JMESPath

JMESPath is a query language for JSON. Examples and a JMESPath tutorial can be found at the JMESPath site. It is used by the field code Action response details and the JWT parser function getFromJSON().

We describe how to get data from JSON objects which are returned as response from an action in after sending a REST request defined by the Atlassian Jira REST API.

How to work with JMESPath in general

We introduce the main search terms that allow you to extract the relevant information from a response. For more complex ones, please have a look at the JMESPath specification.

```
{
  "firstLevelAttribute": "This is the value of a first level attribute",
  "boolean": true,
  "jsonAttribute": {
    "firstName": "John",
    "lastName": "Doe"
  },
  "emptyList": [],
  "emptyJSON": {
  },
  "number": 42,
  "null": null,
  "listWithJSON": [
    {
      "key": "1",
      "name": "Sue",
      "city": "Sydney"
    },
    {
      "key": "2",
      "name": "Jeff",
      "city": "Washington"
    },
    {
      "key": "3",
      "name": "Sergej",
      "city": "Sydney",
    },
    {
      "key": "4",
      "name": "Ayse",
      "city": "Washington",
    }
  ],
  "listWithComplexJSON": [
    {
      "key": "1",
      "name": {
        "first": [
          "Tom",
          "Bob"
        ],
        "last": "Miller"
      }
    },
    {
      "key": "2",
      "name": {
```

```
        "first": [
          "John",
          "Jane"
        ],
        "last": "Doe"
      }
    }
  ],
  "differentNameStructure": [
    {
      "key": "value1",
      "name": {
        "first": [
          "Tom",
          "Bob"
        ],
        "last": "Miller"
      }
    },
    {
      "key": "value2",
      "person": {
        "first": [
          "John",
          "Jane"
        ],
        "last": "Doe"
      }
    }
  ]
}
```

you can access the values as described in the table below.

<table>
<tr><th>What to get</th><th>JMESPath</th><th>Example</th></tr>
<tr><td>Attributes on the first level</td><td>Use the name of the attribute.</td><td>
<table>
<tr><th>JMESPath</th><th>Result</th><th>Notes</th></tr>
<tr><td><code>firstLevelAttribute</code></td><td>This is the value of a first level attribute</td><td></td></tr>
<tr><td><code>jsonAttribute</code></td><td>{ "firstName": "John","lastName": "Doe" }</td><td></td></tr>
<tr><td><code>emptyList</code></td><td></td><td>The empty list is converted to an empty string</td></tr>
<tr><td><code>emptyJSON</code></td><td>{}</td><td></td></tr>
<tr><td><code>boolean</code></td><td>true</td><td></td></tr>
<tr><td><code>number</code></td><td>42</td><td></td></tr>
<tr><td><code>null</code></td><td></td><td>The null value is converted to an empty string</td></tr>
<tr><td><code>notExisting</code></td><td></td><td>If the attribute does not exist, an empty string is returned</td></tr>
</table>
</td></tr>
<tr><td>Attributes on a lower JSON level</td><td>Use the path to this attribute.</td><td>
<table>
<tr><th>JMESPath</th><th>Result</th></tr>
<tr><td><code>jsonAttribute.firstName</code></td><td>John</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td>Access a certain element of a list attribute</td>
<td>Use the index of the list</td>
<td>
<table>
<tr><th>JMESPath</th><th>Result</th><th>Notes</th></tr>
<tr><td><code>listWithComplexJSON[0]</code></td><td>{ "key": "1", "name": { "first": ["Tom", "Bob"], "last": "Miller" } }</td><td>The first element of a list has the index 0</td></tr>
<tr><td><code>listWithJSON[1]</code></td><td>{ "key": "2", "name": "Jeff" }</td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>Get attributes of all list elements</td>
<td>Use the * instead of an index and add the attribute name/path</td>
<td>
<table>
<tr><th>JMESPath</th><th>Result</th><th>Notes</th></tr>
<tr><td><code>listWithJSON[*].name</code></td><td>Sue,Jeff,Sergej,Ayse</td><td></td></tr>
<tr><td><code>listWithJSON[*]</code></td><td>{ "key": "1", "name": "Sue" }, { "key": "2", "name": "Jeff" }, { "key": "3", "name": "Sergej" }, { "key": "4", "name": "Ayse" }</td><td>You can achieve the same result by <code>listWithJSON</code></td></tr>
<tr><td><code>listWithComplexJSON[*].name.last</code></td><td>Miller,Doe</td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>Get a certain attribute of all elements and all level</td>
<td>Use <code>[*]</code>and <code>.*.</code> in combination

