Smart text fields

The value of this **read-only** custom field is dynamically calculated from a custom text expression. It can be used to show a text which aggregates information from one or more field values returned by field codes.

You can also create any custom text value depending on the values of other fields in the current issue or in any other issues (linked issues, sub-tasks, epics, stories, JQL selected issues, etc).

Configuration

Smart fields can be configured in two different ways. You can either use the Template view to choose from a set of preconfigured templates from our use case library, or you can switch to the Expert mode where calculations can be configured freely with our expression editor, best known from our Jira app Jira Workflow Toolbox (Cloud).

Template

The Template view is the default configuration view of Smart Fields for Jira. It provides a set of preconfigured templates which are represented by cards. Each card shows a short name describing its purpose and the icon of the template category.

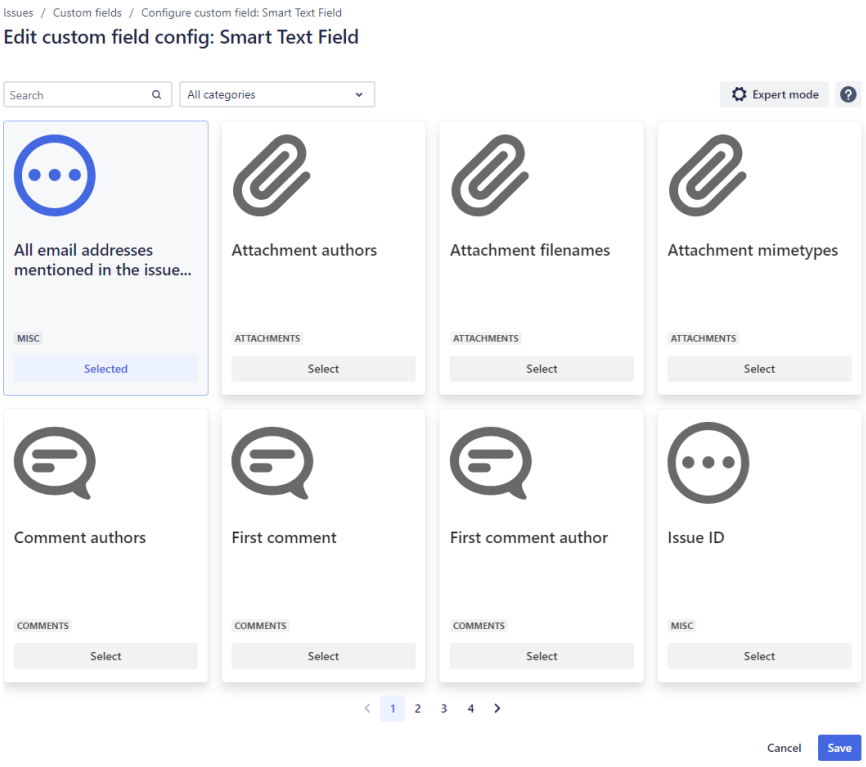
From here, you can switch to the Expert mode at any time. If you had selected a template, its configuration is taken over to the Expert mode, where you can alter the corresponding parser expression. Be aware that you cannot switch back to the Template view once you have made any changes in the Expert mode, even if you reverse them.

1

Select a use case from Template

Select a use case by **clicking on a template card** in the overview, or by clicking on **Select** at the button of the card. The selected template will be highlighted.

You can also **search for templates** by name or drill down in the overview by selecting a specific category, only showing templates from this category.



2

Click **Save** in the bottom right corner to finish your configuration.

Expert mode

The Expert mode allows you to configure custom Smart fields with the expression editor, best known from our best-rated Jira app Jira Workflow Toolbox (Cloud). You are free to edit the entire expression down to the logical elements.

You can choose a card from the Template view, switch to the Expert mode and edit the underlying expression of this template, but you cannot switch back to the Template view once you have made any changes, even if you reverse them.

