ATJ Release Notes 1.1.0

> **Data Center approved**
>
> This is the first Atlassian approved Data Center compatible version of Automation Toolbox for Jira.

New features

- **Manual** trigger to configure rules that are executed manually
- **Log Work** action to create automatic work logs
- **Create Issue Link** action to create issue links, choosing the issue link type and issues to be linked to
- **Delete Issue Link** action to delete issue links, optionally filtering by issue link type(s) and/or custom issue selection
- Possibility to **import and export rules**

Improvements

- Added the possibility to **search by the field id**'s in the field injector drop-down
- **New style for the expressions** of the elements in the rule view
- Added **confirm dialog** to close the rule sidebar if the configuration has been changed
- Added **Run now** button to **Actions menu** of **activated Scheduled Rules** in the **Automation Rules** view
- Added the possibility to **set and inherit issue links** in the **Create Issue** action
- Added the possibility to **inherit issue fields** in the **Create Issue**action
- Improved the view of the inherit fileds option in the **Create Issue** action
- General **UI improvements** in rule sidebar and updated descriptions

Bug fixes

- **User selectors** were broken, if the selected user was renamed
- **Seed issue** was mistakenly displayed in the **Create Issue** action
- **User condition** used to throw a **Nullpointer** when trying to check against required **Selected user(s)**
- **Syntax highlighting** didn't work for the **cascading field key** when a **value** of a cascading list was selected
- **Cascading fields** didn't work properly
- Removed **Seed issue** option from the issue selection in the **Create Issue** action

Direct download from Marketplace

New expressions in the parser

Function	Input	Output
setBoolean(string **variable_name**, boolean **value**)	STRING BOOLEAN	Creates a variable named **variable_name** for storing a BOOLEAN **value**, and assigns it a **value**, which is also returned in order to be used within an expression. Example: `setBoolean("myBoolean",true)`
getBoolean(string **variable_name**)	BOOLEAN	Returns the value stored in a BOOLEAN variable named **variable_name**, which was previously created using the `setBoolean()` function. Example: `getBoolean("myBoolean")`
setNumber(string **variable_name**, number **value**)	STRING NUMBER	Creates a variable named **variable_name** for storing a NUMBER, and assigns it a **value**, which is also returned in order to be used within an expression. Example: `setNumber("myNumber",100)`
getNumber(string **variable_name**)	STRING NUMBER	Returns the NUMBER value stored in a numeric variable named **variable_name**, which was previously created using the `setNumber()` function. Example: `getNumber("myNumber")`
setString(string **variable_name**, string **value**)	STRING STRING	Creates a variable named **variable_name** for storing a STRING, and assigns it a **value**, which is also returned in order to be used within an expression. Example: `setString("myString","Hello World!")`

