2024

# Jira

Project Management Bootcamp

# Contents

# Jira Fundamentals

## Introduction to Jira

JIRA is a powerful project management tool developed by Atlassian. It is widely used by software development teams for bug tracking, issue tracking, and project management. JIRA supports agile methodologies like Scrum and Kanban, making it a popular choice for teams adopting agile workflows.

**Key Features of JIRA**

- Issue Tracking:
  JIRA allows users to create and manage issues, which can represent tasks, bugs, or any work item.

- Agile Support:
  Provides dedicated tools for Scrum and Kanban, including boards, backlogs, and sprints.

- Customizable Workflows:
  Teams can define workflows to match their processes, with customizable states, transitions, and rules.

- Reports and Dashboards:
  Offers real-time insights with reports like Burndown Charts, Velocity Charts, and customizable dashboards.

- Integrations:
  Integrates with other Atlassian tools (e.g., Confluence, Bitbucket) and third-party apps through the Atlassian Marketplace.

- Collaboration:
  Facilitates team collaboration with comments, mentions, and notifications.

- Permissions and Security:
  Provides granular permission controls for projects, issues, and actions.

**JIRA Editions**

- JIRA Software:
  Focused on software development with agile features.

- JIRA Core:
  A general-purpose project management tool suitable for business teams.

- JIRA Service Management:
  Designed for IT teams to manage service requests, incidents, and changes.

**Benefits of JIRA**

- ✓ Centralized Tracking: Tracks all project-related activities in one place.
- ✓ Enhanced Team Productivity: Facilitates task prioritization and streamlined workflows.
- ✓ Improved Visibility: Provides transparency into progress, bottlenecks, and team performance.
- ✓ Scalable for Any Team: Adaptable to small teams and large organizations.

**Common Use Cases**

- Software Development:
  Managing development tasks, bugs, and features in sprints.

- Project Management:
  Tracking project progress with Gantt charts and Kanban boards.

- IT Service Management:
  Handling support tickets and IT workflows.

- Business Process Management:
  Automating and managing routine business processes.

## Setting up a new Jira Instance

1. Go to https://www.atlassian.com/ and create a Jira Account
2. You should be able to get your site name like **johndoe.atlassian.net**
3. Describe work (this allows jira to suggest templates)
4. Create project (provide a project name), select project type (team-managed / company-managed) and choose access type (open. limited and private)

**Step 1 - Create a project**

Log into your Jira site. In the top navigation, select the “Projects” dropdown and select “Create project”.

**Step 2 - Pick a template**

There are dozens of Jira templates, each of which is designed to get your team started quickly and successfully. Today, there are three templates specifically for software teams:

- **Scrum**
  For agile teams that work from a backlog, plan and estimate their work in sprints, and deliver work on a regular schedule.

- **Kanban**
  For agile teams that monitor work in a continuous flow (rather than in sprints), with a focus on managing in-progress work. (Includes the option of a kanban backlog.)

- **Bug tracking**
  For teams that don't need boards and prefer to manage development tasks and bugs in a list view.

**Project Types**

For the scrum and kanban templates only, you will also be prompted to choose a project type. The fundamental difference between the two project types is how they are administered, and whether that occurs at the team level or at a company/Jira admin level.

- Team-managed projects are suited for independent teams who want to control their own working processes and practices in a self-contained space.
- Company-managed projects are set up and maintained by Jira admins. This project type is designed for teams who want to standardize a way of working across many teams, such as sharing a workflow.

**Step 3 - Set up your columns**

In Jira, the board displays a selection of issues in columns, with each column representing a step in your team's workflow for taking work through completion. Although there are many things you can configure on your board, we suggest just setting up columns for now. When you're getting started on a new Jira project, it's important to make your board reflect the way your team works.

<u>In team-managed projects:</u>

a. Go to your board. Select (•••) in the top right and click Configure board.
b. Add a new column, change the name of column, delete a column, or a move a column as necessary.