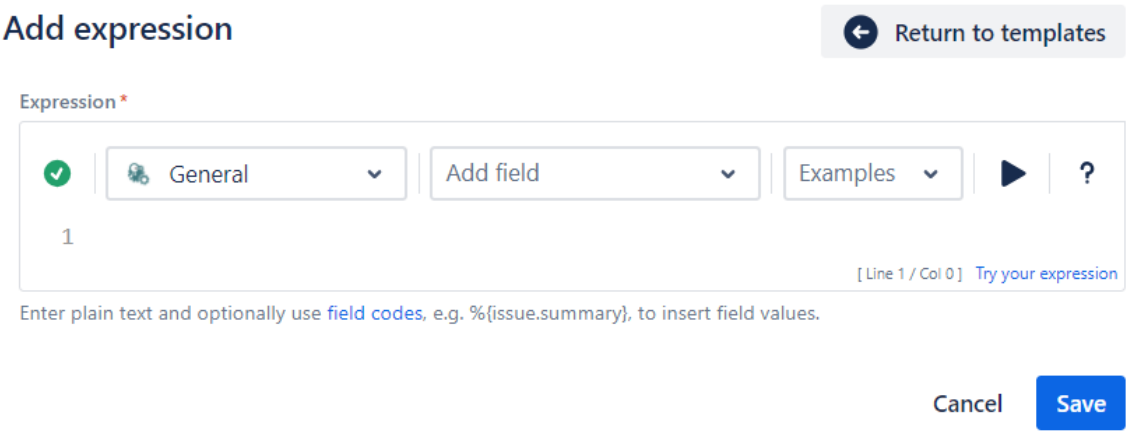
1

Select "Expert mode"

From the Template view, click the **Expert mode** button in the top right corner.

2

Enter a valid Expression



Need some inspiration? We have prepared some really helpful use cases and expressions for you. Check them out here: Use case library

Want to learn more about Expressions? Check out these pages: Expression Parser & Jira Expressions

3

Click **Save** in the bottom right corner to finish your configuration.

Use cases and examples

Use case	Built-in	Parsing mode	Expression
All email addresses mentioned in the issue description		General	`%{findPattern(%{issue.description},"(?<=mailto:)([a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z0-9_-]+)")}`
Assignee groups		Jira expression	`issue.assignee?.groups || ""`
Attachment authors		Jira expression	`issue?.attachments?.reduce((result,att) => result.set(att.author.displayName,""), new Map()).entries().flatten().filter(entry => entry != "")`
Attachment filenames	✓	General	`%{issue.attachments}`

Attachment mimetypes		Jira expression	`issue?.attachments?.reduce((result,att) =>`
			`result.set(att.mimeType,""),`
			`new Map()).entries().flatten().filter(entry => entry != "")`

Comment authors		Jira expression	`issue.comments.reduce((result, comment) =>`
			`result.set(comment.author.displayName, ""),`
			`new Map()).entries().flatten().filter(entry => entry != "")`

Components from all sub-tasks		General	`%{distinct(fieldValue(%{issue.components}, subtasks()))}`
Creation date of the linked epic		Jira expression	`issue?.epic ? issue?.epic.created.toCalendarDate().toString() : ""`
Field value validation		General	`%{matches(%{issue.cfnnnnn},"6|7|8|9") ? "Valid":"Invalid"}` The values entered as arguments in the *matches()* parser function can be replaced with different values or with a regular expressions to validate differently the value of the custom field. Please, note that it is necessary to replace *nnnnn* with the ID of the custom field.
First comment		Jira expression	`issue?.comments[0]?.body?.plainText == null ? '' : issue?.comments[0]?.body?.plainText` The text will be returned using wiki markup, e.g. **bold text** will be shown as *bold text*.
First comment author		Jira expression	`issue?.comments[0]?.author?.displayName`

Historical due dates		Jira expression	`issue.changelogs.filter(changelog => changelog.items.some(item => item.fieldId == 'duedate' && item.from != null)).`
			`map(changelog => {duedateentry: changelog.items.filter(item => item.fieldId == 'duedate' )}).`
			`map(e => new CalendarDate(e.duedateentry[0].from).toString()).join(", ")`

