

03-选择列表转换

编写Behaviour，在表单初始化时将文本字段转换为选择字段或多选字段。

还可以指定字段的可用选项，包括键入搜索时执行的选择器函数。

以下我们以：从Jira问题中选取 来演示 一个例子

链接文本字段的显示是来自于JQL查询的问题的可搜索列表内容。比如希望单文本字段 “GIT项目名称”，需要一个服务器的返回列表中

第一步

验证一个过滤器，使唤之能够返回正确的数据，比如内容如下

```
project ="DEMO ORDER BY key ASC"
```

第二步

创建一个Behaviour，并编写脚本如下：

```
getFieldByName("GIT项目名称").convertToSingleSelect([
  ajaxOptions: [
    url : getBaseUrl() + "/rest/scriptrunner-jira/latest/issue/picker",
    query: true,
    data: [
      currentJql : "project =EMO ORDER BY key ASC",
    ],
    formatResponse: "issue"
  ],
  css: "max-width: 500px; width: 500px",
])
```

第三步

将此Behaviour的内容关联到WXJ项目，Story的问题类型中，如

Behaviours		
Name	Mappings	Operations
hidefield	 DEMO  Story Delete	Fields Add Mapping Advanced Edit Delete Behaviour
server-side-check		Fields Add Mapping Advanced Edit Delete Behaviour
toSelectList	 WXJ  Story Delete	Fields Add Mapping Advanced Edit Delete Behaviour
Add Behaviour		
To create a new Behaviour please specify a name and optionally the description and press Add .		

验证

接下来，我们可以来对之进行验证

创建问题

配置域

可视化 文本

GIT项目名称

GIT仓库名称

GIT源分支

GIT分支名称

修复的版本

Pick high priority issue in Support project (显示 10 匹配的问题 (共 22个))

DEMO-1 - As an Agile team, I'd like to learn about Scrum >> Click the "...

DEMO-2 - As a product owner, I'd like to express work in terms of actua...

DEMO-3 - As a product owner, I'd like to rank stories in the backlog so I...

DEMO-4 - As a team, I'd like to estimate the effort of a story in Story Po...

DEMO-5 - As a team, I'd like to commit to a set of stories to be complet...

ajaxOptions说明

```

ajaxOptions: [
  url : getBaseUrl() + "/rest/scriptrunner-jira/latest/issue/picker",
  query: true, //在进行选择时，每次按键结束都将后到以进行查询
  //以下信息是请求后台的数据请求内容
  data: [
    currentJql : "后台查询的JQL语句",
    label      : "获得结果显示时首行的提示",
    showSubTasks: false, //是否显示查询结果包含的子任务
    max        : 5, //指定查询结果显示的最大问题条数

  ],
  minQueryLength: 4, //最输入字个字符的时候进行查询
  keyInputPeriod: 500, //每次输入停顿多长毫秒的时候开始并底
  formatResponse: "general||issue", //如果不是issue数据的时候，需要填写 general
],

```

其它例子

当然，我们也可以根据不同项目，不同的问题类型，设置动态的JQL来获得对应数据，如下

```

def selectedProject = getFieldById("所属项目").value as Project //获得表单中指定的一个项目，这个字段项目字段类型
def jqlSearchField = getFieldByName("GIT项目名称")

if (selectedProject) {
    //设置GIT项目名称为只读，并给一个默认的提示
    jqlSearchField.setReadOnly(false).setDescription("Select an issue in the ${selectedProject.name} project")

    jqlSearchField.convertToSingleSelect([
        ajaxOptions: [
            url          : getBaseUrl() + "/rest/scriptrunner-jira/latest/issue/picker",
            query        : true,
            data         : [
                currentJql : "project = ${selectedProject.key} ORDER BY key ASC",
                label      : "Pick high priority issue in ${selectedProject.name} project",
            ],
            formatResponse: "issue"
        ],
        css             : "max-width: 500px; width: 500px",
    ])
} else {
    jqlSearchField.convertToShortText()
    jqlSearchField.setReadOnly(true).setDescription("Please select a project before entering the issue")
}

```

对数据校验

```
def selectedIssueField = getFieldById(getFieldChanged())
def selectedIssue = selectedIssueField.value as String
log.debug("selectedIssue changed: ${selectedIssue}")
def selectedProject = getFieldByName("ProjectPicker").value as Project

if (selectedIssue && selectedProject) {

    def jqlQueryBuilder = JqlQueryBuilder.newBuilder()
    def searchService = ComponentAccessor.getComponent(SearchService)
    def user = ComponentAccessor.jiraAuthenticationContext.getLoggedInUser()

    def query = jqlQueryBuilder.where().project(selectedProject.id).and().issue
(selectedIssue).buildQuery()
    if (searchService.searchCount(user, query) == 1) {
        selectedIssueField.clearError()
    } else {
        selectedIssueField.setError("Issue not found in the selected project")
    }
}
```