<code>*</code> in a path means "all values at this level"

<code>[]</code> flattens the result if it is a list (or a list of lists)</td>
<td>
<table>
<tr><th>JMESPath</th><th>Result</th><th>Notes</th></tr>
<tr><td><code>differentNameStructure[*]</code></td><td><code>[{"key":"value1","name":{"first":["Tom","Bob"],"last":"Miller"}},{"key":"value2","person":{"first":["John","Jane"],"last":"Doe"}}]</code></td><td>Get all elements of the list <code>differentNameStructure</code></td></tr>
<tr><td><code>differentNameStructure[*].*</code></td><td><code>[["value1",{"first":["Tom","Bob"],"last":"Miller"}],
["value2",{"first":["John","Jane"],"last":"Doe"}]]</code></td><td>Now get all elements of the next level (i.e. all values of the <code>"key"</code>, <code>"name"</code> and <code>"person"</code> elements)

This is a list of lists</td></tr>
<tr><td><code>differentNameStructure[*].*.last</code></td><td><code>[["Miller"],["Doe"]]</code></td><td>Getting only the last names of the structure (if available)

This is still a list of lists (with only one element)</td></tr>
<tr><td><code>differentNameStructure[*].*.last[]</code></td><td><code>Miller,Doe</code></td><td>The operator <code>[]</code> flattens the result</td></tr>
</table>
</td>
</tr>
<tr>
<td>Filter data</td>
<td>Add a comparison as index for an array</td>
<td>
<table>
<tr><th>JMESPath</th><th>Result</th><th>Notes</th></tr>
<tr><td><code>listWithJSON[?city == 'Washington']</code></td><td><code>[{"key":2,"name":"Jeff","city":"Washington"},{"key":4,"name":"Ayse","city":"Washington"}]</code></td><td></td></tr>
<tr><td><code>listWithJSON[?city == 'Washington'].{key: key, name:name}</code></td><td><code>[{"key":2,"name":"Jeff"},{"key":4,"name":"Ayse"}]</code></td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>Compose new JSON out of the existing one</td>
<td>Add the structure of the new JSON</td>
<td>
<table>
<tr><th>JMESPath</th><th>Result</th><th>Notes</th></tr>
<tr><td><code>listWithJSON.{index: to_string(key), name: name}</code></td><td><code>[{"index":"1","name":"Sue"},
{"index":"2","name":"Jeff"},
{"index":"3","name":"Sergej"},{"index":"4","name":"Ayse"}]</code></td><td>Build a different structure where the key is called "index" and is a text and not a number. Leave out the attribute "city". This structure may be requested from a different function.</td></tr>
<tr><td><code>listWithJSON[?city == 'Washington'].{index: key, name:name}</code></td><td><code>[{"index":2,"name":"Jeff"},
{"index":4,"name":"Ayse"}]</code></td><td>Combine a filter with a structure.</td></tr>
</table>
</td>
</tr>
</table>

Attribute is not available	If an attribute/path is entered which does not exist in the JSON, null is returned	**JMESPath**
		`doesNotExist`
		`firstLevelAttribute.doesNotExist`
		`jsonAttribute.doesNotExist`
		`jsonAttribute.first.doesNotExist`
		`listWithJSON[200]`

How to work with JMESPath in JWT expressions

JMESPath is used by the field code Action response details and the parser function getFromJSON().