getString(string **variable_name**)	STRING	Returns the value stored in STRING variable named **variable_name**, which was previously created using the **`setString()`** function. Example: **`getString("myString")`**
setNumberList(string **variable_name**, number list **value**)	STRING NUMBER []	Creates a variable named **variable_name** for storing a NUMBER [], and assigns it a **value**, which is also returned in order to be used within an expression. Example: **`setNumberList("myNumberList",[1,2,3])`**
getNumberList(string **variable_name**)	STRING	Returns the **value** stored in NUMBER [] variable named **variable_name**, which was previously created using the **`setNumberList()`** function. Example: **`getNumberList("myNumberList")`**
setStringList(string **variable_name**, string list **value**)	STRING STRING []	Creates a variable named **variable_name** for storing a STRING [], and assigns it a **value**, which is also returned in order to be used within an expression. Example: **`setStringList("myStringList",["Hello","World"])`**
getStringList(string **variable_name**)	STRING	Returns the value stored in STRING [] variable named **variable_name**, which was previously created using the **`setStringList()`** function. Example: **`getStringList("myStringList")`**
setIssueList(string **variable_name**, issue list **value**)	STRING ISSUE []	Creates a variable named **variable_name** for storing an ISSUE [], and assigns it a **value**, which is also returned in order to be used within an expression. Example: **`setIssueList("myIssueList",["KEY-1","KEY-2"])`**
getIssueList(string **variable_name**)	STRING	Returns the value stored in ISSUE [] variable named **variable_name**, which was previously created using **`setIssueList()`** function. Example: **`getIssueList("myIssueList")`**
weekOfTheYear(number **t**, number **firstDayOfTheWeek**, number **minimalDaysInFirstWeek**, timeZone **time_zone**)	NUMBER NUMBER NUMBER TIMEZONE	Returns the **week of the year** of the date-time t in a certain **time_zone** as NUMBER. The parameter **firstDayOfTheWeek** represents the first day of the week, e.g.: **{SUNDAY}** in the U.S., and **{MONDAY}** in Germany. The parameter **minimalDaysInFirstWeek** represents the minimal number of days required in the first week of the year, e.g., if the first week is defined as the one that contains the first day of the first month of the year, value **1** should be used. If the minimal number of days required must be a full week (e.g. all days of the week need to be in that year), value **7** should be used. Example: **`weekOfTheYear(2023/01/03, {SUNDAY}, 1, LOCAL)`** returns **1**. Example: **`weekOfTheYear(2023/01/03, {MONDAY}, 1, LOCAL)`** returns **2**. Example: **`weekOfTheYear(2023/01/03, {MONDAY}, 7, LOCAL)`** returns **1**.
dayOfTheYear(number **t**, timeZone **time_zone**)	NUMBER TIMEZONE	Returns the **day of the year** of date-time **t** in a certain **time_zone** as NUMBER, e.g. for January 1st the value returned will be **1**. Example: **`dayOfTheYear(2019/02/01, LOCAL)`** returns **32**
timeInValue(string field **field**, boolean expression **predicate**)	STRING BOOLEAN	Returns the NUMBER of milliseconds a string **field** with code **%{nnnnn}** of the current issue has had a value satisfying a boolean expression **predicate**, where the string value of the **field** with code **%{nnnnn}** is represented by **^%**. Example: **`timeInValue(%{00000}, ^% ~~ "ERROR" OR ^% ~~ "WARNING")`** returns the number of milliseconds the field summary (field code **%{00000}**) of the current issue has contained any of the words "ERROR" or "WARNING", ignoring the case. Example: **`timeInValue(%{00094}, count(toStringList(^%, ",")) > 1)`** returns the number of milliseconds the field components (field code **%{00094}**) of the current issue has contained more than one selected component. Example: **timeInValue(%{00017}, ^% in ["Critical", "High"])** returns the number of milliseconds the field priority (field code **%{00017}**) of the current issue has had a value of Critical or High.

