

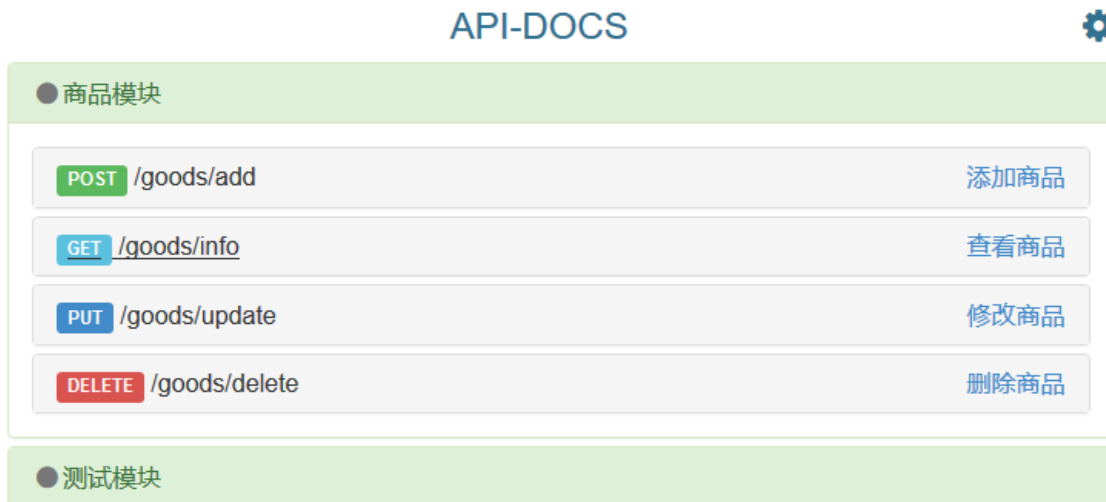
# API-DOSC使用说明

作者：xiaojian 2017年8月3日

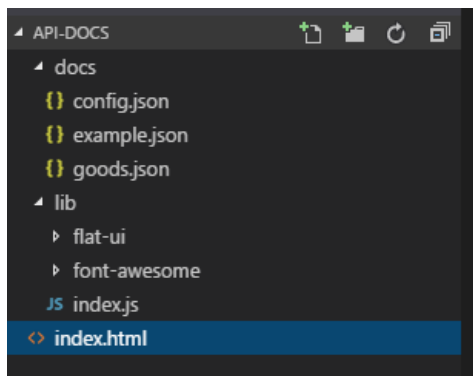
## 简介

提供API测试功能，方便前端查看接口，了解数据结构

## 界面



## 工具结构



## 安装方法

将API-DOCS文件夹放置在Laravel/Lumen项目的public目录下，浏览器中输入对应访问地址即可访问API文档，（如网站地址是<http://127.0.0.1>，请输入<http://127.0.0.1/API-DOCS>）

## 文档编写规则

需要编写的文档都在API-DOCS目录中的docs文件夹中

必须文件：`config.json`（文档配置文件）

文件内容：

```
{
  "server": "", //服务器地址，一般此处不填写
  "title": "API-DOCS", //API文档标题，可修改
}
```

```

"headers": [
  "token", "account" //接口调用是如需要附带header, 在此处设置
],
"docs": [
  "example.json" //引入的API文档, 只有此处有的文档才会被加载
]
}

```

文档文件（多个）：xxxxx.json

说明：文档文件为API文档的主要部分，每个文件就是一个API模块，如默认创建了一个 example.json文件（测试模块）

内容：（可以goods.json文件为参考）

```

{
  "title": "商品模块",          //模块名称
  "forms": [
    ..... //模块包含的接口列表
  ]
}

```

title:这是模块的名称，对文档来说必须是唯一的，也就是说不能有相同的两个模块名称，其对应文档的模块标题，如图：

### ● 商品模块

forms:这是接口列表，forms数组的每一个元素为一个接口，形式如：

```
forms:[ {..接口1..},{..接口2..},{..接口3..} ]
```

