

# BIMT写作指导插件API V0.4

---

## 目录

---

- 修订记录
- 文档简介
  - 名词解释
    - Rest
    - RESTful
    - Jersey
  - 文档目的
  - 读者对象
- 接口规约
  - URL前缀
  - 请求头
  - 请求方法和状态码
  - 数据验证和错误报告
- API
  - 用户相关
    - 用户登入
    - 用户登出
    - 获取用户信息
  - 搜索相关
    - 关键词搜索
    - 搜索索引检索
    - 文献推荐
    - 获取文献
  - 格式相关
    - 格式列表
    - 获取格式

- 增加格式
- 更新格式

# 修订记录

版本	修订人	修订时间	编写/修订说明	备注
V0.1	yx1@bimt.com	2016.03.31	1. 新建文档并完成第一版本API编写	后期可能会有修订之处，定稿后再发布V1.0
V0.2	yx1@bimt.com	2016.04.04	1. 文献推荐接口加入文档数量参数； 2. 修改部分字段名称； 3. 去掉格式具体内容定义；	由于格式内容复杂多变，所以改由客户端进行定义
V0.3	yx1@bimt.com	2016.04.13	1. 搜索索引检索，添加页码参数； 2. 文献推荐，URL增加了"analysis"	
V0.31	yx1@bimt.com	2016.04.14	1. 获取文献，添加摘要，额外信息两个字段； 2. 整理排版	
V0.32	yx1@bimt.com	2016.04.20	1. recommendDocs改成recommandDocs	拼写错误
V0.33	yx1@bimt.com	2016.04.26	1. 增加文献搜索可排序字段说明	
V0.4	yx1@bimt.com	2016.05.03		

			1. 根据客户端插件编写要求，文献返回字段中增加一些字段	这些字段是数据库中没有的，均返回空。后期其内容及格式也可能发生改变
--	--	--	------------------------------	-----------------------------------

# 文档简介

---

本文档主要针对《BIMT写作指导插件功能需求（第一期）》中标明的功能，设计相应的后台REST API接口，规范接口URL以及其支持的操作。

## 名词解释

---

### Rest

Representational State Transfer, or REST, was introduced and defined in 2000 by the doctoral dissertation of Roy Fielding, one of the principal authors of the HTTP specification versions 1.0 and 1.1. The most important concept in REST is resources, which are identified by global IDs-- typically using URIs. Client applications use HTTP methods (GET/ POST/ PUT/ DELETE) to manipulate the resource or collection of resources.

### RESTful

A RESTful Web service is a Web service implemented using HTTP and the principles of REST.

### Jersey

Jersey RESTful Web Services framework is open source, production quality, framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation. More information please refer to Jersey official website - <https://jersey.java.net/> <https://jersey.java.net/>.

## 文档目的

---

本文档是BIMT WORD插件后台研发人员必须参考和遵照的手册。本文档设计的接口主要供给

WORD插件端软件研发人员调用，也可供给手机APP研发人员调用。

# 读者对象

---

软件研发人员。

# 接口规约

---

本文档中接口都是基于Jersey框架，使用JAVA开发的RESTful Web Service，使用这些REST API与BIMT WORD插件端、iOS APP、Android APP、Web前端以及其他应用交互，传输JSON格式的搜索结果、推荐数据等到相应的数据请求方，并对数据访问的安全性进行控制。API是基于REST原则的，可以很方便的客户端使用几乎任何编程语言与REST API进行交互。

## URL前缀

---

本说明书中所有涉及的URL都必须加上如下URL前缀：<https://bimt.com/api/plugin/v1> 其中bimt.com主域名是比美特医护在线域名，/api/plugin代表这是给插件使用的API，v1为API版本号。这里使用https协议进行数据传输，为了确保数据隐私和安全，同时也是由于iOS APP后台数据传输苹果逐渐倾向只支持https协议的缘故。

## 请求头

---

在HTTP协议到请求头中常常包含着一些全局信息，如Cookie等。这样服务器在接收到请求时，只需要解析其请求头，就可以对这个请求进行一些预处理。例如我们常常在用户登录后，将一些标识信息写入到浏览器Cookie存储区域。每次浏览器发起请求时，就需要将本地Cookie读取并加入到请求头中，服务器解析其头，就可以通过头内容配合服务端数据来判断用户是否已经登录。而一般在基于HTTP协议到API设计中，我们也会设计一个头字段来存储全局信息。对于本文中，定义头"BIMT-WA-KEY"来表示用户登录标识。用户进行登入操作后，所有到请求都需要将这个标识放入请求头。具体参考用户登入