timeInValue(number field **field**, boolean expression **predicate**)	NUMBER BOOLEAN	Returns the NUMBER of milliseconds a number or date-time **field** with code **{nnnnn}** of the current issue has had a value satisfying a boolean expression **predicate**, where the numeric value of the **field** with code **{nnnnn}** is represented by **^**. Example: **`timeInValue({00012}, ^ != null)`** returns the number of milliseconds the field Due date (field code **{00012}**) of the current issue has had a value. Example: **`timeInValue({10001}, ^ >= 5 AND ^ <= 10)`** returns the number of milliseconds a hypothetical numeric field called Passengers (field code **{10001}**) of the current issue has remained between 5 and 10. Example: **`timeInValue({10001}, modulus(^, 2) = 0)`** returns the number of milliseconds a hypothetical numeric field called Passengers (field code **{10001}**) of the current issue has had an even value (2, 4, 6,...).
timeInValue(string field **field**, issue list **issues**, boolean expression **predicate**)	STRING ISSUE [] BOOLEAN	Returns the sum of milliseconds a string **field** with code **%{nnnnn}** has had a value satisfying a boolean expression **predicate** in distinct issues as NUMBER, where the string value of the field with code **%{nnnnn}** is represented by **^%**. Example: **`timeInValue(%{00000}, subtasks(), ^% ~~ "ERROR" OR ^% ~~ "WARNING")`** returns the sum of milliseconds the summary fields (field code **%{00000}**) of all subtasks of the current issue have contained any of the words "ERROR" or "WARNING", ignoring the case. Example: **`timeInValue(%{00094}, epic(), count(toStringList(^%, ",")) > 1)`** returns the number of milliseconds the components fields (field code **%{00094}**) in a linked Epic issue have contained more than one selected component. Example: **`timeInValue(%{00017}, filterByIssueType(linkedIssues(), "Bug, New Feature"), ^% in ["Critical", "High"])`** returns the sum of milliseconds all linked Bugs and New Features of the current issue have had a priority (field code **%{00017}**) value of Critical or High.
timeInValue(number field **field**, issue list **issues**, boolean expression **predicate**)	NUMBER ISSUE [] BOOLEAN	Returns the sum of milliseconds a number or date-time **field** with code **{nnnnn}** has had a value satisfying a boolean expression **predicate** in distinct issues as NUMBER, where the numeric value of the **field** with code **{nnnnn}** is represented by **^**. Example: **`timeInValue({00012}, subtasks(), ^ != null)`** returns the number of milliseconds the field due date (field code **{00012}**) of all subtasks of the current issue have had a value. Example: **`timeInValue({10001}, epic(), ^ >= 5 AND ^ <= 10)`** returns the number of milliseconds a hypothetical numeric field called Passengers (field code **{10001}**) of an Epic issue has had a value between 5 and 10. Example: **`timeInValue({10001}, filterByIssueType(linkedIssues(), "Bug, New Feature"), modulus(^, 2) = 0)`** returns the number of milliseconds a hypothetical numeric field called Passengers (field code **{10001}**) has had an even value in any linked Bug or New Feature.
timeInValue(string field **field**, boolean expression **predicate**, string **schedule_name**, timeZone **time_zone**)	STRING BOOLEAN STRING TIMEZONE	Returns the NUMBER of milliseconds a string **field** with code **%{nnnnn}** of the current issue has had a value satisfying a boolean expression **predicate**, where the string value of the **field** with code **%{nnnnn}** is represented by **^%**. The time being calculated by this function is only counted during a defined schedule with name **schedule_name** for time zone **time_zone**. Example: **`timeInValue(%{00000}, ^% ~~ "ERROR" OR ^% ~~ "WARNING", "schedule_name", LOCAL)`** returns the number of milliseconds the field summary (field code **%{00000}**) of the current issue has contained any of the words "ERROR" or "WARNING", ignoring the case, within a schedule named **schedule_name** for the server's default **time_zone**. Example: **`timeInValue(%{00094}, count(toStringList(^%, ",")) > 1, "schedule_name", LOCAL)`** returns the number of milliseconds the field components (field code **%{00094}**) of the current issue has contained more than one selected component, within a schedule named **schedule_name** for the server's default **time_zone**. Example: **`timeInValue(%{00017}, ^% in ["Critical", "High"], "schedule_name", LOCAL)`** returns the number of milliseconds the current issue has had a priority value of Critical or High (field code **%{00017}**), within a schedule named **schedule_name** for the server's default **time_zone**.