<u>In company-managed projects:</u>

a. Go to your board. Select (•••) in the top right  and click Board settings.
b. Click the Columns tab.
c. Add a new column, change the name of column, delete a column, or a move a column as necessary.

**Step 4 - Create an issue**

Issues are the building blocks of your Jira project. An issue can represent a story, epic, bug, feature to be built, or any other task in your project.

Select "Create" in the top navigation. Your issue will appear in the backlog or board of the project.

**Step 5 - Connect your tools**

You and your teams can spend less time managing work and more time building great software with over 3,000 Jira apps that can be custom-tailored to fit any and every use case. Build a unified workspace with 3000+ apps and integrations from the Atlassian Marketplace.

a. Select the Cog in the right corner of the top navigation > Apps.
b. Click Find new apps.
c. Search by the app name, or choose a category.
d. Follow the prompts to Install, Buy now, or start a Free trial.

**Step 6 - Invite your team**
It’s time to get the party started! Once you have enough work represented on your board, start inviting team members.

**Step 7 - Move work forward**
Now that your team has joined your Jira site, you're ready to collaborate and track work together. If you're in a scrum project, you'll need to create and start a sprint to begin tracking work. If you're in a kanban project, you can start tracking work on the board. To track work items, move an issue from one column to another as it progresses through your team's workflow.

## User management - creating groups

1. Access Admin Panel
   Once logged in, go to the Admin Panel by clicking on the gear icon in the top-right corner.

2. Navigate to User Management:
   Go to Admin Panel > User Management.

3. Create a New Group:
   a. Click on the Groups tab and select Create Group.
   b. Enter a name for the group (e.g., "Developers" or "QA Team").

4. Add Members to the Group:
   a. Open the group you just created.
   b. Click Add Members and search for users by their email or name.

5. Assign Permissions to Groups:
   In Permission Schemes, assign specific roles or permissions to the group.

## Projects and categories

1. Create a New Project:
   a. Go to Projects > Create Project.
   b. Select a template (Scrum, Kanban, or custom).
   c. Configure the project name, key, and type (Team-managed or Company-managed).

2. Organize with Categories:
   a. Go to Admin Panel > Projects > Project Categories.
   b. Click on Add Category, name it (e.g., "Web Development"), and assign projects to this category.

## Setting a project as an admin

1. Access Project Settings:
    a. Go to the project and click Project Settings in the left menu.

2. Add Project Admins:
    a. Navigate to People.
    b. Add users and assign them the Administrator role for the project.

## Changing project settings

1. Modify Project Details:
    a. Go to Project Settings > Details.
    b. Edit the project name, URL, or description as needed.

2. Customize Workflows:
   Go to Workflows and modify states, transitions, and rules.

3. Enable/Disable Features:
   Toggle features like Backlogs, Components, and Releases in the Features section.

## Managing components and versions

1. Create Components:
    a. Go to Project Settings > Components.
    b. Click Add Component, provide a name, and optionally assign a default assignee.

2. Create Versions:
    a. Go to Project Settings > Versions.
    b. Click Create Version, name it (e.g., "v1.0"), and add a description or release date.

3. Use in Issues:
   Assign components and versions while creating or editing issues.

## Managing project permissions

1. Access Permission Schemes:
   Go to Admin Panel > Issues > Permission Schemes.

2. Modify Permission Scheme:
    a. Select the scheme assigned to your project.
    b. Edit permissions like Browse Projects, Create Issues, and Edit Issues.

3. Assign Roles or Groups:
   Map permissions to specific roles or groups (e.g., Developers can Edit Issues).

4. Apply to Project:
   Ensure the scheme is associated with the desired project under Project Settings > Permissions.

# Jira Customization and Advanced Settings

## A few words about customization

JIRA allows customization to fit unique business processes and workflows. Some common customization areas include:

- Workflows: Modify issue states and transitions.
- Custom Fields: Add specific data fields relevant to your projects.
- Screens: Customize what users see when creating, viewing, or editing issues.
- Triggers and Automations: Automate transitions or actions based on conditions.
- Permissions and Notifications: Tailor who can access what and receive updates.

