

Forest Resource

Forest Resource is responsible for serving forests and forest updates and receiving the forest actions (change commands) from the client.

Retrieving Forest

Request

```
GET $baseUrl/rest/structure/2.0/forest/latest?s=$forestSpec
POST $baseUrl/rest/structure/2.0/forest/latest
```

Returns the hierarchical issue list (forest) of the specified structure.

Parameters:

\$forestSpec	<i>required</i>	The URL-encoded JSON representation of ForestSpec . See also: RestForestSpec .
POST content	<i>required</i>	While GET method is preferred, POST is more robust because there's no risk of exceeding URL length with large forest specifications. The content is the same JSON object (but not URL-encoded, obviously).

Example:

```
GET /rest/structure/2.0/forest/latest?s={%22structureId%22:113}
```

Retrieves latest forest for structure #113.

Response

```
{
  "spec": {"structureId": 113},
  "formula": "10394:0:4/356,10332:0:14707,10374:1:5/240,10348:2:14717",
  "itemTypes": {
    "4": "com.almworks.jira.structure:type-generator",
    "5": "com.almworks.jira.structure:type-folder"
  },
  "version": {
    "signature": -1659607419,
    "version": 1
  }
}
```

In this reply, the most important part is "formula", which contains serialized information about the forest.

Each component (delimited by comma) represents a row and looks like this: 10374:1:5/240. In this example, the numbers are:

- 10374 is the row ID,
- 1 is the row depth,
- 5/240 is the item identity.

If the row contains an issue, it's just issue ID, otherwise it has the format of <item type>/<long item id>, or <item type>//<string item id>. Item type is a number, which is expanded in the "itemTypes" map in the reply.

Changing Forest

To change a forest, you POST one or more change actions to `/forest/update` resource. Each action is a serialized version of `ForestAction` – for more information about the actions, see [Changing Structure Content](#).

```
POST $baseUrl/rest/structure/2.0/forest/update
```

Parameters:

```
{
  "spec": { "structureId": <id> }, // use structure ID
  "version": { "signature": <signature>, "version": <version> }, // use last seen signature and version
  "actions": [
    {
      "action": "add",
      "under": 0,           // at the top level
      "after": 123,        // after row ID 123 (not issue id!)
      "before": 456,       // before row ID 456
      "forest": "-100:0:10001" // insert issue 10001, -100 is the temporary row ID which will be mapped
into the real row ID when the method returns
    },
    {
      "action": "move", // works like previously, only row IDs instead of issue IDs
      "rowId": 123,
      "under": 456,
      "after": 0,
      "before": 124
    },
    {
      "action": "remove",
      "rowId": 442
    }
  ]
}
```