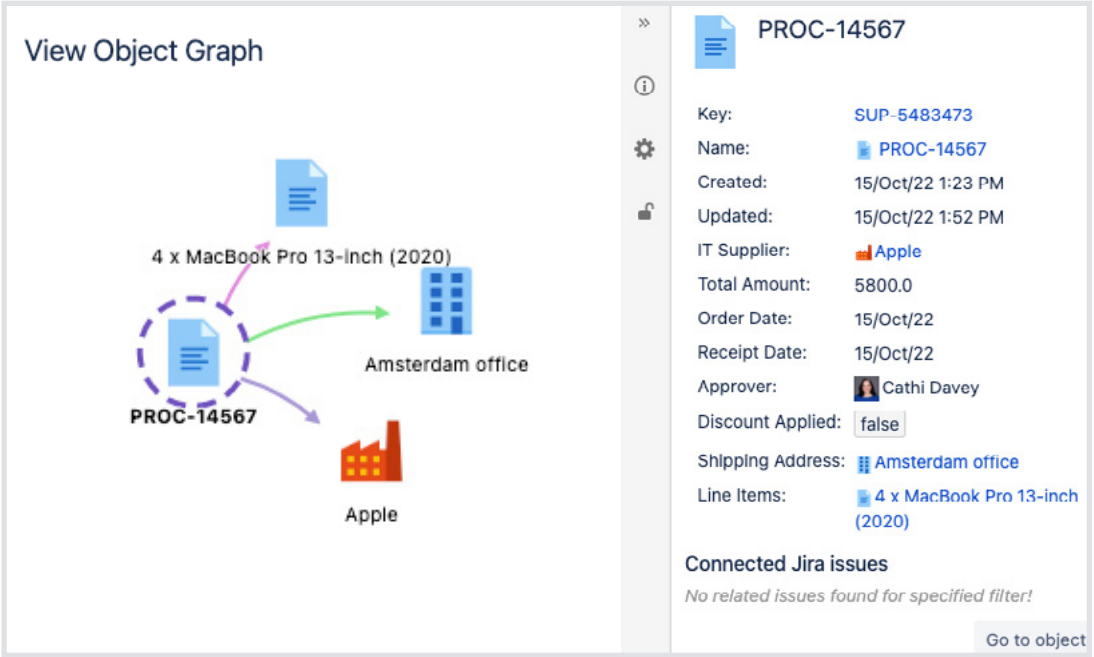
Object schema graph - see how all object types are knitted together.



Object type graph - see an object type's relations with other object types.



Object graph - see an object's relations with other objects and object types.



Because each reference is a link between an object and an attribute on another object, they are divided into two types: outbound references and inbound references.

- Outbound references point from the current object to another object (e.g., from printer asset to cubicle location)
- Inbound references point from another object towards the current object. (e.g., from stockroom to all assets stored in the location)

The direction of a reference is relative; it will change depending on which object you are examining. Additionally, each reference can have a 'Reference type', which describes the type of relationship between two objects.

PRO TIP

When defining ***Assets*** data structure, we recommend building meaningful sentences and setting the Object Type, Attribute and Reference Names accordingly. For example:

"The printer is located in the room" is transformed into:

Object Type: Printer

Reference Name: Located in

Attribute Name: Room

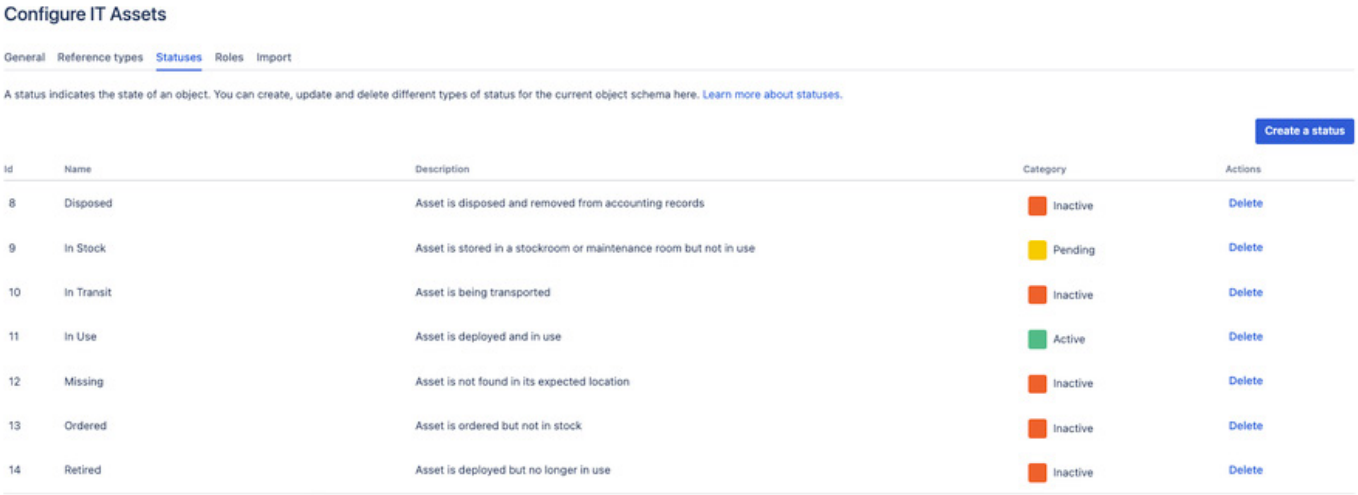
Status

A **status** is a discrete state that could apply to an object. For example, a server could have the status “Running” or “Stopped” depending on if the server is working or not.

Assets in Jira Service Management includes a set of default statuses, but you can also add new statuses to represent the different states of objects in your environment.

Statuses can be global, or they can apply only to a specific object schema. Each status includes an optional description and a general category - active, pending, or inactive.

The status category is especially useful in tracking asset/CI lifecycles and developing automation.



Role

A **role** is a set of permissions granted to Jira users or groups to view or modify data in ***Assets***. Roles can have three levels:

Global - Allows you to configure the entire ***Assets*** application (Jira admin)

Single object schema - Allows you to configure and execute actions on the object schema level and all object types within that schema (Object Schema Manager, Developer, or User)

Single object type - Allows you to execute actions on an individual object type (Object Type Manager, Developer, or User)

Role	Description
Assets Administrator	This role can perform all actions in ***Assets*** in Jira Service Management. This includes: • Configuring ***Assets*** globally • Managing individual schemas Note: ***Assets*** Administrator is a role given to all Jira Administrators by default. It is not possible to revoke this permission.
Assets Managers	This role can execute the following tasks on an object schema. This includes: • Configure all schema details • View objects • Search objects using basic and advanced search Create and edit objects • Export objects • Print QR codes and labels
Assets Developers	This role means builders or creators, rather than e.g. software developers. It can execute the following tasks on an object schema: • View objects • Search objects using basic and advanced search Create and edit objects • Export objects • Print QR codes and labels
Assets Users	This role can do the following on an object schema: • View objects • Search objects using basic and advanced search Export objects • Print QR codes and labels

Special permissions apply to users when viewing or editing object custom fields (which we'll go over soon):

- Any user - even those who are not licensed for Jira Service Management or any Atlassian products - is granted a temporary "User" role when an object custom field is added to a request type which can be accessed by end-users on a portal. This allows them to view the object fields and their values.
- Jira Software, Jira Service Management, and Jira Work Management users can have temporary "Object Schema User" roles that allow them to view and edit the contents of an object custom field within issues where they already have existing edit permissions.

These roles will not count towards the total number of users on your license.

PRO TIP

- Object type permissions take precedence over object schema permissions. For example, a user might be assigned to the User role on the schema (to not make any changes), but then have a more powerful Developer role on a specific object type, so they can create objects.
- If you want users to work with object types (and objects) you need to grant them permissions for object types, but also for the schema (at least User permissions, so they can view it). Without giving them any permissions for the schema, they won't be able to access it.
- If you don't specify permissions for object types, they will be inherited from the object schema.

Service registry

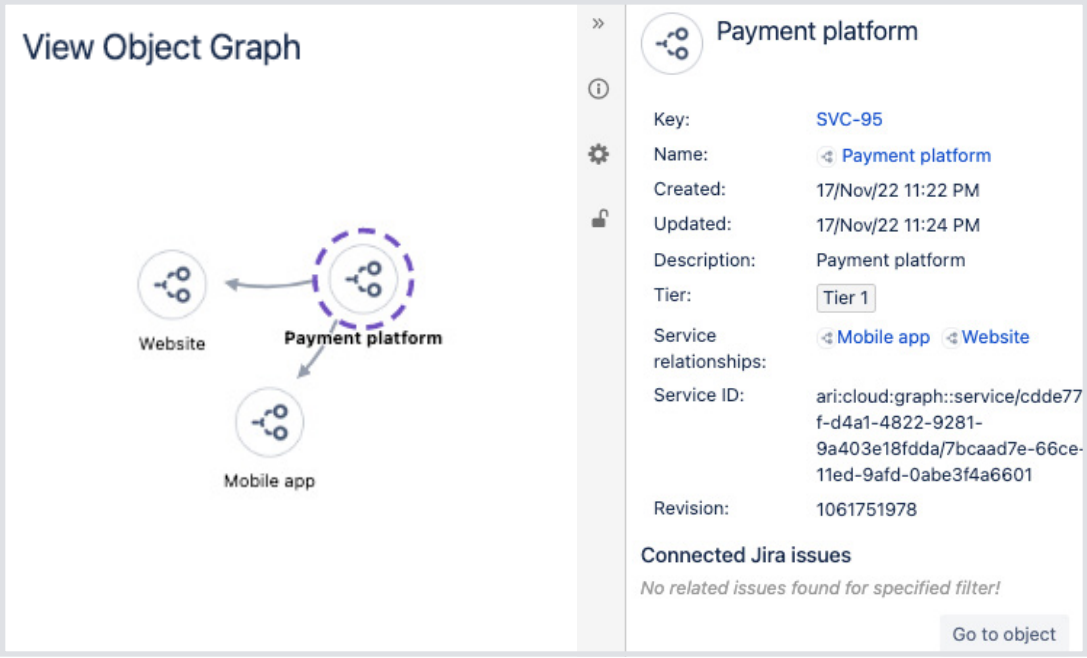
In ITSM, a service is a system, platform or infrastructure that provides value to your business or customers. Services can include things like payment platforms, servers, teams of people (for example, a legal team), websites, products, or application stacks.

In ***Assets,*** you map, organize, and manage your services in the Services object schema. Services behave like 'connectors' in Jira Service Management and can be applied to your entire Jira site and used across all of your service projects.

For example, let's say you set up three services in Jira Service Management: payment platform, website, and mobile app. You can set up their relationships: the website and mobile app both depend on the payment platform. Now, a change for the payment platform will include the mobile app and website as affected services.

Link

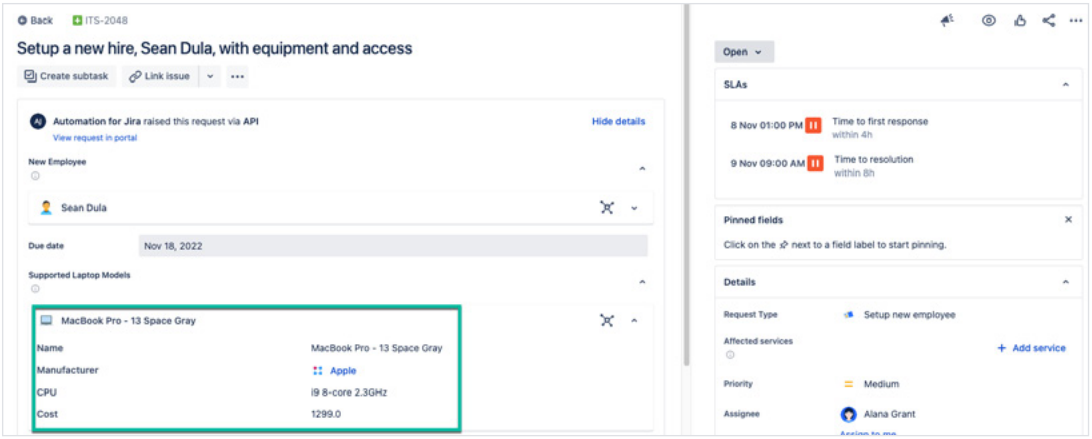
How services work with *Assets* in Jira Service Management



Assets custom fields

In addition to the service registry, you can create ***Assets*** custom fields which allow your team to access assets directly from the issue view. This is a powerful feature that can help your agents get the context they need to resolve issues or requests quickly and effectively.

Using a custom field creates a link between an issue and an object. Adding an object (i.e., as a value) to the field allows you to see all of the connected issues from the object view.



This is useful for incident management because you can use the graph to traverse through dependencies and understand where things have gone wrong. It's also useful for change management because it allows you to see the bigger picture and evaluate risk - easier to do when you can see what depends on the item you're making changes to.

Link

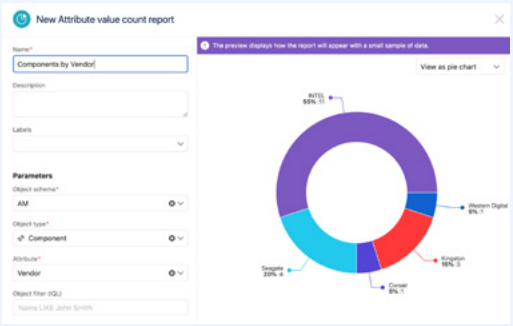
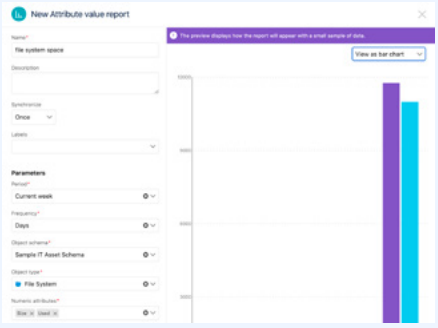
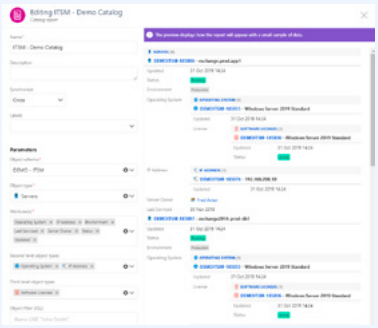
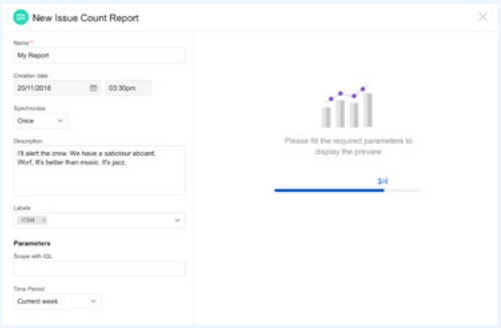
What is the Assets objects field? | Jira Service Management Cloud | Atlassian Support

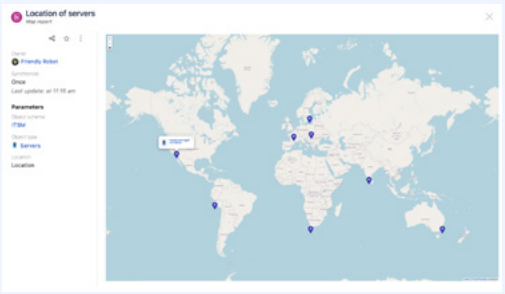
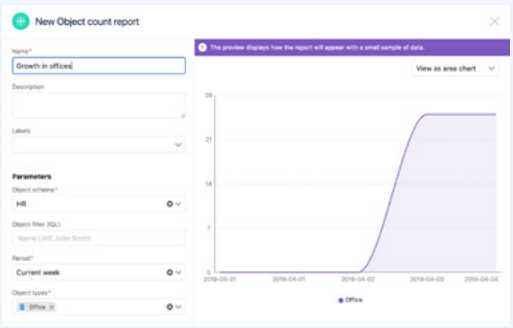
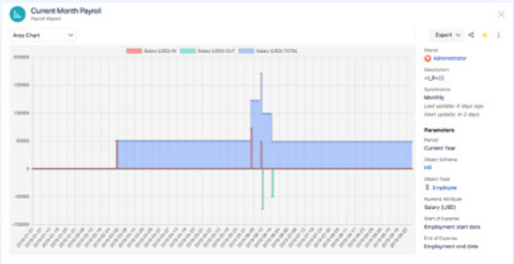
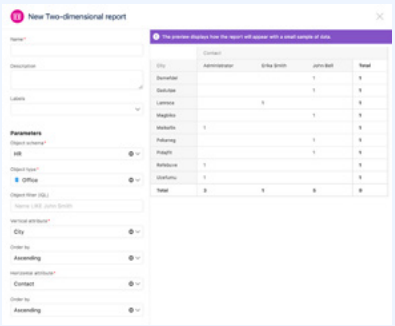
Assets reports

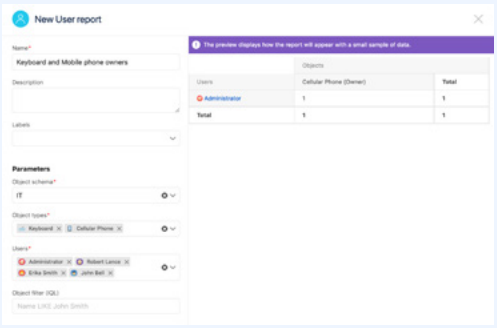
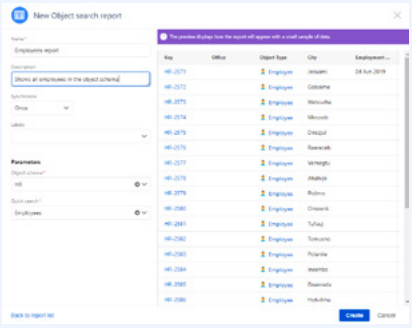
You can use dynamic reports to view information about assets in ***Assets*** in Jira Service Management object schemas (asset-based reports) or Jira issues with asset custom fields (issue-based reports). For example, you can use asset-based reports to monitor the health of your asset/CI data and assess the accuracy and completeness of the data. Issue-based reports summarize Jira issues that include asset-related data (e.g., incidents impacting laptop assets).

Atlassian Analytics provides asset-based and issue-based reporting capabilities for **Jira Service Management Cloud** users (Enterprise only). Additional issue-based reports are available in the cloud environment through out-of-box Jira Service Management reporting or integrations with analytics tools available through Atlassian Marketplace.

The following asset-based reports are available with ***Assets*** in Jira Service Management Data Center/Server application. Out-of-box issue-based reporting is also available in Jira Service Management.

Report	Description	Example
Attribute value count report	Pie chart showing how the objects of an object type are distributed based on attributes.	
Attribute value report	Chart showing attributes totals reported for one or more objects of a specific type.	
Catalog report	Catalog of objects and their attributes.	
Issue count report	Line or bar chart showing historical totals of issues reported for one or more object types or objects.	

Report	Description	Example
Map report	World map showing the location of objects based on an attribute with geographical data. The objects must have a text attribute with a comma-separated latitude and longitude, for example -33.8,151.2 (Sydney).	
Object count report	Line chart showing historical totals of objects for one or more object types.	
Payroll report	Chart showing payroll expenditures.	
Two-dimensional report	Returns a table showing the objects of an object type and their attributes.	

Report	Description	Example
User report	Returns a table showing the total of one or more objects assigned to one or more users.	
Object search	Report is based on a saved quick search (filter) from the ***Assets*** Object Search View. It's a table showing the objects of an object type and their attributes. When you open an object search report, you can select the Quick search link to view the filter that generates the table.	

If you use ***Assets*** in Jira Service Management and include all of its data in an **Atlassian Data Lake** connection (along with Atlassian Analytics), you can utilize the starter dashboard that comes with that Data Lake connection; see the link for details:

Link

Starter dashboards for asset and configuration management | Atlassian Analytics

Automation

You can drive efficiency with automated actions that follow conditional rules. Rules allow you to automate actions within your system based on criteria that you set. Automation rules are made up of three parts:

- Triggers that kick off the rule
- Conditions that refine the rule
- Actions that perform tasks in your site

You can create automation rules that automatically perform actions (for example, notify users) based on specific events (for example, object updated) for all objects, or only a group of objects in a schema. When an asset meets certain criteria, you can update an asset, create issues, send email notifications and execute a script or an HTTP request.

The following Jira automation components are available for ***Assets***:

- Create issue (If the rule is triggered by an object, you can select Insert object to automatically add the triggered object via AQL)
- Edit field attributes
- Edit object
- Lookup objects

Rules can be tailored to your needs using the Asset Query Language (AQL). Use post functions to trigger automatic actions that follow Jira workflows. For instance, automatically assign the issue to a service owner upon confirmation. Leverage powerful automation tools to further slash down resolution time and boost operational efficiency.

For more information, visit the following links:

Use Assets Query Language (AQL) | Jira Service Management Cloud | Atlassian Support

Jira Software Automation: Basics | Atlassian

Jira automation actions | Cloud automation Cloud | Atlassian Support

Assets Discovery

Assets* Discovery** is a network scanning tool that can be used with or without an agent. It detects hardware and software that is connected to your local network, and extracts detailed information about each asset. This data can then be imported into ***Assets to help you manage all of the devices and CIs within your local network. You can choose which assets, and which attributes, you pull into your object schemas and you can create your own scanning patterns to find more specific details. If you run it on a schedule it will pick up changes to keep data updated. With automation rules you can even trigger Jira issues, email notifications, and more based on detected changes!

Assets Discovery is free of charge and includes three separate tools:

- ***Assets*** Discovery is an agent-less scanner to help you discover devices and CIs in your local network.
- ***Assets*** Discovery Agent is an agent-based scanner that can help you discover data from systems that are not always online, or collect data from Windows systems without opening the inbound WMI Port and the Dynamic DCOM Ports.
- ***Assets*** Discovery Collector is a tool that allows you to run multiple instances of Discovery in parallel, or to run a scan remotely and transfer the resulting data to a different location.

For more information visit the following link:

What is *Assets* Discovery? | Jira Service Management Cloud | Atlassian Support

Integrations and data imports

Ensure your Jira Service Management system can scale with a single source of truth that always provides an accurate, real-time picture of your infrastructure.

- Leverage a wide range of free integrations to sync with industry-leading cloud services, asset managers, and other CMDB tools and applications
- Complete data imports in a variety of formats.

Integrations and import functions allow you to connect all the dots and work with up-to-date information that grows with your business. ***Assets*** provides a solution to federate an array of data repositories and link together all the data about an IT resource.

Assets has several built-in importers that let you import your data from CSV, database, JSON, LDAP, and so on. You'll use these importers by creating an import configuration, specifying its details, and then mapping the data you're importing to object types and attributes. Such an import configuration can be then synced on a regular basis, so your assets stay up to date.

For more information visit the following link:

What are imports? | Jira Service Management Cloud | Atlassian Support

Additionally, ***Assets*** integrate with industry-leading tools such as AWS, Google Cloud, Azure, Jamf, and SCCM. You can also migrate from and connect with third-party applications like ServiceNow, Device42, Snow, and NVD.

While Jira Service Management has all these tools, it's not recommend you bring in every bit of data you have into ***Assets*** unless you plan to depreciate the tool. Bring in what you need to use in Jira Service Management, you can always bring in more later.

Jira Service Management plans and *Assets* functionality

Assets in Jira Service Management is built on the Jira Service Management platform, so teams can quickly and easily tie assets/CIs to service requests, incidents, problems, changes, and workloads.

Unlike legacy CMDB applications, ***Assets***' flexible and open data structure allows teams to manage any kind of asset that's important to support their ITSM practices. HR, sales, marketing, legal, facilities, and other functions can also use ***Assets*** to track and manage their assets and resources.

Assets in Jira Service Management is a Premium and Enterprise only feature. It is compatible only with company-managed projects.

Feature	***Assets*** **in Jira Service Management Data Center (v 4.15 and later)** ***Assets*** **- Jira Service Management Data Center/Server app**	***Assets*** **in Jira Service Management Cloud Premium or Enterprise**
Objects and object modeling		
Object schemas Use object schemas to organize the structure of your object types, objects, and attributes.	✓	✓
Object schema templates Create an object schema from a template that contains a pre-created object types and objects.	✓	
Object types Use object types to specify assets/CIs by defining attributes and references between other object types.	✓	✓
Objects A specific instance of an object type. E.g. 'Laptop' would be an object type and 'MacBook-4523' would be an object.	✓	✓
Number of objects	Unlimited Subject to the performance parameters you set for your Jira Service Management environment.	1 million
Attributes Use attributes to manage what kind of information is stored for each object type.	✓	✓
References Use references to define how objects are related to one another.	✓	✓

User Roles Use roles to manage object schema permissions for different users and groups.	✓	✓
Data, Importing, & Exporting		
***Assets* Custom Field** Select objects from fields in Jira issues.	✓ Via ***Assets*** custom fields	✓ Via ***Assets*** custom fields
Imports - CSV Bring data into ***Assets*** from CSV and JSON files.	✓ Normalization for duplicate entries.	✓ Normalization for duplicate entries.
Imports - JSON CSV Bring data into ***Assets*** from CSV and JSON files.	✓ Normalization for duplicate entries.	✓ Normalization for duplicate entries.
Imports – Databases, LDAP, and Jira Users Bring data into ***Assets*** from external databases, Active Directory, or from the Jira environment itself.	✓	This is a priority area for future development cycles.
Asset Discovery Network scanner that can be used to discover IP-enabled assets/CIs and bring them into Jira Service Management.	✓	✓

Integrations Integrate with a third party tools to keep data up to date. Includes: • Cloud providers (AWS, Azure, Google Cloud) • Mobile device and software management (JAMF, SCCM, Snow) • Other CMDBs (ServiceNow, Device42) • Atlassian ecosystem (Jira & Bitbucket, Confluence, Tempo) • Others (NVD)	✓	This is a priority area for future development cycles.
Export objects Export data from ***Assets*** as a backup or to be used in other systems.	✓	✓
Reporting, viewing, and searching		
Object graph Use the object graph to view the relationship and hierarchy between different objects and object types.	✓	✓
AQL search Use AQL (Assets Query language) to search ***Assets*** for specific objects (e.g. what computers are not assigned to a user). Note: this was previously IQL (Insight Query Language) and will continue to function following the rebrand to ***Assets***.	✓	✓

JQL search Use the ***Assets*** JQL function to search for Jira issues that have objects linked to them.	✓	✓
Bulk edit objects Make changes to a large number of objects at once.	✓	✓
Reports View your ***Assets*** information in myriad different ways.	✓	✓ Via Atlassian Analytics (Enterprise only) Via integration with analytics tool available from Atlassian Marketplace
Widgets Use an ***Assets*** widget to view ***Assets*** information within a Jira Dashboard or on a Confluence page.	✓	
Print QR Codes Generate printable QR codes for each object in ***Assets***.	✓	
Label templates Generate printable, customizable label templates for each object in ***Assets***.	✓	

Automation		
Workflow Transitions Automate ***Assets*** related tasks when a particular transition in a workflow is triggered.	✓ Uses ***Assets*** specific post-functions.	✓ Uses Jira Automations rather than post-functions. Similar functionality to Data Center.
Object Automations Create rules that automate simple tasks in ***Assets***. Rules are automatically triggered upon certain events.	✓ Uses ***Assets*** specific automations.	✓ Uses Jira Automations rather than ***Assets*** specific automations. Similar functionality to Data Center.
Extending *Assets* functionality		
Scripting Ability to extend automation actions with Groovy scripting	✓ Uses ***Assets*** specific automations.	Can use Jira Automations or ScriptRunner instead
REST API	✓	✓

03

Good Practices for IT Asset and Service Configuration Management

Why use ***Assets*** in Jira Service Management?

Digital transformation of businesses, a top tech initiative, changes how we look at management of IT assets–including information and lifecycle–from on-premises hardware and software to SaaS apps and services in the cloud. Technology management as a whole requires clear visibility into the entire IT landscape, and that all starts with offloading the baggage of historically troublesome terminology, like CMDB - a term that often conjures feelings of inaccuracy and untrustworthiness. As Gartner previously documented, only 25 percent of organizations derive value from their current CMDB investment. [3]

However, CMDBs can provide valuable insights and enable IT to make better decisions more quickly for service delivery. Recent Forrester research indicated that:

> "A CMDB is an integrated operational data store that contains key IT/digital assets, inventories, and their dependencies. It can play an essential role in enabling impact analysis and managing IT portfolios for risk, efficiency, and performance. In our survey, 67% of respondents said their organization has a CMDB; of those, 91% agreed that their CMDB is essential to their operations."

Forrester further hypothesizes that:

> "... organizations investing in this capability have a better understanding of their digital estate, leading to higher performance on multiple dimensions. Notably, high-performing organizations overcome the data quality and completeness concerns that have plagued CMDBs and led to their failure; respondents in these groups are also more likely to report that they have automated their CMDB data maintenance as much as possible." [4]

The key to successful technology management in the current era is aligning key objectives and shedding legacy perceptions of what CMDB means for IT inventory and assets. So how can an organization reassess its current and future landscape? By looking at what's needed – providing the right data to the right stakeholders at the right time.

Assets in Jira Service Management provides a modern-world database for asset and service configuration management that is extensible throughout the business environment.

3 Gartner, Inc. "Break the CMDB Failure Cycle With a Service Asset and Configuration Management Program." Published 5 May 2020.

4 Forrester Research, Inc. "The State of Service Management, 2022." Published 22 July 2022.

When saying “modern” world, it means a technology landscape that is agile, hybrid, and changing – based on emerging DevOps practices, provided by SaaS products, and hosted on a combination of mobile and on-premise devices and cloud platforms.

Assets was built with this complex, transforming landscape in mind and has a few advantages over other vendors' offerings:

Tool Structure - By standardizing on a single tool across both asset and service configuration management, users benefit from tighter process integration and gain richer information context through shared data elements. Customers can also realize cost savings through faster implementation and simpler maintenance of a single tool.

IT Asset Management CMDB Service Configuration Management

Flexibility - The open data structure provides customers with more control over their assets and CI types. Customers can track asset and CI data required to support key business processes which reduces implementation time and maintenance effort.

Accuracy - No-code/low-code automation keeps data updated and reduces the manual workflow significantly.

Centralization - A wide range of integrations so customers can use a single entry point to their data. The source copy of the data is stored elsewhere. (Note: Due to the number of integrations available, this currently applies more to Jira Service Management Data Center than Cloud.)

Reputation - With Jira Software’s reputation with software developers, we have an opportunity to improve the reputation of CMDBs from a “barrier” to an “enabler” within the DevOps world.

How do I get started with ***Assets*** in Jira Service Management?

Assets in Jira Service Management' tool structure allows teams to adopt an adaptable and scalable strategy for building their IT asset and service configuration management system.

Use the ITIL 4 Guiding Principles to observe and understand the services, practices, people, and tools that you already have.

Consider the following ITIL 4 principles:

- Start where you are
- Focus on value
- Progress iteratively with feedback
- Keep it simple and practical
- Optimize and automate
- Collaborate and promote visibility
- Think and work holistically

While no one likes “homework”, it is required for a successful asset and service configuration management implementation. Most customers conduct a series of workshops to outline the company’s key business initiatives and establish clear goals for their asset and service configuration implementation. The following sections include activities for successfully implementing ***Assets***.

Activity 1 - List objectives and measurable outcomes that support business goals and strategies

Start by identifying a team and constructing a playbook. The team should include members of various groups – development, IT operations, and business (legal, finance, etc.) – to ensure that the organization's goals are articulated and priorities are defined.

The team's playbook should outline the business value of IT asset and service configuration management using the following items:

- Stakeholders (Who)
- Scope of the work (What)
- Approach, constraints, assumptions (How)
- Expected business outcomes (Why)
- Success measurements (Evaluate)

For example:

Who

- Service desk
- DevOps
- SecOps
- Enterprise architecture

What

Implement a system to manage IT assets and service configurations

How

- Provides support for cloud computing and a cloud-driven demand model
- Provides improved data availability to IT Information Security
- Enhances change collision detection / proactive change management impact analysis
- Provides additional data for incident management, problem management and event management
- Facilitates improved collaboration and cooperation across the organization
- Provides data to support contracts with external service providers

Why

- Improve overall system availability
- Be better positioned to support audits / regulatory requirements
- Contribute to a cohesive strategy across IT organizations
- Provide a Single Source of Truth describing how the IT infrastructure supports the business
- Provide better insight to IT operational environments

Evaluated by

- Improved visibility of planned and unplanned changes as a percentage of total changes
- Increased successful change rate
- Decreased incident mean-time-to-repair
- Improved perception of IT as an enabler of the business

Activity 2 - Develop a top-down, lean approach to design your service model architecture

Here are some common questions a company might ask to give some inspiration of where to start and what info to include in the team's playbook. Which questions come up most for you? Which answers take the longest to find out? What answers cost your organization the most if you don't have them?

IT asset management	Service configuration management
• What type of IT devices are important for us to track and manage? • What do we need to understand about our IT devices to track them effectively? • What do we need to track when onboarding and offboarding employees and contractors? • What types of software licenses do we track (cloud vs physical)? Do we have a good understanding of the subscription license purchased versus assigned? • How can leadership and IT teams find the total assets deployed, who owns or is assigned the assets, where is the asset's associated purchase order and contract data, etc. to make key business decisions? • What types of information do we need to track to support financial audit requirements?	• What are the top services that are important to our business? • Who manages these services? • What percentage of services are deployed to the cloud (AWS, Azure, Google etc.)? • Do we have a good understanding of the service taxonomy (the supporting service applications/ infrastructure and their relationships)? • What types of information do we need to track to support compliance requirements?

If you can't easily answer these questions, then you likely have blindspots in your asset and service configuration oversight. If you have unused licenses, maybe you're paying too much for your software agreement. If you're not sure which operating systems you have running, how do you make sure every device is updated if there's a security patch required?

It is recommended to keep your approach simple and practical by focusing on your most critical services and systems. These are easy to spot – if there is even a hint of a disruption, your service desk is flooded with calls and senior management starts sending nervous messages to their staff. List out central problems and questions for these capabilities in the team's playbook.

This information will determine what data you need to answer your questions, help solve your problems, and are the foundation for your initial use cases to be included in the team's playbook. Identify your critical service and the supporting infrastructure; specifically, the applications and the related servers. You can then prioritize developing these use cases based on the value the functionality will bring to the business and ease of implementation. You'll get a quicker and more visual realization of the available benefits, and the rest of the business will quickly notice the positives such as quicker incident resolution, less failed changes, and less downtime.

Activity 3 - Identify data, workflow, and roles that support key business processes

After your initial use cases are defined, the next action is to consider what data asset and service configuration management must provide to other processes – incident, change, request fulfillment, etc.

For example, what data is needed to repair an employee's laptop?

- Asset tag
- Model
- PO
- Assigned user
- Location

This data will be the attributes for your assets/CIs. Include only the attributes you need to support the data needs of the use cases and determine the source for the information. Some customers may have configuration data available through current discovery tools and other asset data tracked in various spreadsheets or databases across separate organizations. Use your existing data as the baseline for ***Assets*** implementation and document which attributes require manual entry and which ones can be updated automatically via discovery.

Also, outline the workflows and roles between processes so that everyone knows how they should be working with each other and who owns the data completeness and accuracy.