## Creating a custom workflow

1. Access Workflows:
   Go to Admin Panel > Issues > Workflows.

2. Create a Workflow:
   a. Click Add Workflow.
   b. Name the workflow and add a description.

3. Design the Workflow:
   a. Use the Diagram Mode to define:
   b. Statuses: E.g., "To Do," "In Progress," "Done."
   c. Transitions: Link statuses using transitions (e.g., "Start Progress" connects "To Do" to "In Progress").

4. Save and Publish:
   Save the workflow and associate it with a workflow scheme.

5. Assign Workflow to Project:
   Apply the workflow scheme to the desired project under Project Settings > Workflows.

## Custom workflow: triggers, conditions and post-functions

1. Edit Workflow:
   Go to the workflow and select Edit Workflow > Diagram Mode.

2. Add a Trigger:
   a. Select a transition.
   b. Click Add Trigger.

   c. Choose a trigger (e.g., automatically transition an issue when a linked branch is created in Bitbucket).

3. Add Conditions:
   a. Conditions restrict who can execute transitions.
   b. Select a transition > Add Condition.
   c. Example: Only users in the "Developers" group can move issues to "In Progress."

4. Add Post-Functions:
   a. Post-functions execute actions after a transition.
   b. Click Add Post-Function.
   c. Example: Automatically assign an issue to a specific user after transitioning.

## Creating and managing custom fields

1. Access Custom Fields:
   Go to Admin Panel > Issues > Custom Fields.

2. Create a Custom Field:
   a. Click Create Custom Field.
   b. Choose a field type (e.g., Text Field, Date Picker, Dropdown).
   c. Provide a name and description.

3. Add Field to Screens:
   After creating, associate the field with specific screens.

4. Edit or Delete Fields:
   Manage existing fields by editing their properties or removing them from screens.

## Creating custom screens

1. Access Screens:
   Go to Admin Panel > Issues > Screens.

2. Create a Screen:
   a. Click Add Screen.
   b. Name the screen (e.g., "Bug Creation Screen").

3. Add Fields to the Screen:
   a. Select the screen.
   b. Click Add Field to include existing custom or system fields.

4. Associate Screen with Operations:
   a. Go to Screen Schemes.
   b. Map the custom screen to operations like Create Issue, Edit Issue, or View Issue.

5. Apply to Project:
   Assign the screen scheme to a project under Project Settings > Screens.

# Using Jira Query Language

## JQL basic functions and operators

Basic Syntax: JQL is structured as:

```
<field> <operator> <value>
```

Operators:

| | | |
|---|---|---|
| = | : | Exact match |
| != | : | Does not match |
| ~ | : | Contains |
| !~ | : | Does not contain |
| IN | : | One of multiple values |
| NOT IN | : | None of the values |
| IS | : | Field is empty or not empty |
| AND/OR | : | Combine conditions |
| ORDER BY | : | Sort results |
| > / < / >= / <= | : | Compare values |

Examples:

(Find all issues in the "Marketing" project.)

```
project = "Marketing"
```

(Find all issues not marked as "Done.")

```
status != "Done"
```

(Find issues with "bug" in the summary.)

```
summary ~ "bug"
```

(Find issues that are assigned.)

```
assignee IS NOT EMPTY
```

(Find issues with high or critical priority.)

```
priority IN ("High", "Critical")
```

(Find issues without "UI" or "UX" labels.)

```
labels NOT IN ("UI", "UX")
```

(Find issues due after the start of the month.)

```
duedate >= startOfMonth()
```

(Find issues created after January 1, 2024.)

```
created >= "2024-01-01"
```

(Sort bugs by priority.)

```
type = Bug ORDER BY priority DESC
```

(Find issues assigned to the current user and not closed.)

```
status != Closed AND assignee = currentUser()
```

## Advanced JQL: queries related to people

Fields: **assignee, reporter, watcher**

Examples:
**assignee = currentUser()**
(Issues assigned to you.)

