getIssueType

Signature

Helper	nodeHelper		
Signature	getIssueType(String issueTypeName)	getIssueType(String issueTypeName, String projectKey)	getIssueType(String issueTypeName, IHubProject project)
Description	Returns an issuetype object based on the issueTypeName Returns a null when the issueTypeName is not found	Returns an issuetype object based on the issueTypeName and a projectKey. Returns a null when the issueTypeName is not found	
Introduced in	0.8	4.0.0	

Example use

```
issue.issueType = nodeHelper.getIssueType("Bug")
```

For Exalate 4.0.0 and higher

We retrieve a project key from the issue variable to find a set of issue types associated with a certain project.

Example use

Get the issue type, based on the replica issueTypeName and projectKey.

Incoming sync for new issues(create/change processor)

```
issue.typeName = nodeHelper.getIssueType(replica.type?.name, (issue.projectKey ?: issue.project.key))?.name ?: "Task"
```

Asana Usage Examples

Retrieve an issue type by its name in Asana:

```
entity.type = nodeHelper.getIssueType("task")
```

Freshdesk / Freshservice Usage Examples

Retrieve an issue type by its name in Freshdesk / Freshservice:

```
entity.type = nodeHelper.getIssueType("Incident")
```

Mapping Remote Issue Types to Freshdesk / Freshservice Issue Types

If the issue type names differ between Freshdesk / Freshservice and the remote system, you can use a mapping to ensure correct synchronization:

```
def typeMapping = [
    "Task": "Feature Request",
    "[System] Service Request": "Question",
    "[System] Incident": "Incident",
    "[System] Problem": "Problem",
    "[System] Change": "Refund"
]

entity.type = nodeHelper.getIssueType(typeMapping[replica.typeName] ?: "Incident")
```

In this example incoming issue types from the remote system are mapped to appropriate Freshdesk / Freshservice issue types. If no match is found, it defaults to `"Incident"` .

Comments

- The name is case sensitive
- Use issue.issueTypeName as shortcut function

```
issue.issueTypeName = "Bug"
```