Issue ID		General	`%{issue.id}`

Issue type description		Jira expression	`issue?.issueType?.description`
Keys of linked issues in current project		General	`%{filterByProject(linkedIssues(), %{issue.project.key})}`
Keys of non-estimated sub-tasks		Jira expression	`issue.subtasks.filter(s => s?.originalEstimate == null && s?.customfield_nnnnn == null).map(i => i.key).join(", ")` Please, note that it is necessary to replace *nnnnn* with the ID of the custom field displaying the original estimate or story points.
Last assignee change author		Jira expression	`issue.changelogs.filter(changelog => changelog.items.some(item => item.fieldId == 'assignee')).map(changelog => changelog.author.displayName)[0] || ""`
Last comment		General	`%{issue.lastComment}`
Last comment author		Jira expression	`issue?.comments[issue?.comments?.length-1]?.author?.displayName || ""`
Last status change author		Jira expression	`issue.changelogs.filter(c=>c.items.some(i=>i.field=="status"))[0]?.author?.displayName || ""`
Last status change date		Jira expression	`issue.changelogs.filter(changelog => changelog.items.some(item => item.fieldId == 'status')).map(changelog => changelog.created).concat(issue.created)[0].toString()`

Lowest date and time from linked issues		Jira expression	`let minDate = issue?.links.map(link => link.linkedIssue.customfield_nnnnn).filter( d => d != null);`
			`minDate.length > 0 ? new Date(minDate.reduce((a, b) => a < b ? a : b )).toString() : ""`
			
			Please, replace *nnnnn* with the ID of a Date Time Picker custom field.

Lowest date from linked issues		Jira expression	`let minDate = issue?.links.map(link => link.linkedIssue.customfield_nnnnn).filter( d => d != null);`
			`minDate.length > 0 ? new CalendarDate(minDate.reduce((a, b) => a < b ? a : b )).toString() : ""`
			
			Please, replace *nnnnn* with the ID of a Date Picker custom field.

Parent assignee	✓	General	`%{parent.assignee.displayName}`
Parent priority	✓	General	`%{parent.priority}`
Parent reporter	✓	General	`%{parent.reporter.displayName}`
Parent status	✓	General	`%{parent.status}`
Percentage of completed child issues		Jira expression	`let children = (issue.isEpic ? issue.stories : issue.subtasks);` `children.length ? (children.filter(child => child.resolution).length / children.length * 100 + '').slice(0, 4) + '%' : ''`
Previous assignee		Jira expression	`issue.changelogs.map(i=>i.items).flatten().filter(i=>i.field=="assignee").length>0 ?` `issue.changelogs.map(i=>i.items).flatten().filter(i=>i.field=="assignee").map(a=>a.fromString)[0] : ""`
Previous issue status	✓	General	`%{issue.status.previous}`

Previous reporter	✓	Jira expression	`let reporters= issue.changelogs.map(i=>i.items).flatten().filter(i=>i.field=="reporter");`
			`reporters.length>0 ?`
			`reporters.map(a=>a.fromString)[0] == null ?`
			`""`
			`:reporters.map(a=>a.fromString)[0]`
			`: ""`

Priority description		Jira expression	`issue?.priority?.description`
Project category of issue		General	`%{issue.project.category}`

Project description		General	`%{issue.project.description}`
Project lead		General	`%{issue.project.leadDisplayName}`
Status description		Jira expression	`issue?.status?.description`
Sub-task assignees		General	`%{distinct(fieldValue(%{issue.assignee.displayName}, subtasks()))}`
Sub-task labels		General	`%{distinct(fieldValue(%{issue.labels}, subtasks()))}`
Sub-task reporters		General	`%{distinct(fieldValue(%{issue.reporter.displayName}, subtasks()))}`
Timely resolution assessment		General	`%{{issue.resolutionDate} != null AND {issue.dueDate} != null ? ({issue.resolutionDate} <= {issue.dueDate} ? "Yes" : "No") : (datePart({system.currentDateTime}, RUN_AS_LOCAL) > {issue.dueDate} ? "No" : "")}`