Syntactic rules

Escaping certain characters

When used in Action response details, the curly brackets have to be escaped:

character	escaped character
{	\{
}	\}

Example:

```
%{action.response.listWithComplexJSON[].\{"fullname":name.last\}}
```

When used in getFromJSON(), the quote has to be escaped

character	escaped character
"	\"

Example:

The custom field with the custom ID 10010 holds the value

```
"listWithComplexJSON": [
    {
      "key": "1",
      "name": {
        "first": [
          "Tom",
          "Bob"
        ],
        "last": "Miller"
      }
    },
    {
      "key": "2",
      "name": {
        "first": [
          "John",
          "Jane"
        ],
        "last": "Doe"
      }
    }
  ]
```

```
%{getFromJSON(%{issue.cf10010},"listWithComplexJSON[].{\"fullname\":name.last}")}
```

Parser expressions in JMESPath

You cannot use JWT field codes or parser expressions in JMESPath itself.

Conversion from JSON types to string

They **always** return a **text**, e.g. the boolean value `true` will be converted to the string value `true`, an array with elements of a flat data type (e.g. number) will be a comma separated string of the array elements. The `null` value is represented as an empty text.

If the result is not

- null
- boolean
- string
- number

but a JSON object itself, this JSON object will be converted to a text with `JSON.stringify`

The same is done if an array does not hold flat values but JSON objects.

These results may be stored in a custom field and being read with the new parser function getFromJSON() in a different post function.

Cast the text results

In case the result is not a text but e.g. a number or an array, you can cast it using the known functions from the parser like toNumber() or toStringList() and used as input for other parser functions or to set a field in post function like Create issue.

JWT parser expression	Result	Note
`%{toStringList(%{action.response.fields.fixVersions[*].name})}`	v2,v5	
`%{first(toStringList(%{action.response.fields.fixVersions[*].name}))}`	v2	This is equivalent to `%{action.response.fields.fixVersions[0].name}`
`%{toNumber(%{action.response.fields.customfield_10042})}`	21	