For example, incidents can be created only for servers that are connected to the company network and operational. The data center team is responsible

for receiving the server and updating asset data; the server support team installs the server and a discovery tool detects the new server and sets the configuration status to operational.

Activity 4 - Outline your asset/configuration dataset

In most cases, companies typically start with tracking infrastructure assets they need to support incident and change management, as well as, service requests. These items typically represent IT components such as:

- Servers and virtual machines
- Applications
- Laptops/desktops
- Printers
- Network equipment
- Storage arrays
- Databases
- Security appliances
- Microservices

Decide which asset types and attributes you need to support your use cases. Again, it is recommended that organizations start simple and make incremental improvements as they gain experience.

PRO TIP

- Start by populating ***Assets*** with a solid inventory of assets and CIs focused on specific use cases. If you find yourself populating with items that do not tie back to your goal or use case, you are off track.
- Assets/CIs should have unique identifiers that do not change. The identifier needs to be unique so it can be differentiated from other assets/CIs, and it mustn't change so the asset/CI can be tracked over time. Establish a consistent naming convention to improve usability of asset data. Serial number, asset tag, asset name or external system identifier can be used depending on the asset type.
- Assets/CIs should have relationships. An infrastructure asset/CI represents a component that needs to be managed to deliver or support a service. In other words, each infrastructure asset/CI has a direct or indirect relationship with one or more service CIs.

Activity 5 - Develop metrics that demonstrate improvement in key business outcomes

You can show value to the organization by tying back to the goals and objectives the team set earlier. The value to the company is that you can provide a link between strategic business drivers to the services offered to your customers to the operational infrastructure used to deliver your services and the associated total cost of your services.

For example, ***Assets*** can help your IT organization track:

- Improved mean time to identification (MTTI) and mean time to resolution (MTTR) for incidents by using CI dependency information
- Reduced device misconfigurations which can contribute to system downtime and cyberattack vulnerability
- Increased service availability and change deployment success through more visibility of system relationships and improved risk assessments

- Improved asset utilization and budgeting accuracy with a single source of asset lifecycle data and the associated costs
- Faster system isolation and remediation after security incident with accurate infrastructure dependency data
- Increase regulatory compliance through streamlined IT asset and CI tracking procedures and improved data quality and reporting

These metrics provide a more data-driven approach to new software and hardware investment for the organization.

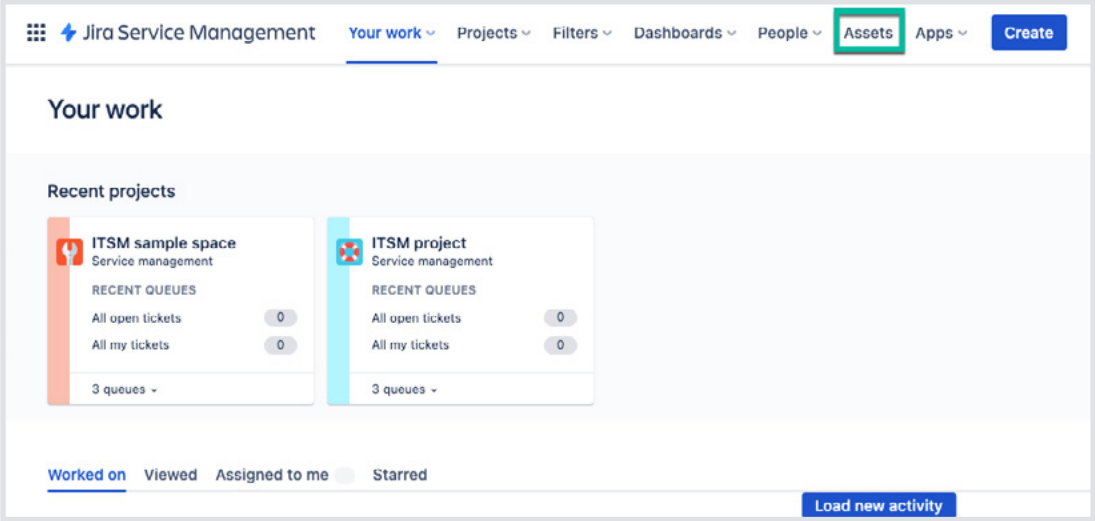
How do I build assets using *Assets* in Jira Service Management?

Assets is included in Jira Service Management Premium, Enterprise, and Data Center plans, allowing teams to track their assets, CIs, and resources to gain visibility into critical relationships between services, infrastructure, and other key assets. ***Assets*** is built on Jira Service Management, giving teams a simple and quick way to tie assets and CIs to service requests, incidents, problems, changes, and other issues to gain valuable context and the ability to automate workflow to boost operational efficiency.

To start your free trial of Jira Service Management Cloud Premium, get in touch with your local Atlassian Solution Partner.

Access *Assets* in Jira Service Management

Whether you are on a licensed or trial version of Jira Service Management Premium or Enterprise, you can access ***Assets*** in Jira Service Management by clicking on the ***Assets*** option in the Jira Service Management main navigation bar

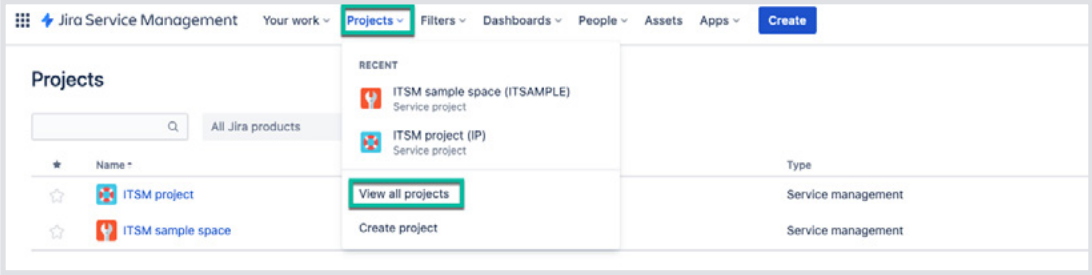


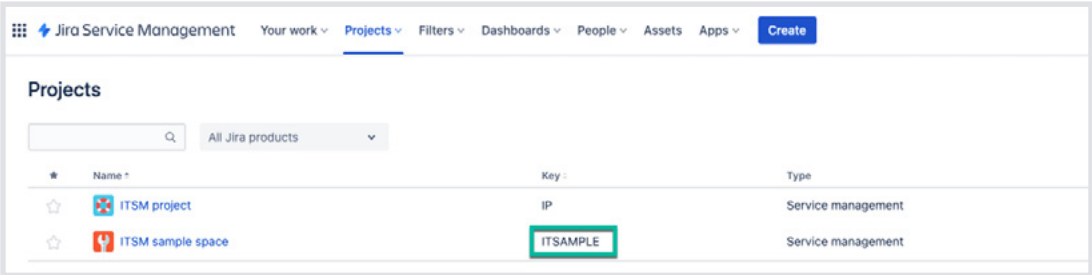
PRO TIP

If you are new to Jira Service Management, it is recommended you create a project using the IT service management template. In a new instance, this template creates 2 projects:

- ITSM sample space to test, explore, and learn how ITSM projects work by creating new requests, adding custom fields, and assigning them to people – or play around with the sample requests already created for you.
- ITSM project to handle service requests, resolve incidents, approve changes and fix problems.

Whether you use a default project or create a new project, note the project's Key data because we will use the information later when creating a custom field.





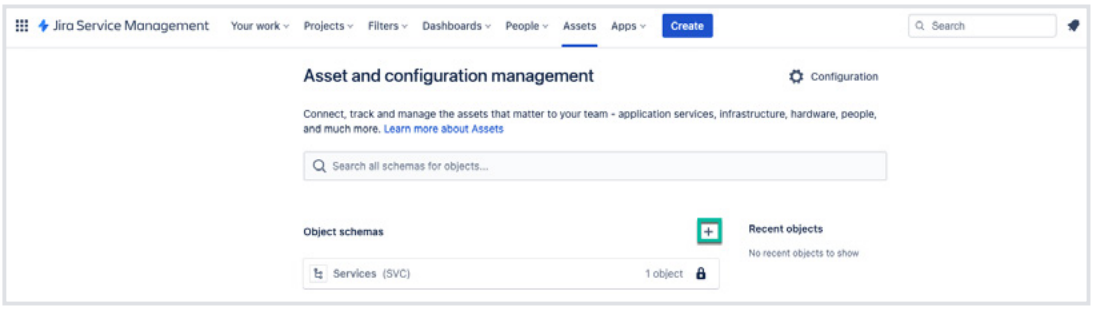
Use Case - Tracking a Jira request for an employee laptop

Managing laptops is a common use case for every organization.This will provide a step-by-step guide for tracking laptop requests in Jira Service Management for explaining the features and capabilities of***Assets*** better.

Step 1 - Create an object schema for Facilities

Facility data will be included in employee records, so a Facilities object schema must be created first, so information can be referenced by the Employees object schema created next. Floors and rooms will be tracked as nested object types with the information displayed in a flexible tree hierarchy.

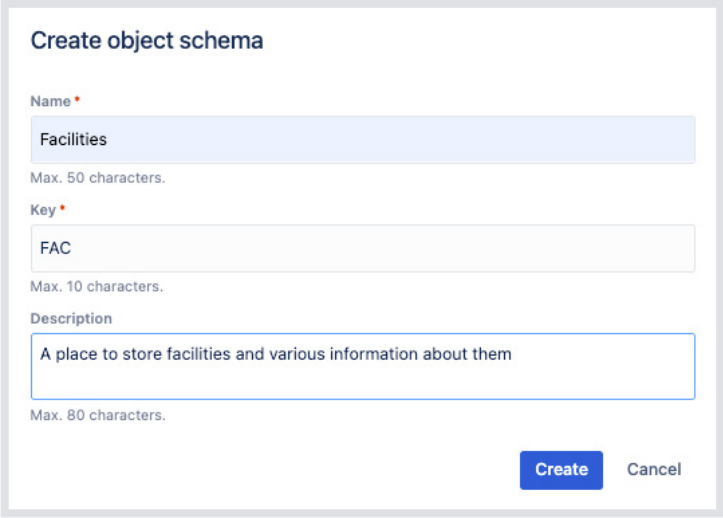
If your organization does not need to track multiple buildings or facility data, skip to Step 5.

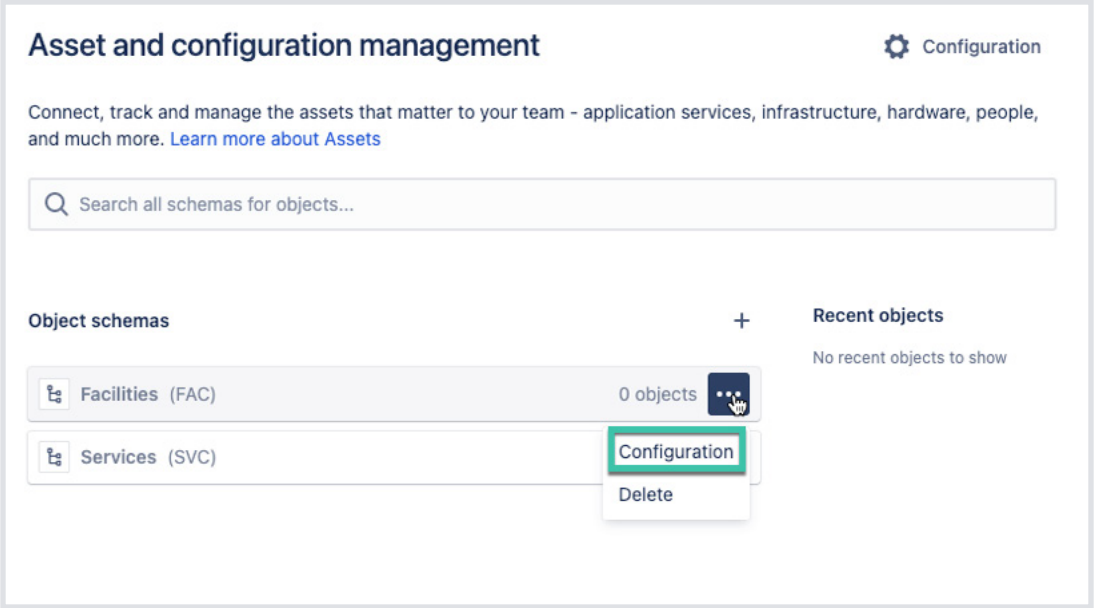


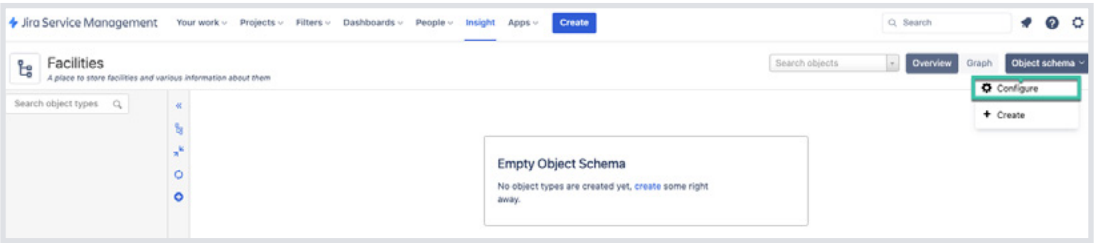
Note: There is a default, read-only object schema named 'Services'; this object schema is system generated and tracks Services created in Jira Service Management. The Services object schema will be covered further in a later section.

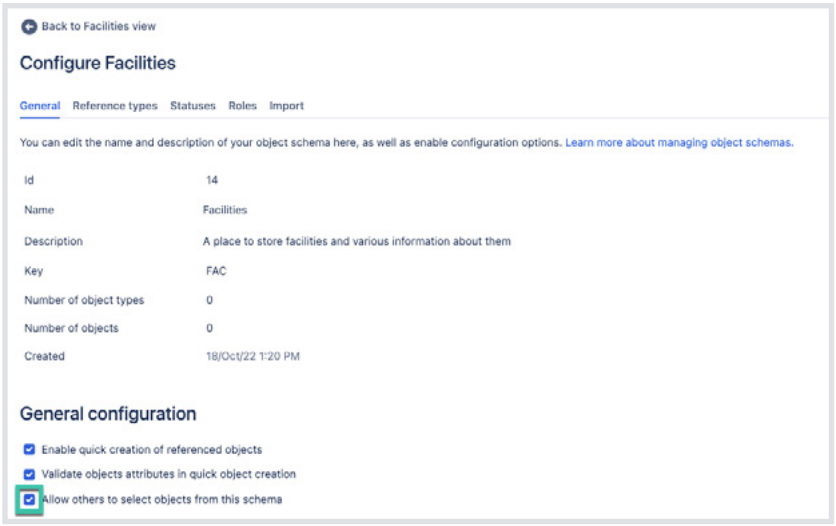
Click the "**+**" to display the **Create object schema** window, enter object schema information, and create the record

Configure the new object schema and select *Allow others to select objects from this schema* because the facility data should be reference-able by other objects.




Asset and configuration management
Configuration
Connect, track and manage the assets that matter to your team - application services, infrastructure, hardware, people, and much more. Learn more about Assets
Search all schemas for objects...
Object schemas
Recent objects
No recent objects to show
Facilities (FAC)
0 objects
Services (SVC)
Configuration
Delete


Jira Service Management
Your work
Projects
Filters
Dashboards
People
Insight
Apps
Create
Search
Facilities
A place to store facilities and various information about them
Search object types
Search objects
Overview
Graph
Object schema
Configure
Create
Empty Object Schema
No object types are created yet, create some right away.


Back to Facilities view
Configure Facilities
General
Reference types
Statuses
Roles
Import
You can edit the name and description of your object schema here, as well as enable configuration options. Learn more about managing object schemas.
Id 14
Name Facilities
Description A place to store facilities and various information about them
Key FAC
Number of object types 0
Number of objects 0
Created 18/Oct/22 1:20 PM
General configuration
Enable quick creation of referenced objects
Validate objects attributes in quick object creation
Allow others to select objects from this schema

PRO TIP

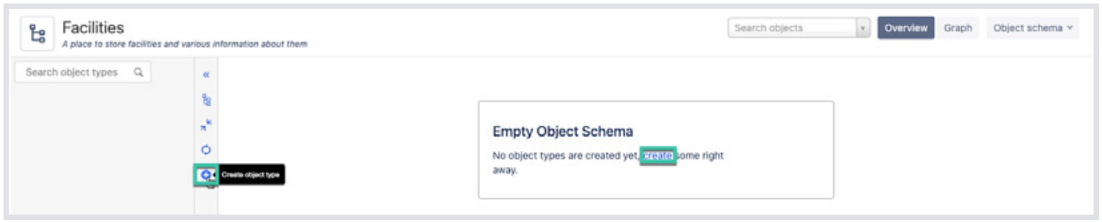
Enable quick creation of referenced objects and *Validate objects attributes in quick object creation* are set by default.

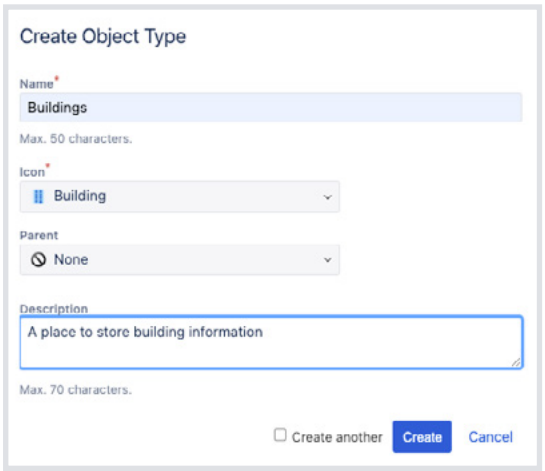
When you are creating a new object, you can enter details about that object in the 'Create Object' dialog box. These details could include text or numerical information, or even references to other objects. If you Enable quick creation of referenced objects you can both create and reference an object in one action, simply by entering a new label into any object reference field on the 'Create Object' dialog box.

Because these newly created objects may have mandatory fields or validations in place, you can select Validate object attributes in quick object creation to enforce any requirements, and block the creation of new objects.

Step 2 - Create an object type and objects for Buildings

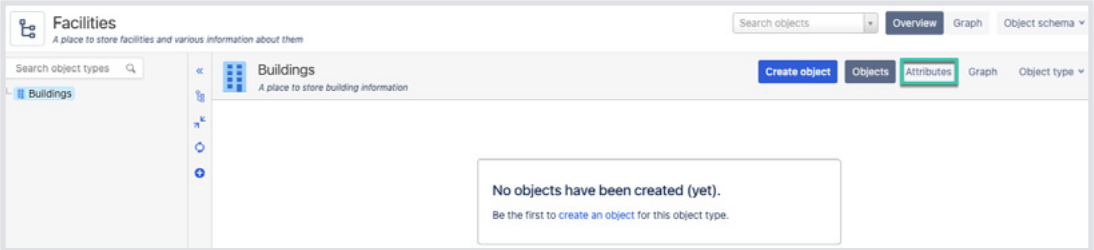
Click the **+** icon or the **Create** link to display the **Create Object Type** window, enter the object type information, and create the record





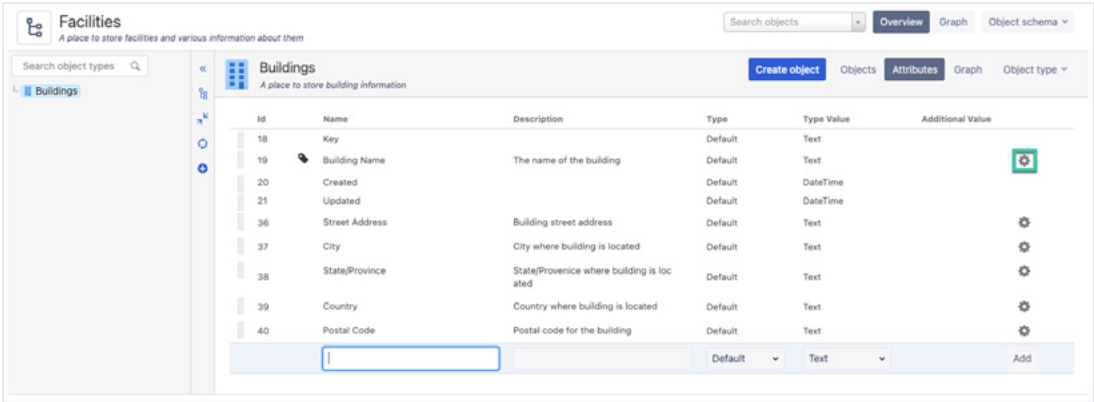
PRO TIP

Add new icons as you wish and use for object types. This way you can have your own organization's colors/theme and improve readability of asset data.

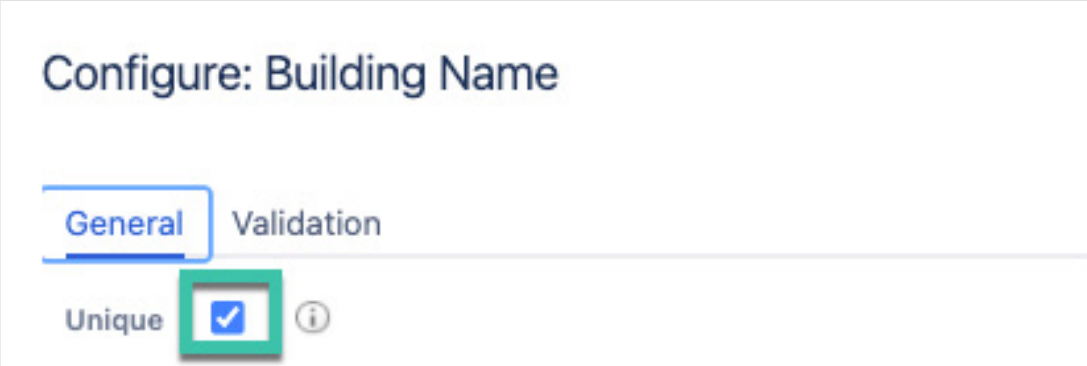


Select the **Attributes** option for **Buildings** object type and add options important to your organization:

Name	Type	Value	Additional Value
Street Address	Default	Text	
City	Default	Text	
State/Province	Default	Text	
Country	Default	Text	
Postal Code	Default	Text	

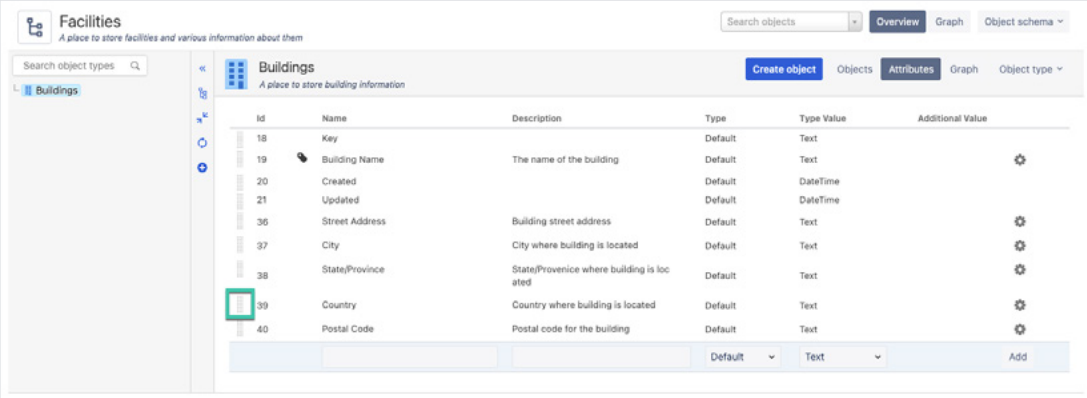


For the **Name** attribute, update the name data to “Building Name”, click the cog icon and configure the **Name** attribute to be unique, so you avoid duplicate building names

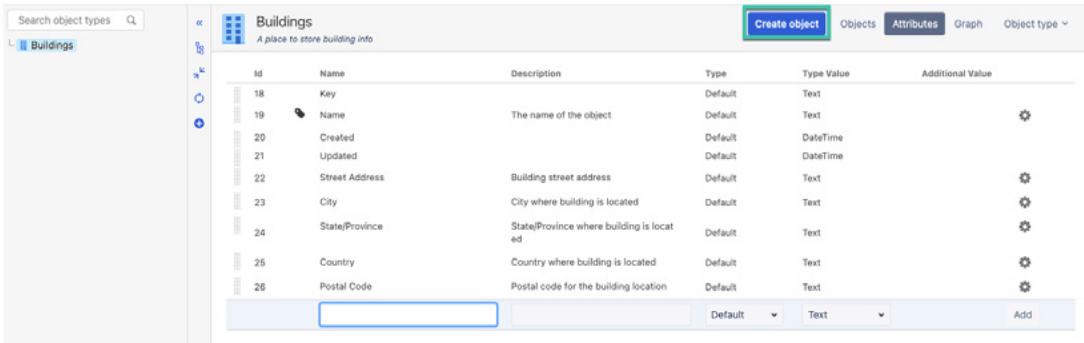


PRO TIP

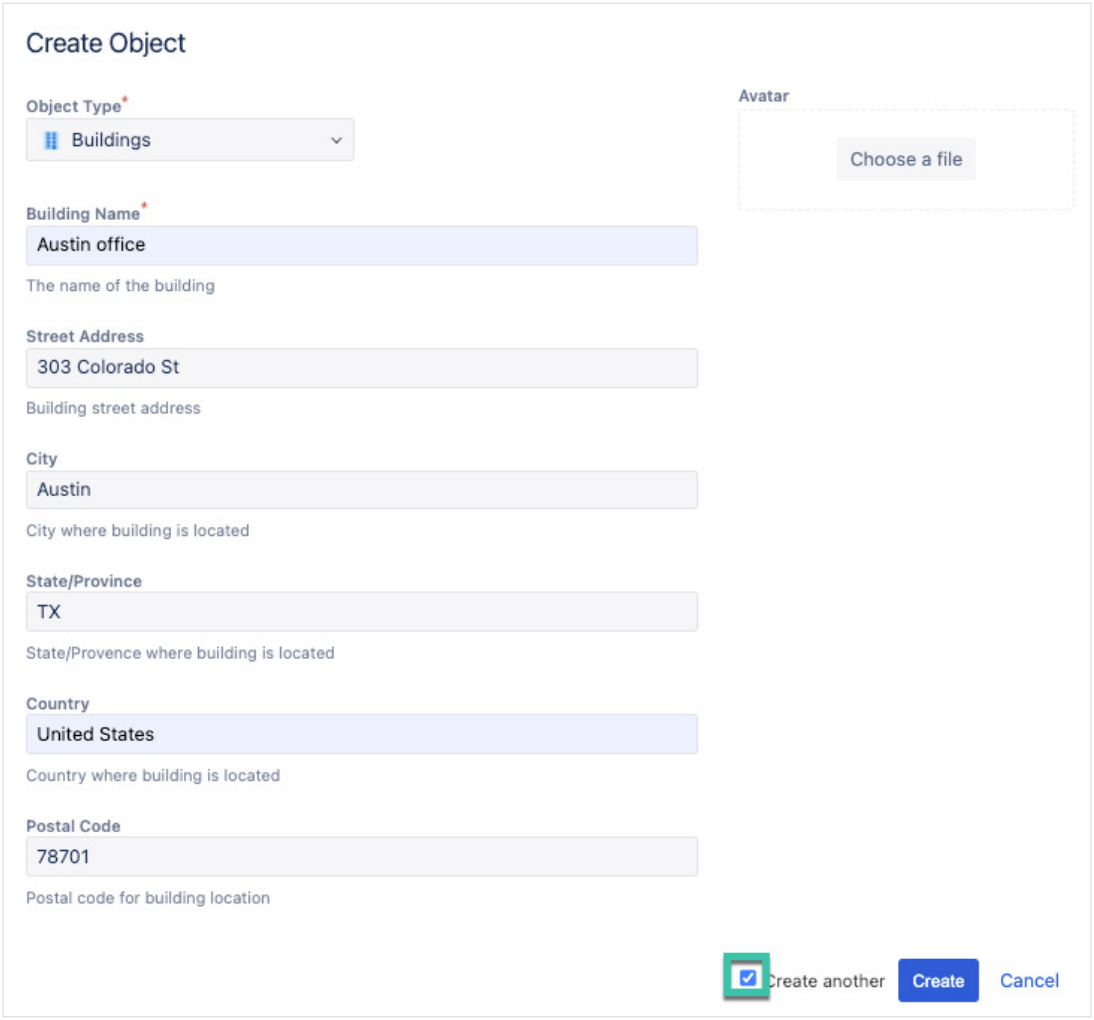
You can easily reorder attributes by selecting the grid icon and dragging the item to a different location.



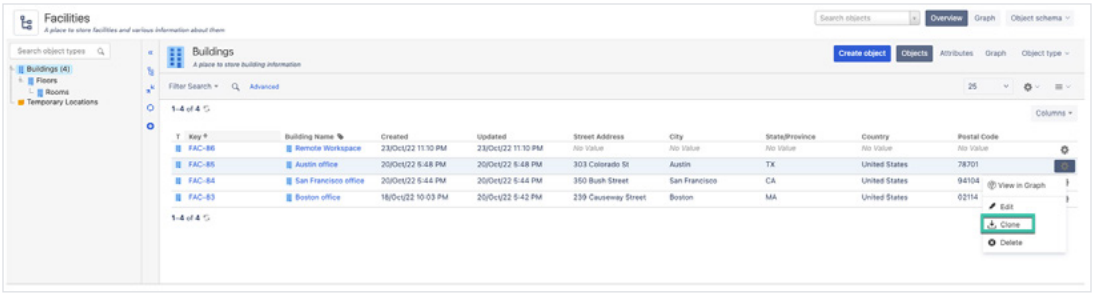
Click the **Create object** button to display the **Create Object** window



Add building data, select **Create another** to create multiple records using the same window, and create the record



You can also use the **Clone** feature to create records

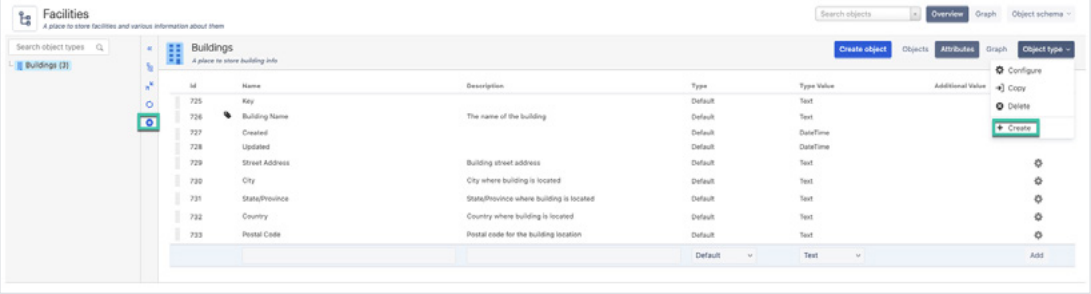


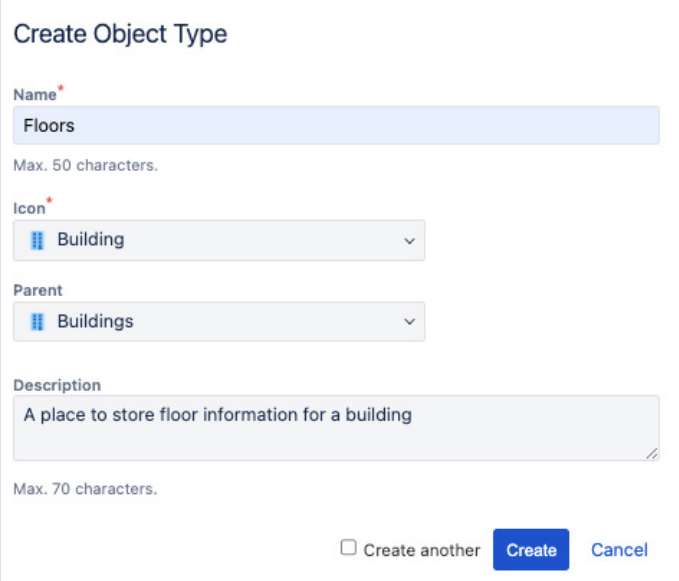
Add your primary building data; additional building records can be created when needed.

Step 3 - Create Floor and Room object types

Some companies may need to track facility data at a specific level (e.g., data center stockroom, laboratory room, etc.), so create a hierarchical data structure to support this requirement. To create a hierarchical data structure, create object types with parent object types and include the parent as an attribute.

Create an object type for **Floors** and select **Buildings** as the **Parent**.

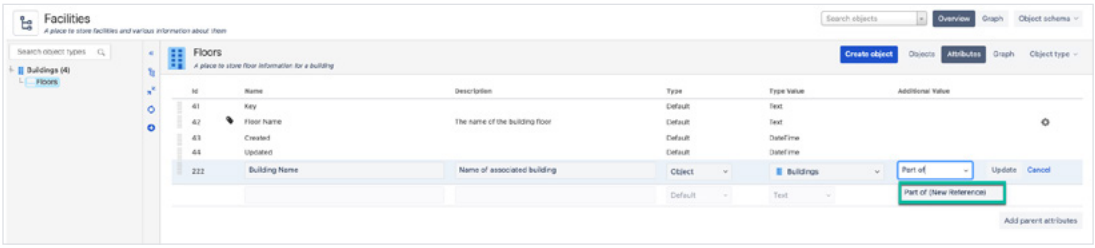




Select the **Attributes** option for **Floors** object type and add the following item:

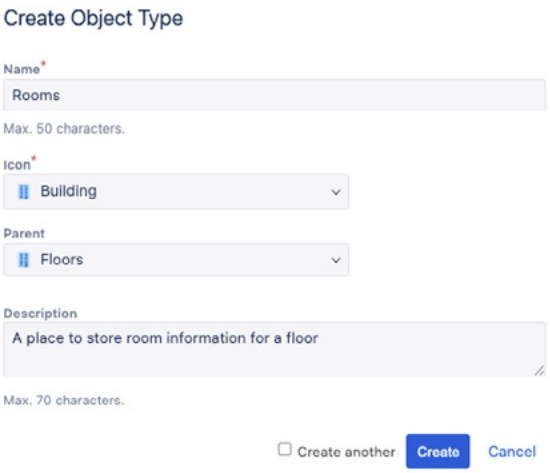
Name	Type	Value	Additional Value
Building Name	Object	Buildings	Part of

For the **Name** attribute, update the name data to “Floor Name”, click the **cog** icon and configure the **Name** attribute to be unique, so you avoid duplicate floor names



For the **Building Name** attribute, you use an existing value or enter a new reference value; simply click on the item to enter the value. If you enter a new reference value, the data is automatically added to the **Facilities** object schema **Reference types** where you can add more information and update the color.

Create an object type for **Rooms** and select **Floors** as the **Parent**.

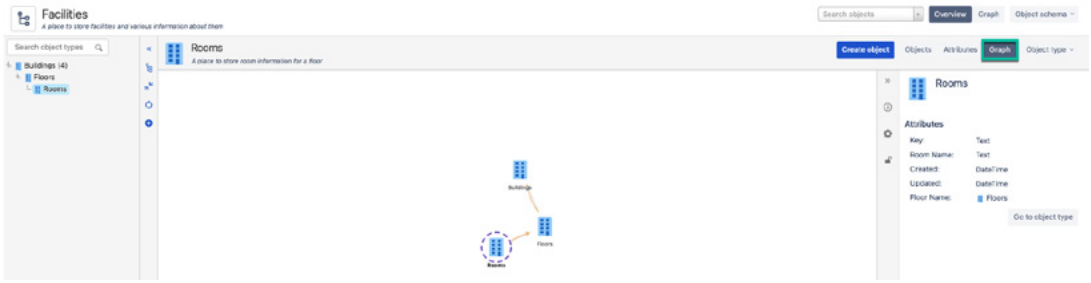


Select the **Attributes** option for **Rooms** object type and add the following item:

Name	Type	Value	Additional Value
Floor Name	Object	Floors	Part of

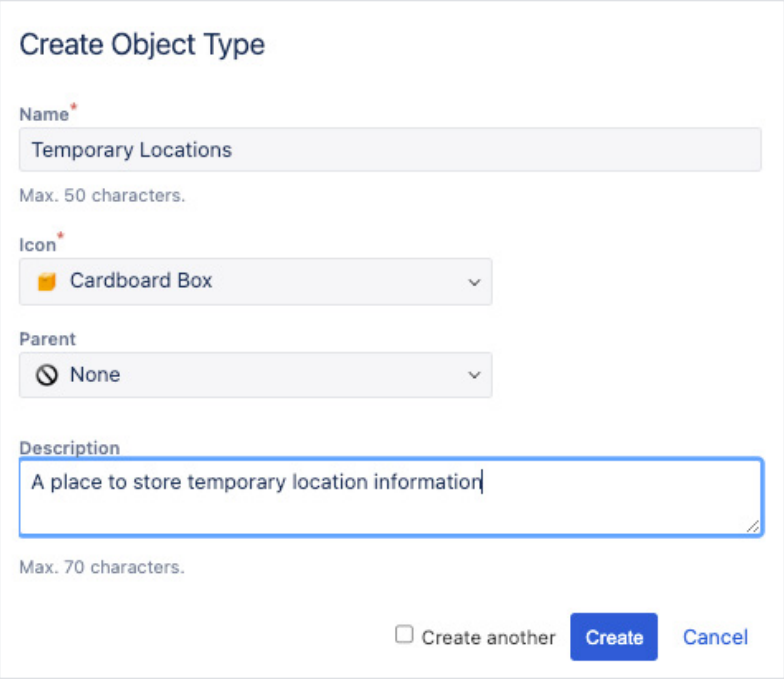
For the **Name** attribute, update the name data to “Room Name”, click the **cog** icon and configure the Name attribute to be unique, so you avoid duplicate room names

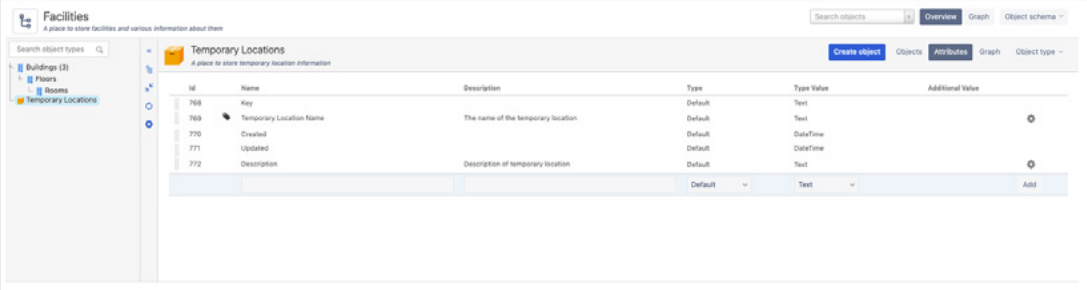
Click the **Graph** option to display the object types and their relationships. As you can see in the relationship arrows, **Rooms** are part of **Floors**, and **Floors** are part of **Buildings**.



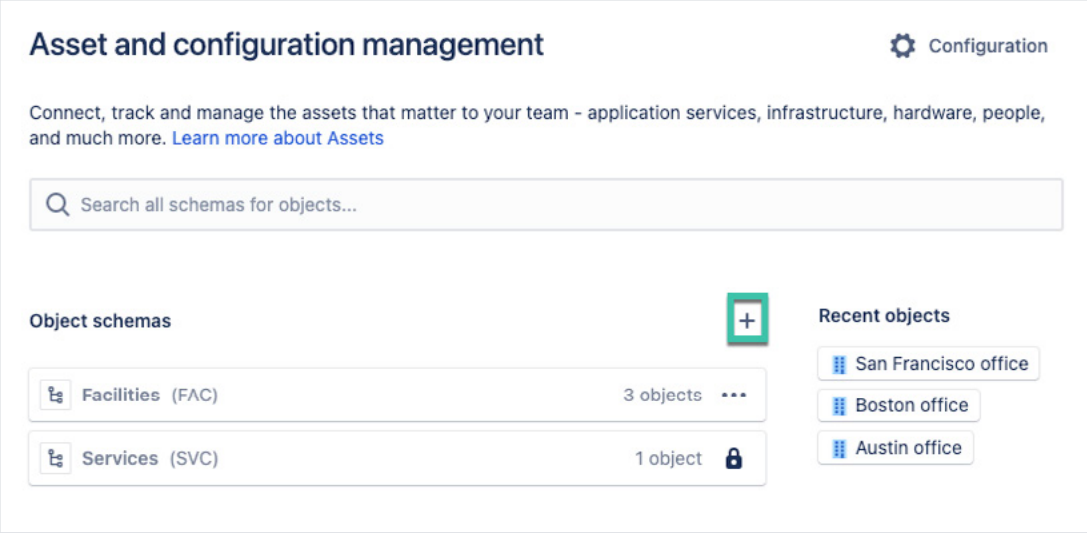
Step 4 - Create Temporary Locations object type

Some companies set up temporary operations in parking lots or partner facilities for special events. This data can be simply added as an object type along with any needed information.

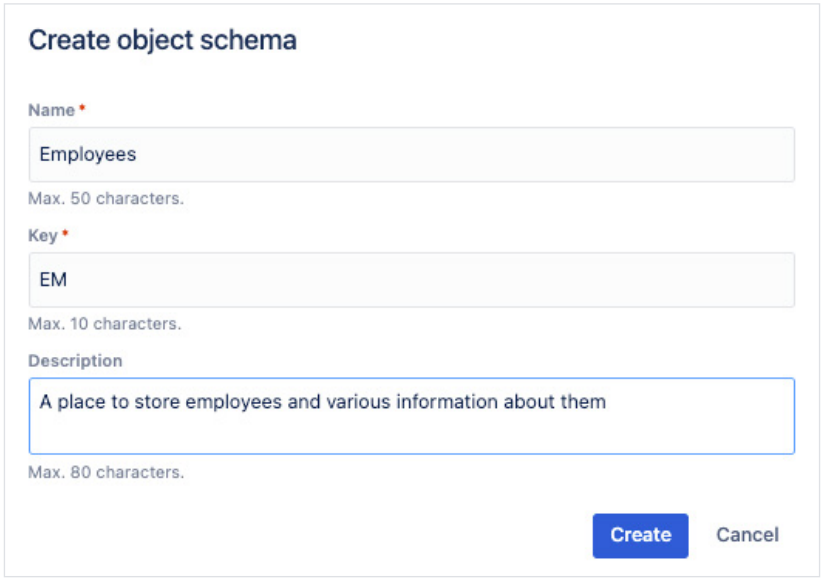


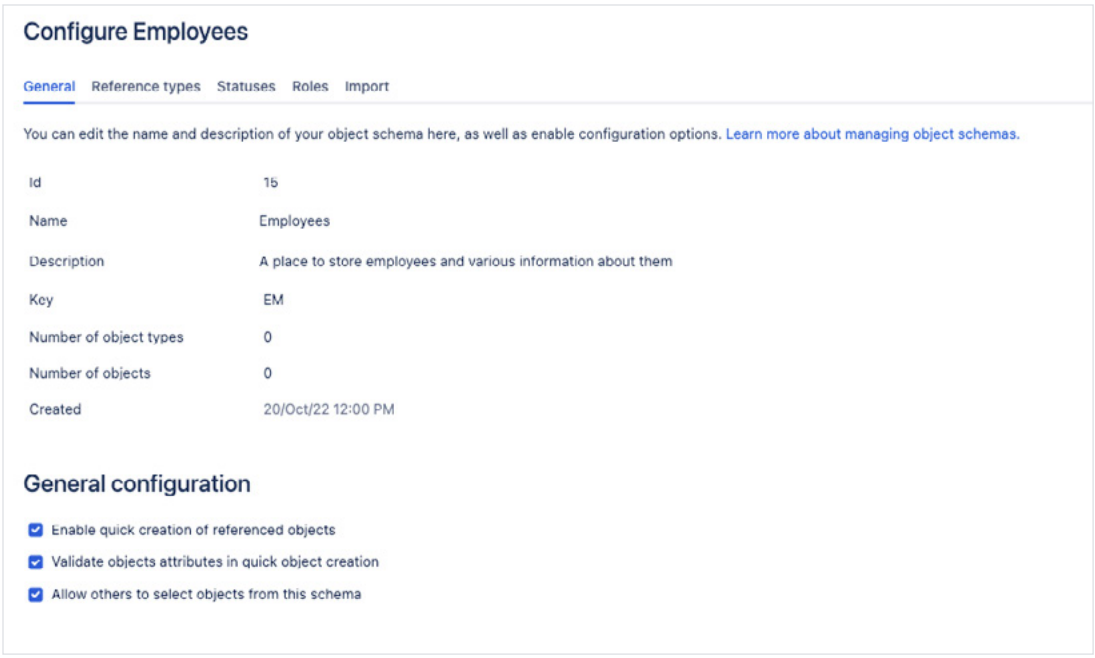


Step 5 - Create an object schema, object type, and objects for Employees

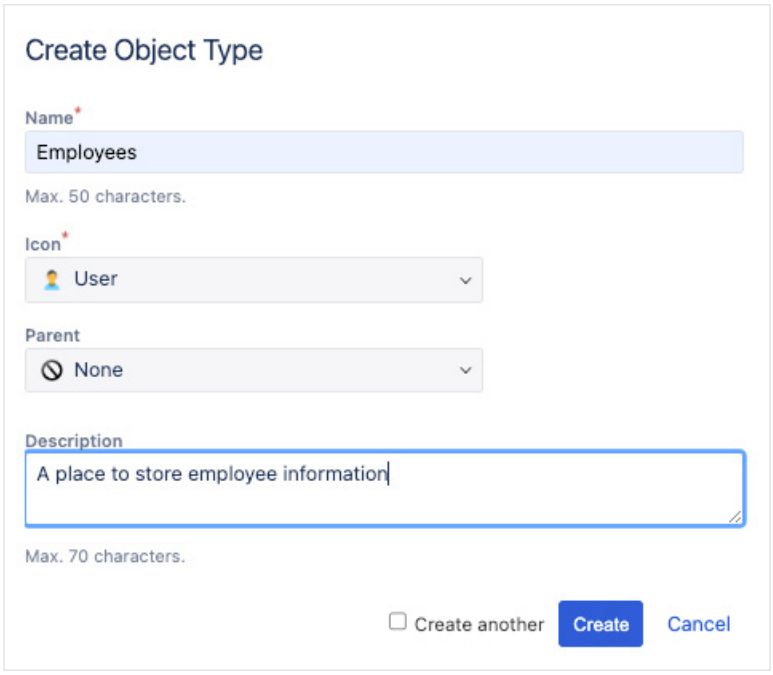


Display the **Create Object Schema** window, enter object schema information, then configure the object schema to select *Allow others to select objects from this schema*.





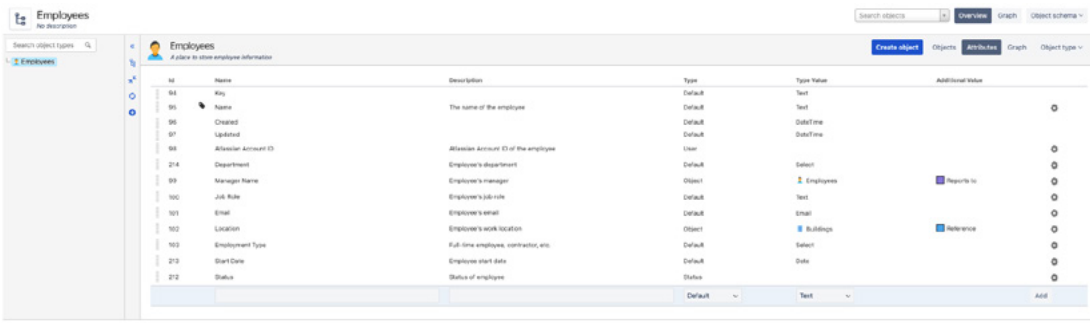
Create an object type for Employees.



Select the **Attributes** option for **Employees** object type and add options important to your organization:

Name	Type	Value	Additional Value
Atlassian Account ID	User		
Department	Default	Select	HR IT Finance Marketing Operations R&D
Manager Name	Object	Employees	Reports to
Job Role	Default	Text	
Email	Default	Email	
Location	Object	Buildings	Reference Skip this attribute if your organization does not need to track multiple buildings
Employment Type	Default	Select	Full-time Employee Contractor
Start Date	Employee start date	Date	
Status	Status		

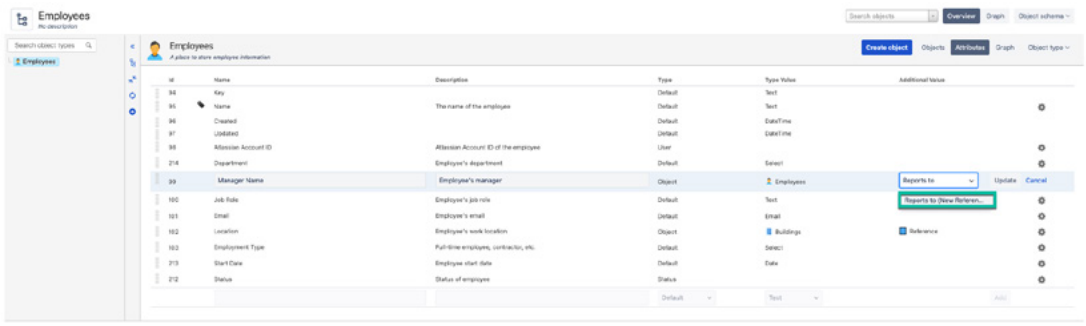
For the **Name** attribute, update the name data to “Employee Name,” click the cog icon, and configure the Name attribute to be unique, so you avoid duplicate employee names.



Configure the **Department** attribute and add options important to your organization:

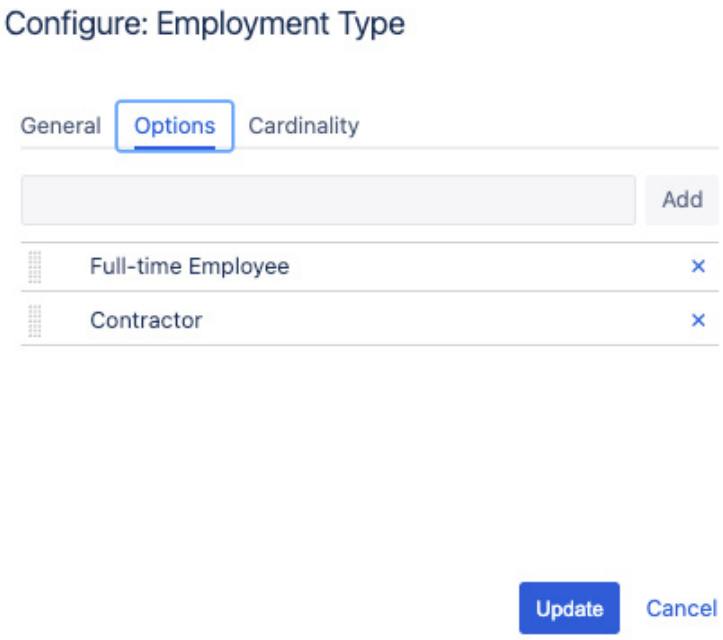
- Finance
- HR
- IT
- Marketing
- Operations
- R&D

For the **Manager Name** attribute, use an existing reference value or enter a new reference value by clicking on the item and entering the new value.



Configure the Employment Type attribute and add relevant options:

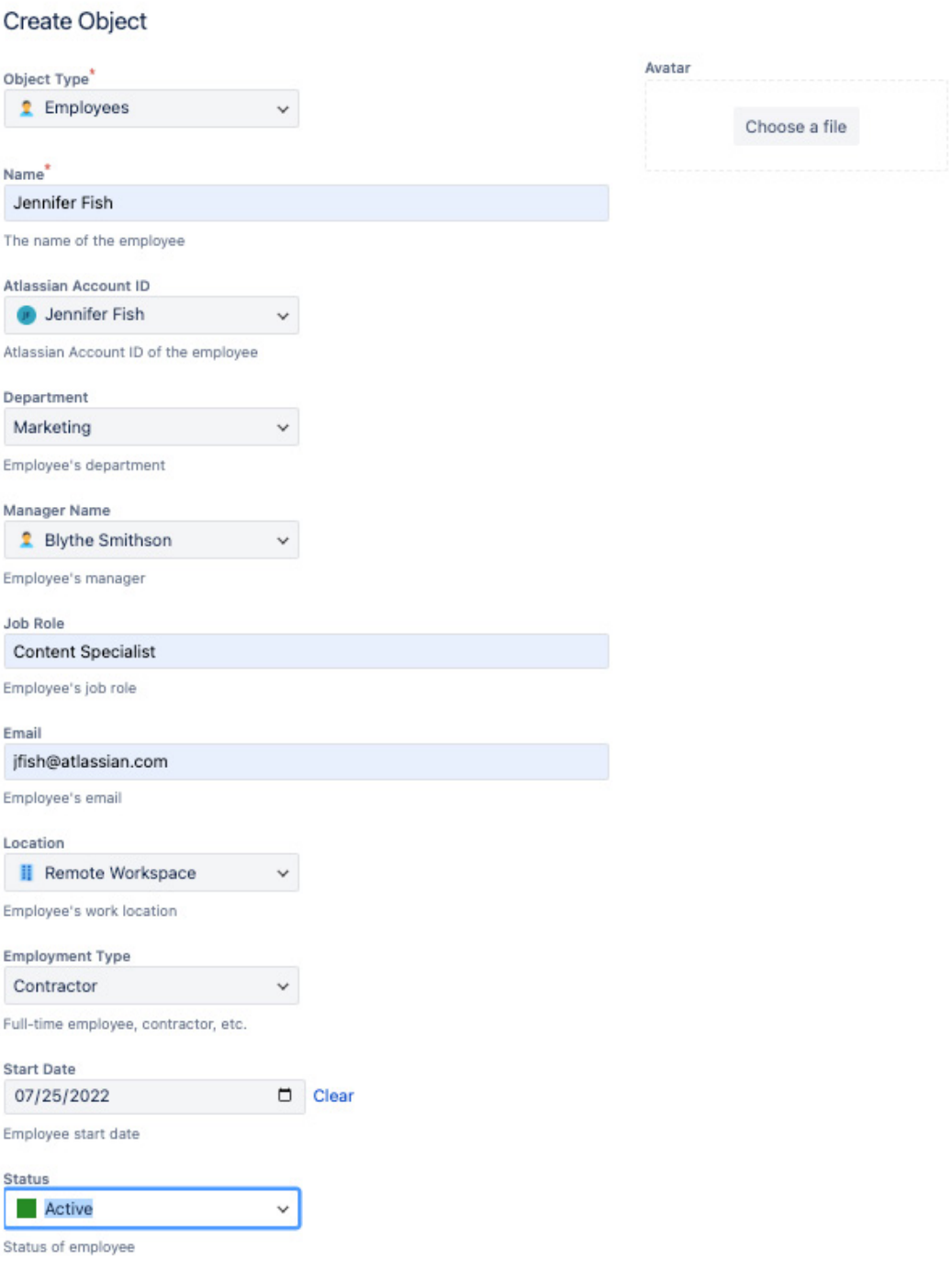
- Full-time Employee
- Contractor



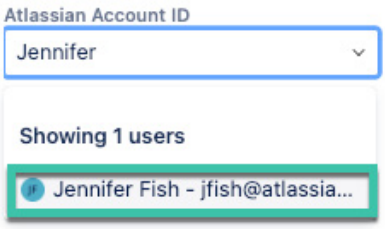
PRO TIP

Do not enter a **Type Value** for the **Status** field so that all values are allowed. Additional functionality will be developed for the **Employees** object type and you will want the flexibility to add more status values.

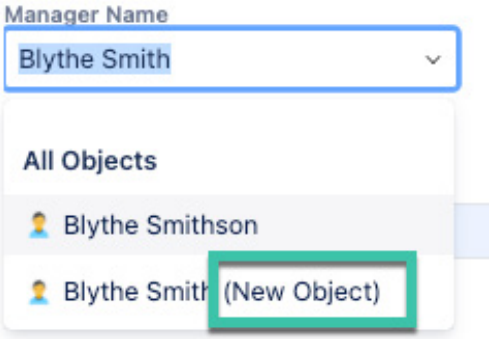
Click the **Create Object** button to display the **Create Object** window.



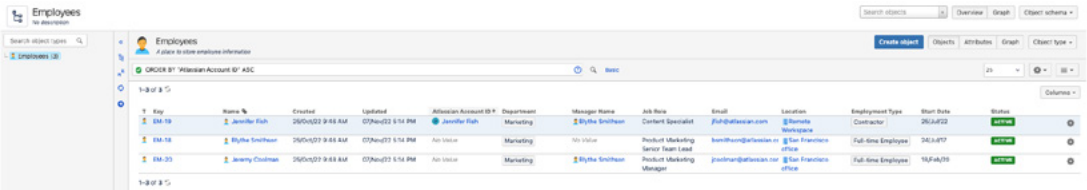
In the **Atlassian Account ID** attribute, type a few characters of the employee's ID, and a list of Jira accounts is provided; click on the appropriate item to enter the value.



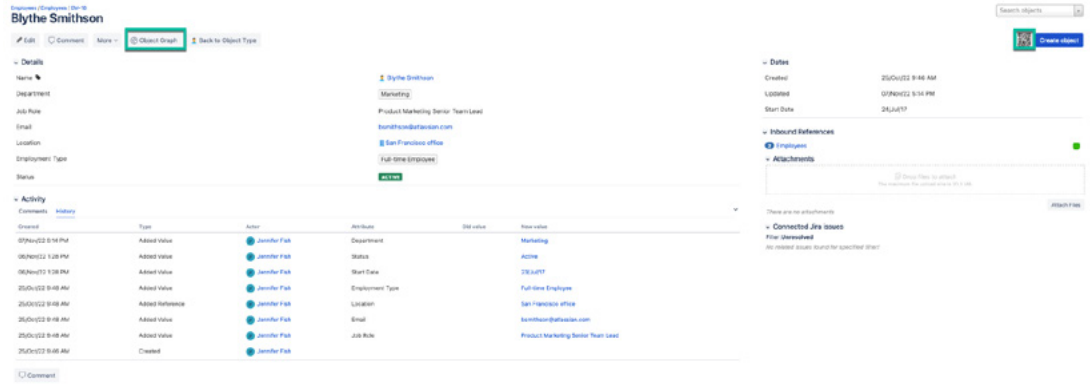
In the **Manager** attribute, type in the manager's name and a list of matching objects with matching names is provided. An existing reference object can be selected or a new reference object can be created by clicking on the item and entering the value. Be sure to update the new object with additional data, if needed.



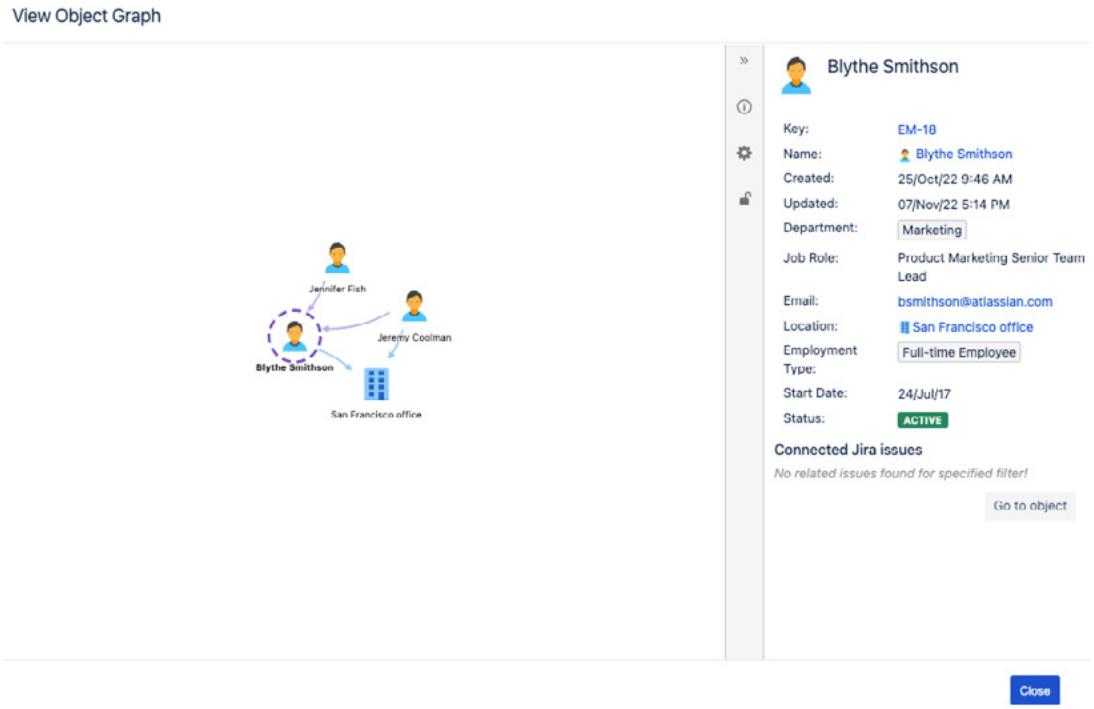
Add your primary team member data; additional employee objects can be created when needed.



Select an **Employee** object and display the object record; the display contains the defined attributes, references, update history, comments, and printable QR code.

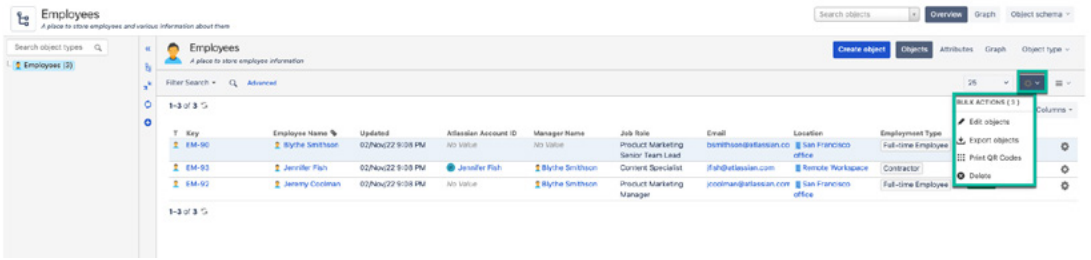


Click the **Object Graph** option to display the object and relationships.



PRO TIP

You can quickly update, export, delete, and print QR codes for assets through the bulk actions function. Query for the objects you want to modify, then click on the cog icon to display the bulk action options.



Bulk edit (3)

Here you can modify multiple objects at once. All objects currently filtered will be changed.

Keep — Name
The name of the employee

Keep — Atlassian Account ID
Atlassian Account ID of the employee

Keep — Manager Name
Employee's manager

Keep — Job Role
Employee's job role

Keep — Email
Employee's email

Keep — Location
Employee's work location

Keep — Employment Type
Full-time employee, contractor, etc.

Keep — Start Date — mm/dd/yyyy — Clear
Employee start date

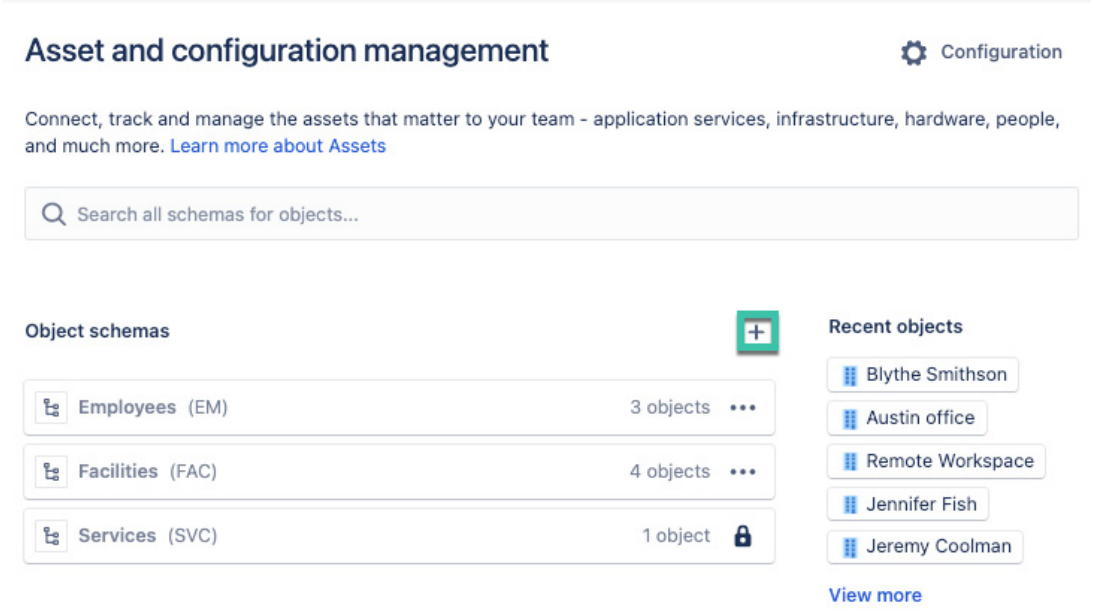
Keep
✓ Change
Add
Remove
Status — Active
Status of employee

Confirm Cancel

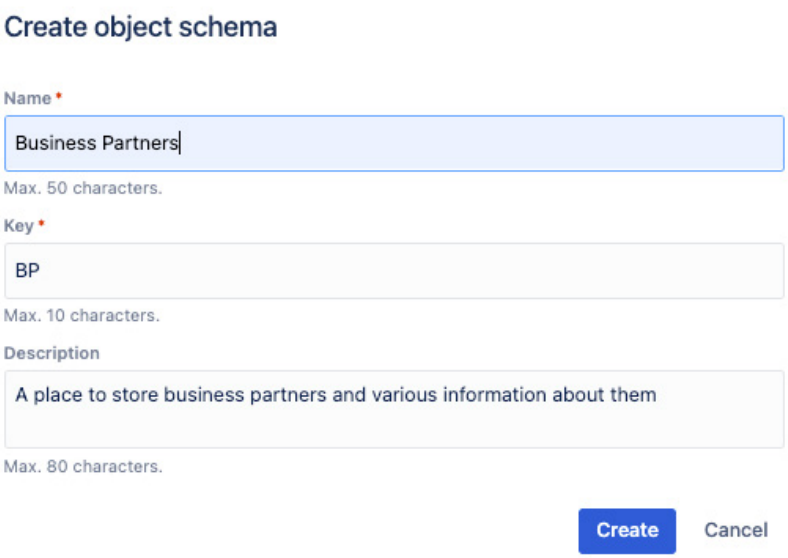
Step 6 - Create an object schema and object type for Business Partners

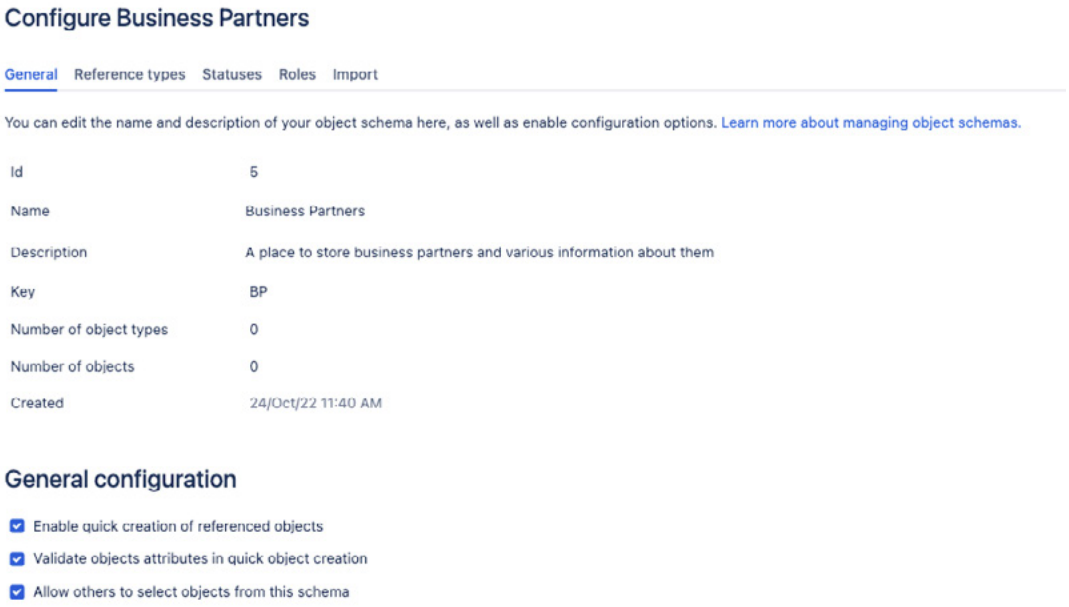
Manufacturer data will be included in IT asset records, so a **Business Partners** object schema must first be created, so information can be referenced by the **IT Assets** object schema created next. Object types will also be created for vendors and suppliers which can be used in future cases for tracking vendor and supplier information.

If your organization does not need to manage business partners at this level, skip to Step 9; the functionality can be added and data updated when needed.

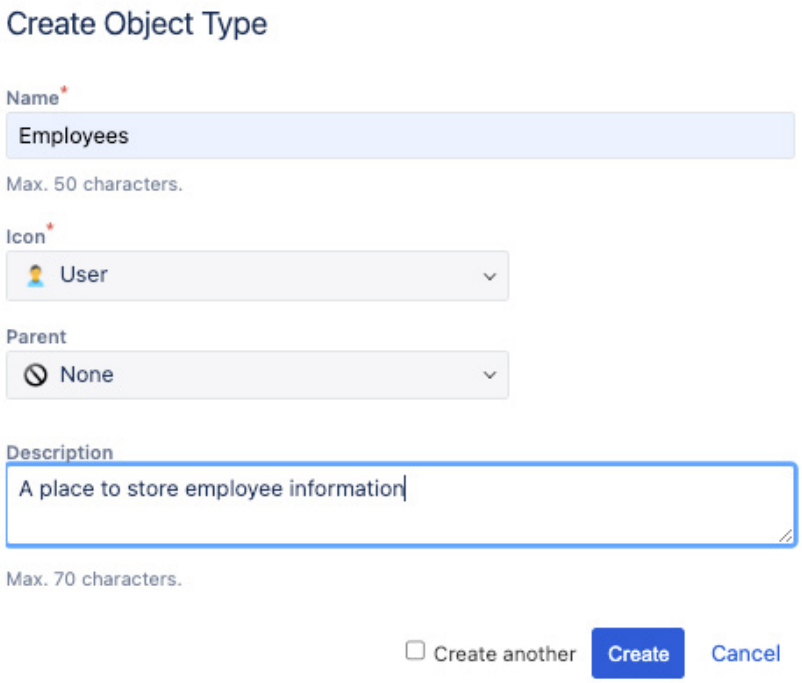


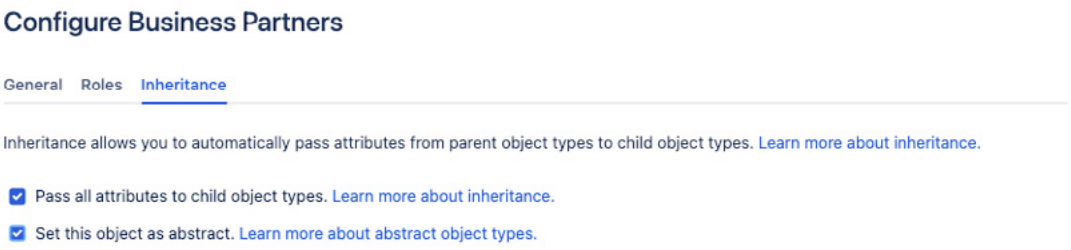
Display the **Create Object Schema** window, enter object schema information, then configure the object schema to select *Allow others to select objects from this schema*.





Create an object type for **Business Partners** and configure the object type with *Pass all attributes to child object types* and *Set this object as abstract*





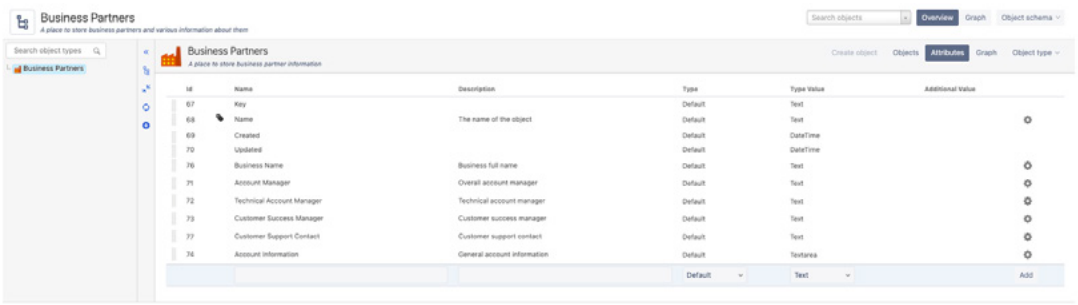
PRO TIP

Create a parent object type when you have similar data sets with different sources and/or uses. Using a parent object type, you can create multiple child object types and maintain the same structure. As you add or update attributes in the parent object type, the attributes will also be added / updated to the child object types.

Note that all attributes will be copied from the parent object type and cannot be modified.

Select the **Attributes** option for **Business Partners** object type and add relevant options:

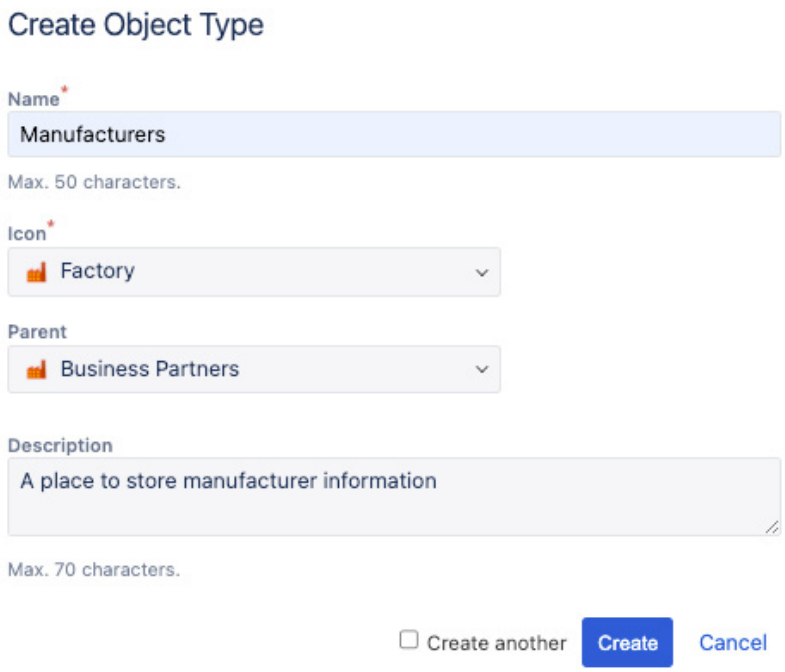
Name	Type	Value	Additional Value
Business Name	Default	Text	
Account Manager	Default	Text	
Technical Account Manager	Default	Text	
Customer Success Manager	Default	Text	
Customer Support Contact	Default	Text	
Account Information	Default	Textarea	

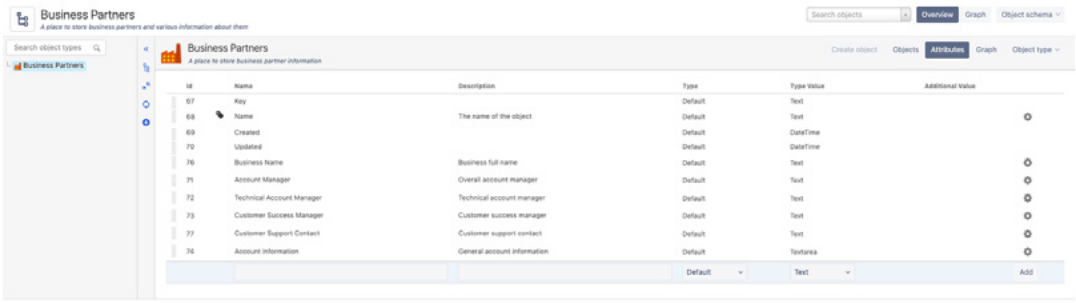


For the **Name** attribute, click the **cog** icon and configure the **Name** attribute to be unique, so you avoid duplicate object names.

Step 7 - Create an object type and objects for Manufacturers

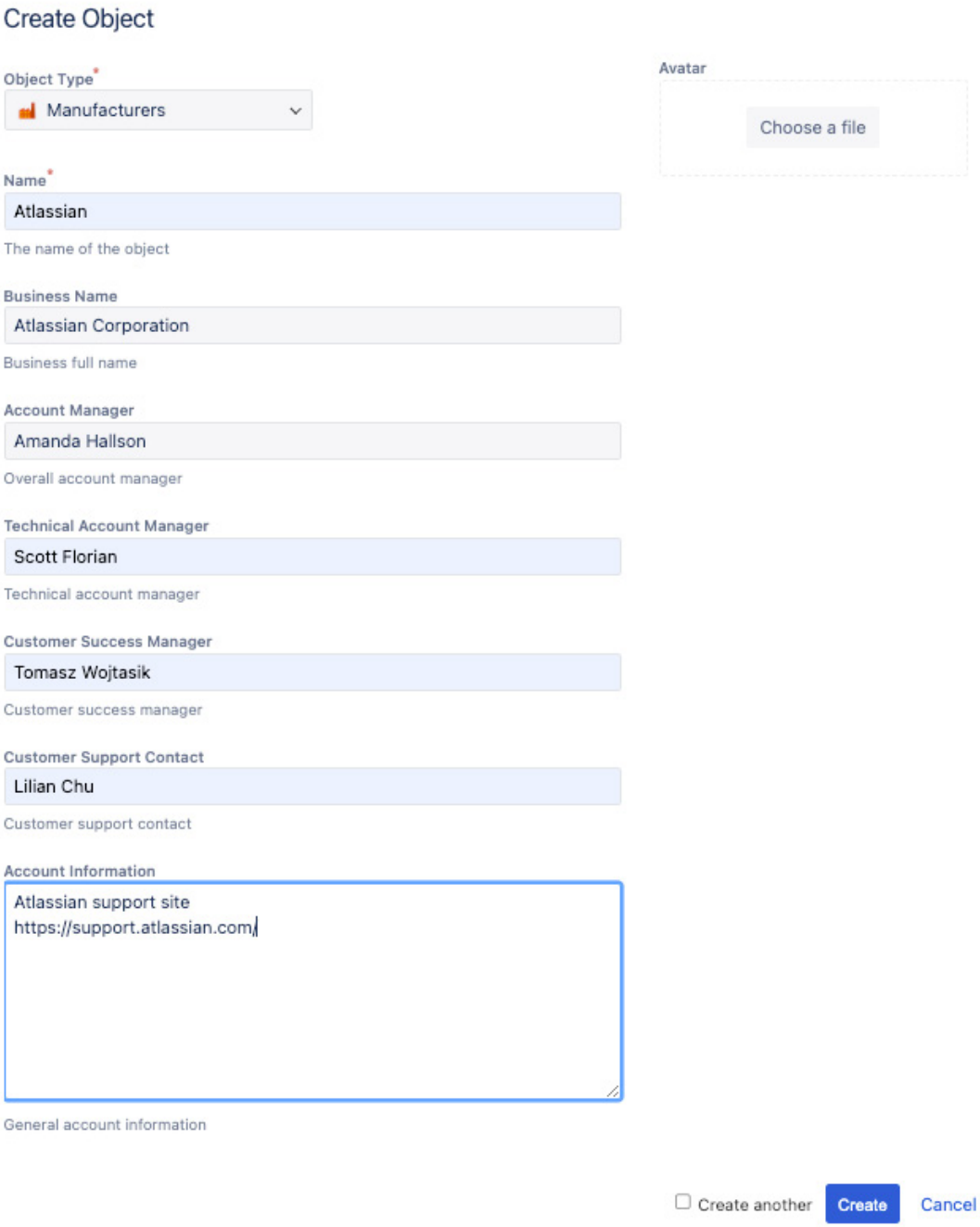
Create an object type for **Manufacturers** and select **Business Partners** as the **Parent**.



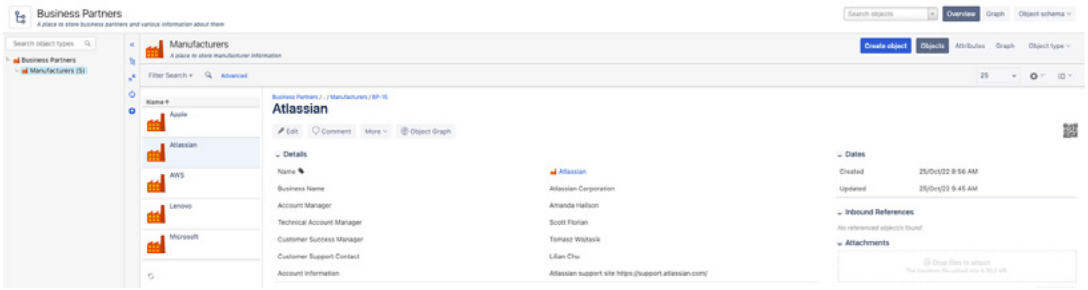


There is no need to add or modify any attributes because the data was copied from the parent object type.

Click the **Create object** button to display the **Create Object** window.

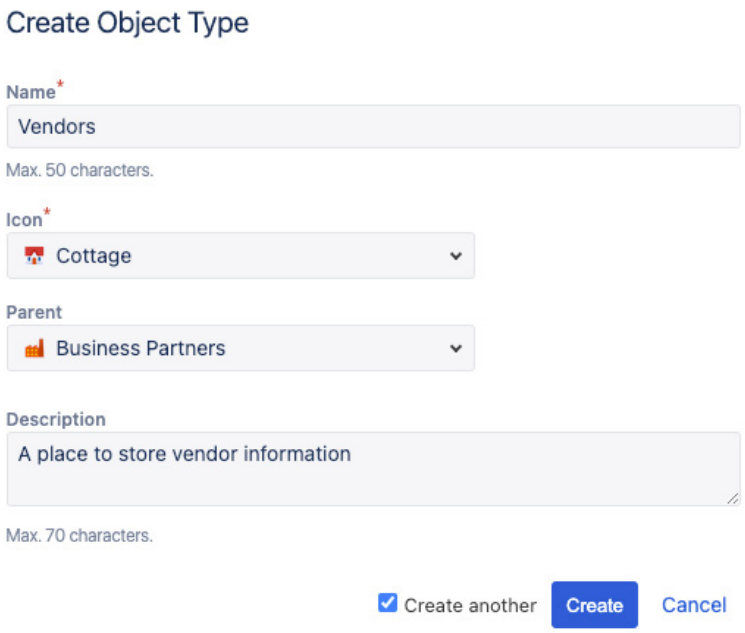


Add your primary asset manufacturer data; additional manufacturer objects can be created when needed.

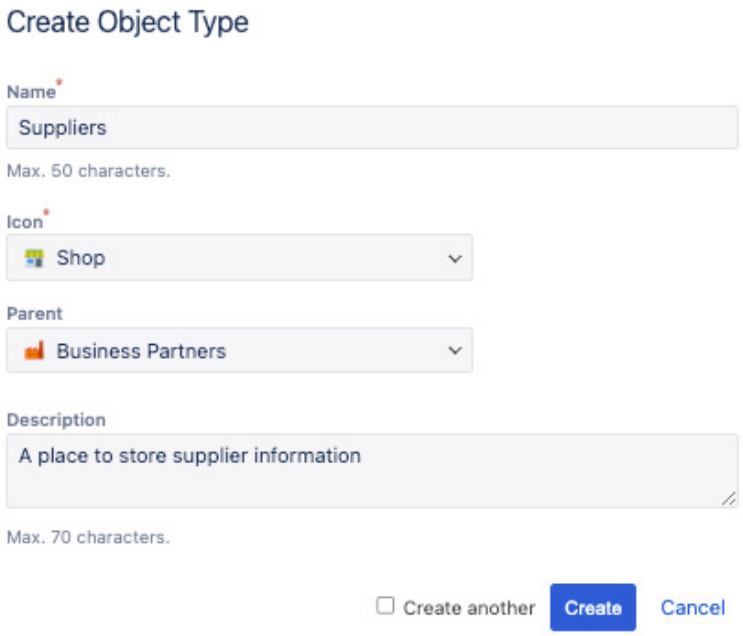


Step 8 - Create an object types and objects for Vendors and Suppliers

Create an object type for **Vendors** and select **Business Partners** as the **Parent**.

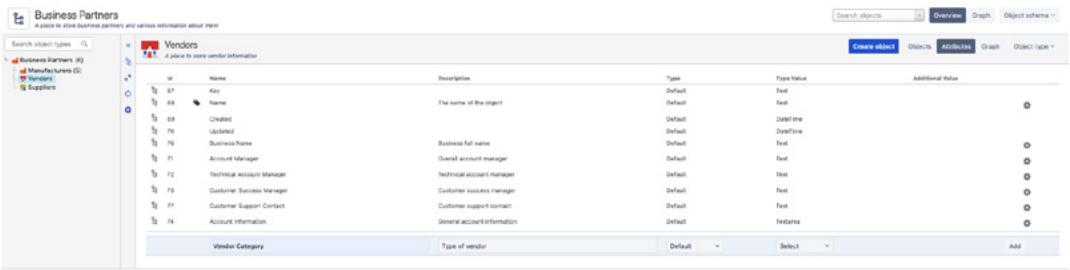


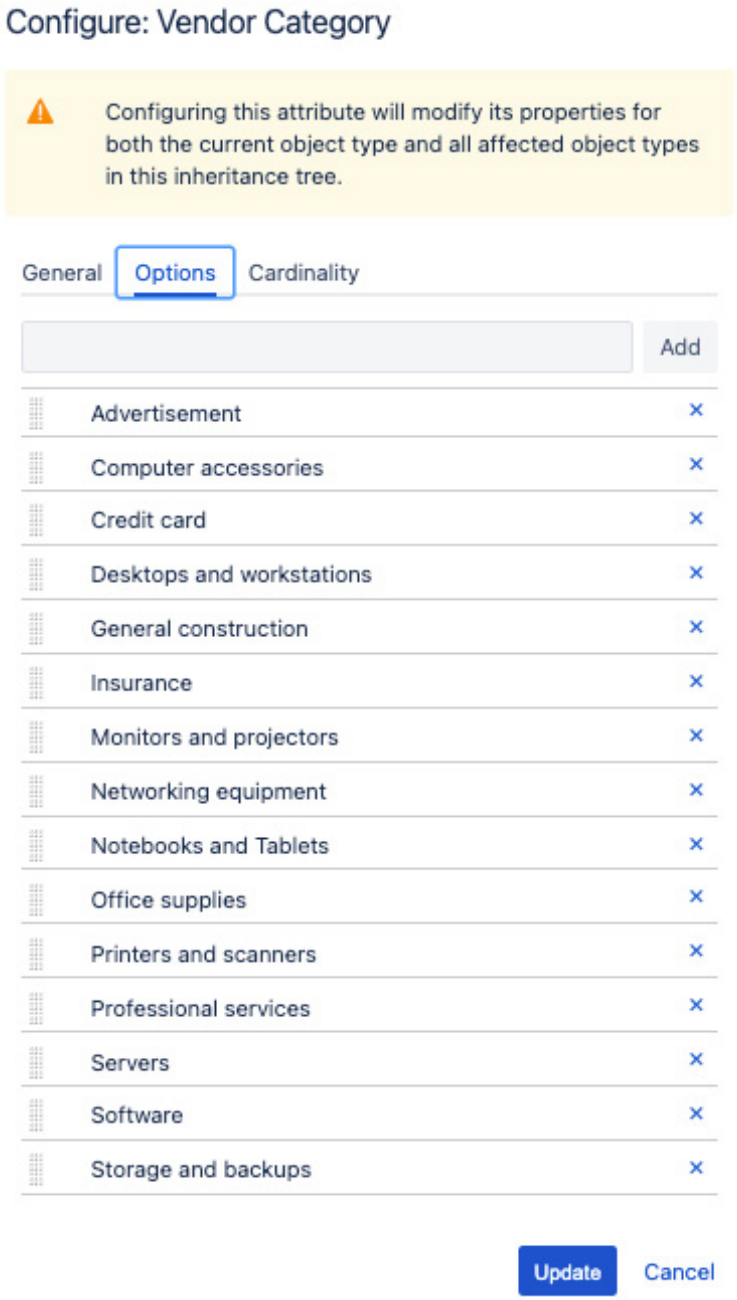
Create an object type for **Suppliers** and select **Business Partners** as the **Parent**.



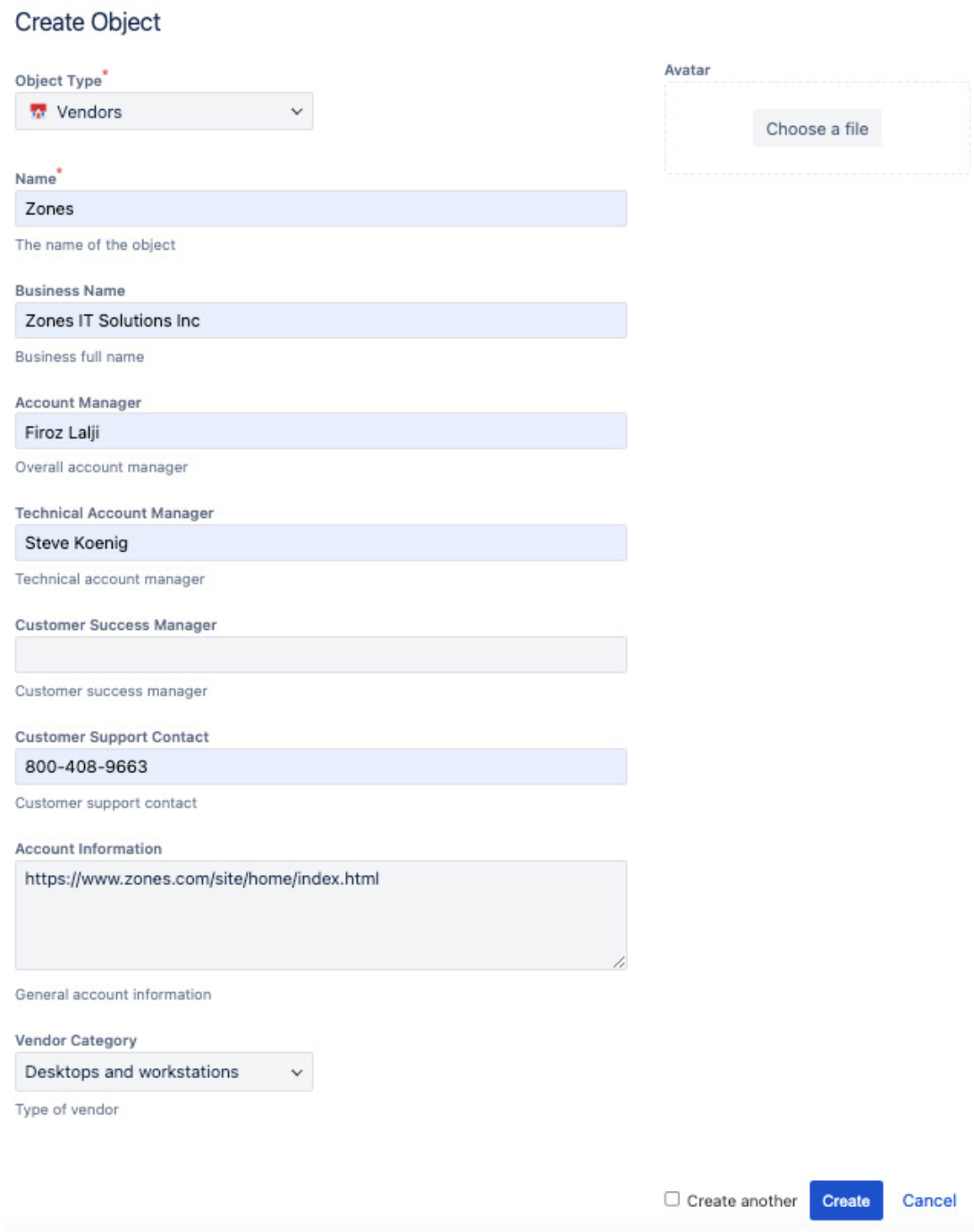
All attributes are copied from the parent object type to the children. If you want to track more information in children object types, additional attributes are simple to include.

Select the **Vendors** object type, click on **Attributes**, and add another attribute.



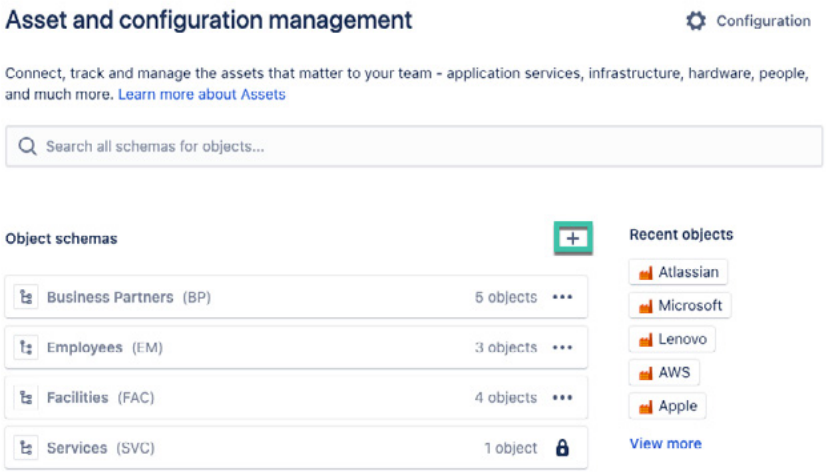

Configure: Vendor Category
Configuring this attribute will modify its properties for both the current object type and all affected object types in this inheritance tree.
General
Options
Cardinality
Add
Advertisement
Computer accessories
Credit card
Desktops and workstations
General construction
Insurance
Monitors and projectors
Networking equipment
Notebooks and Tablets
Office supplies
Printers and scanners
Professional services
Servers
Software
Storage and backups
Update
Cancel

Add your primary vendor data; additional supplier and vendor objects can be created when needed.

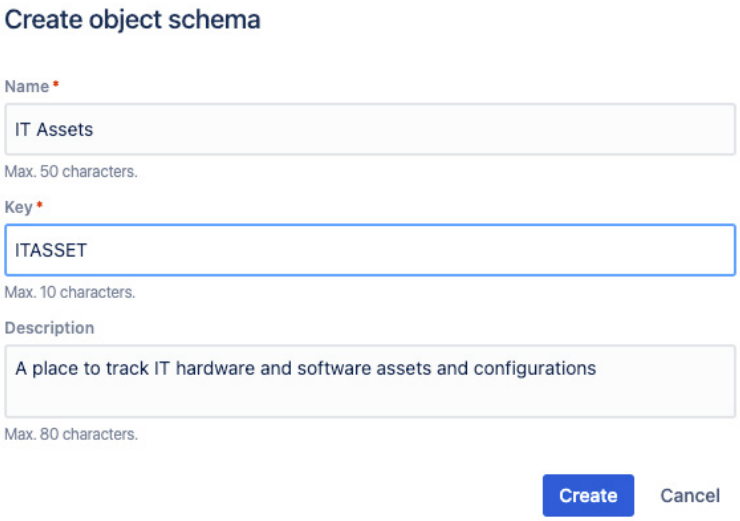


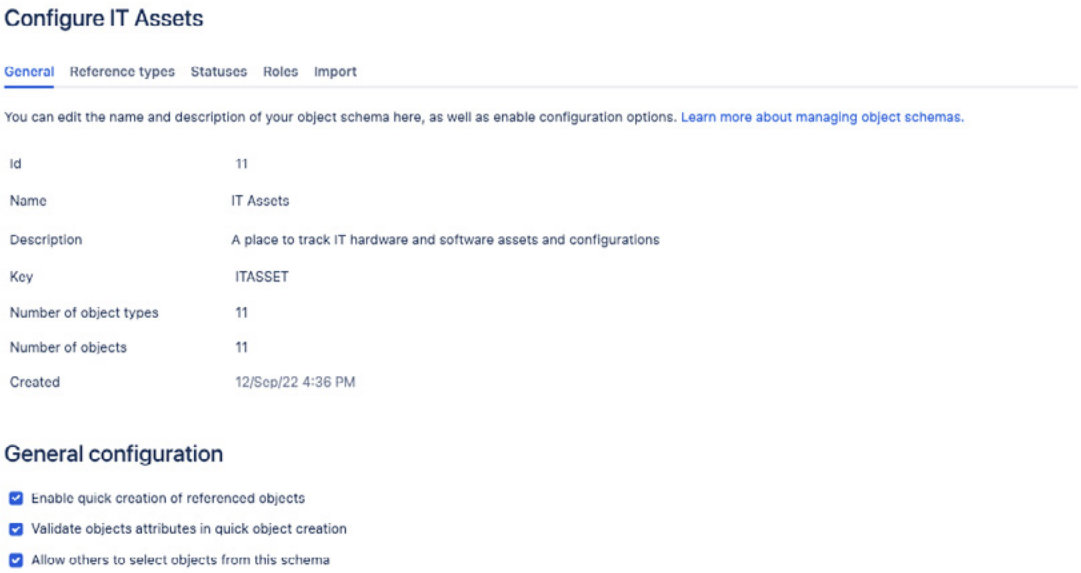
Step 9 - Create an object schema for IT Assets

Now let's create an **IT Assets** object schema that will reference information in the **Employees** and **Manufacturer** object types.



Display the **Create object schema** window, enter object schema information, then configure the object schema to select *Allow others to select objects from this schema*.



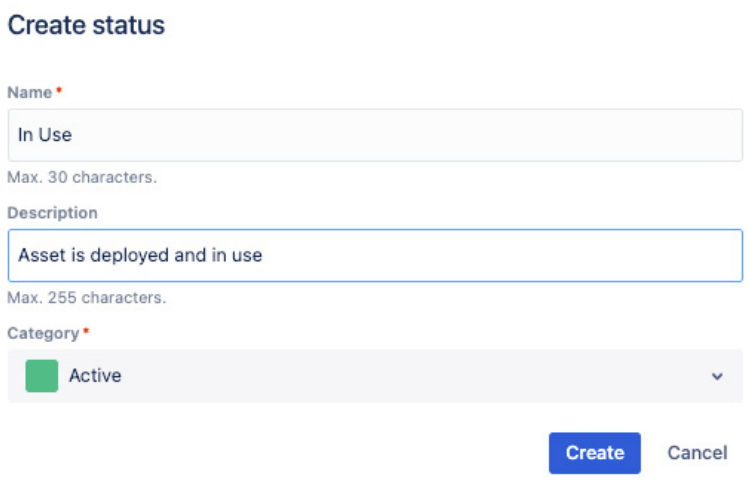


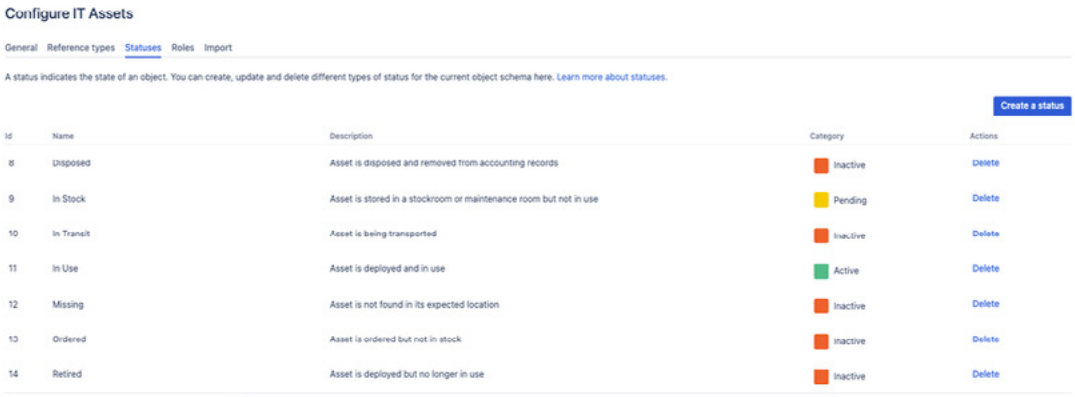
PRO TIP

The object schema key is a series of alphanumeric characters and identifies the data related to your object schema. You can enter a **Key** value for an object schema when you create the record; however, the data cannot be modified later.

Select the **Statues** option for Asset object schema and add options:

Name	Category	Description
Ordered	Inactive	Asset is ordered but not in stock
In Transit	Inactive	Asset is being transported
In Stock	Pending	Asset is in stock but not in use
In Use	Active	Asset is deployed and functioning
Retired	Inactive	Asset is deployed but no longer in use
Disposed	Inactive	Asset is disposed and removed from accounting records
Missing	Inactive	Asset is not found in its expected location

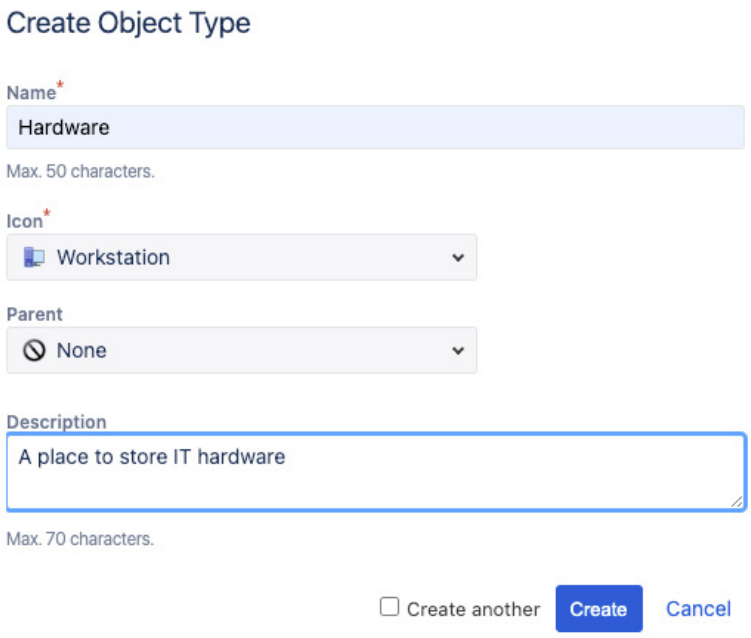




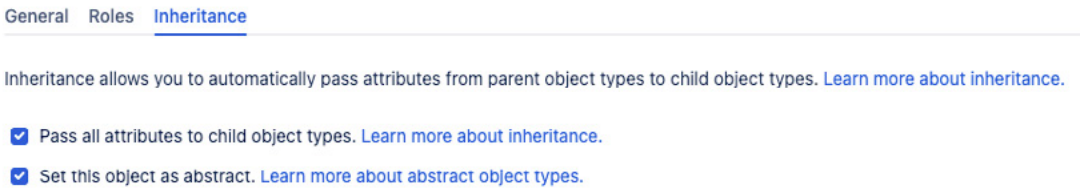
PRO TIP

This use case is focusing on status values for assets; however, ***Assets*** can extend functionality as the organization's needs change and grow. For example, additional status values (or use existing values) can be included to track asset/CI operational statuses and link the value to discovery operations.