timeInValue(number field **field**, boolean expression **predicate**, string **schedule_name**, timeZone **time_zone**) : number	NUMBER BOOLEAN STRING TIMEZONE	Returns the NUMBER of milliseconds of a number or date-time **field** with code **{nnnnn}** of the current issue has had a values satisfying a boolean expression **predicate**, where the numeric value of the **field** with code **{nnnnn}** is represented by **^**. The time being calculated by this function is only counted during a defined schedule with name **schedule_name** for time zone **time_zone**. Example: **`timeInValue({00012}, ^ != null, "schedule_name", LOCAL)`** returns the number of milliseconds the field due date (field code **{00012}**) of the current issue has had a value, ignoring the case, within a schedule named "my_schedule" for the server's default **time_zone**. Example: **`timeInValue({10001}, ^ >= 5 AND ^ <= 10, "schedule_name", LOCAL)`** returns the number of milliseconds a hypothetical numeric field called Passengers (field code **{10001}**) of the current issue has had a value between 5 and 10, within a schedule named **schedule_name** for the server's default **time_zone**. Example: **`timeInValue({10001}, modulus(^, 2) = 0, "schedule_name", LOCAL)`** returns the number of milliseconds a hypothetical numeric field called Passengers (field code **{10001}**) in current issue has had an even value, within a schedule named **schedule_name** for the server's default **time_zone**.
timeInValue(string field **field**,issue list **issues**, boolean expression **predicate**, string **schedule_name**, timeZone **time_zone**)	STRING ISSUE [] BOOLEAN STRING TIMEZONE	Returns the NUMBER of milliseconds a string **field** with code **%{nnnnn}** has had a value satisfying a boolean expression **predicate** in distinct issues, where the value of the **field** with code **%{nnnnn}** is represented by **^%**. The time being calculated by this function is only counted during a defined schedule with name **schedule_name** for time zone **time_zone**. Example: **`timeInValue(%{00000}, subtasks(), ^% ~~ "ERROR" OR ^% ~~ "WARNING", "my_schedule", LOCAL)`** returns the sum of milliseconds the fields summary (field code **%{00000}**) of all subtasks of the current issue have have contained any of the words "ERROR" or "WARNING", ignoring the case, within a schedule named "**schedule_name**" for the server's default **time_zone**. Example: **`timeInValue(%{00094}, epic(), count(toStringList(^%, ",")) > 1, "my_schedule", LOCAL)`** returns the number of milliseconds the field components (field code **%{00094}**) in the linked Epic issue has contained more than one selected component, within a schedule named **my_schedule** for the server's default **time_zone**. Example: **`timeInValue(%{00017}, filterByIssueType(linkedIssues(), "Bug, New Feature"), ^% in ["Critical", "High"], "my_schedule", LOCAL)`** returns the sum of milliseconds all linked Bugs and New Features of the current issue have had a priority (field code **%{00017}**) value of Critical or High., within a schedule named **my_schedule** for the server's default **time_zone**.
timeInValue(number field **field**, issue list **issues**, boolean expression **predicate**, string **schedule_name**, timeZone **time_zone**)	NUMBER ISSUE [] BOOLEAN STRING TIMEZONE	Returns the NUMBER of milliseconds number or date-time **field** with code **{nnnnn}** has had a value satisfying a boolean expression **predicate** in distinct issues, where the numeric value of the **field** with code **{nnnnn}** is represented by **^**. The time being calculated by this function is only counted during a defined schedule with name **schedule_name** for time zone **time_zone**. Example: **`timeInValue({00012}, subtasks(), ^ != null, "schedule_name", LOCAL)`** returns the number of milliseconds the field due date (field code **{00012}**) of all subtasks of the current issue have had a value, within a schedule named "my_schedule" for the server's default **time_zone**. Example: **`timeInValue({10001}, epic(), ^ >= 5 AND ^ <= 10, "schedule_name", LOCAL)`** returns the number of milliseconds a hypothetical numeric field called Passengers (field code **{10001}**) in the linked Epicissue has had a value between 5 and 10, within a schedule named "**schedule_name**" for the server's default **time_zone**. Example: **`timeInValue({10001}, filterByIssueType(linkedIssues(), "Bug, New Feature"), modulus(^, 2) = 0, "schedule_name", LOCAL)`** returns the number of milliseconds a hypothetical numeric field called Passengers (field code **{10001}**) has had an even value in any linked Bug or New Feature, within a schedule named **schedule_name** for the server's default **time_zone**.
fieldChangeTimes(string field **field**, boolean expression **predicate**)	STRING BOOLEAN	Returns the timestamps as NUMBER [] of when a string value of **field** with code **%{nnnnn}** has changed satisfying a certain **predicate** that depends on the values of the **field** before and after the value change. The string value before the change is represented by **`^0%`**, and after the change by **`^1%`**. The timestamps are returned as a number list sorted in ascending order. Example: **`fieldChangeTimes(%{00000}, ^0% !~~ "IMPORTANT" AND ^1% ~~ "IMPORTANT")`** returns the list of timestamps when word "IMPORTANT" has been added to the current issue's summary (field code **%{00000}**) ignoring the case. Example: **`fieldChangeTimes(%{00017}, ^0% = null AND ^1% != null)`** returns the list of timestamps of when the issue priority (field code **%{00017}**) of the current issue has been set. Example: **`fieldChangeTimes(%{00017}, ^0% not in ["Critical", "High"] AND ^1% in ["Critical", "High"])`** returns the list of timestamps when current issue's priority (field code **%{00017}**) has become Critical or High.