How to use JMESPath for a Jira REST API response

The following JSON is an excerpt of the response returned by a GET /rest/api/3/issue call for issue PU-670


```
{
  "id": "11986",
  "self": "https://your-domain.atlassian.net/rest/api/2/issue/11986",
  "key": "PU-670",
  "fields": {
    "summary": "Get the data from the database",
    "labels": [
      "Backend",
      "Frontend"
    ],
    "creator": {
      "self": "https://your-domain.atlassian.net/rest/api/2/user?accountId=557058%3A145e0983-5707-439c-80e4-
1160dd57f113",
      "accountId": "557058:145e0983-5707-439c-80e4-1160dd57f143",
      "emailAddress": "not.me@decadis.de",
      "avatarUrls": {
        "48x48": "https://avatar-management--avatars.us-west-2.prod.public.atl-paas.net/557058:145e0983-5707-
439c-80e4-1160d57f114/9f9c7a9a-460d-41e2-825e-859866ae81c6/48",
        "24x24": "https://avatar-management--avatars.us-west-2.prod.public.atl-paas.net/557058:145e0983-5707-
439c-80e4-1160d57f114/9f9c7a9a-460d-41e2-825e-859866ae81c6/24",
        "16x16": "https://avatar-management--avatars.us-west-2.prod.public.atl-paas.net/557058:145e0983-5707-
439c-80e4-1160d57f114/9f9c7a9a-460d-41e2-825e-859866ae81c6/16",
        "32x32": "https://avatar-management--avatars.us-west-2.prod.public.atl-paas.net/557058:145e0983-5707-
439c-80e4-1160d57f114/9f9c7a9a-460d-41e2-825e-859866ae81c6/32"
      },
      "displayName": "Who am I",
      "active": true,
      "timeZone": "Europe/Berlin",
      "accountType": "atlassian"
    },
    "statuscategorychangedate": "2021-11-11T00:58:58.197-1100",
    "fixVersions": [
      {
        "self": "https://your-domain.atlassian.net/rest/api/2/version/10003",
        "id": "10003",
        "description": "",
        "name": "v2",
        "archived": false,
        "released": true,
        "releaseDate": "2020-04-28"
      },
      {
        "self": "https://your-domain.atlassian.net/rest/api/2/version/10006",
        "id": "10006",
        "description": "",
        "name": "v5",
        "archived": false,
        "released": false
      }
    ],
    "issuelinks": [
      {
        "id": "10292",
        "self": "https://your-domain.atlassian.net/rest/api/2/issueLink/10292",
        "type": {
          "id": "10000",
          "name": "Blocks",
          "inward": "is blocked by",
          "outward": "blocks",
          "self": "https://your-domain.atlassian.net/rest/api/2/issueLinkType/10000"
        },
        "outwardIssue": {
          "id": "12013",
          "key": "PU-684",
          "self": "https://your-domain.atlassian.net/rest/api/2/issue/12013",
          "fields": {
            "summary": "a",
            "status": {
              "self": "https://your-domain.atlassian.net/rest/api/2/status/10001",
              "description": "",
              "iconUrl": "https://your-domain.atlassian.net/",
              "name": "To Do",
              "id": "10001",
              "statusCategory": {
                "self": "https://your-domain.atlassian.net/rest/api/2/statuscategory/2",
                "id": 2,
                "key": "new",
                "colorName": "blue-gray",
                "name": "To Do"
              }
            },
```

```
            "priority": {
              "self": "https://your-domain.atlassian.net/rest/api/2/priority/3",
              "iconUrl": "https://your-domain.atlassian.net/images/icons/priorities/medium.svg",
              "name": "Medium",
              "id": "3"
            },
            "issuetype": {
              "self": "https://your-domain.atlassian.net/rest/api/2/issuetype/10003",
              "id": "10003",
              "description": "A small piece of work that's part of a larger task.",
              "iconUrl": "https://your-domain.atlassian.net/rest/api/2/universal_avatar/view/type/issuetype
/avatar/10316?size=medium",
              "name": "Sub-task",
              "subtask": true,
              "avatarId": 10316,
              "hierarchyLevel": -1
            }
          }
        }
      },
      {
        "id": "10292",
        "self": "https://your-domain.atlassian.net/rest/api/2/issueLink/10292",
        "type": {
          "id": "10000",
          "name": "Blocks",
          "inward": "is blocked by",
          "outward": "blocks",
          "self": "https://your-domain.atlassian.net/rest/api/2/issueLinkType/10000"
        },
        "outwardIssue": {
          "id": "12013",
          "key": "PU-685",
          "self": "https://your-domain.atlassian.net/rest/api/2/issue/12013",
          "fields": {
            "summary": "a",
            "status": {
              "self": "https://your-domain.atlassian.net/rest/api/2/status/10001",
              "description": "",
              "iconUrl": "https://your-domain.atlassian.net/",
              "name": "To Do",
              "id": "10001",
              "statusCategory": {
                "self": "https://your-domain.atlassian.net/rest/api/2/statuscategory/2",
                "id": 2,
                "key": "new",
                "colorName": "blue-gray",
                "name": "To Do"
              }
            },
            "priority": {
              "self": "https://your-domain.atlassian.net/rest/api/2/priority/3",
              "iconUrl": "https://your-domain.atlassian.net/images/icons/priorities/medium.svg",
              "name": "Medium",
              "id": "3"
            },
            "issuetype": {
              "self": "https://your-domain.atlassian.net/rest/api/2/issuetype/10003",
              "id": "10003",
              "description": "A small piece of work that's part of a larger task.",
              "iconUrl": "https://your-domain.atlassian.net/rest/api/2/universal_avatar/view/type/issuetype
/avatar/10316?size=medium",
              "name": "Sub-task",
              "subtask": true,
              "avatarId": 10316,
              "hierarchyLevel": -1
            }
          }
        }
      },
      {
        "id": "10292",