## 请求方法和状态码

---

在客户端发出HTTP请求时，API根据具体的上下文环境和操作类型，返回不同的状态码、数据实体。对于每一个请求，客户端需要明确其请求到资源ID、请求方法、请求参数，服务器需要返回其请求结果，结果中需要能够明确标示出请求是否成功、失败原因等内容。例如，客户端需要在资源users上使用POST方法来创建一个新用户，同时需要提供新建用户所必需到参数；服务器可能会返回状态码400 Bad Request，即请求中没有提供请求所需的参数。

客户端的HTTP请求方法：

HTTP方法	描述
GET	请求获取一个或一组资源数据，并以JSON格式返回数据
POST	请求创建一个资源。如果创建成功，服务器将返回201 Created状态码，并在数据中包含新资源引用位置
PUT	请求更新一个资源的数据，如果更新成功，服务器将返回200 OK状态码；如果该资源不存在，服务器把这个请求当成一个POST请求处理
DELETE	请求删除一个或一组资源，服务器直接返回200 OK状态码，即使该资源不存在

服务器返回的HTTP状态码：

HTTP状态码	描述
200 OK	请求成功
201 Created	创建资源请求成功，返回新资源引用位置
400 Bad Request	请求中必需参数缺失
401 Unauthorized	用户请求未授权，一般为缺少Key，参考用户登入小节
403 Forbidden	不被允许的请求，例如用户不能删除别人到项目
404 Not Found	资源不存在
405 Method Not Allowed	请求方法不被支持
409 Conflict	资源已存在，例如创建项目时，项目名称已存在
422 Unprocessable	请求实体未被处理
500 Server Error	处理请求时，服务端出现错误

## 数据验证和错误报告



API在处理接收到请求时，首先会对请求中数据项和数据格式进行验证，如果出现错误，则返回400 Bad Request状态码。具体包含以下两种情况：

- 必需的请求参数缺失，例如发送短信时，未提供发送到的电话号码；
- 参数格式错误，例如项目名称太长；

当必需的请求参数缺失时，服务器返回结果如下：

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "error": {
    "author": [{
      "code": 1,
      "message": "not given"
    }],
    "title": [{
      "code": 1,
      "message": "not given"
    }]
  }
}
```

当参数格式错误时，服务器返回结果如下：

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "error": {
    "bio": [{
      "code": 15,
      "message": "is too long (maximum is 255 characters)"
    }]
  }
}
```

综上，API返回到错误报告数据格式可统一为：

```
{
  "error": {
    "<property-name>": [
      {
```

```
        "code": "<error-code>",
        "message": "<error-message>"
    }, {
        "code": "<error-code>",
        "message": "<error-message>"
    },
    ...
],
"<embed-entity>": {
    "<property-name>": [
        {
            "code": "<error-code>",
            "message": "<error-message>"
        }, {
            "code": "<error-code>",
            "message": "<error-message>"
        },
        ...
    ],
    ...
},
...
}
```

Parameters :

- embed-entity - 参数中的复杂对象；
- property-name - 参数名称；
- code - 错误类型；
- message - 对于错误的具有可读性到描述；

其中，错误类型Code定义如下：

Code	描述
0	未知错误
1	必需参数缺失
2	参数格式错误
3	增加中。。。

# API

---

这里只针对《BIMT写作指导WORD插件PRD文档》中标明优先级为"高"的功能进行后台RESTful Web Service的设计。

## 用户相关

---

### 用户登入

```
POST <baseUrl>/users/login
```

Parameters:

- username (required) - 用户登录名
- password (required) - 用户密码
- timestamp (required) - 客户端时间戳

当用户登录成功后，服务器将返回一个32位的Key。

```
{  
  "WAKey": "e57941ff9000aedb44eb2fa13f6e3e3c"  
}
```

在登录后的每次通信中，客户端都需要将这个Key放到HTTP请求头中。

```
"BIMT-WA-KEY": "<WAKey>"
```

### 用户登出

---

```
POST <baseUrl>/users/logout
```

Parameters:

- timestamp (required) - 客户端时间戳

## 获取用户信息

```
GET <baseUrl>/users/me
```

Parameters:

- 无参数

服务端返回的数据如下：

```
{
  "nickname": "oscer",
  "userLevel": 4
}
```

Parameters:

- nickname - 用户昵称
- userLevel - 用户会员等级。当前用户等级从1 ~ 9，值越大用户等级越高

# 搜索相关

## 关键词搜索

通过关键词进行文献搜索。在使用关键词搜索时，服务器会返回一个搜索索引给客户端。这样在搜索条件不变的情况下，客户端可直接通过该搜索索引获取下一页数据。由于服务器对客户端调用关键词搜索接口频率有所限制，所以一般情况下，首先进行关键词搜索，在搜索条件不变到情况下，应继续调用搜索索引检索接口获取后面几页到数据。

```
POST <baseUrl>/documents/search
{
  "conditions": [{
    "oper": "AND",
    "key": "Bimt",
    "type": "Author"
  }, ...],
  "paging": {
    "page": 0,
    "size": 10
  },
  "sorting": [{
    "property": "Time",
    "direction": "DESC"
  }, ...]
}
```