**reporter = "john.doe"**
(Issues reported by "John Doe.")

**watchers = "jane.smith"**
(Issues watched by "Jane Smith.")

**assignee IN (membersOf("Developers"))**
(Issues assigned to developers group.)

**reporter != currentUser()**
(Issues not reported by you.)

**assignee IS EMPTY**
(Unassigned issues.)

**assignee WAS "john.doe" ON "2024-11-01"**
(Issues assigned to John Doe on a specific date.)

**assignee WAS IN (membersOf("QA")) DURING ("2024-01-01", "2024-01-31")**
(Issues assigned to QA group during January 2024.)

**voters > 5**
(Issues with more than 5 votes.)

**creator = currentUser()**
(Issues created by you.)

## Advanced JQL: queries related to statuses

Fields: **status**

Examples:
**status = "In Progress"**
(Find issues in progress.)

**status != "To Do"**
(Issues not in "To Do" status.)

**status CHANGED FROM "Open" TO "In Progress"**
(Issues moved from Open to In Progress.)

**status CHANGED AFTER "2024-01-01"**
(Status changed after January 1, 2024.)

**status WAS "Done"**
(Issues that were previously Done.)

**status WAS NOT "Blocked"**
(Issues never marked as Blocked.)

**status CHANGED DURING ("2024-01-01", "2024-01-15")**
(Status changed during a specific date range.)

**status IN ("Closed", "Resolved")**
(Issues that are either Closed or Resolved.)

**status NOT IN ("Open", "In Progress")**
(Exclude Open or In Progress issues.)

**statusCategory = "Done"**
(Issues in the Done category.)

## Advanced JQL: queries related to links

Fields: **issueLinkType**

Examples:
**issueLinkType = "blocks"**
(Find issues that block others.)

**issueLinkType = "is blocked by"**
(Find issues blocked by others.)

**linkedIssues = "JIRA-123"**
(Find issues linked to "JIRA-123.")

**issueLinkType != "duplicates"**
(Exclude issues with duplicate links.)

**linkedIssues IN ("JIRA-123", "JIRA-456")**
(Find issues linked to JIRA-123 or JIRA-456.)

**issueLinkType WAS "relates to"**
(Previously linked as "relates to.")

**issueLinkType IN ("clones", "duplicates")**
(Find cloned or duplicated issues.)

**linkedIssues IS NOT EMPTY**
(Issues with any links.)

**linkedIssues IS EMPTY**
(Issues with no links.)

**linkedIssues = currentUser()**
(Issues linked to those reported by the current user.)

## Advanced JQL: queries related to time

Examples:
**created >= startOfDay()**
(Issues created today.)

**updated <= endOfWeek()**
(Issues updated before the week ends.)

**due < now()**
(Overdue issues.)

**created > -7d**
(Issues created in the last 7 days.)

**resolved >= "2024-10-01"**
(Resolved after October 1, 2024.)

**created <= "2023-12-31"**
(Created before the end of 2023.)

**updated DURING ("2024-01-01", "2024-01-31")**
(Updated during January 2024.)

**resolved IS NOT EMPTY**
(Resolved issues.)

**created >= startOfMonth(-1)**
(Created during the previous month.)

**due = now()+7d**
(Due 7 days from today.)

## Advanced JQL: multilevel queries

Examples:
**(project = "HR" AND status = "Open") OR assignee = currentUser()**
(HR issues or issues assigned to you.)

**type = Bug AND (priority = "High" OR severity = "Critical")**
(Critical or high-priority bugs.)

**(reporter = "john.doe" AND status = "Blocked") OR labels IN ("urgent")**
(Blocked issues by John Doe or labeled as urgent.)