        "self": "https://your-domain.atlassian.net/rest/api/2/issueLink/10292",
        "type": {
          "id": "10000",
          "name": "Blocks",
          "inward": "is blocked by",
          "outward": "blocks",
          "self": "https://your-domain.atlassian.net/rest/api/2/issueLinkType/10000"
        },
        "inwardIssue": {
          "id": "12013",
          "key": "PU-677",
          "self": "https://your-domain.atlassian.net/rest/api/2/issue/12013",
```

```
          "fields": {
            "summary": "a",
            "status": {
              "self": "https://your-domain.atlassian.net/rest/api/2/status/10001",
              "description": "",
              "iconUrl": "https://your-domain.atlassian.net/",
              "name": "To Do",
              "id": "10001",
              "statusCategory": {
                "self": "https://your-domain.atlassian.net/rest/api/2/statuscategory/2",
                "id": 2,
                "key": "new",
                "colorName": "blue-gray",
                "name": "To Do"
              }
            },
            "priority": {
              "self": "https://your-domain.atlassian.net/rest/api/2/priority/3",
              "iconUrl": "https://your-domain.atlassian.net/images/icons/priorities/medium.svg",
              "name": "Medium",
              "id": "3"
            },
            "issuetype": {
              "self": "https://your-domain.atlassian.net/rest/api/2/issuetype/10003",
              "id": "10003",
              "description": "A small piece of work that's part of a larger task.",
              "iconUrl": "https://your-domain.atlassian.net/rest/api/2/universal_avatar/view/type/issuetype
/avatar/10316?size=medium",
              "name": "Sub-task",
              "subtask": true,
              "avatarId": 10316,
              "hierarchyLevel": -1
            }
          }
        }
      }
    ],
    "subtasks": [
      {
        "id": "12193",
        "key": "PU-805",
        "self": "https://your-domain.atlassian.net/rest/api/3/issue/12193",
        "fields": {
          "summary": "one",
          "status": {
            "self": "https://your-domain.atlassian.net/rest/api/3/status/10001",
            "description": "",
            "iconUrl": "https://your-domain.atlassian.net/",
            "name": "To Do",
            "id": "10001",
            "statusCategory": {
              "self": "https://your-domain.atlassian.net/rest/api/3/statuscategory/2",
              "id": 2,
              "key": "new",
              "colorName": "blue-gray",
              "name": "To Do"
            }
          },
          "priority": {
            "self": "https://your-domain.atlassian.net/rest/api/3/priority/3",
            "iconUrl": "https://your-domain.atlassian.net/images/icons/priorities/medium.svg",
            "name": "Medium",
            "id": "3"
          },
          "issuetype": {
            "self": "https://your-domain.atlassian.net/rest/api/3/issuetype/10003",
            "id": "10003",
            "description": "A small piece of work that's part of a larger task.",
            "iconUrl": "https://your-domain.atlassian.net/rest/api/2/universal_avatar/view/type/issuetype
/avatar/10316?size=medium",
            "name": "Sub-task",
            "subtask": true,
            "avatarId": 10316,
            "hierarchyLevel": -1
          }
        }
      },
      {
        "id": "12194",
        "key": "PU-806",
        "self": "https://your-domain.atlassian.net/rest/api/3/issue/12194",
        "fields": {
          "summary": "two",
          "status": {
```

```
            "self": "https://your-domain.atlassian.net/rest/api/3/status/10001",
            "description": "",
            "iconUrl": "https://your-domain.atlassian.net/",
            "name": "To Do",
            "id": "10001",
            "statusCategory": {
              "self": "https://your-domain.atlassian.net/rest/api/3/statuscategory/2",
              "id": 2,
              "key": "new",
              "colorName": "blue-gray",
              "name": "To Do"
            }
          },
          "priority": {
            "self": "https://your-domain.atlassian.net/rest/api/3/priority/3",
            "iconUrl": "https://your-domain.atlassian.net/images/icons/priorities/medium.svg",
            "name": "Medium",
            "id": "3"
          },
          "issuetype": {
            "self": "https://your-domain.atlassian.net/rest/api/3/issuetype/10015",
            "id": "10015",
            "description": "A very small piece of work that's part of a larger task.",
            "iconUrl": "https://your-domain.atlassian.net/rest/api/2/universal_avatar/view/type/issuetype
/avatar/10338?size=medium",
            "name": "Small Sub-task",
            "subtask": true,
            "avatarId": 10338,
            "hierarchyLevel": -1
          }
        }
      }
    ]
  },
  "customfield_10050": {
    "self": "https://your-domain.atlassian.net/rest/api/3/user?accountId=5fb3d09bf8b01200694cffb1",
    "accountId": "5fb3d09bf8b01200694cffb1",
    "avatarUrls": {
      "48x48": "https://secure.gravatar.com/avatar/29d7f88795ae29f2ce7b66e42005f49?d=https%3A%2F%2Favatar-
management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FU-1.png",
      "24x24": "https://secure.gravatar.com/avatar/29d7f88795ae29f2ce7b66e42005f49?d=https%3A%2F%2Favatar-
management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FU-1.png",
      "16x16": "https://secure.gravatar.com/avatar/29d7f88795ae29f2ce7b66e42005f49?d=https%3A%2F%2Favatar-
management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FU-1.png",
      "32x32": "https://secure.gravatar.com/avatar/29d7f88795ae29f2ce7b66e42005f49?d=https%3A%2F%2Favatar-
management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FU-1.png"
    },
    "displayName": "The cutest user",
    "active": true,
    "timeZone": "Pacific/Midway",
    "accountType": "atlassian
  },
  "customfield_10042": 21
}
```


