目录

附件：GitBook使用说明

概述

本文档由云粒智慧科技有限公司撰写，旨在作为文档编写模板，辅助工程文档的编写。

GitBook安装和使用

GitBook在Windows和Mac电脑上均可使用。

安装NodeJs和NPM

从这里安装：Node.js installer

安装GitBook工具

```
npm install gitbook-cli -g
```

安装PDF转换工具

从这里安装：Calibre application

在Mac上还需要建一个符号链接：

```
sudo ln -s ~/Applications/calibre.app/Contents/MacOS/ebook-convert /usr/local/bin
```

链接创建的位置 /usr/local/bin 可以替换成在 $PATH 里面的任意位置。

【可选】安装封面自动生成的依赖软件

封面自动生成插件需要额外的依赖软件，详见 https://github.com/GitbookIO/plugin-autocover

> This module use node-canvas. You need to install some modules on your system before being able to use it: Wiki of node-canvas.

这个插件的安转比较麻烦，不建议使用。

GitBook的使用

拿到文档样例后，解压。在Mac上使用unar解压工具进行解压，解决中文目录名的问题。

```
sudo brew install unar
unar -e GBK xxx.zip
```

然后参照样例中用户中心的目录结构，根据所需编写的文档创建自己的文档目录结构，并编写md格式的文章。

使用md格式编写一篇文章

使用标准的md格式编写一篇文章，建议使用Typora，非常优美高效。

文档撰写工具

统一使用GitBook生成在线电子文档和离线PDF文档，便于交流。

MarkDown文档的编写，推荐使用Typora编辑器，支持Mac和Windows。a decent markdown editor for free，下载地址：https://www.typora.io/。

下面是文档编写中常用的元素的示例：

代码

```
class Test {
    private string name;
}
```

```
require(['gitbook', 'jQuery'], function (gitbook, $) {
  var url = ''
  var title = ''
  var style = ''
  var insertLogo = function (url, title, style) {
    var logo = url ? '<img src="' + url + '" style="' + style + '">' : ''
    var ttl = title ? '<h1 class="summary-title">' + title + '</h1>' : ''
    $('.book-summary').children().eq(0).before('<div class="book-logo">' + logo + ttl


  }
})
```

```
ls
mkdir test
```

```
dir
md help
```

图片

PDF

文档样例

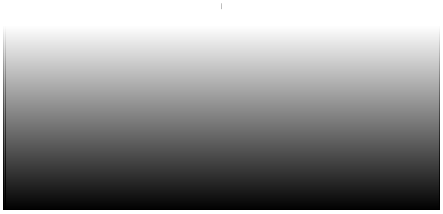
不支持嵌入的PDF对象: 文档样例

视频

链接方式：

https://yunlizhihui.yuque.com/preview/yuque/0/2020/mp4/550992/1579425293405-966a9c16-351a-4b49-b900-9b22ae880a04.mp4

嵌入方式：



面板

Panel without title.

This is a panel with title

Panel with title and default style.

This is a danger panel

Panel with title and danger style.

This is an info panel

Panel with title and info style.

This is a success panel

Panel with title and success style.

This is a warning panel

Panel with title and warning style.

文档目录的生成

GitBook支持通过手工编写SUMMARY.md文件的方式生成文档目录，也支持通过插件的方式自动生成文档目录。

手工撰写文档目录

手工撰写文档目录比较灵活，只需要按照类似下面的格式编写SUMMARY.md并放置在文档根目录下即可。

手工撰写文档目录，文档在文件系统上的存储结构可以比较随意灵活，不必遵循严格的规则。

```
- [用户中心](应用中台/1-用户中心/README.md)

    - [组件简介](应用中台/1-用户中心/0-组件简介/README.md)

        - [什么是租户](应用中台/1-用户中心/0-组件简介/0-什么是租户.md)
        - [什么是业务](应用中台/1-用户中心/0-组件简介/1-什么是业务.md)
        - [什么是组织](应用中台/1-用户中心/0-组件简介/2-什么是组织.md)
    - [快速入门]()
        - [创建租户](应用中台/1-用户中心/1-快速入门/0-创建租户.md)
        - [创建业务](应用中台/1-用户中心/1-快速入门/1-创建业务.md)
        - [创建组织](应用中台/1-用户中心/1-快速入门/2-创建组织.md)
    - [接口文档](应用中台/1-用户中心/2-接口文档/README.md)

        - [用户中心 API](应用中台/1-用户中心/2-接口文档/0-用户中心 API.md)
        - [用户中心 SDK](应用中台/1-用户中心/2-接口文档/1-用户中心 SDK.md)
    - [用户指南]()
    - [常见问题]()
    - [其他资源]()
```

