Declaring a New Generic Item Type

Generic items are managed by Structure and are similar to folders but may also contain an icon and a description. You can define a generic item type in your app to allow Structure users to create and work with domain-specific items, e.g. milestones or test cases. A new generic item type is defined in your app's `atlassian-plugin.xml` by declaring a `<structure-item-type>` module and using `com.almworks.jira.structure.api.item.generic.GenericItemType` as its implementation class.

Example

```
<structure-item-type key="type-milestone" name="itemtype:Milestone" weight="100"
                     class="com.almworks.jira.structure.api.item.generic.GenericItemType">
 <icon spanClass="my-app-milestone-icon-class"/>
 <displayable key="my.app.milestone.displayable"/>
 <title key="my.app.milestone.title"/>
 <newItemTitle key="my.app.milestone.new"/>
</structure-item-type>
```

Element	Required	Description
`@key`	Yes	Unique module key within the plugin. Full module key will define the `itemType` part of `ItemIdentity`.
`@name`	No	A human-readable name of the plugin module.
`@weight`	No	Determines the order in which generic item types appear in menus and lists. Items with the 'lightest' weight are displayed first and the 'heaviest' items sink to the bottom.
`@class`	Yes	Module class. Must be `GenericItemType`.
`icon`	No	An icon shown in the item row. At this point the only supported option is using a single icon, associated with a CSS class, for all items of the given type. If you're using your own icons, make sure the appropriate CSS styles are loaded everywhere Structure can be used (see Loading Additional Web Resources For Structure Widget).
`displayable`	Yes	An I18N key used to generate a textual representation of an item for activity streams, decision panels, and elsewhere. The value should contain the type name and a placeholder for the item name, e.g. "milestone {0}".
`title`	Yes	An I18N key for the item creation panel, e.g. "Milestone".
`newItemTitle`	Yes	An I18N key for the Structure toolbar's "Add item" drop-down menu, e.g. "New Milestone".

Programmatic Access to Generic Items

Use `GenericItemService` or `GenericItemManager` to create, retrieve, and update generic items in your plugin code. `GenericItemService` is a higher-level component which checks users' permissions and performs other validation tasks as needed. `GenericItemManager` is a low-level component which queries and updates the database, throwing exceptions if anything goes wrong.

Generic Item Permissions

Each generic item is associated with the structure that contains it, and that structure's permissions are used to determine who can see and update the item.

- Any user can create a new generic item programmatically. **Edit** access level is required to add the new item to a structure. When the item is added to a structure, it becomes associated with that structure.
- As with issues and folders, **Edit** access level is required to create a generic item using Structure UI. The item is associated with the structure it was created in.
- If a generic item from one structure is copied or moved to a different structure, a copy of the item is created and associated with the new structure.
- All users having the **View** access level to a structure can view all generic items in that structure.
- All users having the **Edit** access level to a structure can update and delete all generic items in that structure.
- As with any other item, **Edit** access level is required to remove a generic item from a structure. When a generic item is removed from a structure it is **not** deleted from the database. It can still be seen in structure history, accessed or updated programmatically, and re-inserted into the structure.