Step 10 - Create an object type for Hardware



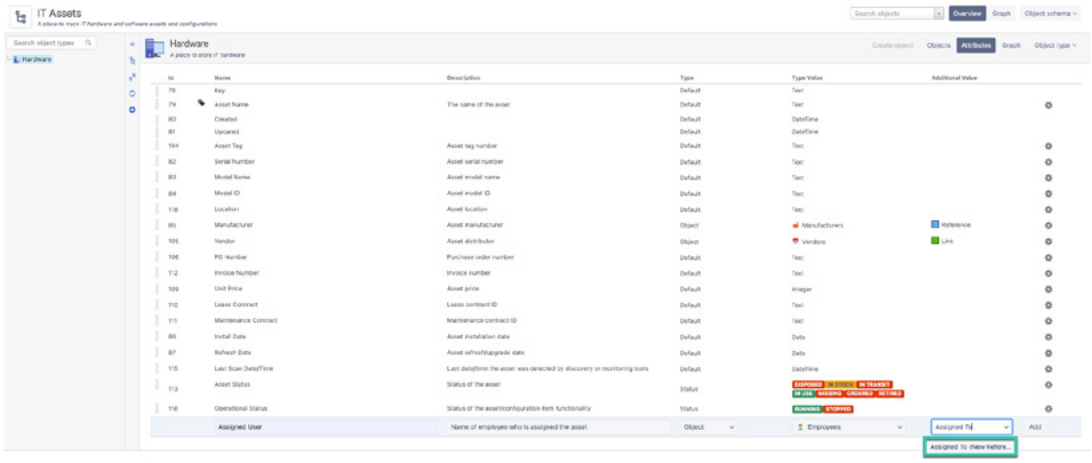
Create an object type for hardware and configure the object type with *Pass all attributes to child object types* and *Set this object as abstract*



Select the **Attributes** option for Hardware object type and add the following items:

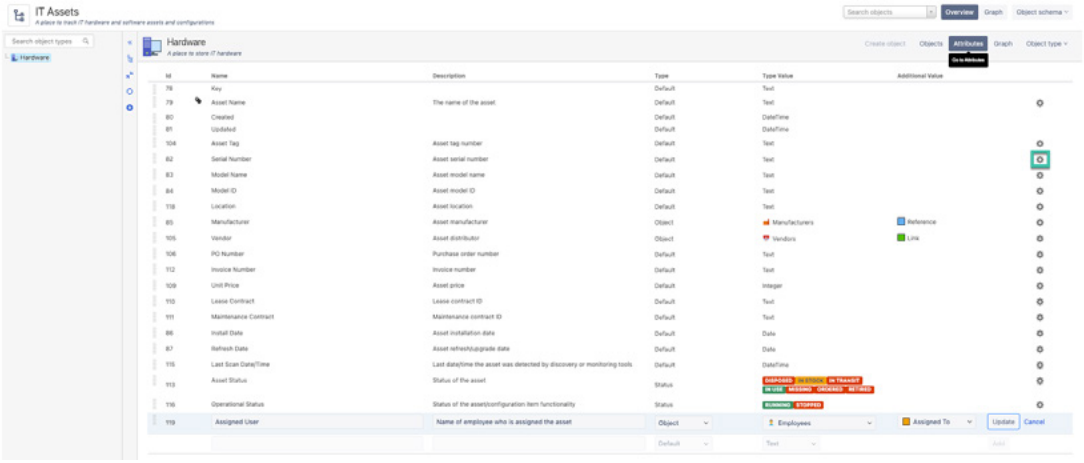
Name	Type	Value	Additional Value
Asset Tag	Default	Text	
Serial Number	Default	Text	
Model Name	Default	Text	
Model ID	Default	Text	
Location	Default	Text	
Manufacturer	Object	Manufacturers	Referece
Vendor	Object	Vendors	Link
PO Number	Default	Text	
Invoice Number	Default	Text	
Unit Price	Default	Text	
Lease Contract	Default	Text	
Maintenance Contract	Default	Text	
Purchase Date	Default	Date	
Refresh Date	Default	Date	
Last Scan Date/Time	Default	DateTime	

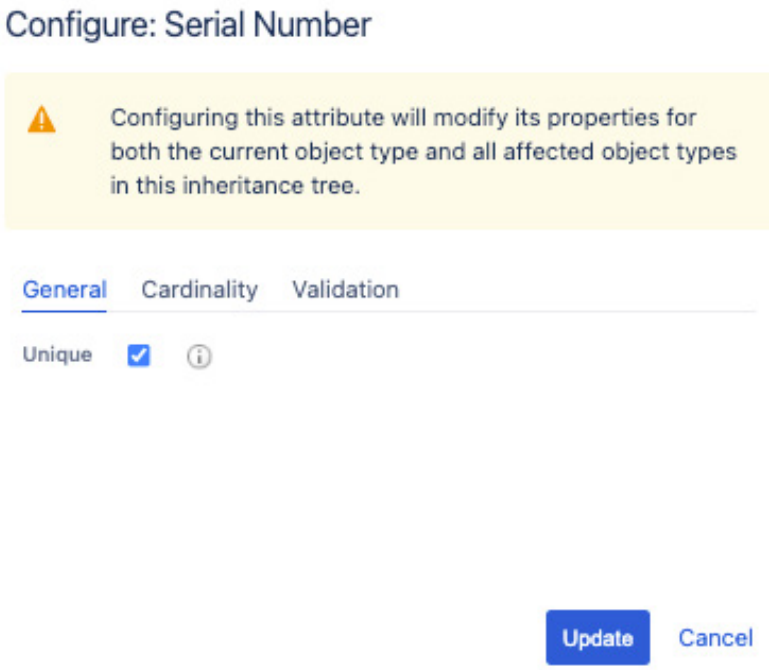
Asset Status	Status	Ordered In Transit In Stock In Use Missing Retired Disposed	
Operational Status	Status	Running Stopped	
Assigned User	Object	Employees	Assigned To



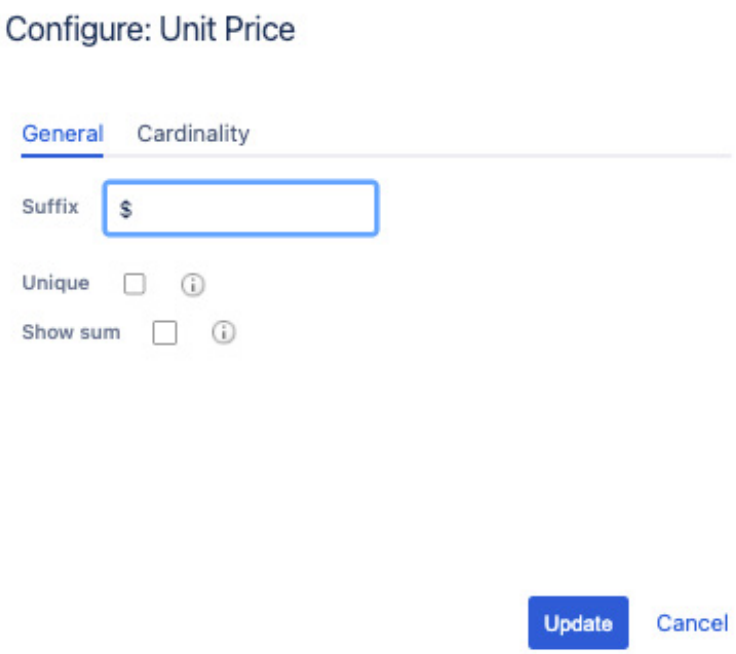
For the **Name** attribute, update the name data to “Asset Name.”

Click the **cog icon** and configure the **Asset Tag**, and **Serial Number** attributes to be unique, so you avoid duplicate asset data.


IT Assets
Hardware


Configure: Serial Number
Configuring this attribute will modify its properties for both the current object type and all affected object types in this inheritance tree.
General
Cardinality
Validation
Unique
Update
Cancel

For **Unit Price** attribute, configure the suffix to include a currency symbol.

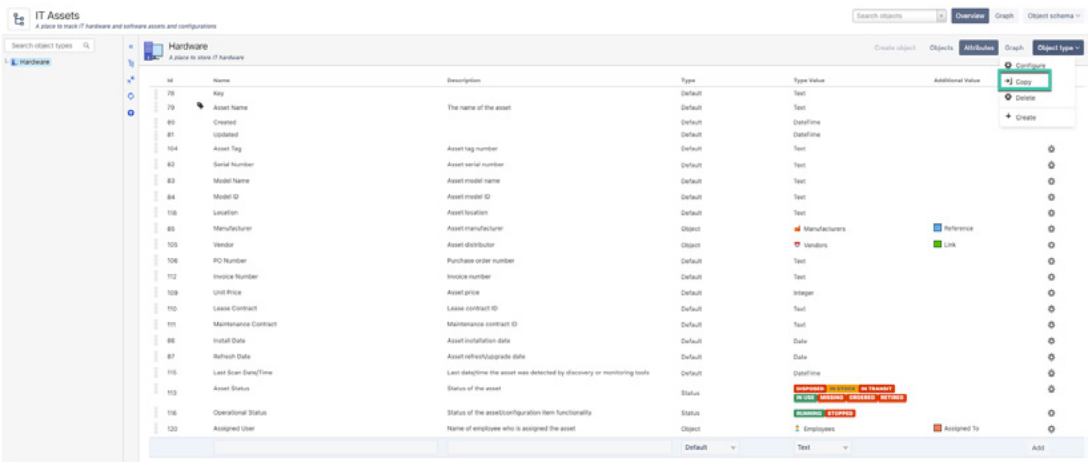


For the **Assigned User** attribute, you use an existing reference value or enter a new reference value; simply click on the item to enter the new value.

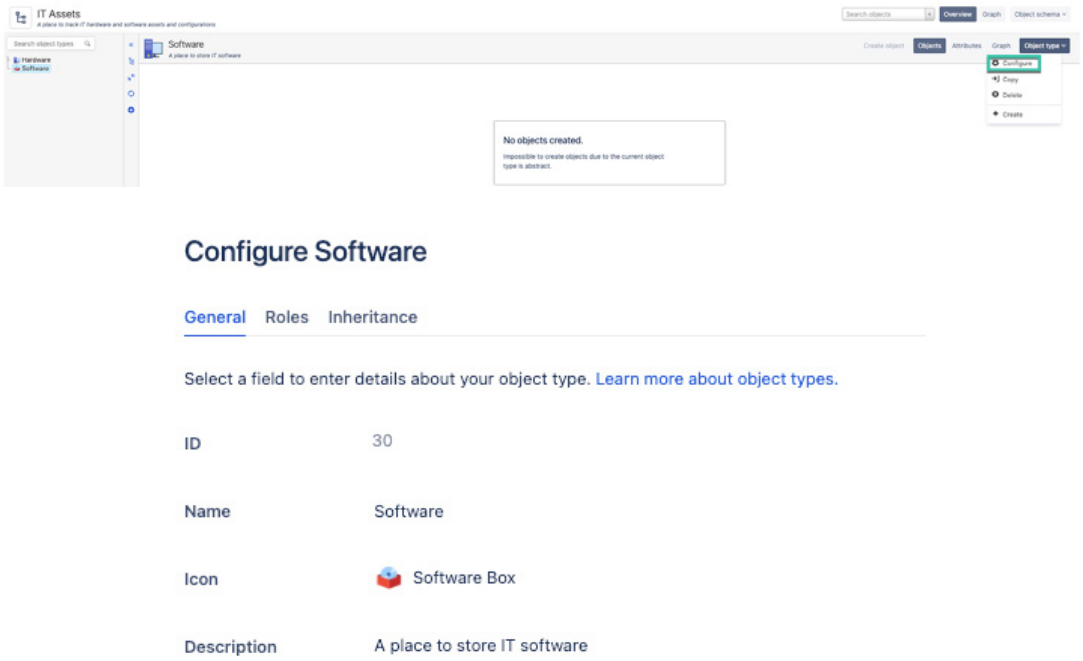
PRO TIP

Although this use case does not include software IT assets, we can copy and update the Hardware object type for future use managing software assets.

Copy the Hardware object type and create a new object type for Software.

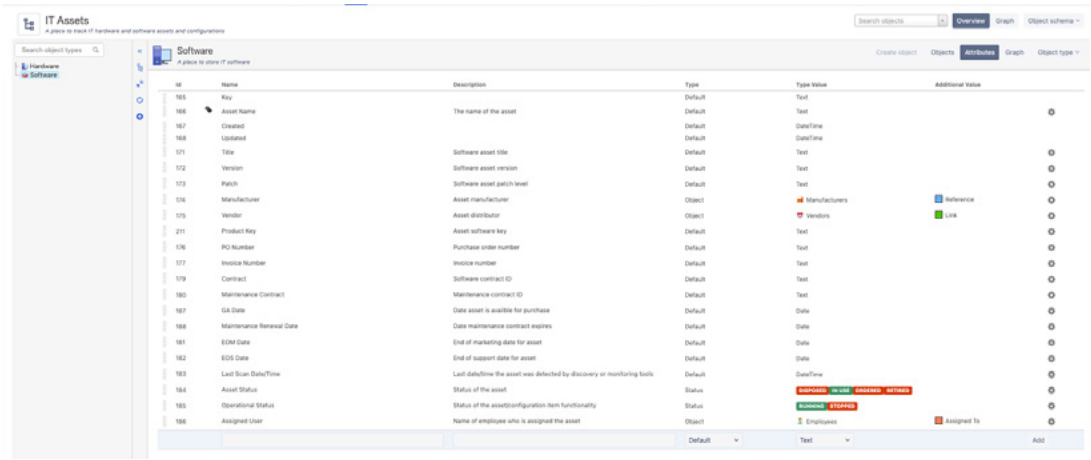


Configure the Software object type and update the **Icon** and **Description** values.



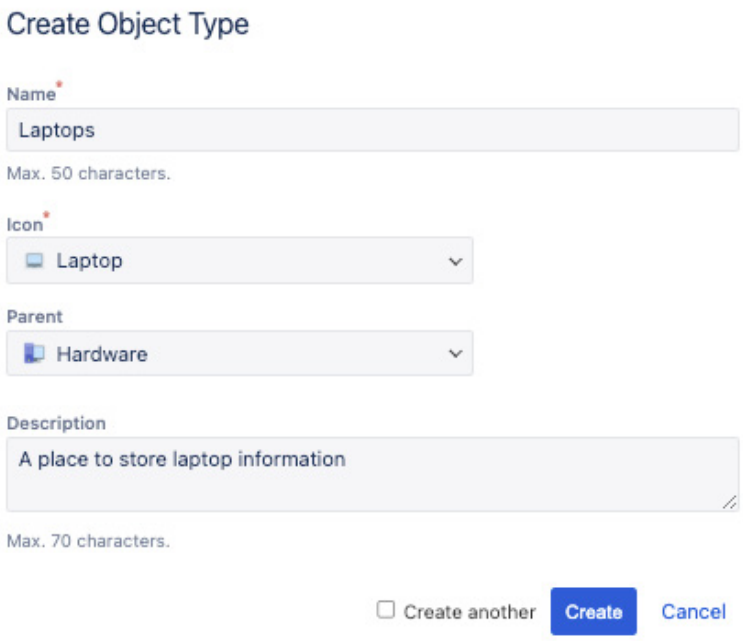
Select the **Attributes** option for Software object type and include relevant options and delete any unnecessary items:

Name	Type	Value	Additional Value
Title	Default	Text	
Version	Default	Text	Default
Patch	Default	Text	
Manufacturer	Object	Manufacturers	Reference
Vendor	Object	Vendors	Link
Product Key	Default	Text	
PO Number	Default	Text	
Invoice Number	Default	Text	
Contract	Default	Text	
Maintenance Contract	Default	Text	
GA Date	Default	Date	
Maintenance Renewal Date	Default	Date	
EOS Date	Default	Date	
Last Scan Date/Time	Default	DateTime	
Asset Status	Status	Ordered In Use	
Operational Status	Status	Running Stopped	
Assigned User	Object	Employees	Assigned To

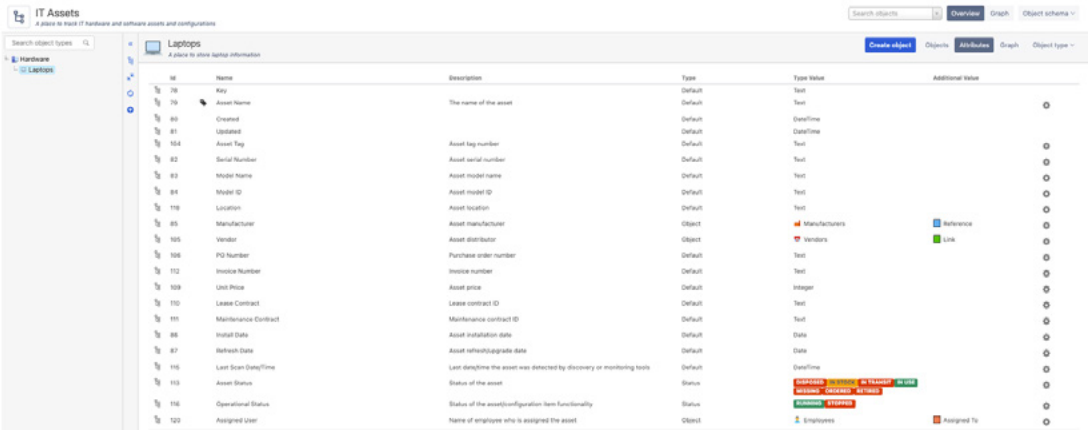


Step 11 - Create an object type for Laptops

Create an object type for Laptops and select Hardware as the Parent.



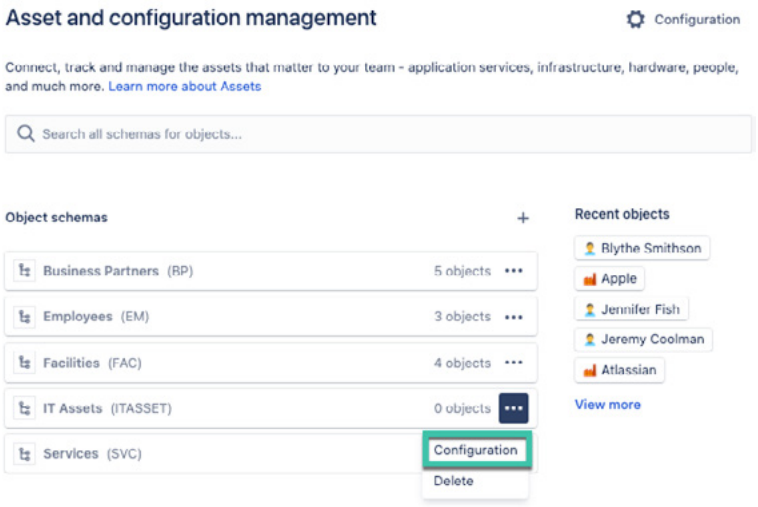
There is no need to add or modify any attributes because the data was copied from the parent object type.

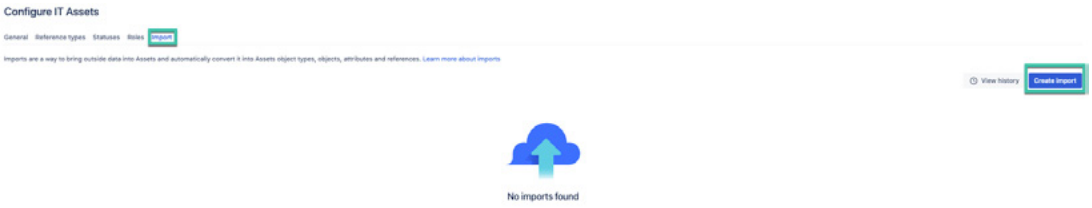


Additional object types can be created for Desktop Software and Applications, when needed.

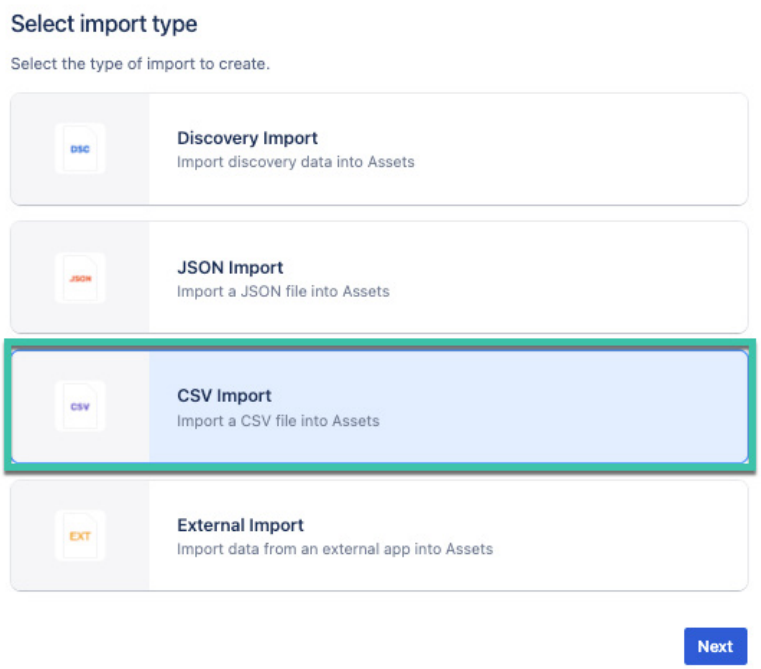
Step 12 - Configure the IT Assets object schema to import laptop data

Update the attached **laptop.csv** file and change the **Assigned User** data to match your Jira users.

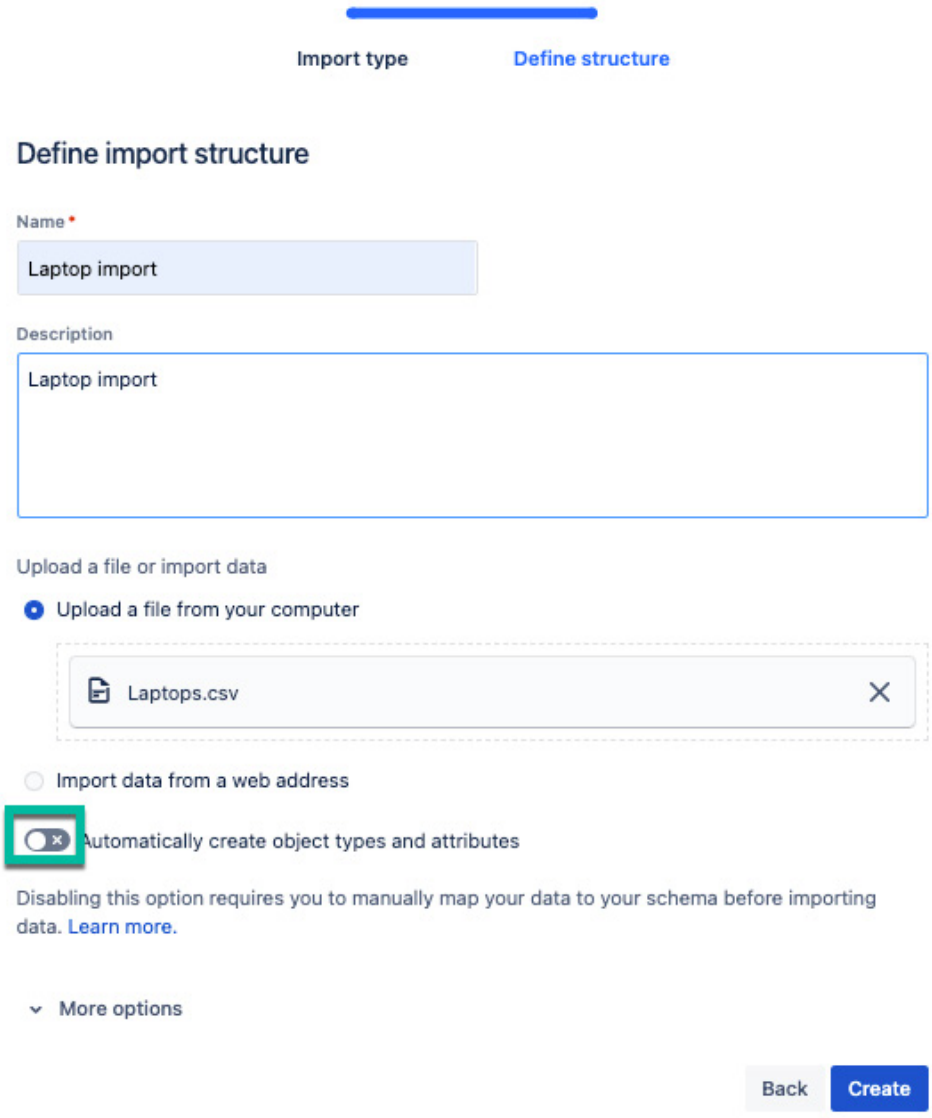




Select the **Import** option, then click **Create import**



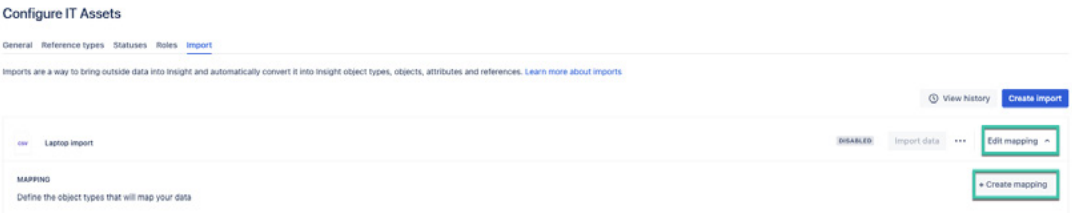
Select **CSV import** type and click **Next**



Enter a name for the import and select your CSV file.

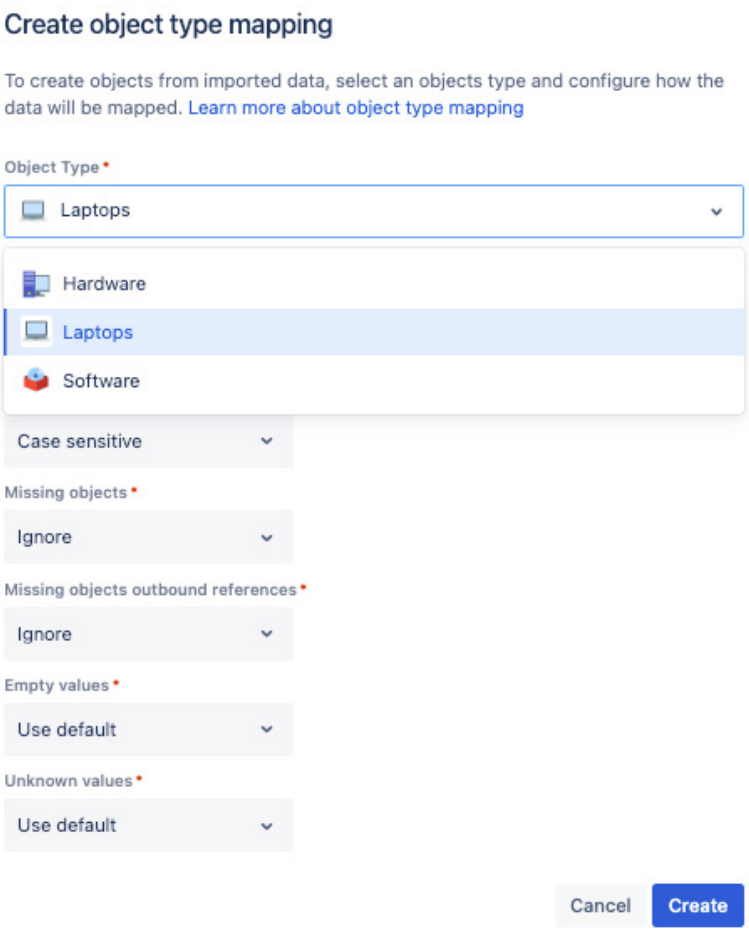
Deselect *Automatically create object types and attributes* because you have already defined the **Laptops** object type and attributes. There is no need to update the additional options.

Click **Create**



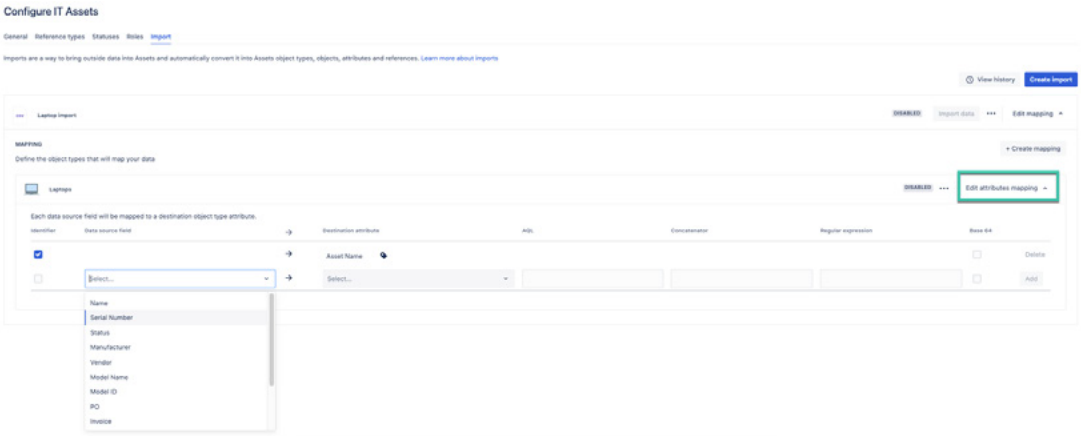
Select **Edit mapping**

Click **Create mapping**



Select Laptops object type and click **Create**.

Select **Edit attributes mapping**, add data from source file mapping to attributes.

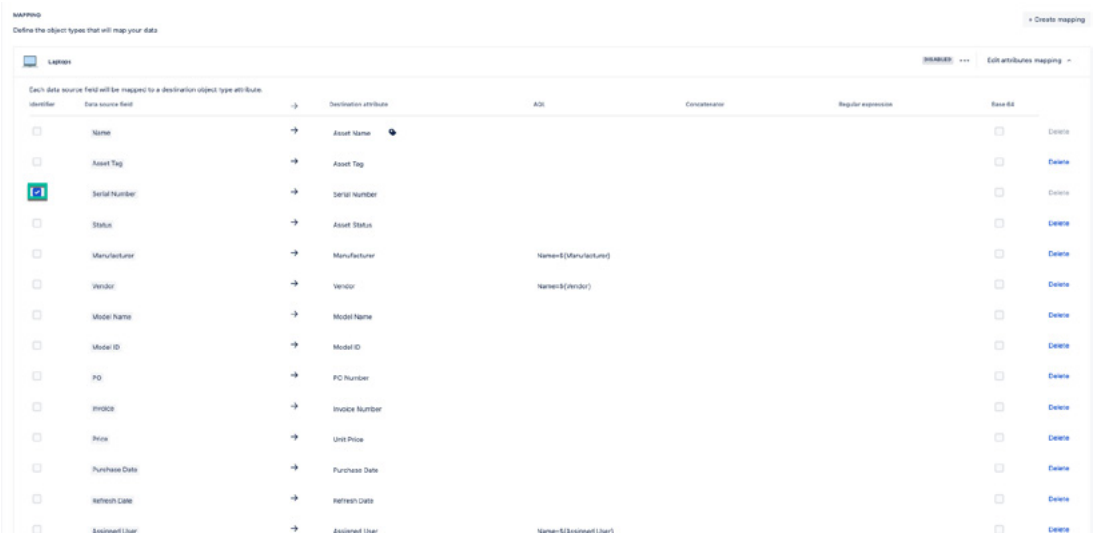


Create the attribute mappings for relevant items :

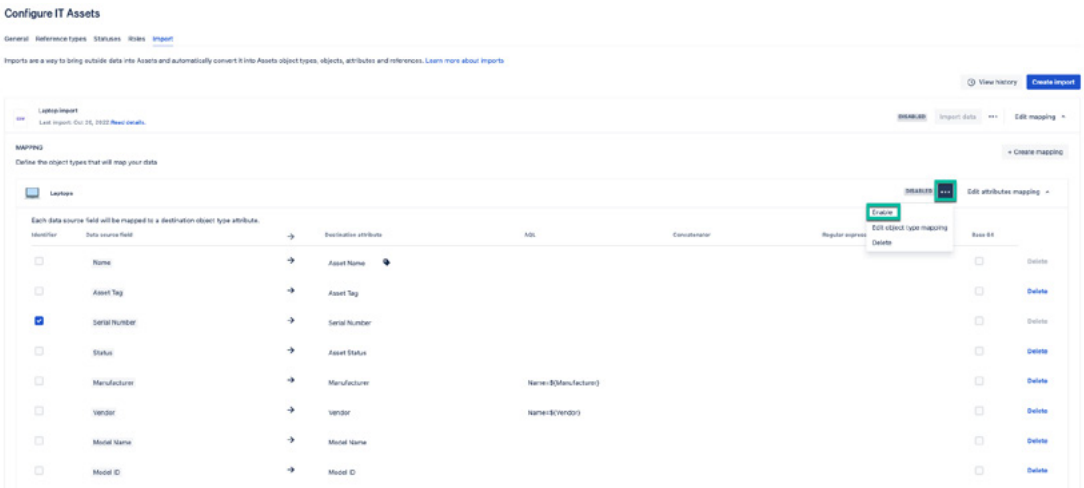
Data source field	Destination attribute	AQL
Name	Asset Name	
Asset Tag	Asset Tag	
Serial Number	Serial Number	
Status	Asset Status	
Manufacturer	Manufacturer	Name=${Manufacturer}
Vendor	Vendor	Name=${Vendor}
Model Name	Model Name	
Model ID	Model ID	
PO	PO Number	
Invoice	Invoice Number	
Price	Unit Price	
Purchase Date	Purchase Date	
Refresh Date	Refresh Date	
Assigned User	Assigned User	Name=${Assigned User}

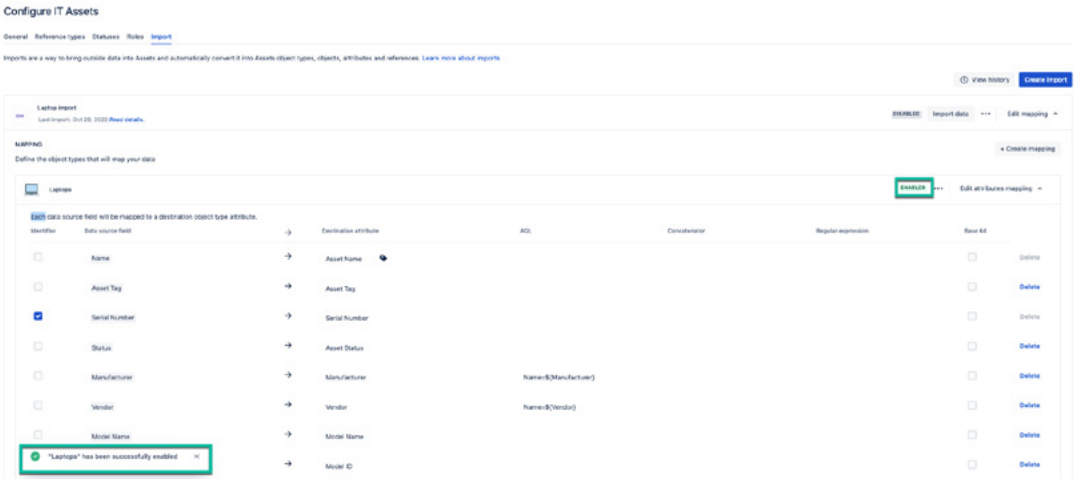
For the **Manufacturer**, **Vendor**, and **Assigned User** mapping, you want to create a relationship between the laptop data and the data you created earlier. The AQL syntax is Attribute=${Name of placeholder}; the placeholder is the column label in the external data source.

Select **Serial Number** as unique identifier.

