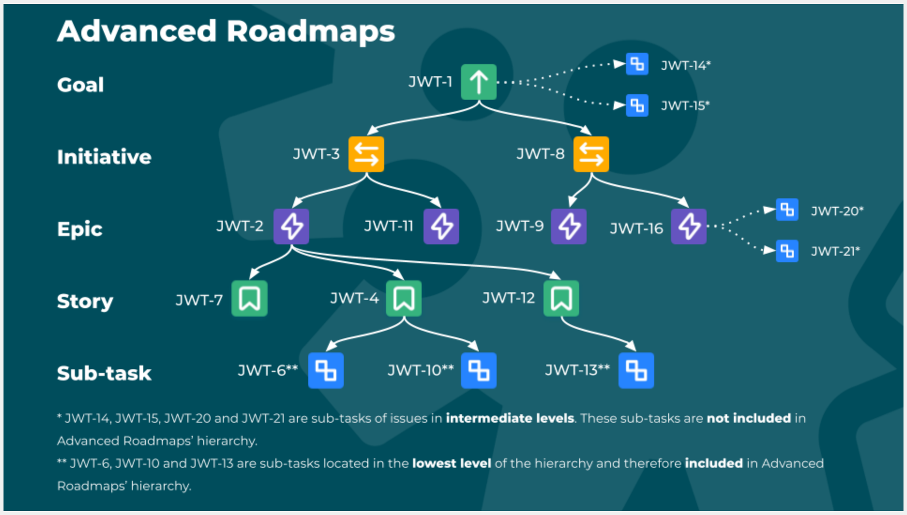
issuesAbove()

This function returns an **issue list** containing all the ancestor issues of a given issue according to **Advanced Roadmaps hierarchy**, i.e., all related issues in levels above a given issue, or in other words: the parent issue, the grandparent issue, the great grandparent issue, etc. The output list is ordered in ascending levels, being the parent issue the first element of the list.

Advanced Roadmaps hierarchy includes:

- Issues linked by **Parent Link** custom field.
- Issues linked by **Epic-Story** link.
- **Sub-tasks** in the **lowest level** of the hierarchy. Sub-tasks under issues in intermediate levels of the hierarchy are not considered to belong to **Advanced Roadmaps hierarchy.**

Example:



* JWT-14, JWT-15, JWT-20 and JWT-21 are sub-tasks of issues in **intermediate levels**. These sub-tasks are **not included** in Advanced Roadmaps' hierarchy.
** JWT-6, JWT-10 and JWT-13 are sub-tasks located in the **lowest level** of the hierarchy and therefore **included** in Advanced Roadmaps' hierarchy.

Syntax

```
issuesAbove() #Output: Issue list
```

Examples

Parser expression	Description
`issuesAbove()`	Considering that current issue is *JWT-13*, this expression returns: **[JWT-12, JWT-2, JWT-3, JWT-1]**

Output

This function returns an ISSUE LIST

Variant where you additionally define **issues**.

Duplicated issues are filtered from the output.

Syntax

```
issuesAbove(issueList) #Output: Issue list
```

Examples

Parser expression	Description
`issuesAbove(linkedIssues("is blocked by"))`	Assuming that expression `linkedIssues("is blocked by")` returns `[JWT-2, JWT-9]`, this example returns all the issues in the level just under issues *JWT-2* and *JWT-9*: **[JWT-3, JWT-1, JWT-8]** To achieve this the following functions are used: - linkedIssues()

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
`issueList`	ISSUE LIST	Any given issue list. Usually this value is retrieved from a function (e.g. linkedIssues() or issuesUnderEpic()).

Output

This function returns an ISSUE LIST

Variant where you define **issue keys** instead.

Duplicated issues are filtered from the output.

Syntax

```
issuesAbove(issueKeys) #Output: Issue list
```

Examples

Parser expression	Description

`issuesAbove("JWT-7, JWT-9")`	This example returns all issues in the level just under issues *JWT-7* and *JWT-9*: **[JWT-2, JWT-3, JWT-1, JWT-8]**

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
`issueKeys`	TEXT	A comma separated list of issue keys.

Output

This function returns an ISSUE LIST

To return **the issues just under an issue** within the Advanced Roadmaps hierarchy, you might want to have a look into the issuesUnder() function.

To return **all issues under an issue** within the Advanced Roadmaps hierarchy, you might want to have a look into the allIssuesUnder() function.

To return **all issues having the same parent** within the Advanced Roadmaps hierarchy, you might want to have a look into the siblingIssues() function.

To return **the parent of an issue** within the Advanced Roadmaps hierarchy, you might want to have a look into the parent() function.

Use cases and examples

Use case
No content found.