fieldChangeTimes(number field **field**, boolean expression **predicate**)	NUMBER BOOLEAN	Returns the timestamps as NUMBER [] of when a numeric / date-time value of **field** with code **{nnnnn}** has changed satisfying a certain **predicate** that depends on the values of the field before and after the value change. The numeric value before the change is represented by **^0**, and after the change by **^1**. The timestamps are returned as a number list sorted in ascending order. Example: **`fieldChangeTimes({00012}, ^0 < ^1)`** returns the timestamps of when the Due date (field code **{00012}**) has been edited to a higher value. Example: **`fieldChangeTimes({10001}, abs(^0 - ^1) / ^0 >= 0.25)`** returns the timestamps of when a hypothetical numeric field called Passengers(field code **{10001}**) has been edited with a variation of at least 25% over its previous value.
fieldChangeTimes(string field **field**, issue list **issues**, boolean expression **predicate**)	STRING ISSUE [] BOOLEAN	Returns the timestamps as NUMBER [] of when a string value of **field** with code **%{nnnnn}** in distinct parameter issues have changed satisfying certain **predicate** that depends on the values of the fields before and after the value change. The string value before the change is represented by **^0%**, and after the change by **^1%**. The timestamps are returned as a number list containing a sequence of sorted numeric values in ascending order for each parameter issue. Example: **`fieldChangeTimes(%{00000}, subtasks(), ^0% !~~ "IMPORTANT" AND ^1% ~~ "IMPORTANT")`** returns the list of timestamps of when the word "IMPORTANT" has been added the the summary (field code **%{00000}**) of all current issue's subtasks, ignoring the case. Example: **`fieldChangeTimes(%{00017}, epic(), ^0% = null AND ^1% != null)`** returns the list of timestamps of when the issue **priority** (field code **%{00017}**) of the current issue's epic has been set. Example: **`fieldChangeTimes(%{00017}, linkedIssues("is blocked by"), ^0% not in ["Critical", "High"] AND ^1% in ["Critical", "High"])`** returns the list of timestamps of when the **priority**(field code **%{00017}**) in all blocking linked issues has become Critical or High.
fieldChangeTimes(number field **field**, issue list **issues**, boolean expression **predicate**)	NUMBER ISSUE [] BOOLEAN	Returns the timestamps as NUMBER [] of when a numeric value of **field** with code **{nnnnn}** in distinct parameter issues have changed satisfying a certain **predicate** that depends on the values of the fields before and after the value change. The numeric value before the change is represented by **^0**, and after the change by **^1**. The timestamps are returned as a number list containing a sequence of sorted numeric values in ascending order for each parameter issue. Example: **`fieldChangeTimes({00012}, subtasks(), ^0 < ^1)`** returns the timestamps of when the due date (field code **{00012}**) has been edited to a higher value in any of the current issue's subtasks. Example: **`fieldChangeTimes({10001}, epic(), abs(^0 - ^1) / ^0 >= 0.25)`** returns the timestamps when a hypothetical numeric field called Passengers (field code **{10001}**) in the current issue's epic has been edited with a variation of at least 25% over its previous value
lastFieldChangeTime(string field **field**)	STRING	Returns the timestamp as NUMBER of most recent value update of a **field** with code **%{nnnnn}**. Example: **`lastFieldChangeTime(%{00000})`** returns the timestamp of the last update of an issue's summary (field code **{00000}**).
unreleasedVersionsBySequence()		Returns a STRING [] with the unreleased versions **in the current project** with the default order. Only non-archived versions are returned. The first version in the list is the lowermost version in the version table.
releasedVersionsBySequence()		Returns a STRING [] with the released versions **in the current project** with the default order. Only non-archived versions are returned. The first version in the list is the lowermost version in the version table.