XML Specifications

XML Specifications

THIS IS AN EXPERIMENTAL FEATURE

User created gadgets are not supported at this time and the specification below could change without notice. If you want to make your own gadget, please get in touch so we can help you.

- XML Specifications
  - XML Specification
    - Implemented Tags
    - Example
    - Fields

XML Specification

The XML gadget specification is based on the Atlassian Gadgets.

Implemented Tags

The implemented tags and corresponding attributes are:

- ModulePrefs

  Description
  - title
  - width - "full" or "half" are the only options for now
- ContentSources
  - type - "file" uses the file given in src, "html" uses the content in the tag (eg. Content type="html"This is the content/Content)
  - src - relative filename or url
- Source
  Fields
  - type - "css" or "javascript"
  - src - relative filename or url
- Field
  - slug

Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<Module>
    <ModulePrefs title="Contact Info" width="half" />
    <Description>This gadget adds a field editor for fields related to contact info (phone, address, ect).<
/Description>
    <Content type="file" src="template.html" />
    <Sources>
       <Source type="javascript" src="script.js" />
    </Sources>
    <Fields>
       <Field slug="6c-resourceholder-phone-main" />
       <Field slug="6c-resourceholder-phone-fax" />
    </Fields>
</Module>
```

Fields

If a gadget uses fields, you can optionally add the slug of the field in this section to hide it from the main field list.

This can be very useful and make your Resource Sections easier to work with. If the fields are not hidden, this can lead to long lists of redundant data in multiple places and can cause confusion. However, all viewing and editing of the field will have to be done through the gadget. If your gadget uses a field in a read-only manner, then you should **not** add it to the gadget's manifest because that would prevent users from editing the field data through the standard edit page.