

Bundle Markdown Encoder maven plugin

- [Create markdown file to include with your App](#)
- [Add Bundle-Description tag to pom.xml](#)
- [Add bundle-markdown-encoder plugin \(from IGB project\)](#)
- [Add plugin repository \(from IGB project\)](#)
- [Supported Markdown](#)
- [How this works](#)
- [Example](#)

To help users decide whether to install your App, create a markdown file to package with your App and use the bundle-markdown-encoder plugin to package it with your App. IGB will then display it in the right side panel of the IGB App Manager.

Create markdown file to include with your App

To add documentation to your App:

- Create a markdown file (e.g., AppDescription.md)
- Save it in the top level of your IGB App project (same level as pom.xml)
- Indicate the name of the Markdown file in your pom.xml file using tag **app.description** under **properties**.

pom.xml

```
<properties>
  ...
  <app.description>AppDescription.md</app.description>
  ...
</properties>
```

Add Bundle-Description tag to pom.xml

Add a Bundle-Description tag within the **instructions** tag of the **org.apache.felix maven-bundle-plugin** section of your pom.xml:

pom.xml

```
<instructions>
  <Bundle-Description>${bundleDescription}</Bundle-Description>
</instructions>
```

Add bundle-markdown-encoder plugin (from IGB project)

Add the following plugin (as a child of the plugins tag) to the pom.xml file in your App project:

pom.xml

```
<plugin>
  <groupId>com.lorainelab</groupId>
  <artifactId>bundle-markdown-encoder</artifactId>
  <executions>
    <execution>
      <goals>
        <goal>encodeMarkdown</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

Add plugin repository (from IGB project)

The `pluginRepositories` node is at the top level of the pom file, like the `repositories` node.

Add the following plugin repository to the `pom.xml` file in your App project:

pom.xml

```
<pluginRepositories>
  <pluginRepository>
    <id>maven-releases</id>
    <url>https://nexus.bioviz.org/repository/maven-releases</url>
  </pluginRepository>
</pluginRepositories>
```

Supported Markdown

The IGB App Manager supports github-flavored markdown, described here: <https://help.github.com/articles/github-flavored-markdown/>

However, before deploying your Markdown you should test it with the App Manager.

How this works

The `bundle-markdown-encoder` module uses [Base64](#) encoding to translate the content of your Markdown document into a sequence of alphanumeric characters. It inserts this sequence into the `description` tag of the `repository.xml` file that you create when you compile your App using maven. When users add a new App repository and then select an App, IGB decodes the contents of the App's `description` tag and displays them in the App Manager. The App Manager contains code that can render the Markdown.

Note that currently, if you would like to reference an image in your `README.md`, you'll need to use an absolute URL to wherever the image is hosted. You'll also need to ensure that the image URL is always accessible. We recommend that if you are hosting your own App repository, you keep the images you need there. This will ensure that if users can download your App to install it, IGB can also download the image.

Example

README.md

```
# Heading 1

## Heading 2

### Heading 3

#### Heading 4

Here is [a link](https://bioviz.org)

An unordered list:

* Item one
* Item two

A table:

| Left-Aligned  | Center Aligned  | Right Aligned |
| :-----:    | :-----:    | -----:    |
| col 3 is     | some wordy text | $1600       |
| col 2 is     | centered       | $12         |

* * *

An image:

![A Plant](https://example.com/Plant.png "A Plant")
```

Here's how it looks in the App Manager:

