

法规系统对外提供接口

版本记录

日期	版本	功能	操作人
2018.8.6	v1.0	new	陈利军

授权认证：

在调用所有业务接口时，都需要在 Http Request Header 中配置以下参数用于授权认证：

参数	格式	说明	必须
X-Appid	string	由被调用方提供的应用id	是
X-CurTime	string	当前UTC时间戳，从1970年1月1日0点0 分0 秒开始到现在的毫秒数	是
X-Checksum	string	令牌，计算方法：MD5(apiKey + curTime)，三个值拼接的字符串，进行MD5哈希计算（32位小写），其中apiKey由被调用方提供，调用方管理。	是

1.提交征询

uri:/v1/api/consultCommit

method:post

请求参数：

bookId：【必填】法规id
userId：【必填】用户id
nodeId：【必填】法规章节id
text：【必填】意见
mark：【非必填】备注

请求示例：

```
{
  "bookId": 123,
  "userId": xxx,
  "consultList": [
    {
      "nodeId": 995,
      "text": "强烈支持995",
      "mark": "备注测试995"
    },
    {
      "nodeId": 995,
      "text": "强烈支持995",
      "mark": "备注测试995"
    },
    {...},
    ...
  ]
}
```

响应示例：

```
{
  "status": "200", //正常返回为200 其他为异常返回，具体异常信息为“errmsg”
  "errmsg": "success", //请求异常时异常信息
  "data": {} //返回的请求数据
}
```

2.获取意见

uri:/v1/api/opinionList

method:get

请求参数：

bookId: 【必填】法规id

请求示例：

```
{
  "bookId": "721",
}
```

响应示例：

```
{
  "statsu": "200",
  "errmsg": "success",
  "data": {
    "list": [
      {
        "bookId": 721, //法规id
        "createTime": 1531747564522,
        "id": 1,
        "msg": "测试意见", //意见
        "nodeId": 1001, //章节id
        "userId": 1 //用户id
      },
      {...},
      ...
    ]
  }
}
```

3.提交用户意见

uri:/v1/api/opinionCommit

method:post

请求参数：

bookId：【必填】法规id
nodeId：【必填】章节id
userId：【必填】用户id
msg：【必填】用户意见

请求示例：

```
{
  "bookId": 721,
  "nodeId": 1001,
  "userId": 123,
  "msg": "测试意见"
}
```

响应示例：

```
{
  "status": "200",
  "errmsg": "success",
  "data": {}
}
```

4.全部法规列表

uri:/v1/api/allBooks

method:get

请求参数：

pageSize：【非必填】每页条数 默认：20
pageNum：【非必填】页数（从1开始） 默认：1
history：【非必填】是否将法规的历史版本也返回，默认：false

说明： 存在隐含条件即只返回处于发布状态的法规，处于其他状态的法规不会进行返回

请求示例：

```
{
  "pageSize": 2,
  "pageNum": 30
}
```

响应示例：

```

{
  "statsu": "200",
  "errmsg": "success",
  "data": {
    "list": [
      {
        "title": "111", //法规名
        "version": "R1", //法规版本
        "abs": "新的摘要", //摘要
        "bookType": "", //位阶类型
        "buhao": "", //部号
        "startDate": 1535731200000, //发布开始时间
        "endDate": 1536422400000, //发布结束日期
        "id": 777, //法规id
        "linghao": "中国民用航空总局令 第49号", //令号/文号
        "managerOrg": "", //管理单位
        "status": 2, //法规状态 1: 提前发布 2: 发布中
        "tagList": [ //自定义关键字
          {
            "id": 1,
            "desc": "关键字1"
          },
          {...},
          ...
        ],
        "bizList": [ //业务分类
          {
            "id": 1,
            "desc": "分类1"
          },
          {...},
          ...
        ]
      },
      {...},
      ...
    ]
  },
  "page": { //分页信息
    "pageSize": 20,
    "pageNum": 1,
    "total": 1000
  }
}

```

5. 获取章节内容

uri:/v1/api/nodeDetail

method:get

请求参数：

nodeIds：【必填】章节ids，多个id用逗号分隔

请求示例：

```
{
  "nodeIds": 2,3,4,5
}
```

响应示例：

```
{
  "statsu": "200",
  "errmsg": "success",
  "data": {
    "list": [
      {
        "id": 1049, //章节Id
        "bookId": 777, //所属法规Id
        "title": "aaaaa", //标题
        "nodeNumber": "111", //标题号
        "content": "<p>1111111</p>\n" //章节正文（html格式）
      },
      {...},
      ...
    ]
  }
}
```

说明：由于章节的内容比较大，建议一次性获取的章节尽量少的

6.获取法规目录（整体目录）

uri:/v1/api/bookContent

method:get

请求参数：

bookId: 【必填】法规Id

请求示例：

```
{
  bookId: 123
}
```

响应示例：

```
{
  "statsu": "200",
  "errmsg": "success",
  "data": {
    "list": [
      {
        "id": 1049, //章节Id
        "bookId": 123, //所属法规Id
        "title": "aaaaa", //标题
        "nodeNumber": "111", //标题号
        "nodeLevel": 1, //章节层级（从1开始）
        "nodeOrder": 1 //节点顺序
      },
      {...},
      ...
    ]
  }
}
```

说明：目前法规的树形目录结构是根据nodelevel 和nodeorder两个字段来进行控制，具体返回的数据结构有待商榷

6.获取章节子节点

uri:/v1/api/bookNodeCld

method:get

请求参数：

bookId：【必填】法规Id
nodeId：【必填】章节Id （如果是获得第一层级的目录则传0）

请求示例：

```
{
  bookId: 123,
  nodeId: 345
}
```

响应示例：

```
{
  "status": "200",
  "errmsg": "success",
  "data": {
    "list": [
      {
        "id": 345, //章节Id
        "bookId": 123, //所属法规Id
        "title": "aaaaa", //标题
        "nodeNumber": "111", //标题号
        "nodeLevel": 1, //章节层级（从1开始）
        "nodeOrder": 1 //节点顺序
      },
      {...},
      ...
    ]
  }
}
```

6.搜索法规或章节

uri:/v1/api/search

method:get

请求参数：

```
bookName: 【非必填】法规名（模糊搜索）
nodeName: 【非必填】章节标题（模糊搜索）
nodeNumber: 【非必填】章节标题号 （模糊搜索）
```

说明： 以上三个参数必须有一个不为空

请求示例：

```
{
  bookName: "航空法"
}
```

响应示例：

```
{
  "statsu": "200",
  "errmsg": "success",
  "data": {
    "list": [
      {
        "id": 345, //章节Id
        "type": 1, //1: 法规 2: 章节
        "title": "aaaaa", //法规名（章节名）
        "nodeNumber": "111", //章节标题号
        "nodeVersion": 1, //章节版本
        "nodeType": 节, //章节类型
        "preview": "预览" //法规（章节）预览
        "idPath": "1,3,345" //id路径
        "namePath": "航空法,第一章,第三节" //描述路径
      },
      {...},
      ...
    ]
  }
}
```