手工编写文档目录，只需要在book.json中禁用（-summary）或者删除自动生成插件summary就可以了：

```
"plugins": [
        "-summary-pro"
    ],
```

自动生成文档目录

使用插件自动生成文档目录，需要在book.json中配置插件如下：

```
"plugins": [
        "summary-pro"
    ],

"pluginsConfig": {
        "summary-pro": {
            "firstpage": {
                "title": "概述"
            }
        },
}
```

这个插件的配置，只有一项：firstpage.title。这个配置的是文档根目录下README.md文件在文档目录中的标题。

有了这个插件，就不必手工编写SUMMARY.md文件，而是需要按照特定的规则来组织文档在文件系统上的存储结构。

如下面的示例，通过目录名称，文档的文件名，前缀序号的等方式，我们就可以生成文档目录结构，并可以通过序号来控制文档排列的顺序。

```
- [产品介绍](应用中台/0-产品介绍.md)
- [用户中心](应用中台/1-用户中心/README.md)

    - [组件简介](应用中台/1-用户中心/0-组件简介/README.md)

        - [什么是租户](应用中台/1-用户中心/0-组件简介/0-什么是租户.md)
        - [什么是业务](应用中台/1-用户中心/0-组件简介/1-什么是业务.md)
        - [什么是组织](应用中台/1-用户中心/0-组件简介/2-什么是组织.md)
    - [快速入门]()
        - [创建租户](应用中台/1-用户中心/1-快速入门/0-创建租户.md)
        - [创建业务](应用中台/1-用户中心/1-快速入门/1-创建业务.md)
        - [创建组织](应用中台/1-用户中心/1-快速入门/2-创建组织.md)
    - [接口文档](应用中台/1-用户中心/2-接口文档/README.md)

        - [用户中心 API](应用中台/1-用户中心/2-接口文档/0-用户中心 API.md)
        - [用户中心 SDK](应用中台/1-用户中心/2-接口文档/1-用户中心 SDK.md)
    - [用户指南]()
    - [常见问题]()
    - [其他资源]()
- [报表引擎](应用中台/2-报表引擎/README.md)

    - [组件简介]()
    - [快速入门]()
    - [接口文档]()
    - [用户指南]()
    - [常见问题]()
    - [其他资源]()
```

文档封面的制作

技术文档可以配置一张图片作为封面，也可以通过插件auto cover自动生成，但这个插件的安转比较麻烦，不建议使用。

自定义图片封面

如果要使用自定义图片作为封面，在文档的根目录下放置 `cover.jpg`，如果想要缩略图可以放置 `cover_small.jpg`，文件格式必须为 jpg。

一个好的封面需要:

- 大小要求 cover.jpg 1800x2360 pixels , cover_small.jpg 200x262
- 不要有边框
- 有清晰的标题
- 任何小的标题需要清晰可见

使用auto cover插件自动生成封面

在book.json里面配置auto cover，示例如下：

```
{
    "title": "My Book",
    "author": "Author",

    "plugins": [
        "autocover"
    ],

    "pluginsConfig": {
        "autocover": {
            "font": {
                "size": null,
                "family": "Impact",
                "color": "#FFF"
            },
            "size": {
                "w": 1800,
                "h": 2360
            },
            "background": {
                "color": "#09F"
            }
        }
    }
}
```

文档的发布

使用GitBook可以将撰写好的技术文档发布为在线文档，或者是PDF格式的电子文档。

发布为在线文档

```
gitbook serve --port 5000
```

在根目录下执行上述命令，将在端口5000上启动web服务，可在线访问文档内容。

发布为PDF格式的文档

```
gitbook pdf .\  ..\用户中心文档.pdf
```

在根目录下执行上述命令，将在上级目录下生成PDF格式的文档。