

Wiki Markup, HTML and Links

- [Fields with Wiki Markup](#)
- [Inserting Hyperlinks](#)
- [Fields with HTML](#)
- [JavaScript with Wiki or HTML](#)

Fields with Wiki Markup

If the fields to be exported are configured to use a [Wiki Markup Renderer](#), you can place the wiki rendering in the result document.

Definition:

```
${wiki:<Field name>}
```

Expand to see the examples on sample code

```
${wiki:Description}
${wiki:Custom Free Text}
```



When creating an Excel template document, you should use: `${wiki:Description}`

Inserting Hyperlinks

Xporter also allows generating hyperlinks on the fly.

Definition:

```
@{title=<Link Title>|href=<Link Address>}
```

Generate a link named with issue key, linking to the issue address on the Jira Server

Building the hyperlink using Jira issue data.

```
@{title=${Key}|href=${BaseURL}/browse/${Key}}
```

Building the hyperlink using an external URL

```
@{title=${Key}|href=http://www.google.com}
```



When creating an Excel template document, you should use: `${link:title=${Key},href=${BaseURL}/browse/${Key}}`

Fields with HTML

If the fields to be exported are configured to use an HTML renderer, such as the one provided by the [JEditor plugin](#), you can place the HTML rendering in the result document.

Definition:

```
${html:<Field name>}
```

Expand to see the examples on sample code

```
${html:Description}
${html:Custom Free Text}
```



When creating an Excel template document, you should use: `${html:Description}`



If the document's output format is HTML, you should use: `${wiki:Description}` instead.

JavaScript with Wiki or HTML

Xporter allows applying Wiki or HTML functions in the text rendered based on JavaScript. For example, if the fields to be exported are configured to use the **Default Text Renderer** provided by Jira, you can use JavaScript to format the field using the notation explained [here](#) and then apply the Wiki or HTML functions. You can also use JavaScript to format fields that are configured to use the [Wiki Markup Renderer](#) or an HTML renderer, such as the one provided by the [JEditor plugin](#) and then apply the Wiki or HTML functions. To achieve that, the JavaScript function should be defined as a field in the template and the function to be applied to that field.

Definition:

```
${wiki:%{<Javascript>}}
or
${html:%{<Javascript>}}
```

Formatting the first word in field "Description" to "Bold", "Italic" or to have a different text color (red in the examples) using HTML function

```
${html:%{var t = '<b>'+${Description}.split(' ')[0] + '</b>'+${Description}.substr('${Description}.indexOf(' ')+1);t.toString();}}
${html:%{var t = '<i>'+${Description}.split(' ')[0] + '</i>'+${Description}.substr('${Description}.indexOf(' ')+1);t.toString();}}
${html:%{var t = '<b>'+<font color="Red">'+${Description}.split(' ')[0] + '</b>'+</font>'+${Description}.substr('${Description}.indexOf(' ')+1);t.toString();}}
```

Formatting the first word in field "Description" to "Bold", "Italic" or to have a different text color (red in the examples) using Wiki function

```
${wiki:%{var t = '*'+${Description}.split(' ')[0] + '*'+ ${Description}.substr('${Description}.indexOf(' ')+1);t.toString();}}
${wiki:%{var t = '_'+${Description}.split(' ')[0] + '_'+ ${Description}.substr('${Description}.indexOf(' ')+1);t.toString();}}
${wiki:%{var t = '{color:red}'+${Description}.split(' ')[0] + '{color}'+ ${Description}.substr('${Description}.indexOf(' ')+1);t.toString();}}
```