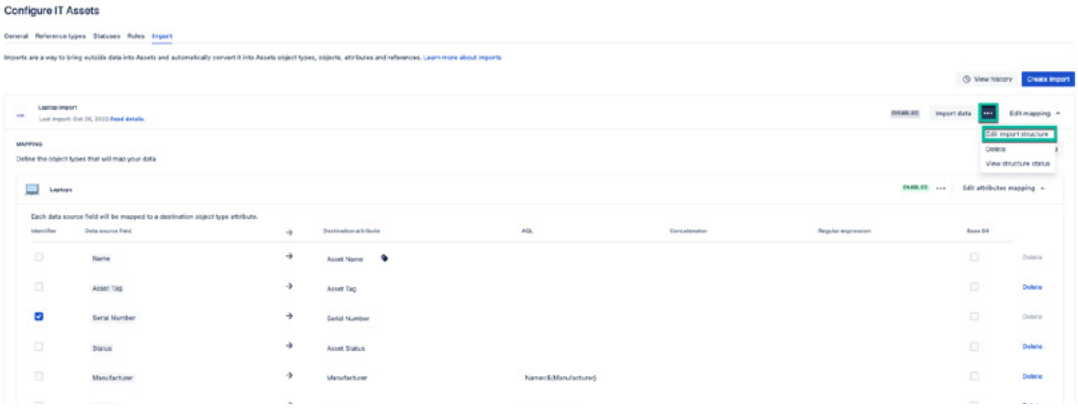


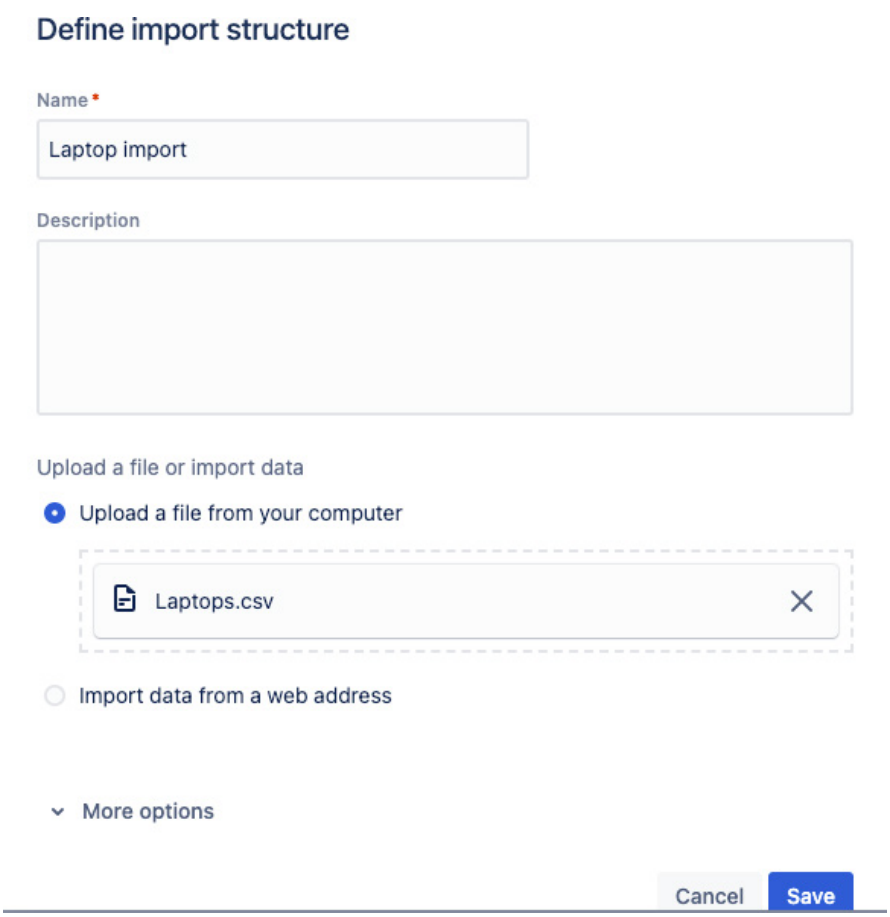
Select **Enable** to activate the attribute mapping.



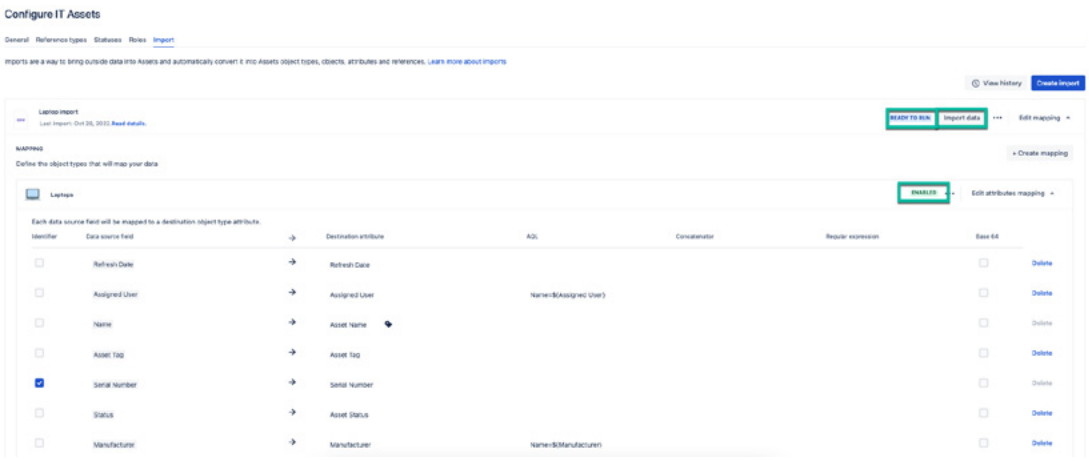


Select **Edit import structure** and save the import structure.

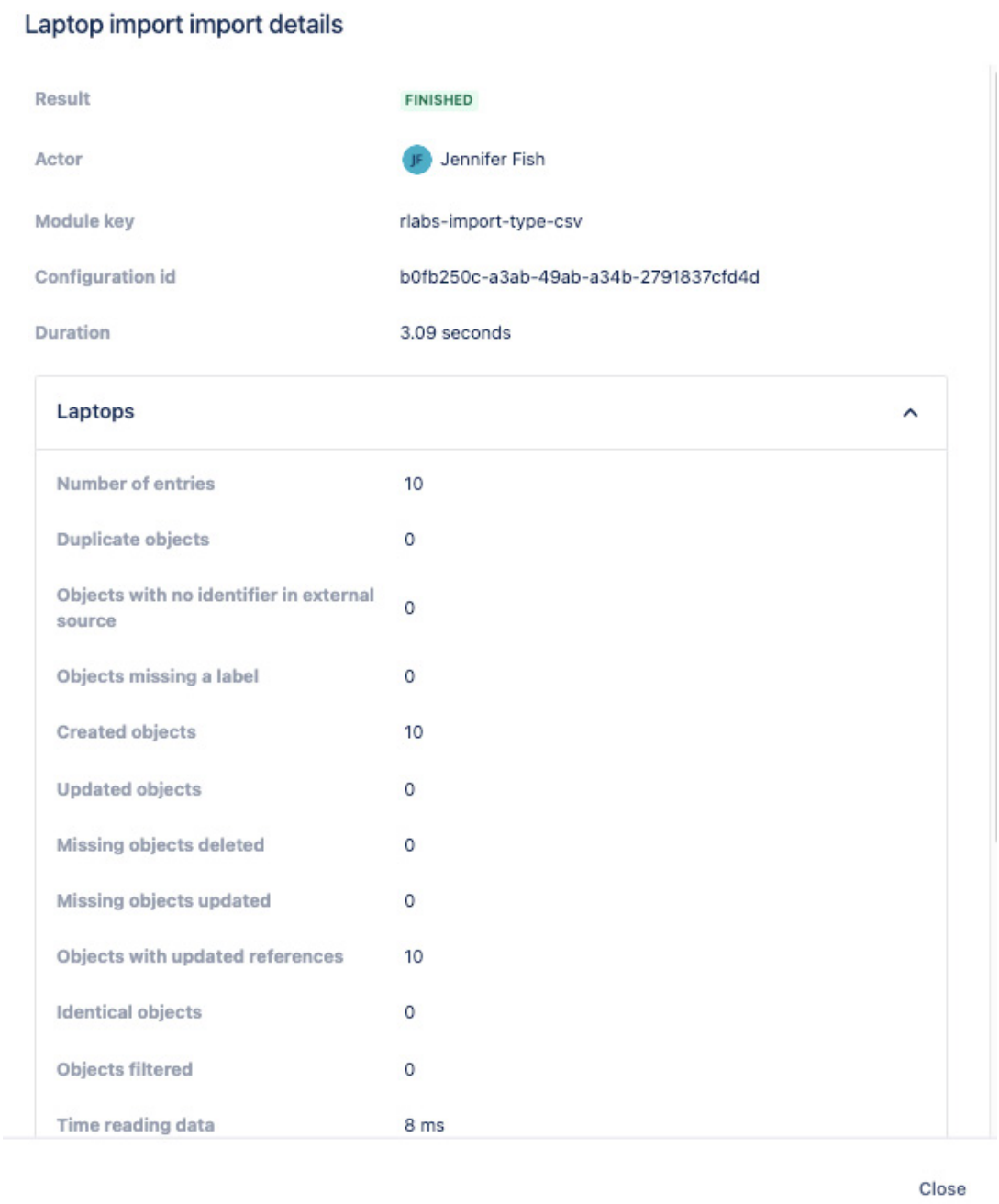




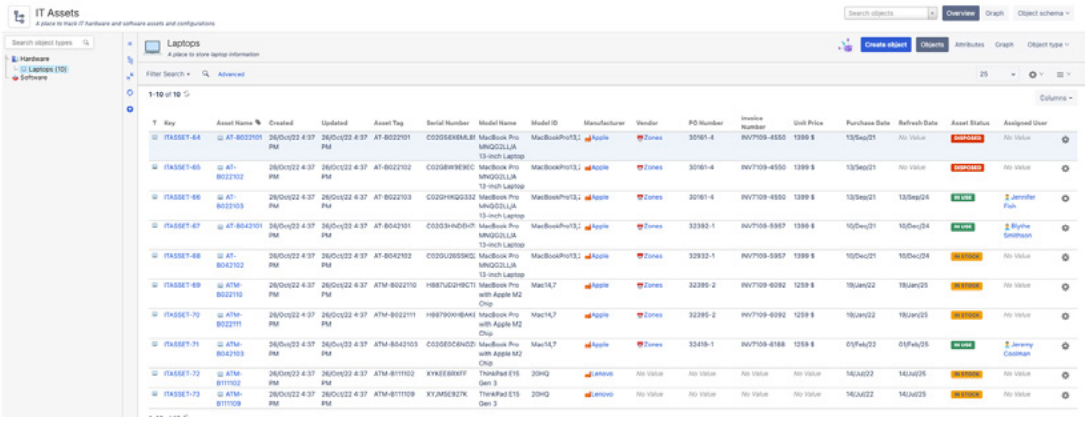
The data is now ready to import, so click **Import data**.



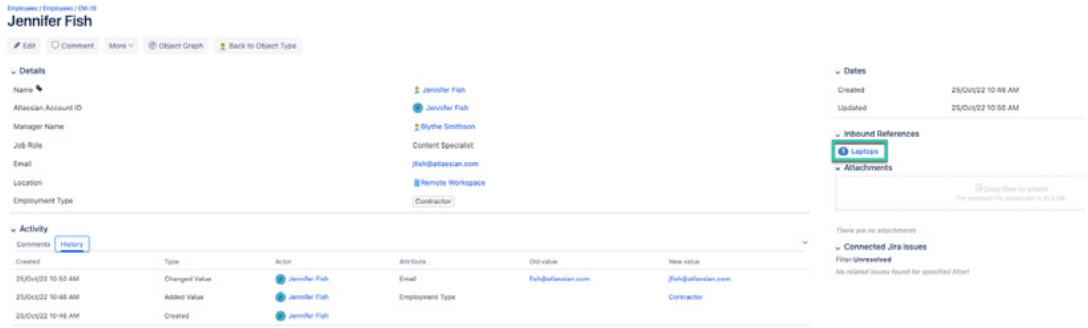
When the data import is complete, click **Read details** to view the results.



The laptop data is now available.

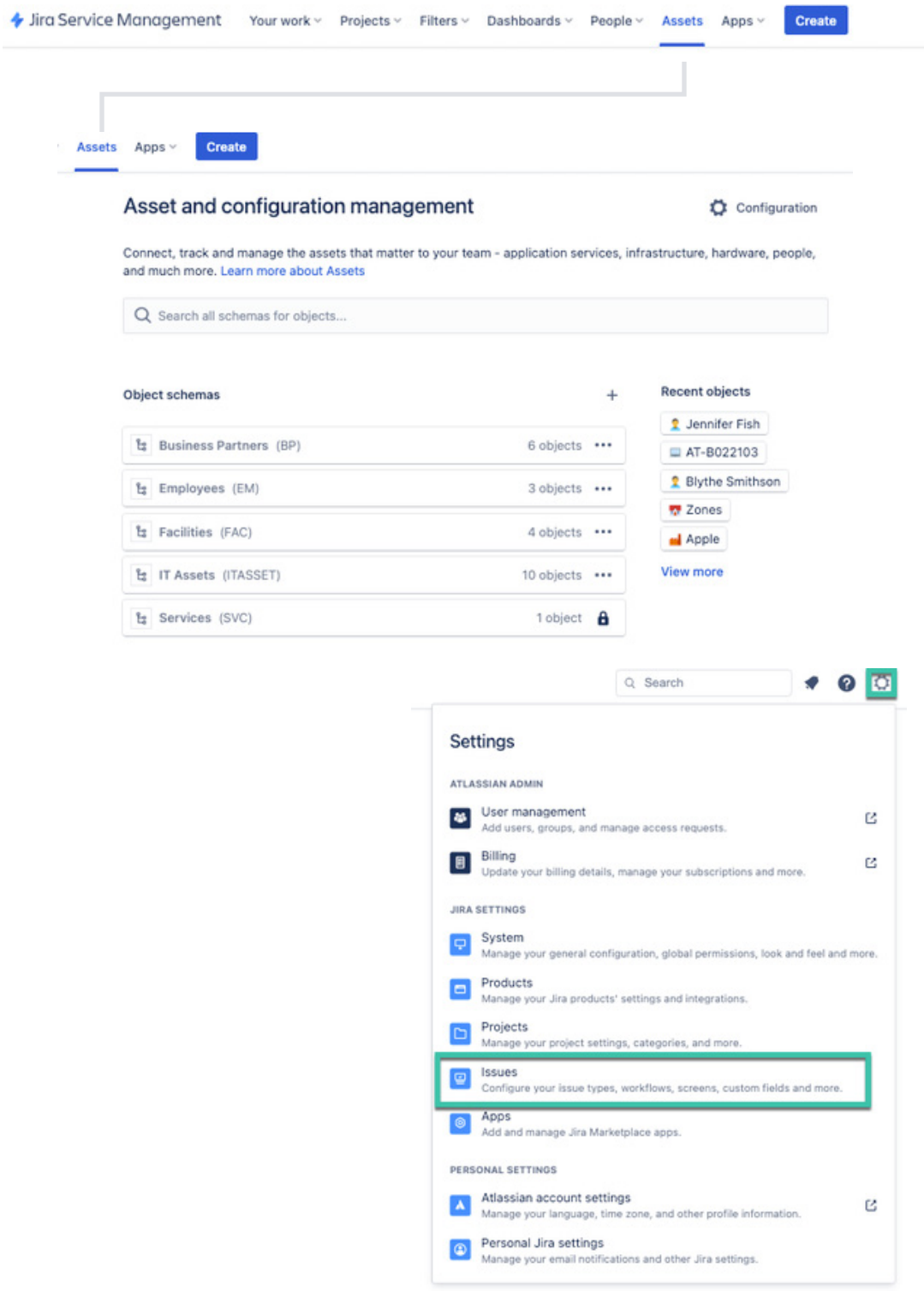


And linked to employees.

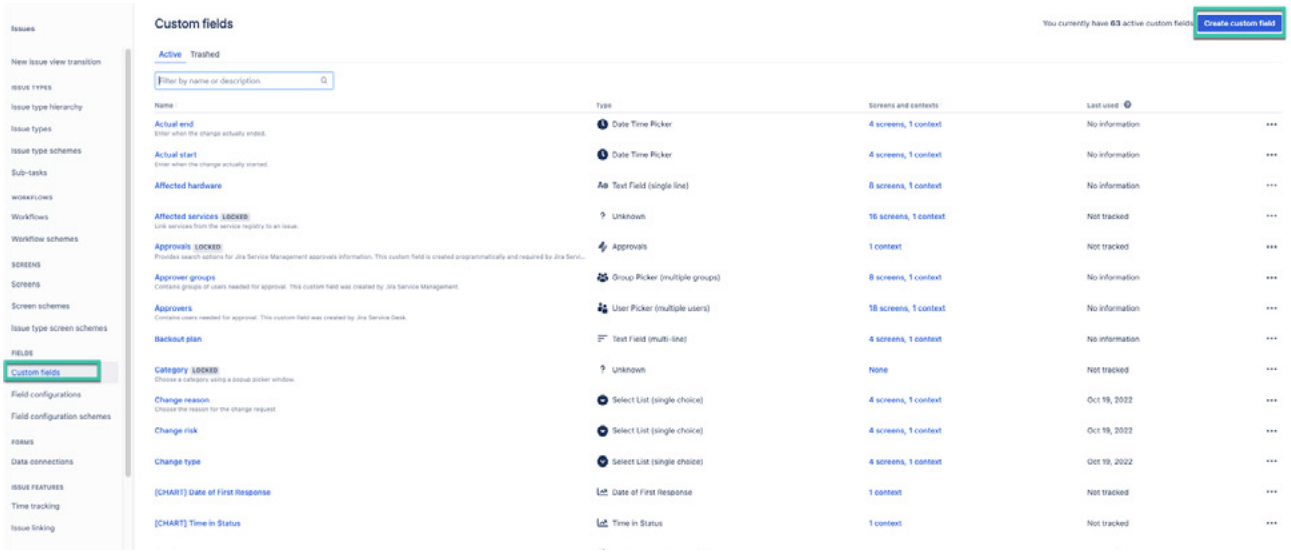


Step 13 - Create custom field to store laptop data in a request

Select the Jira Service Management cog icon and select **Issues**.

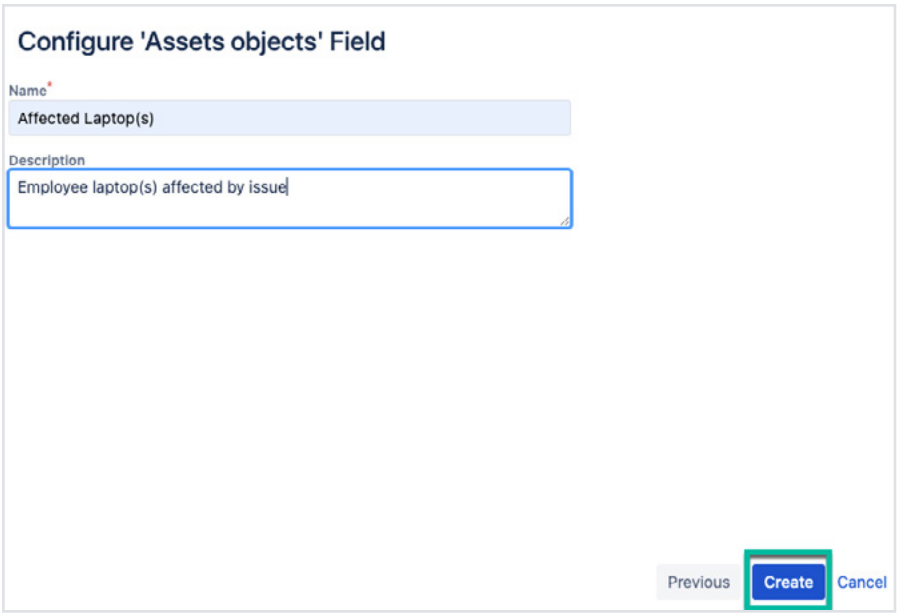


In the **Issues types** pane, select **Custom fields** then click **Create custom field**.

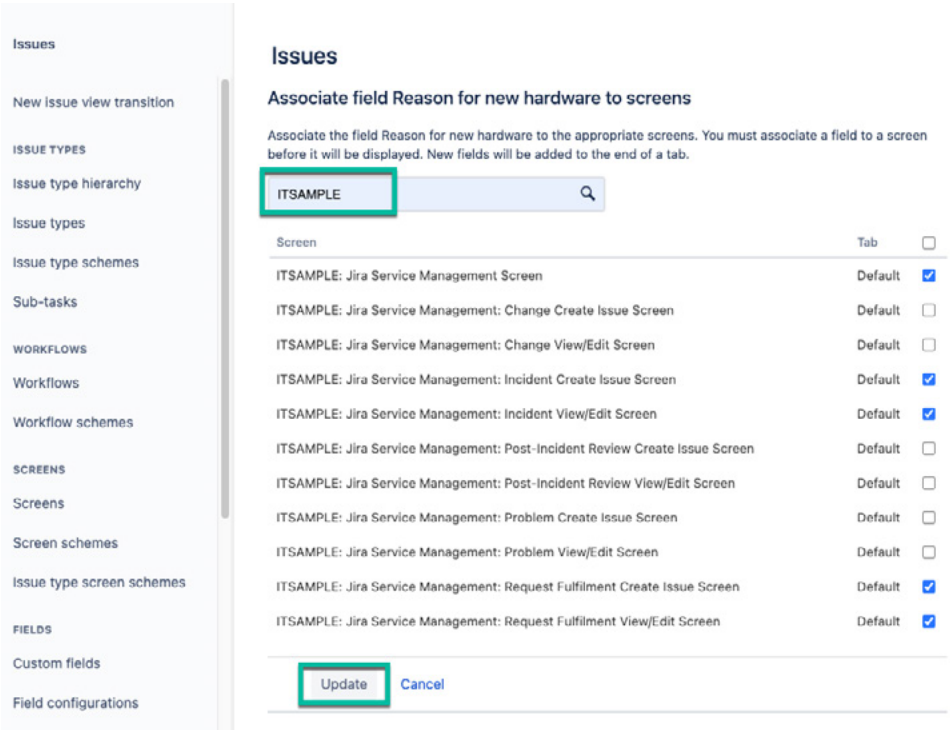


In the **Select a Field Type** window, click **All**, select **Assets objects**, then click **Next**.

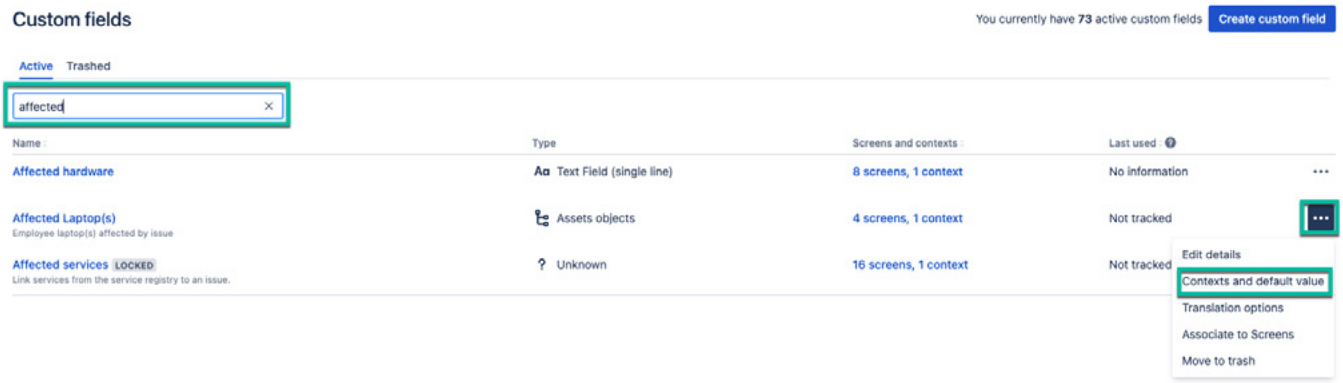
Enter **Name** data for the custom field and create the new field.



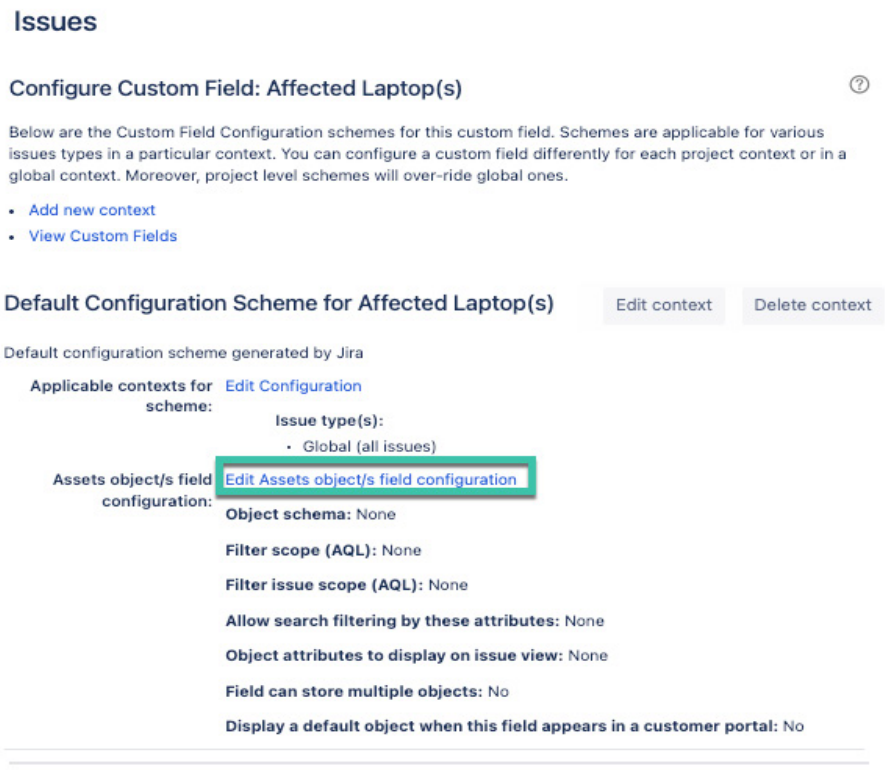
Search for your project and associate the new field to the request and incident screens in your project, then click **Update**.



In the **Custom fields** pane, search for the new field then update the **Context and default values**.



Click **Edit Assets object/s field configuration** to update the configuration context.



Select **IT Assets** as the **Object schema** and enter **objectType="Laptops"** in the **Filter scope (AQL)** field.

To limit the data list to laptops assigned to the issue reporter, include *"Assigned User"."Atlassian Account ID"=${reporter}* in the **Filter issue scope** field.

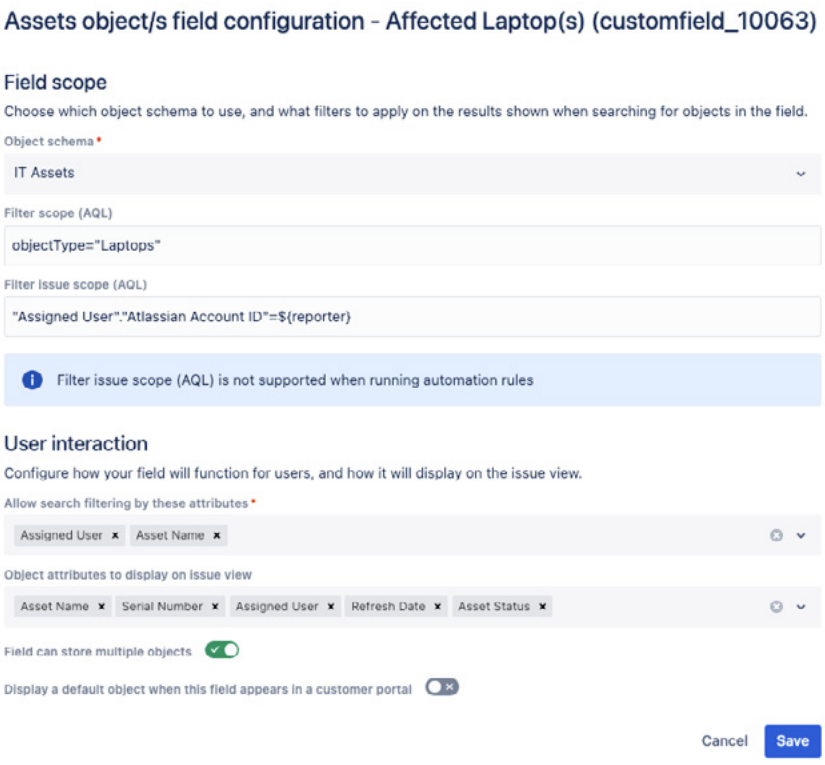
Include the relevant values in the **Allow search** filtering by these attributes field:

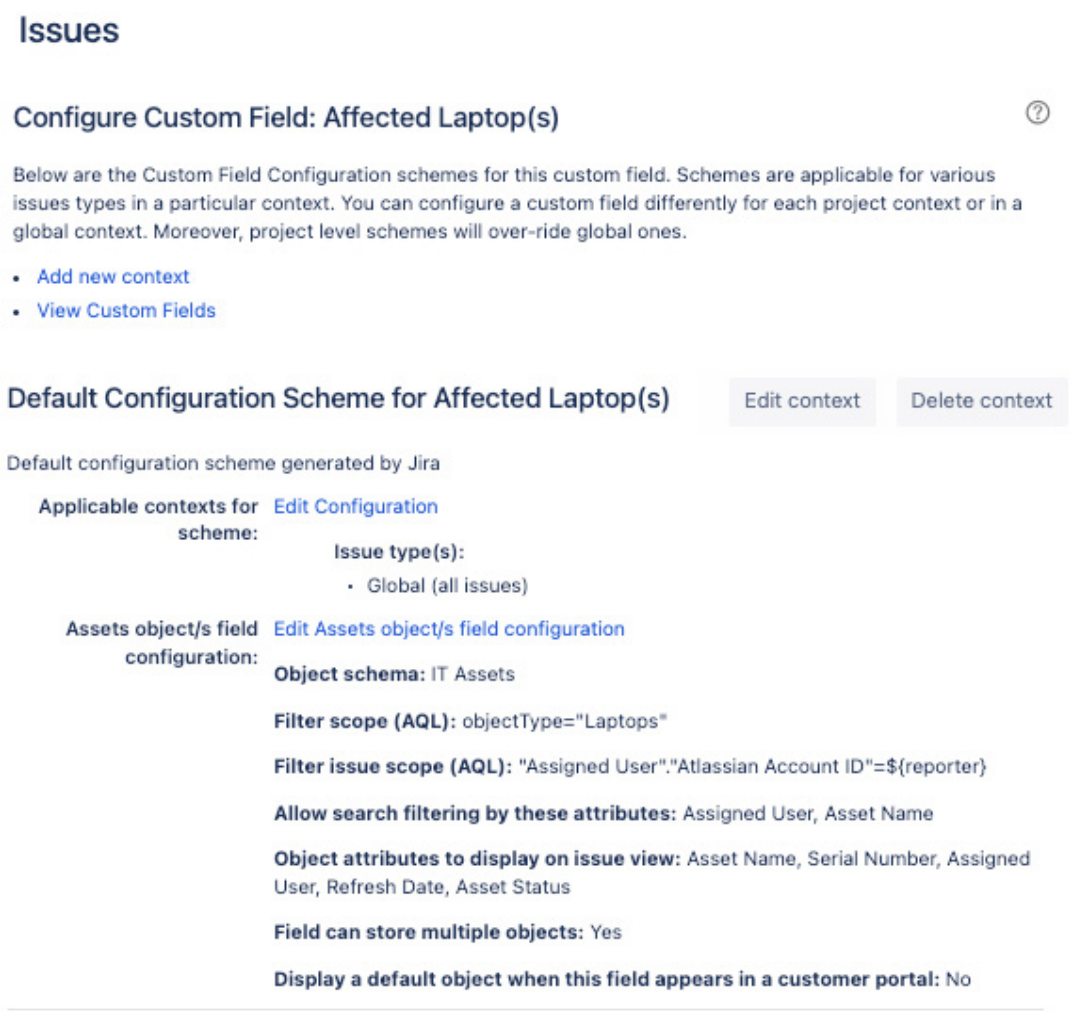
- Assigned User
- Asset Name

Include options in the **Object attributes to display on issue view** field:

- Asset Name
- Serial Number
- Assigned User
- Refresh Date
- Asset Status

Select **Field can store multiple object.**





The custom field is available in the specified project screens.

PRO TIP

AQL Basic syntax

The basic syntax of an AQL query is **<attribute> <operator> <value/ function>**. One or more objects is returned by the query when the attributes of these objects match the operator and value specified.

Example: Owner = “Ted Anderson”

This basic AQL query would return all objects for which the Owner is "Ted Anderson". Note the quotations around "Ted Anderson", since there is a space in the value name.

Dot notation

Dot notation is used in AQL to travel down a reference chain of objects. The format **<attribute>.<attribute> <operator> <value/ function>** will return information based upon objects referenced by the parent object.

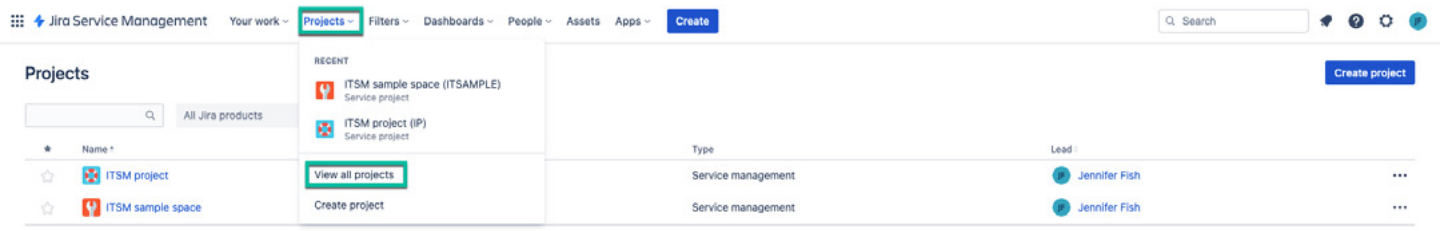
Example: “Belongs to Department”.”Name” = HR

In this case, the Employee object type has a referenced attribute called "Belongs to Department". This query returns all the Employees which belong to the HR department.

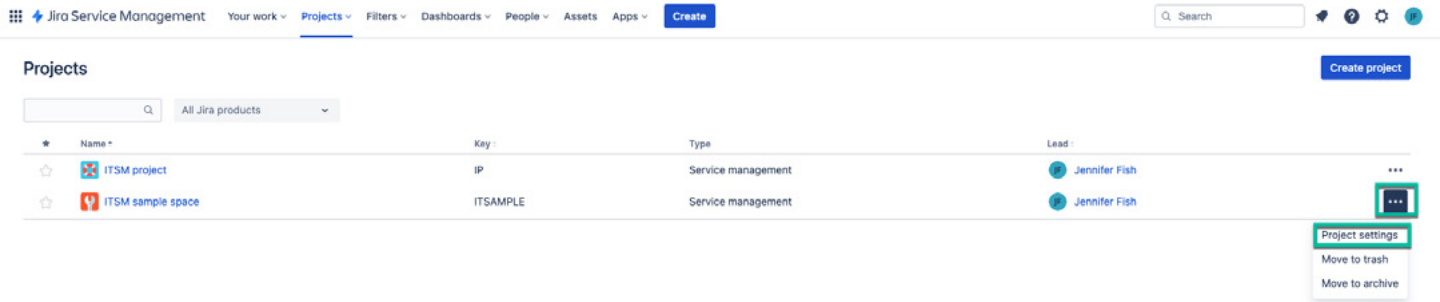
Note that since the referenced attribute contains spaces, it has been enclosed with a pair of double quotes.