**(duedate < now() AND status != Done) AND assignee IN (membersOf("QA"))**
(Overdue QA tasks.)

**project IN ("Finance", "IT") AND created >= startOfYear()**
(Issues from Finance or IT since the start of the year.)

**status IN ("To Do", "In Progress") AND resolution IS EMPTY**
(Unresolved issues in To Do or In Progress.)

**labels NOT IN ("low-priority") AND type IN ("Task", "Bug")**
(Tasks or bugs without low-priority label.)

**status WAS IN ("Open", "Reopened") DURING ("2024-01-01", "2024-01-15")**
(Previously Open/Reopened during a date range.)

**(type = Story OR priority = Critical) AND updated > -30d**
(Recent updates for stories or critical issues.)

**fixVersion = "v1.0" AND (statusCategory != Done OR issueLinkType = "blocks")**
(Issues for version 1.0 not Done or blocking others.)

## Advanced JQL: enhanced search

Requires plugins like **ScriptRunner** or **Enhanced Search**.

Examples:
**text ~ "critical bug"**
(Search issues mentioning "critical bug.")

**lastCommentBy = "john.doe"**
(Issues where John Doe made the last comment.)

**comment ~ "needs review"**
(Issues with comments mentioning "needs review.")

**attachments IS NOT EMPTY**
(Issues with attachments.)

**SLA breached = true**
(SLA breached issues in JIRA Service Management.)

**parentIssue IN ("JIRA-123")**
(Subtasks of a specific issue.)

**worklogAuthor = currentUser()**
(Issues with your work logs.)

**commentedBy = currentUser()**
(Issues you commented on.)

**worklogDate >= startOfWeek()**
(Issues worked on this week.)

**timeSpent > 5h**
(Issues with more than 5 hours logged.)

## Getting JQL filters ready for work

1. Create a Filter:
    a. Run your JQL query.
    b. Click Save As and give it a name.

2. Share Filters:
   Set permissions under Filter Details to share with your team.

3. Use Filters in Boards:
   Configure your board to use the saved filter.

4. Subscribe to Filters:
   Automate email notifications for filter results.

5. Example Filters:
   a. "Open bugs assigned to me."
   b. "Unresolved high-priority tasks."
   c. "Issues updated in the last week."

# Team Management and Administration

## Everything you need to know about a Scrum board

A Scrum board is used to visualize and manage work in Scrum methodology, focusing on sprints.

Key Components:

- Backlog: Holds the product backlog items.
- Active Sprint: Displays tasks for the current sprint.
- Columns: Reflect the workflow (e.g., "To Do," "In Progress," "Done").

Steps to Set Up a Scrum Board:

1. Create a Project:
   a. Go to Projects > Create Project.
   b. Choose a Scrum template.
   c. Provide a name and key.

2. Customize the Board:
   a. Go to Board Settings > Columns.
   b. Map workflow statuses to columns.
   c. Example: Add a "Review" column between "In Progress" and "Done."

3. Plan a Sprint:
   a. Navigate to Backlog.
   b. Drag issues from the backlog into the sprint.
   c. Click Start Sprint.

4. Track Progress:
   Use the Burndown Chart under Reports to monitor sprint progress.

## Everything you need to know about a Kanban board

A Kanban board is ideal for continuous work with no fixed iterations.

Key Components:

- Columns: Represent workflow stages.
- WIP Limits: Control the maximum number of tasks in a column.

Steps to Set Up a Kanban Board:

1. Create a Project:
    a. Go to Projects > Create Project.
    b. Choose a Kanban template.
    c. Provide a name and key.

2. Customize the Board:
    a. Go to Board Settings > Columns.
    b. Add/remove columns to reflect your workflow.
    c. Example: Add columns like "Backlog," "To Do," "In Progress," "Testing," "Done."

3. Set WIP Limits:
    a. In Column Settings, set limits to prevent overloading.
    b. Example: Allow a maximum of 3 tasks in "In Progress."

4. Monitor Flow:
   Use the Cumulative Flow Diagram under Reports to visualize work stages.

## Setting up a dashboard

Dashboards in JIRA provide an overview of project status through widgets.