单独对模块接口列表进行编写

```

"forms": [
  {
    "title": "查看商品",          //接口标题
    "description": "商家查看商品信息", //接口的描述信息
    "url": "/goods/info",
    //接口的路由地址（注意laravel路由开头没有‘/’，是‘goods/info’）
    "method": "get", //api请求方式
    "inputs": [
      { "type": "text", "name": "id", "description": "商品ID" }
    ]
  },
]

```

单独讨论inputs,inputs是输入框列表，inputs每一个子元素对应一个表单控件

```

"inputs": [
  {
    "type": "text", //输入框类型，目前只有text,file两种，text是文本输入，file是文件选择
    "name": "id", //输入框name,也就是API字段名称（参数名）
    "description": "商品ID" //对字段的描述，不要太长
  }
]

```

GET /goods/info

查看商品

id

商品ID

商家查看商品信息

测试接口

编写例子

场景：我需要开发一个商品管理模块，模块有4个功能 查看指定ID的商品，删除指定ID的商品，添加指定的商品，修改商品

第一步：

创建goods.json文件，（文件名称要与模块有联系，不要乱起名称）

第二步：

编写模块的基本信息，如下图

```
{
  "title": "商品模块",
  "forms": [
  ]
}
```

第三步：在forms中添加第一个API接口（查看指定ID的商品），此功能为请求者提供一个商品的ID，

就要获取这个ID的商品所有信息.

1给这个接口起个名字：查看商品

2描述一下这个接口：商家查看商品信息

3接口的访问地址：‘/goods/info’，对应访问链接为 http://...../goods/info

4这是一个简单的数据获取接口，所以使用GET方法，

5接口只需要提供一个ID参数，所以只需要一个文本输入框，类型为text，字段名称为id，描述一下这个字段：商品ID

```
{
  "title": "商品模块",
  "forms": [
    {
      "title": "查看商品",
      "description": "商家查看商品信息",
      "url": "/goods/info",
      "method": "get",
      "inputs": [
        { "type": "text", "name": "id", "description": "商品ID" }
      ]
    }
  ]
}
```

然后我们开始写第二个接口：添加商品接口

```

{
  "title": "商品模块",
  "forms": [
    {
      "title": "查看商品",
      "description": "商家查看商品信息",
      "url": "/goods/info",
      "method": "get",
      "inputs": [
        { "type": "text", "name": "id", "description": "商品ID" }
      ]
    },
    {
      "title": "添加商品",
      "description": "商家添加商品",
      "url": "/goods/add",
      "method": "post",
      "inputs": [
        { "type": "text", "name": "name", "description": "商品名称" },
        { "type": "text", "name": "price", "description": "商品价格" },
        { "type": "file", "multiple": "true", "name": "image", "description": "商品图片" }
      ]
    }
  ]
}

```

可以发现与之前的查看指定ID的接口不一样，inputs中多了几个东东

1. 添加商品需要：商品名称，所以有了

```
{ "type": "text", "name": "name", "description": "商品名称" },
```

2. 添加商品需要：商品价格，所以有了

```
{ "type": "text", "name": "price", "description": "商品价格" },
```

3. 添加商品需要：商品图片，所以有了（此处type为file,multiple为true代表可选多张图片，false为单张）

```
{ "type": "file", "multiple": "true", "name": "image", "description": "商品图片" }
```

然后我们继续写完其他接口（结果请看API-DOCS中docs目录的goods.json文件）

最后，我们要在config.json中引入goods.json

```

{
  "server": "",
  "title": "API-DOCS",
  "headers": [
    "token", "account"
  ],
  "docs": [
    "goods.json",
    "example.json"
  ]
}

```

然后刷新页面，你可以看到

API-DOCS



## ● 商品模块

POST /goods/add

添加商品

GET /goods/info

查看商品

PUT /goods/update

修改商品

DELETE /goods/delete

删除商品