The following table using the field code Action response details.

What to get	JMESPath	Result
Issue key	`%{action.response.key}`	PU-670
Display name of the creator	`%{action.response.fields.creator.displayName}`	Who am I
Atlassian account Id from a single user picker custom field	`%{action.response.fields.customfield_10050.accountId}`	5fb3d09bf8b01200694cffb1
Get all labels	`%{action.response.fields.labels}`	Backend,Frontend
Get the second label	`%{action.response.fields.labels[0]}`	Frontend
Get all sub-tasks	`%{action.response.subtasks}`	`{ "id": "12193", "key": "PU-805", "self": "https://your-domain.atlassian.net/rest/api/3/issue/12193", "fields": { "summary": "one",`

```
      "status": {
        "self": "https://your-domain.
atlassian.net/rest/api/3/status
/10001",
        "description": "",
        "iconUrl": "https://your-domain.
atlassian.net/",
        "name": "To Do",
        "id": "10001",
          "statusCategory": {
            "self": "https://your-domain.
atlassian.net/rest/api/3/statuscategory
/2",
             "id": 2,
            "key": "new",
             "colorName": "blue-
gray",
             "name": "To Do"
             }
        },
      "priority": {
        "self": "https://your-domain.
atlassian.net/rest/api/3/priority
/3",
        "iconUrl": "https://your-domain.
atlassian.net/images/icons/priorities
/medium.svg",
        "name": "Medium",
        "id": "3"
        },
        "issuetype": {
          "self": "https://your-domain.
atlassian.net/rest/api/3/issuetype
/10003",
          "id": "10003",
          "description":
          "A small piece of work that's
part of a larger task.",
          "iconUrl": "https://your-
domain.atlassian.net/rest/api/2
/universal_avatar/view/type/issuetype
/avatar/10316?size=medium",
          "name": "Sub-task",
          "subtask": true,
          "avatarId": 10316,
          "hierarchyLevel": -1
        }
    }
  },
  {
    "id": "12194",
    "key": "PU-806",
    "self": "https://your-domain.
atlassian.net/rest/api/3/issue
/12194",
    "fields": {
      "summary": "two",
      "status": {
        "self": "https://your-domain.
atlassian.net/rest/api/3/status
/10001",
        "description": "",
        "iconUrl": "https://your-domain.
atlassian.net/",
        "name": "To Do",
        "id": "10001",
        "statusCategory": {
          "self": "https://your-domain.
atlassian.net/rest/api/3/statuscategory
/2",
          "id": 2,
          "key": "new",
          "colorName": "blue-
gray",
          "name": "To Do"
        }
      },
      "priority": {
        "self": "https://your-domain.
atlassian.net/rest/api/3/priority
/3",
        "iconUrl": "https://your-domain.
atlassian.net/images/icons/priorities
/medium.svg",
```

```
        "name": "Medium",
        "id": "3"
        },
        "issuetype": {
            "self": "https://your-domain.
atlassian.net/rest/api/3/issuetype/10015"
,
            "id": "10015",
            "description": "A very small
piece of work that's part of a larger
task.",
            "iconUrl": "https://your-
domain.atlassian.net/rest/api/2
/universal_avatar/view/type/issuetype
/avatar/10338?size=medium",
            "name": "Small Sub-task",
            "subtask": true,
            "avatarId": 10338,
            "hierarchyLevel": -1
          }
    }
  }
```