Steps to Create a Dashboard:

1. Create a Dashboard:
    a. Go to Dashboards > Create Dashboard.
    b. Provide a name and description.

2. Add Gadgets:
    a. Click Add Gadget and select from options like Pie Chart, Filter Results, or Sprint Burndown.
    b. Example: Add a Two-Dimensional Filter to view tasks by status and assignee.

3. Customize Layout:
   Adjust the grid layout to organize gadgets.

4. Share the Dashboard:
   Click Share Dashboard to set permissions for teams or groups.

## Roadmaps

Roadmaps help plan and visualize long-term goals.

Steps to Use Roadmaps:

1. Enable Roadmaps:
   Go to Project Settings > Features and enable Roadmaps.

2. Create Epics:
   a. In the Roadmap view, click + Create Epic.
   b. Add details like epic name, description, and timeline.

3. Link Issues to Epics:
   Open an issue, click More > Link Issue, and select the epic.

4. Adjust Timeline:
   Drag and drop epics to adjust timelines.

5. Track Progress:
   View progress bars for epics based on completed tasks.

## Reports and charts

JIRA offers various reports to track project and sprint progress.

Key Reports:

- Burndown Chart: Tracks sprint progress.
- Example: Monitor how many tasks are completed daily.
- Velocity Chart: Shows completed story points per sprint.
- Cumulative Flow Diagram: Tracks task status over time.
- Control Chart: Analyzes cycle time.
- Pie Chart: Visualizes tasks by priority or status.

Steps to Access Reports:

1. Go to Reports in your project.
2. Select a report type.
3. Example: Choose Velocity Chart to compare completed work across sprints.

## Working with story points

Story points measure the effort required to complete a task.

Steps to Use Story Points:

1. Enable Story Points:
   a. Go to Project Settings > Fields.
   b. Add the Story Points field if not visible.

2. Assign Story Points:
    a. Edit an issue and set its story points.
    b. Example: Assign 3 points for a medium-complexity task.

3. Estimate Sprint Workload:
   In Backlog, sum up story points for tasks in the sprint.

4. Track Progress:
   Use Burndown Charts to monitor story point completion.

## Case study: setting up Jira for an HR team

Scenario: The HR team wants to track tasks like recruitment, onboarding, and employee engagement.

Steps:

1. Create a Project:
    a. Use the Kanban template.
    b. Name it HR Tasks.

2. Customize Workflow:
   Add statuses like "New Task," "In Review," "Approved," "Completed."

3. Add Custom Fields:
   Example: Add fields like Candidate Name, Position, or Joining Date.

4. Create Task Types:
   Define issue types like Recruitment, Onboarding, Policy Update.

5. Set Up Boards:
   Customize columns: Backlog, Under Review, Approved, Completed.

6. Use Filters:
    a. Create a filter: type = Recruitment AND status != Done.
    b. Add it to a dashboard gadget.

7. Reports for HR:
    a. Use Pie Chart to display tasks by status.
    b. Use Two-Dimensional Filter for issues by assignee and priority.

8. Automations:
   Example: Automatically assign tasks labeled "Urgent" to the HR manager.

# Jira Advanced

## Jira automations

Automation in JIRA lets you define rules to perform actions automatically.

Steps to Create an Automation Rule:

1. Access Automation Settings:
   Go to Project Settings > Automation.

2. Create a New Rule:
   a. Click Create Rule.
   b. Choose a Trigger (e.g., "When an issue is transitioned").

3. Define Actions:
   a. Add an Action (e.g., "Assign issue to a specific user").
   b. Example: Assign tasks labeled "Critical" to the team lead when their status changes to "In Progress."

4. Add Conditions:
   a. Add conditions to limit actions.
   b. Example: Only apply automation if the issue priority is "High."

5. Publish the Rule:
   Test the rule and click Publish.

Examples:

Auto-assign issues to the reporter:

**Trigger: Issue created > Action: Assign to reporter.**

Send an email for overdue tasks:

**Trigger: Issue due date breached > Action: Send email.**

Update labels when status changes:

**Trigger: Status = Done > Action: Add label "Completed".**

Create sub-tasks for a specific issue type:

**Trigger: Issue created (Type = Bug) > Action: Create sub-task.**

Re-open issues if a comment is added:

**Trigger: Comment added > Action: Transition to "Reopened".**

## Confluence integration

Confluence is a knowledge management tool by Atlassian. Integrating it with JIRA enhances documentation and collaboration.