Step 14 - Associate new custom fields with a project request

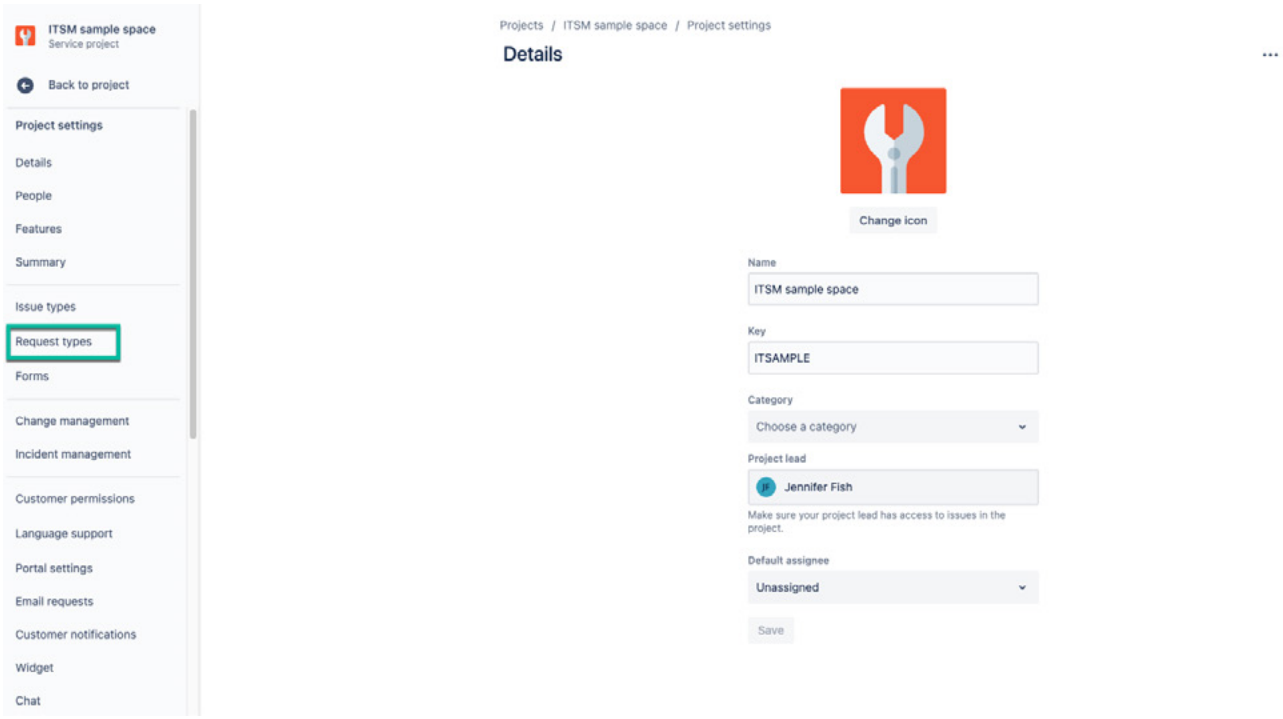
Select **Projects** option in the Jira Service Management main navigation bar and select **View all projects.**

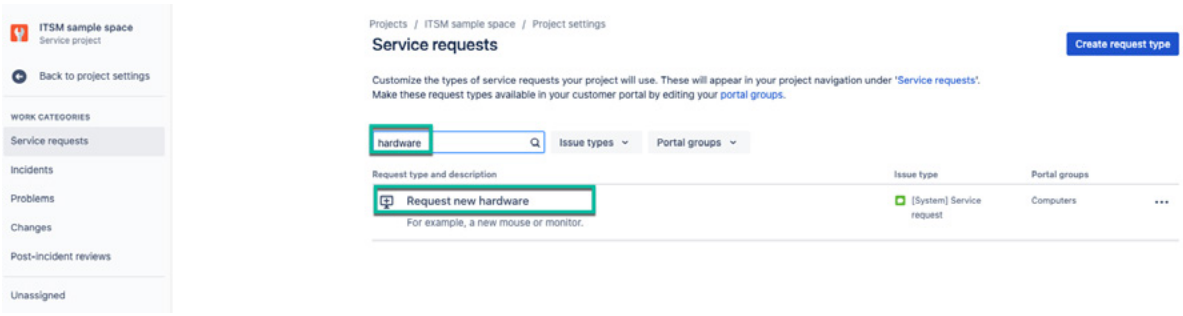


Click on the **ellipse** icon and select **Project settings**.

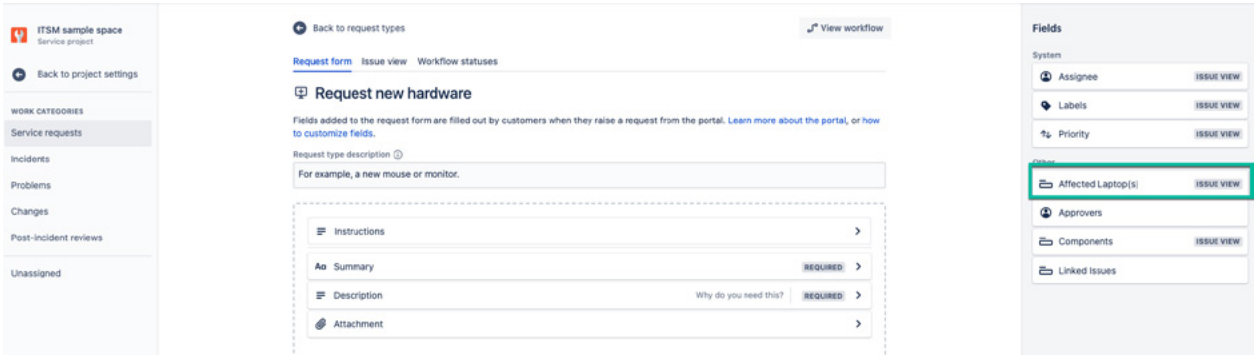


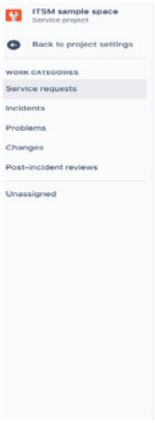
Select **Request types** in the **service project sidebar**, filter on **“hardware”** then click on **Request new hardware** link.

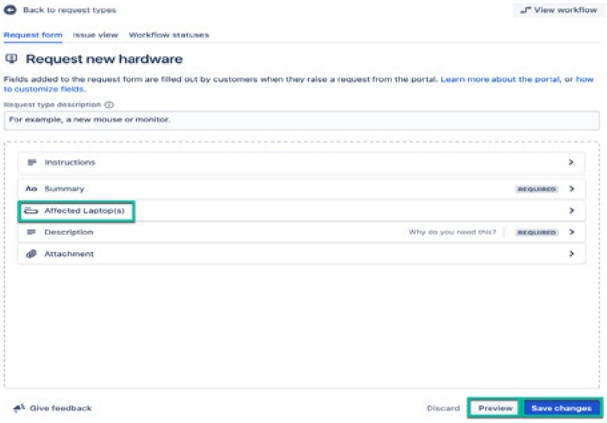


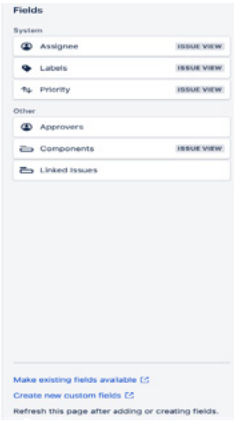


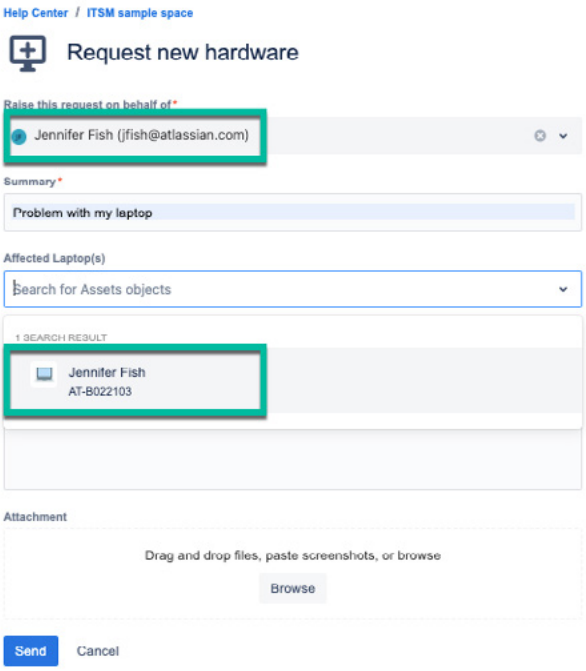
Locate the new custom field in the **Fields** pane, drag-and-drop the field to the **Request new hardware** form, click **Save**, then click **Preview** to see the updated request form.




ITSM sample space
Service project
Back to project settings
WORK CATEGORIES
Service requests
Incidents
Problems
Changes
Post-incident reviews
Unassigned


Back to request types
View workflow
Request form
Issue view
Workflow statuses
Request new hardware
Fields added to the request form are filled out by customers when they raise a request from the portal. Learn more about the portal, or how to customize fields.
Request type description
For example, a new mouse or monitor.
Instructions
Summary
REQUIRED
Affected Laptop(s)
Description
Why do you need this?
REQUIRED
Attachment
Give feedback
Discard
Preview
Save changes

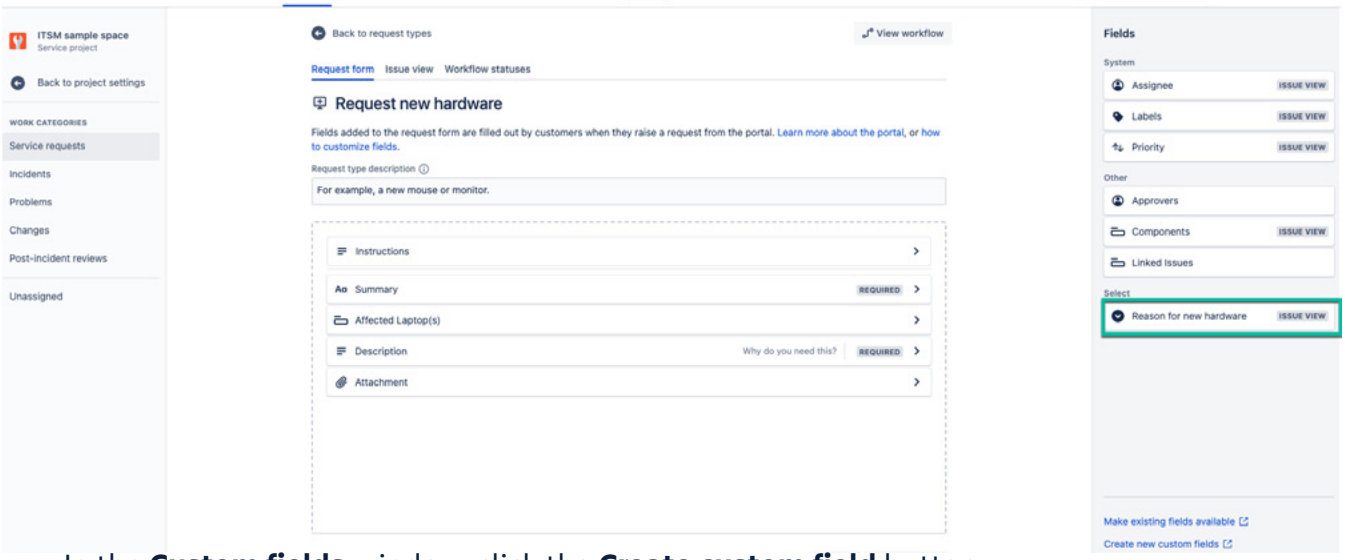

Fields
System
Assignee
ISSUE VIEW
Labels
ISSUE VIEW
Priority
ISSUE VIEW
Other
Approvers
Components
ISSUE VIEW
Linked Issues
Make existing fields available
Create new custom fields
Refresh this page after adding or creating fields.


Help Center / ITSM sample space
Request new hardware
Raise this request on behalf of*
Jennifer Fish (jfish@atlassian.com)
Summary*
Problem with my laptop
Affected Laptop(s)
Search for Assets objects
1 SEARCH RESULT
Jennifer Fish
AT-B022103
Attachment
Drag and drop files, paste screenshots, or browse
Browse
Send
Cancel

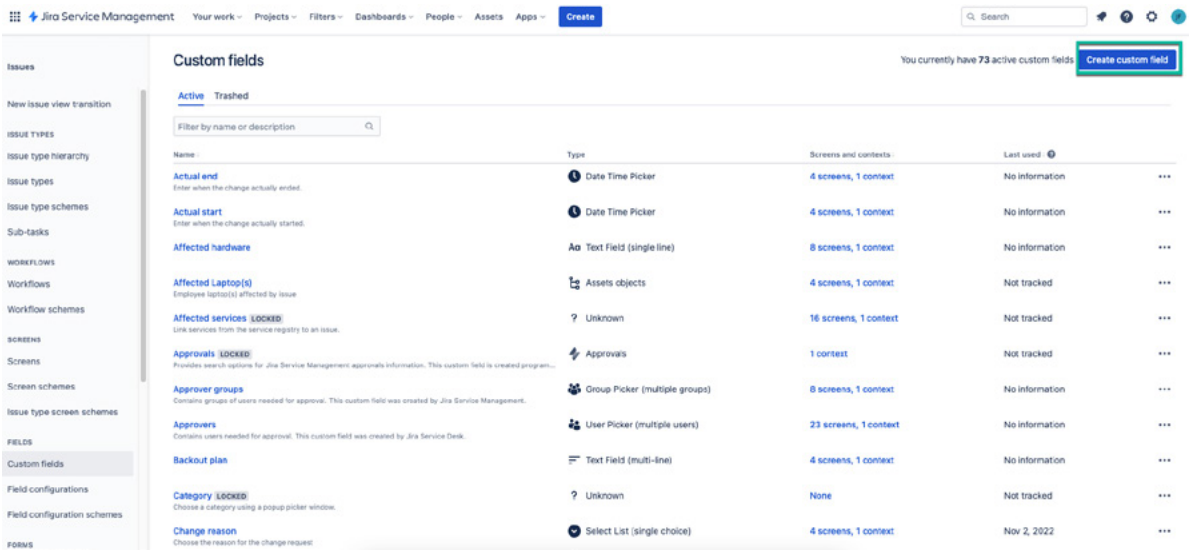
Step 15 - Create automation for updating a laptop

With the **Affected laptop(s)** field associated with the **Request new hardware** form, you can create automation to update assets when requests are submitted. In this step, let's create another custom field and define the automation steps.

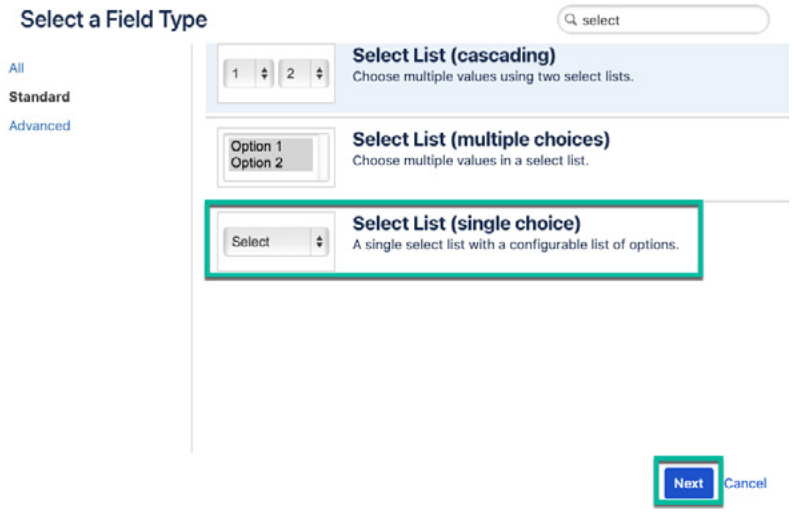
In the **Request new hardware** form window, click on the **Create new custom fields** link.



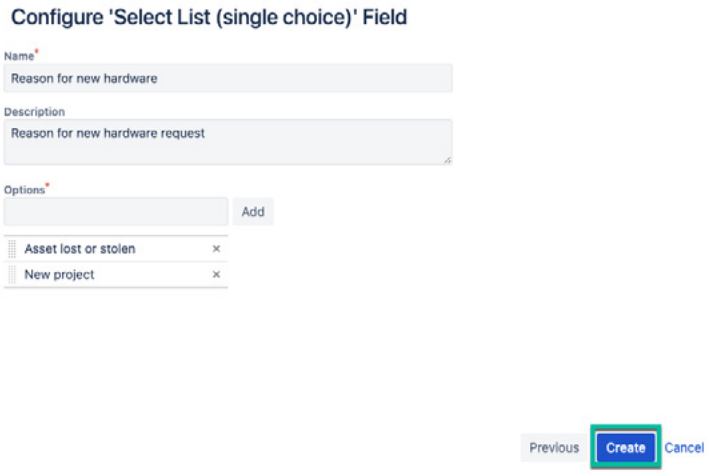
In the **Custom fields** window, click the **Create custom field** button.



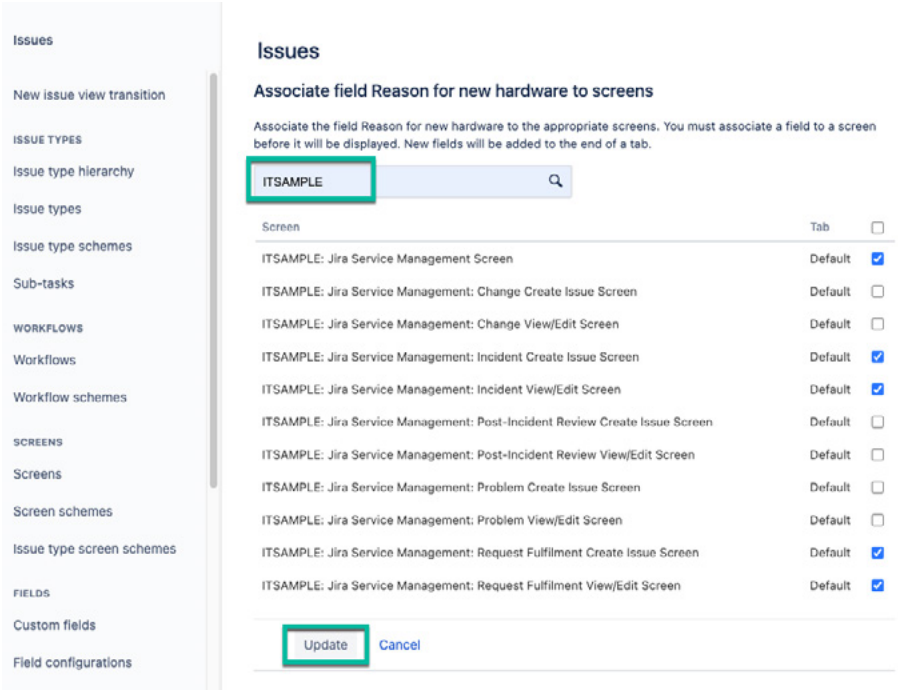
In the **Select a Field Type** window, search on "**select**", select **Select List (single choice)**, then click **Next**.



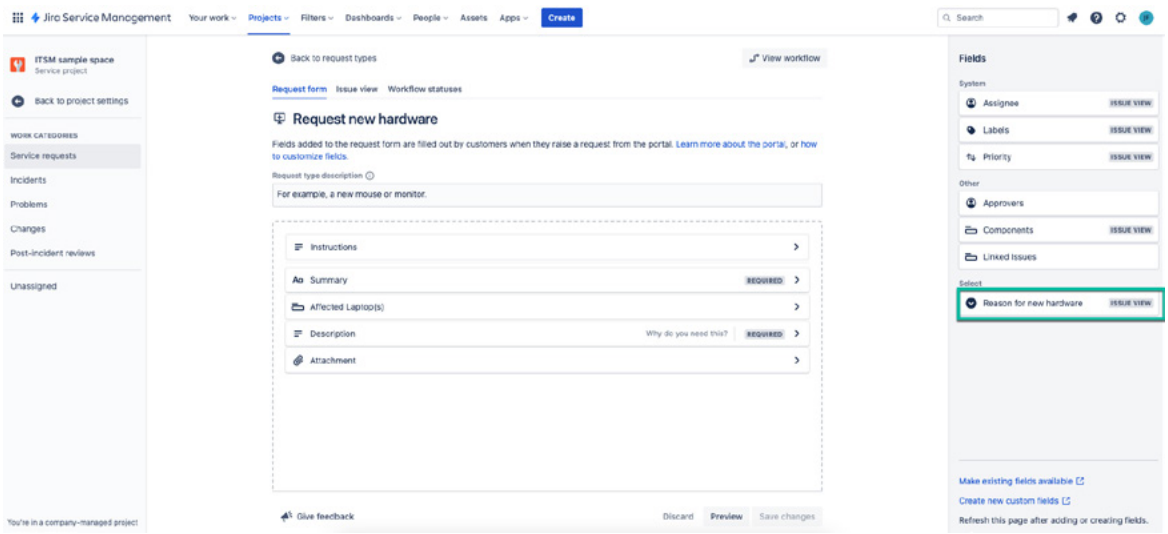
In the **Configure 'Select List (single choice)' Field** window, enter **Name** and **Options** data for the custom field, then create the new field.

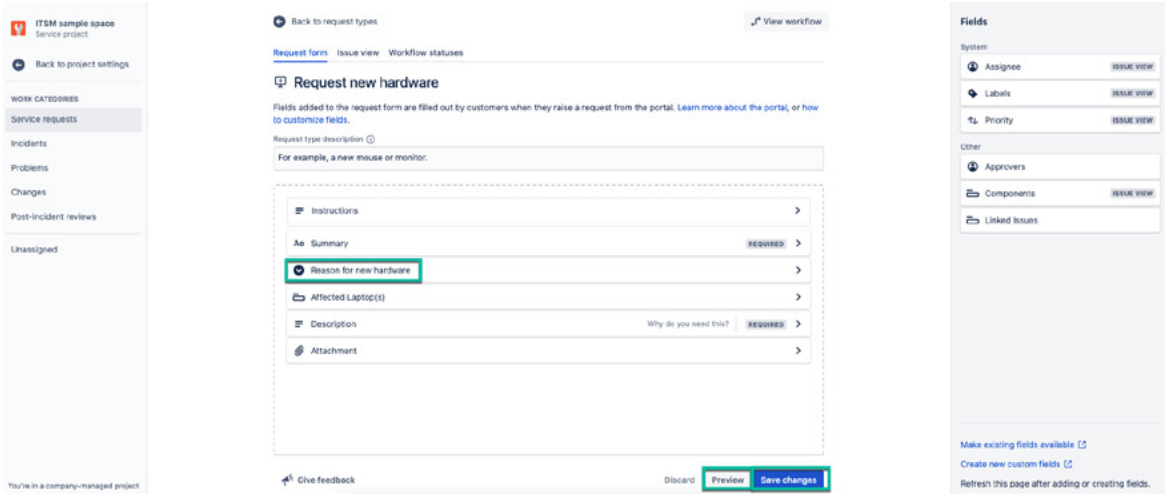


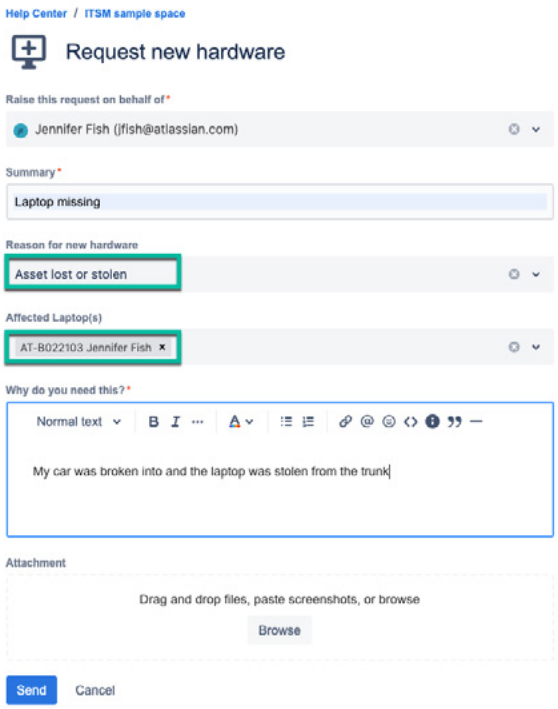
Search for your project and associate the new field to the request and incident screens in your project, then click **Update**.



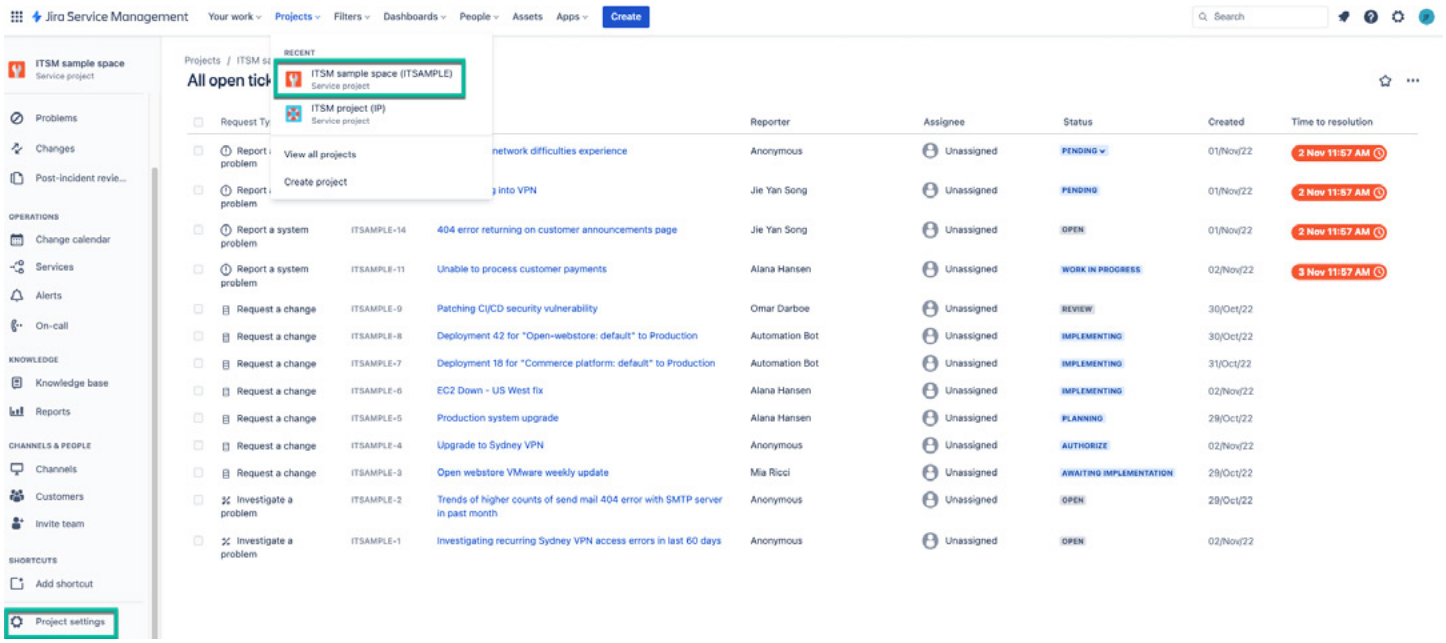
Return to the **Request new hardware** window, drag-and-drop the new field to the form, click **Save**, then click **Preview** to see the updated request form.



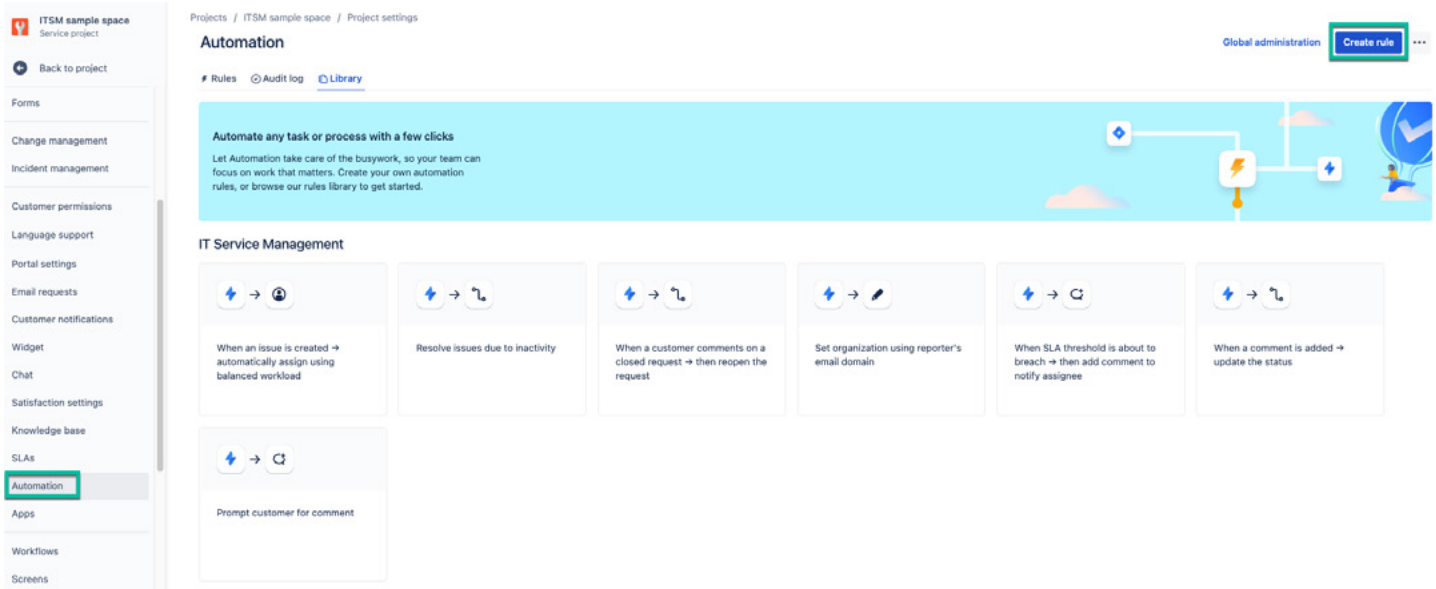

ITSM sample space
Service project
Back to project settings
WORK CATEGORIES
Service requests
Incidents
Problems
Changes
Post-incident reviews
Unassigned
You're in a company-managed project
Back to request types
View workflow
Request form
Issue view
Workflow statuses
Request new hardware
Fields added to the request form are filled out by customers when they raise a request from the portal. Learn more about the portal, or how to customize fields.
Request type description
For example, a new mouse or monitor.
Instructions
Summary
REQUIRED
Reason for new hardware
Affected Laptop(s)
Description
Why do you need this?
REQUIRED
Attachment
Give feedback
Discard
Preview
Save changes
Fields
System
Assignee
ISSUE VIEW
Labels
ISSUE VIEW
Priority
ISSUE VIEW
Other
Approvers
Components
ISSUE VIEW
Linked Issues
Make existing fields available
Create new custom fields
Refresh this page after adding or creating fields.


Help Center / ITSM sample space
Request new hardware
Raise this request on behalf of*
Jennifer Fish (jfish@atlassian.com)
Summary*
Laptop missing
Reason for new hardware
Asset lost or stolen
Affected Laptop(s)
AT-B022103 Jennifer Fish
Why do you need this?*
Normal text
My car was broken into and the laptop was stolen from the trunk
Attachment
Drag and drop files, paste screenshots, or browse
Browse
Send
Cancel

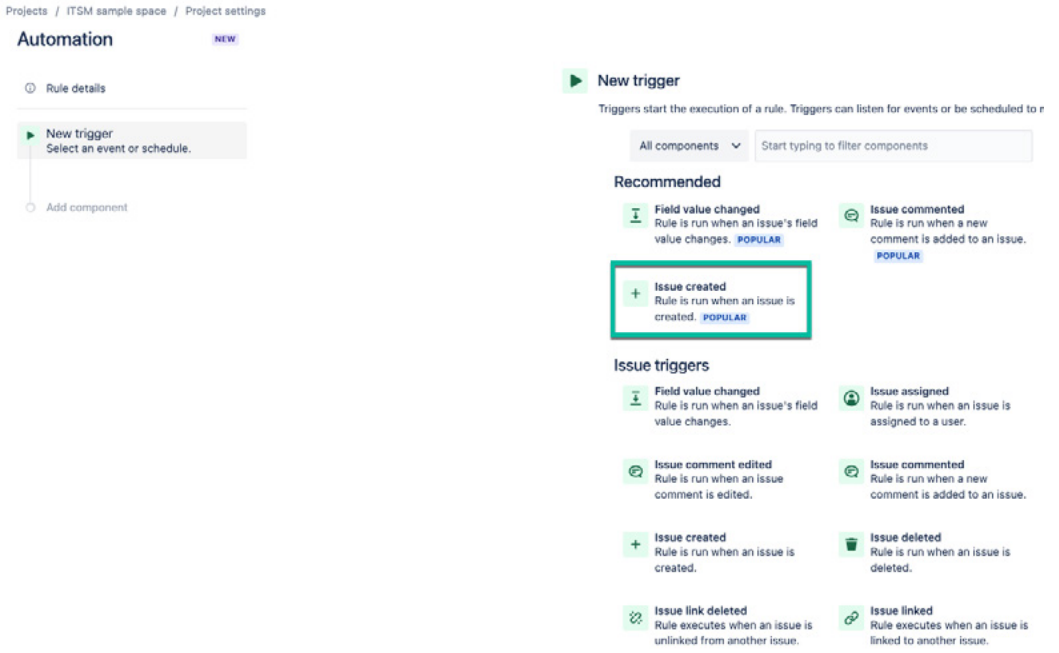
Select your project from the Jira Service Management main navigation bar then select **Project settings** in the **service project sidebar.**



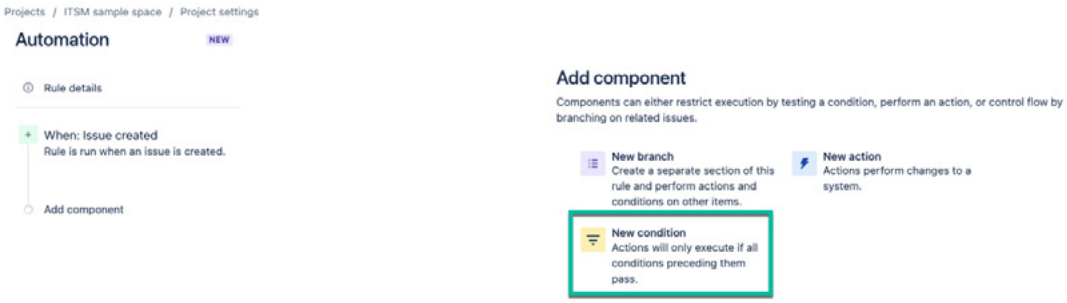
Select **Automation**, then click the **Create** rule button.



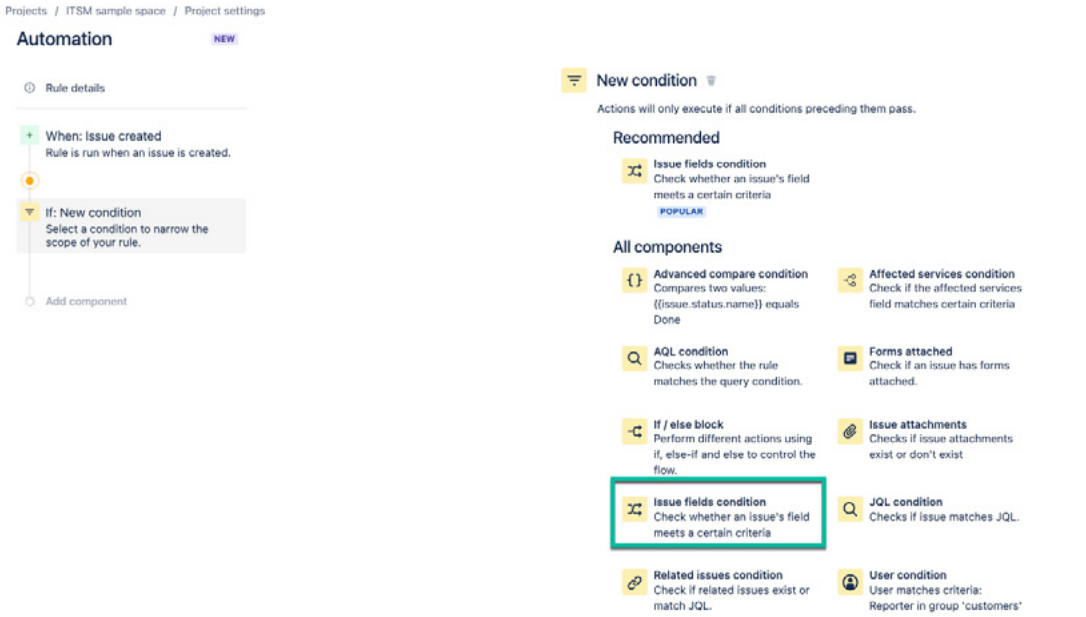
In the **New trigger** pane, select **Issue Created,** then click **Save**.