<table>
<tr><td>Get the first sub-task</td><td>%{action.response.fields.subtasks[0]}</td><td><pre>{
    "id": "12193",
    "key": "PU-805",
    "self": "https://your-domain.
atlassian.net/rest/api/3/issue
/12193",
    "fields": {
      "summary": "one",
      "status": {
        "self": "https://your-domain.
atlassian.net/rest/api/3/status
/10001",
        "description": "",
        "iconUrl": "https://your-domain.
atlassian.net/",
        "name": "To Do",
        "id": "10001",
          "statusCategory": {
            "self": "https://your-domain.
atlassian.net/rest/api/3/statuscategory
/2",
            "id": 2,
           "key": "new",
            "colorName": "blue-
gray",
            "name": "To Do"
            }
        },
      "priority": {
        "self": "https://your-domain.
atlassian.net/rest/api/3/priority
/3",
        "iconUrl": "https://your-domain.
atlassian.net/images/icons/priorities
/medium.svg",
        "name": "Medium",
        "id": "3"
        },
        "issuetype": {
          "self": "https://your-domain.
atlassian.net/rest/api/3/issuetype
/10003",
          "id": "10003",
          "description":
          "A small piece of work that's
part of a larger task.",
          "iconUrl": "https://your-
domain.atlassian.net/rest/api/2
/universal_avatar/view/type/issuetype
/avatar/10316?size=medium",
          "name": "Sub-task",
          "subtask": true,
          "avatarId": 10316,
          "hierarchyLevel": -1
        }
    }
  }</pre></td></tr>
<tr><td>Get the names of all fix versions</td><td>%{action.response.fields.fixVersions[*].name}</td><td>v2,v5</td></tr>
<tr><td>Get the keys of all linked issues</td><td>%{action.response.fields.issuelinks[*].*.key[]}</td><td>PU-684, PU-685, PU-677

The second * eases the access to all linked issues, independent of the direction (<code>inward</code> or <code>outward</code>). <code>[]</code> flattens the result that otherwise would be list of list.</td></tr>
<tr><td>Filter the sub-tasks by their issuetype and return the sub-task's summary and the status</td><td>%{action.response.fields.subtasks[?fields.issueType.name == 'Sub-task'].\{"summarySub-task": fields.summary, "statusSub-Task": fields.status.name\}}</td><td>{"summarySub-task":"one","statusSub-Task":"To Do"}</td></tr>
</table>

If you still have questions, feel free to refer to our support team.