Steps to Integrate Confluence with JIRA:

1. Set Up Application Links:
    a. In JIRA, go to Settings > Applications > Application Links.
    b. Enter your Confluence URL and click Create New Link.
    c. Authenticate with admin credentials for both applications.

2. Embed JIRA Issues in Confluence:
    a. Open a Confluence page.
    b. Click Insert > JIRA Issue/Filter.
    c. Use JQL to display relevant issues.

3. Link Confluence Pages to JIRA Issues:
    a. In a JIRA issue, click More > Link > Confluence Page.
    b. Select or create a Confluence page to link.

Examples:

- Embed a sprint progress report in Confluence using the JQL filter status != Done AND sprint = currentSprint().
- Create a Confluence page for "Bug Fixes" and link relevant JIRA issues.
- Generate project documentation in Confluence and reference specific JIRA tickets.

## Superadmin panel

The Superadmin Panel is the central location for managing global settings and permissions in JIRA.

Key Functions:

- Global Permissions: Define who can create projects, manage dashboards, etc.
- User Management: Add/remove users, assign roles.
- Project Permissions: Control access at the project level.

Steps to Access Superadmin Panel:

1. Go to Administration:
   Click on the gear icon (⚙) > System.

2. Manage Global Permissions:
    a. Navigate to Global Permissions.
    b. Example: Grant "Browse Users" permission to a specific group.

3. Configure Application Settings:
   Adjust Email Notifications, Time Tracking, and Issue Linking.

4. Audit Logs:
   Check changes made to permissions, workflows, or settings.

Examples:

- Restrict project creation to administrators only.
- Add a new custom permission scheme for marketing projects.
- Enable time tracking globally with a custom working week.

## Slack integration

Integrating JIRA with Slack helps teams get real-time updates on JIRA activities.

Steps to Integrate JIRA with Slack:

1. Install the Slack App:
   Go to Slack Apps and install the Jira Cloud for Slack app.

2. Authorize Integration:
   Authenticate with your JIRA credentials and allow Slack access.

3. Configure Notifications:
   Set up which notifications to receive in Slack (e.g., issue created, status changed).

4. Link Channels to Projects:
   a. Use the command /jira connect in Slack.
   b. Link a specific Slack channel to a JIRA project.

Examples:

- Post a message to the Slack channel when a critical issue is created:
- Automation rule: Trigger: Issue created (Priority = Critical) > Action: Send Slack message.
- Notify the QA team channel when a bug is marked as resolved.
- Use /jira create in Slack to create a new JIRA issue directly.

## Jira dos and don'ts

**Dos:**

1. Organize Projects Properly:
   a. Use consistent naming conventions for projects.
   b. Example: Use department prefixes like "DEV-", "HR-".

2. Use Filters and Dashboards:
   a. Set up shared filters for team visibility.
   b. Example: A dashboard showing all "In Progress" issues for a sprint.

3. Leverage Automation:
   Automate repetitive tasks, such as moving issues to "Done" when resolved.

4. Use Versions and Components:
    a. Break down projects for better tracking.
    b. Example: Create "Frontend" and "Backend" components.

5. Regularly Groom Backlogs:
   Ensure the backlog is prioritized and up to date.

**Don'ts:**

1. Don't Overcomplicate Workflows:
   Avoid adding unnecessary statuses or transitions.

2. Don't Assign Tasks to Multiple People:
   Assign one person to maintain accountability.

3. Don't Ignore User Permissions:
   Ensure sensitive projects have restricted access.

4. Don't Forget to Archive Old Projects:
   Archive completed projects to reduce clutter.

5. Don't Rely Solely on Manual Updates:
   Automate status transitions where possible.