In the **Add component pane**, click **New condition**.

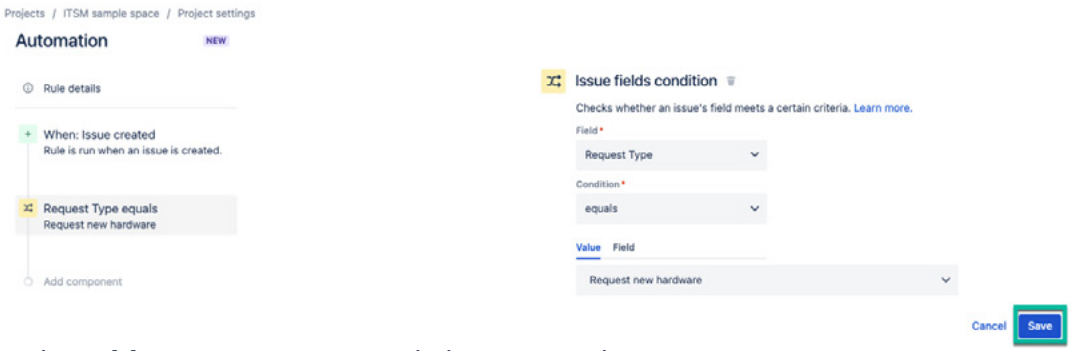


In the **New Condition pane**, select **Issue fields condition**.

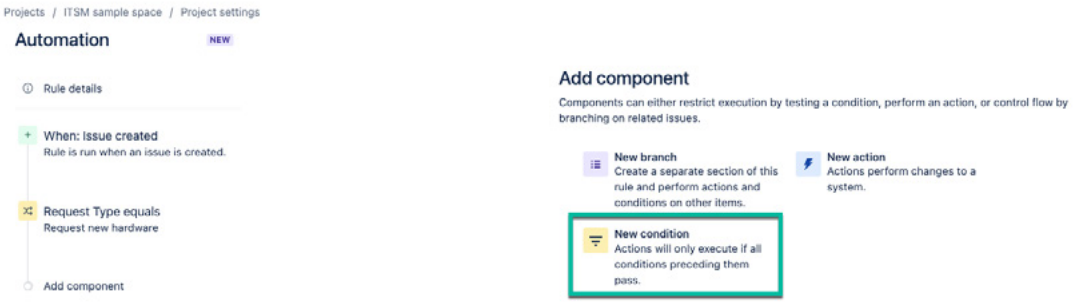


In the **Issue fields condition pane**, enter the following data then click the **Save** button.

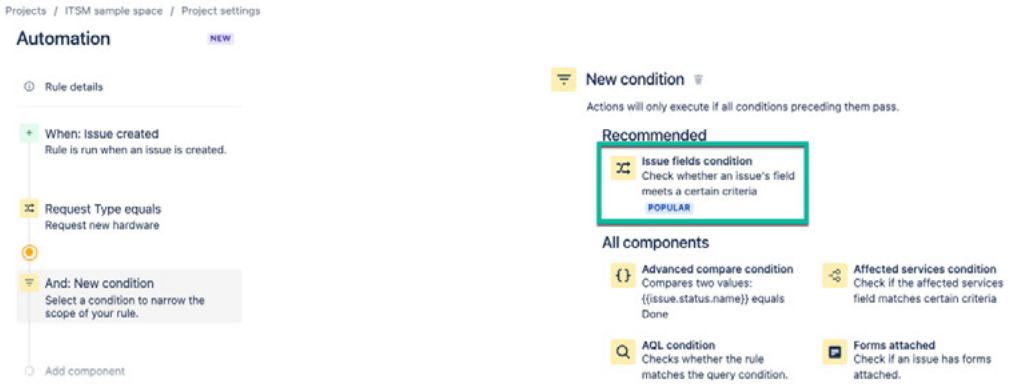
Field	Condition	Value
Request Type	equals	Request new hardware



In the **Add component pane**, click **New** condition.



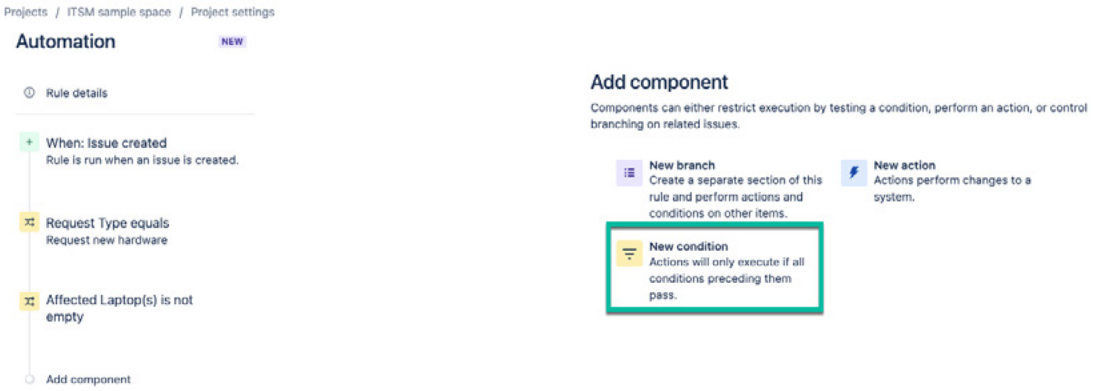
In the **New** condition pane, click **Issue fields condition**.



In the **Issue fields condition pane**, select the **custom asset field** and enter **Condition data**, then click **Save**.

Field	Condition
Affected Laptop(s)	is not empty

In the **Add component pane**, select **New condition**.



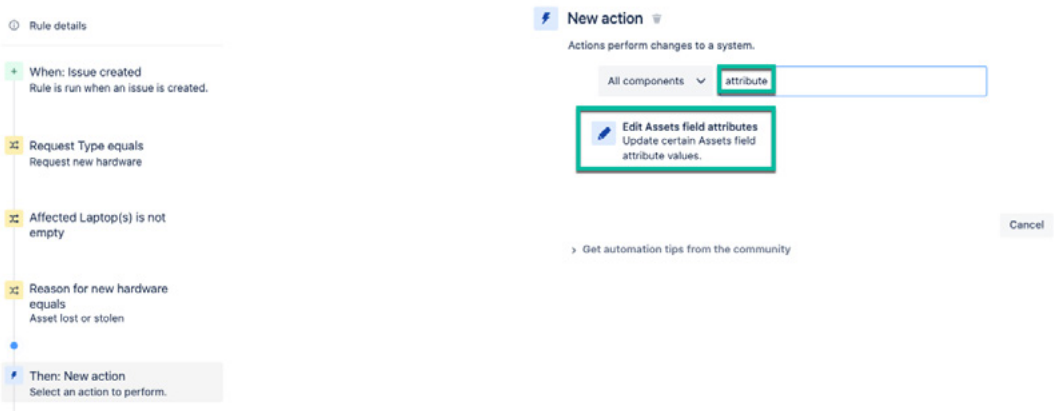
In the **Issue fields condition pane**, select the custom field and enter **Condition** data, then click **Save**.

Field	Condition	Value
Reason for new hardware	equals	Asset lost or stolen

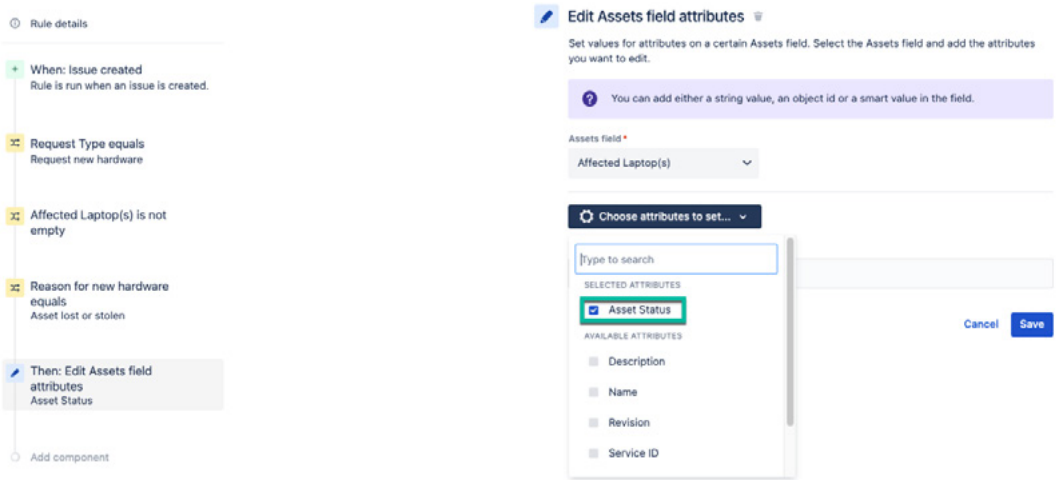
PRO TIP

Triggers are a powerful tool for keeping your Jira issues synchronized with ***Assets*** data. It is recommended that you configure triggers within individual project automation (rather than globally) and define specific conditions to ensure the automation rule works reliably and only in expected conditions.

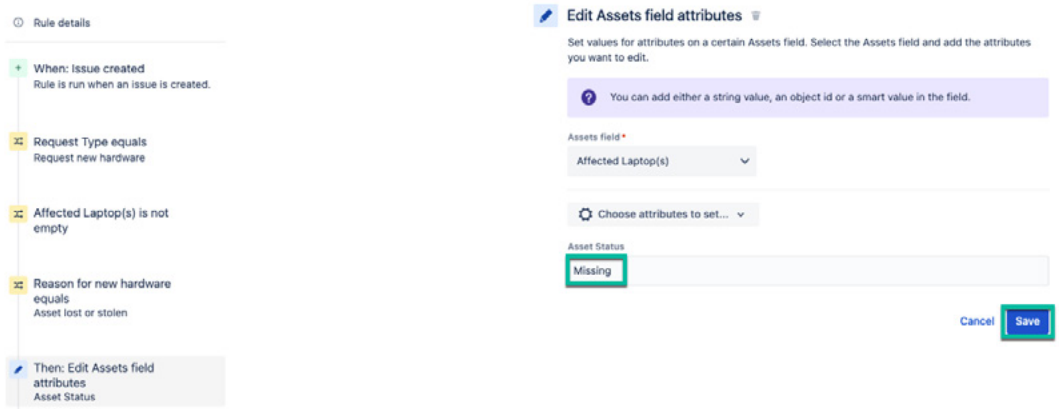
In the **Add component** pane, select **New action**, filter on "**attribute**", then select **Edit Assets field** attributes.



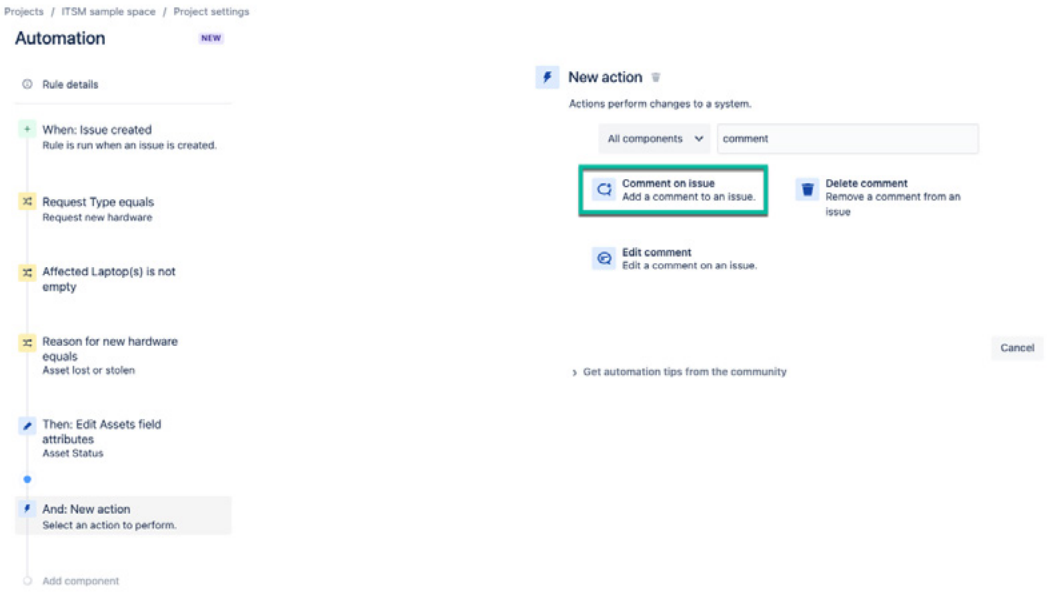
In the **Edit Assets field attributes** pane, **Affected Laptop(s)** is pre-populated as the **Asset Field** value, so click on the **Choose attribute to set** link, then select **Asset Status**.



Enter **Missing** in the **Asset Status** field and click **Save**.



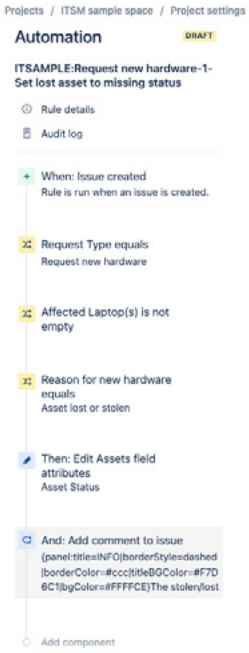
In the **New component** pane, select **New action**, filter on **"comment, "** then select **Comment on issue**.



In the **Comment on issue** pane, enter an informational message, ensure the **checkbox** for *Prevent duplicates by only adding this comment once to a particular issue* is selected, select **Share with customer** in the **Comment Visibility** field, then click **Save**.

Comment

{panel:title=INFO|borderStyle=dashed|borderColor=#ccc|titleBGColor=#F7D6C1|bgColor=#FFFFCE}The stolen/lost laptop '{{issue.fields.Affected Laptop(s)}}' was | updated in the IT Assets schema and the Asset Status is set to Missing{panel}



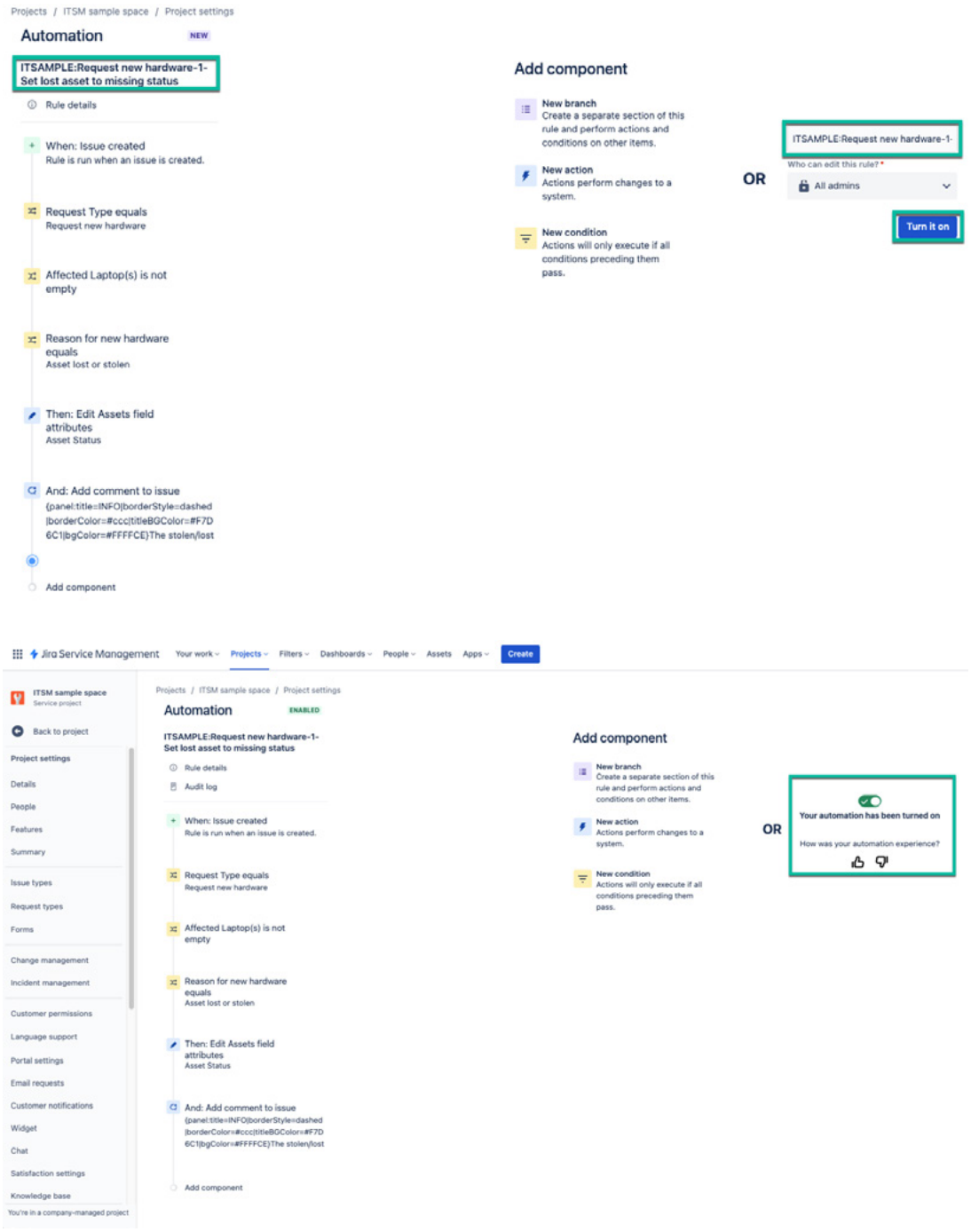
PRO TIP

You can create more detailed, informative messages by including Jira smart values. Smart values are placeholders that let you pull in dynamic data. You can use them to access and manipulate almost any issue data from Jira.

For more information, visit the following link:

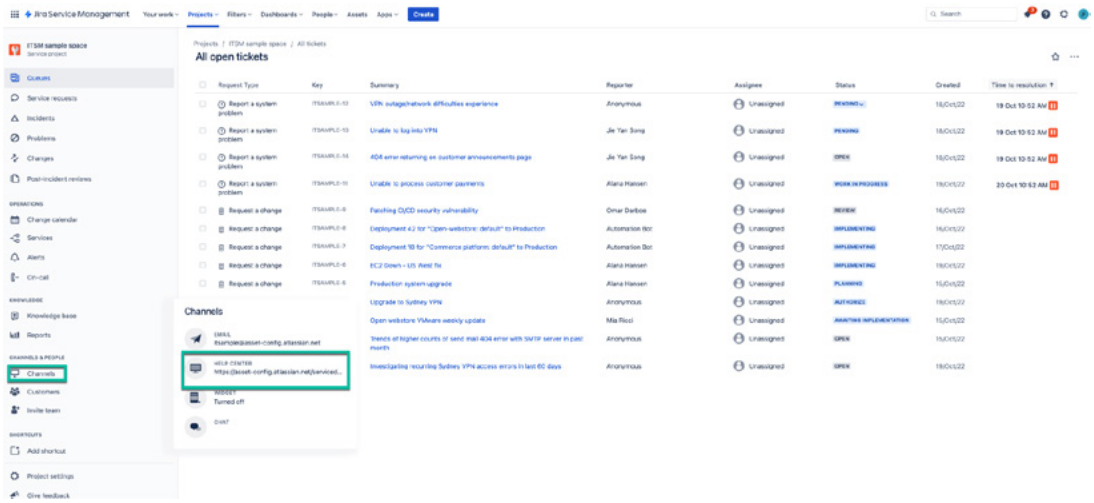
Jira smart values - issues | Cloud automation Cloud | Atlassian Support

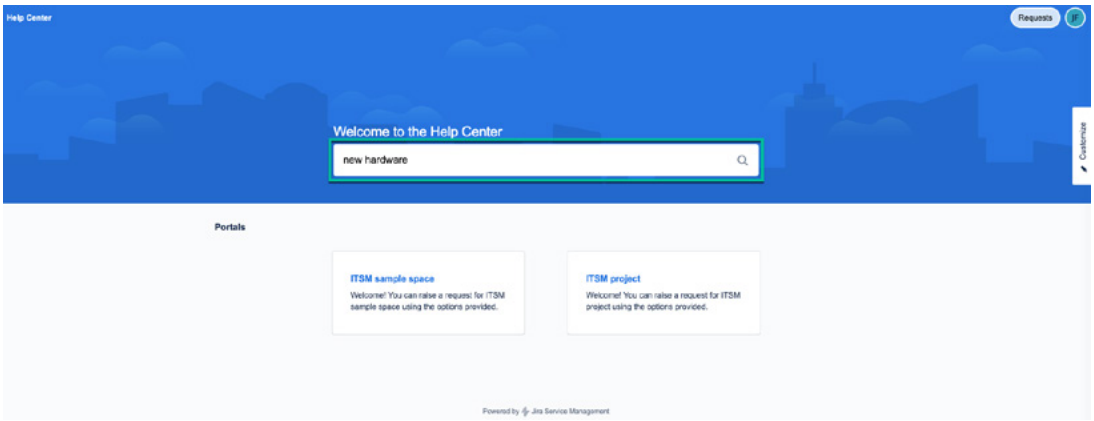
The automation rule is complete, so add a name for the automation and click **Turn it on**.

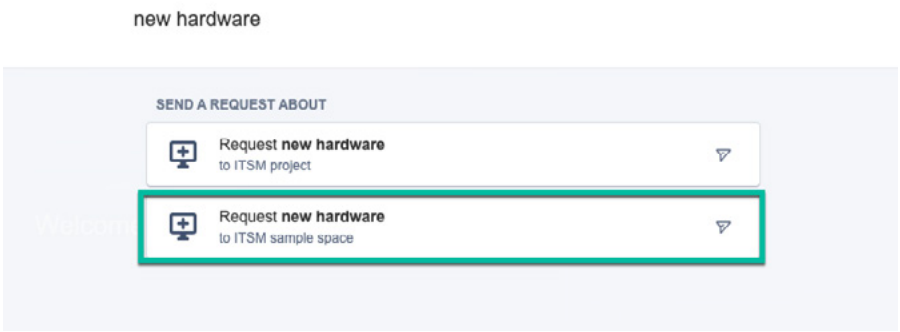

Projects / ITSM sample space / Project settings
Automation
NEW
ITSAMPLE:Request new hardware-1-
Set lost asset to missing status
Rule details
When: Issue created
Rule is run when an issue is created.
Request Type equals
Request new hardware
Affected Laptop(s) is not
empty
Reason for new hardware
equals
Asset lost or stolen
Then: Edit Assets field
attributes
Asset Status
And: Add comment to issue
{panel:title=INFO|borderStyle=dashed
|borderColor=#ccc|titleBGColor=#F7D
6C1|bgColor=#FFFFCE}The stolen/lost
Add component
Add component
New branch
Create a separate section of this
rule and perform actions and
conditions on other items.
New action
Actions perform changes to a
system.
New condition
Actions will only execute if all
conditions preceding them
pass.
OR
ITSAMPLE:Request new hardware-1-
Who can edit this rule? *
All admins
Turn it on
Jira Service Management
Your work
Projects
Filters
Dashboards
People
Assets
Apps
Create
ITSM sample space
Service project
Back to project
Project settings
Details
People
Features
Summary
Issue types
Request types
Forms
Change management
Incident management
Customer permissions
Language support
Portal settings
Email requests
Customer notifications
Widget
Chat
Satisfaction settings
Knowledge base
You're in a company-managed project
Projects / ITSM sample space / Project settings
Automation
ENABLED
ITSAMPLE:Request new hardware-1-
Set lost asset to missing status
Rule details
Audit log
When: Issue created
Rule is run when an issue is created.
Request Type equals
Request new hardware
Affected Laptop(s) is not
empty
Reason for new hardware
equals
Asset lost or stolen
Then: Edit Assets field
attributes
Asset Status
And: Add comment to issue
{panel:title=INFO|borderStyle=dashed
|borderColor=#ccc|titleBGColor=#F7D
6C1|bgColor=#FFFFCE}The stolen/lost
Add component
Add component
New branch
Create a separate section of this
rule and perform actions and
conditions on other items.
New action
Actions perform changes to a
system.
New condition
Actions will only execute if all
conditions preceding them
pass.
OR
Your automation has been turned on
How was your automation experience?

Step 16 - Create a request for an employee laptop

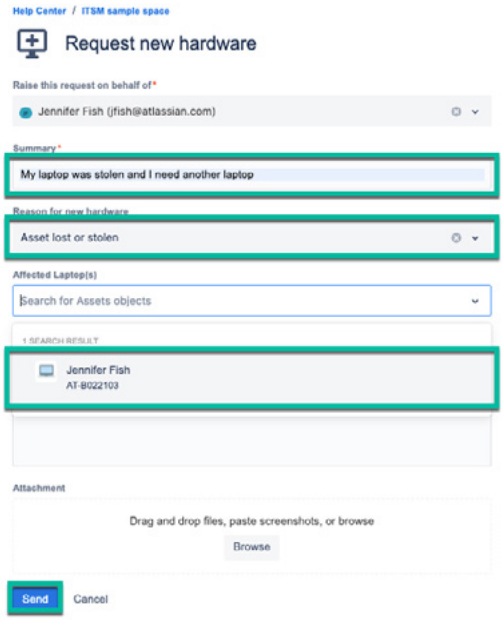
Now you can create a request for an employee and select the employee's laptop. Access your customer portal by selecting **Channels** in the **service project sidebar** and clicking **Help Center/Open**. In the Help Center, search for **"new hardware",** and select **Request new hardware** in your project.

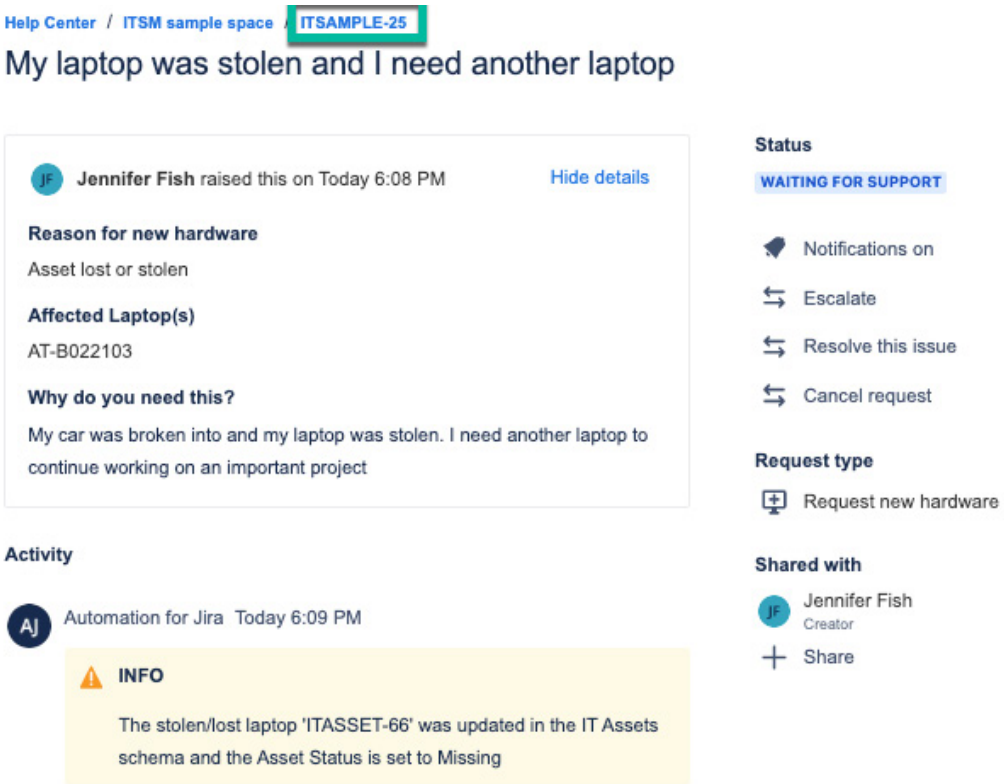




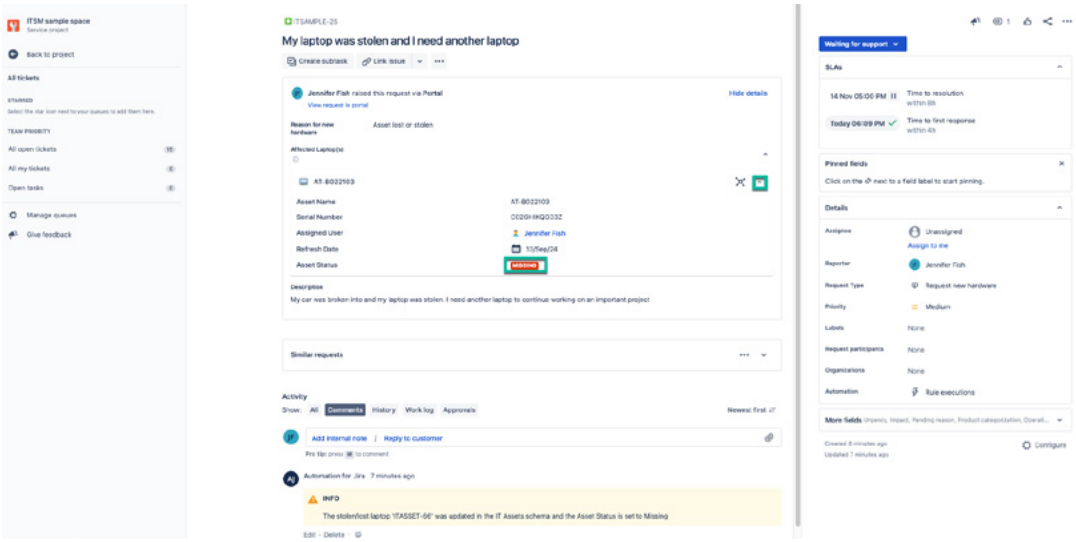


Enter information in the **Summary** field, select **“Asset lost or stolen”** in the **Reason for new hardware** field, select an **Affected Laptop(s)** value, enter information in the **Why do you need this?** field, then click the **Send** button.





You can view the issue by selecting the issue ID and see the request details, note about the asset record update, SLAs for issue response and resolution, etc. Additionally, you can click on the asset field **Show details** toggle and see that the **Asset Status** is set to **Missing**.



Appendix

Checklist of next steps

This list is designed as a high-level overview of next steps to take when embarking on your ITAM and service configuration management journey. By starting small and focused, organizations can implement a valuable asset and/or configuration system in just a few months.

Build your business case

- ☐ Pick a current business problem that will deliver value to the business if solved. We recommend starting to solve just one or two problems for the first iteration of asset and service configuration management.
- ☐ Assemble a team to tackle this problem. The team members you choose will depend heavily on the business problem.
- ☐ Consider which teams interact with the problem area and pull stakeholders from each relevant team.
- ☐ Choose a sponsor from your organization who is responsible for the outcomes you're trying to improve.
- ☐ Ensure everyone has the same understanding of IT asset and service configuration management.
- ☐ Outline the business problem in detail, how asset and/or service configuration management can help overcome it, and the business outcomes that it will lead to.
- ☐ Define your goals such as reducing mean time to resolve by 10% or increasing customer satisfaction by 15%.
- ☐ Build a business case using the problem statement and goals to get buy-in from stakeholders and budget approval.

Understand what data you need

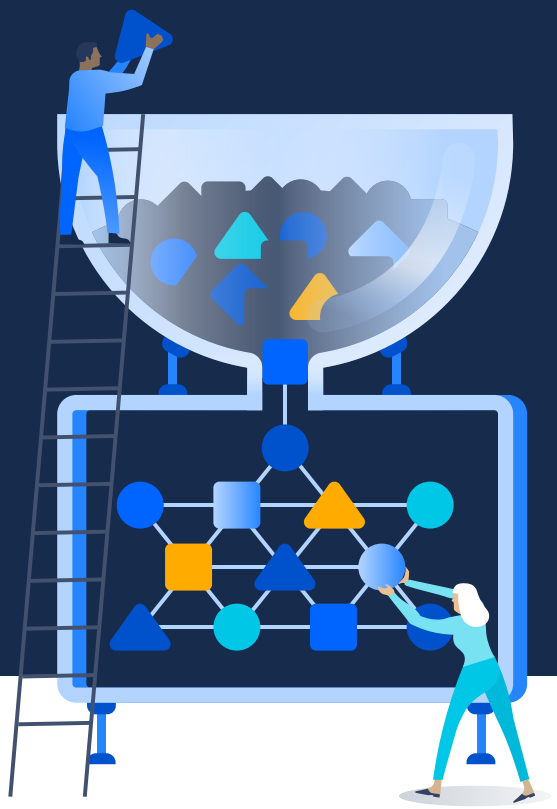
- ☐ Understand what information you need to solve your chosen problem.
- ☐ List relevant asset and CI categories (e.g. laptops, servers, databases) and what information (attributes) you need to know about each category.
- ☐ Understand where that information is located today (e.g. spreadsheets, in people's heads, external databases).
- ☐ Decide what data to leave in its current tool and what should be moved entirely into the CMDB. It's definitely time to leave those spreadsheets behind.
- ☐ Understand which integrations to third party tools or file imports you will need based on the above.
- ☐ Understand how often data is changed to inform how often integrations need to run to keep your CMDB up to date.
- ☐ Understand if any governance, compliance, or audit requirements are required.
- ☐ Do a final check. Does every piece of data have a purpose? If not, remove it.

Implement

- ☐ Carry out any relevant product training for your team members that will be building, maintaining, and interacting with your chosen tool.
- ☐ Learn about data modeling best practices. There are plenty of resources available, even in-depth YouTube lectures.
- ☐ Map out the structure you want on paper or a whiteboard. Ensure that your chosen structure can support the access permissions you require.
- ☐ Import data and organize it according to your plan. Integrate your assets and CIs with your service desk.
- ☐ Set up relevant automation rules to keep data up to date. Trial and iterate.
- ☐ Audit data periodically to keep it up to date.
- ☐ Select the next problem to solve and continue expanding.

Whether you're already in the Atlassian ecosystem or you're making a switch from legacy CMDB tools, ***Assets*** in Jira Service Management can help you modernize your asset and service configuration management practices.

To take the next steps in your modernization **reach out to your local Atlassian Solution Partner today.**



T4S partners

p: 720-379-4070 | **w:** t4spartners.com | **e:** info@T4SPartners.com
7935 E Prentice Ave., Suite 400, Greenwood Village, CO 80111

ATLASSIAN