Parameters :

- conditions (required) - 搜索条件数组
  - oper (optional) - 逻辑关系，与上一个条件的逻辑关系。值可为AND / OR / NOT，默认值为AND
  - key (required) - 条件关键词
  - type (optional) - 搜索的字段类型
- paging (optional) - 分页参数
  - page (optional) - 需要返回的页码，默认为1，即第一页。当值大于总页数时，返回最后一页
  - size (optional) - 分页大小，默认值为10，即每页10条记录。取值范围为(1,100]
- sorting (optional) - 排序参数，数组中的先后顺序影响其排序顺序

- property (required) - 进行排序到字段名称
- direction (optional) - 排序方向，值可为ASC / DESC，默认值为ASC

搜索的字段类型值参考下表：

下表中带星号的字段为可排序字段

值	描述
*Topic	主题（默认项）
*Keyword	关键词
*Author	作者
Abstract	摘要
*Organization	作者单位
*Time	日期
*DOI	DOI
*PMID	PMID
*PublishSource	期刊名称
PublishIssue	期刊刊期

服务端返回数据如下：

```
{
  "meta": {
    "sid": "49ba59abbe56e057",
    "pageNo": 1
    "pageCount": 31
  },
  "list": [{ ... }, ...]
}
```

Parameters:

- meta - 搜索结果元信息

- sid - 搜索结果索引
- pageNo - 当前结果页序号
- pageCount - 搜索结果总页数
- list - 搜索结果内容，其格式参考获取文献小节请求返回的数据。

## 搜索索引检索

通过搜索索引获取文献列表。

```
GET <baseUrl>/documents/search/{sid}?pageNo={pageNo}
```

Parameters:

- sid (required) - 关键词检索返回到搜索结果索引
- pageNo (required) - 关键词检索结果分页的页码

服务器端返回数据参考关键词检索小节。

## 文献推荐

通过一句话作为输入，系统通过算法返回相关度最高的N篇(默认20篇)文献给用户。

```
POST <baseUrl>/documents/analysis/recommandDocs
{
  "words": "公立医院试行IOT模式的实践",
  "docAmount": 20
}
```

Parameters:

- words (required) - 作为输入到那句话
- docAmount (optional) - 需要返回的文献数量，默认为20篇

服务器返回数据如下：

```
{
  "meta": {
    "words": "公立医院试行IOT模式的实践",
    "keywords": ["公立", "医院", "试行", "IOT", "模式", "实践"],
```

```

    "docAmount": 15
  },
  "list": [{ ... }, ...]
}

```

Parameters :

- meta - 匹配结果元数据
  - words - 用户提交到那句话
  - keywords - 从那句话中提取到关键词
  - docAmount - 匹配到的总数
- list - 匹配结果内容，具体参考获取文献小节

## 获取文献

```
GET <baseUrl>/documents/{did}
```

Parameters:

- did (required) - 文献ID

服务器返回数据格式如下：

```

{
  "titles" : [ {
    "cn" : "小儿消化性溃疡合并出血36例的临床护理观察",
    "en" : "Clinical nursing observation of 36 cases of peptic ulcer
complicated with hemorrhage in children"
  } ],
  "shortTitle" : null,
  "otherTitle" : null,
  "topics" : null,
  "keywords" : {
    "cn" : [ "消化", "内科", "溃疡", "护理" ],
    "en" : [ "Digestion", "internal medicine", "ulcer", "nursing" ]
  },
  "authors" : {
    "names" : [ {
      "last" : null,
      "fore" : null,
      "full" : "马姝艳"
    }, {

```



```

    "last" : null,
    "fore" : null,
    "full" : "刘晓红"
  } ],
  "organizations" : [ "一汽总医院儿科病房/吉林大学第四医院, 长春, 130011",
    "一汽总医院儿科病房/吉林大学第四医院, 长春, 130011" ],
  "addresses" : [ ]
},
"publishInfo" : {
  "periodicalInfo" : {
    "name" : {
      "cn" : "中国实用医药",
      "en" : "CHINA PRACTICAL MEDICAL"
    },
    "place" : null,
    "press" : null,
    "originalPress" : null,
    "reprintEdition" : null,
    "ISBN/ISSN" : null
  },
  "publishYear" : 2012,
  "volumeCount" : null,
  "volumeInfo" : "7卷",
  "issueInfo" : "36期",
  "column" : null,
  "pageRange" : null,
  "wordsCount" : null,
  "price" : null
},
"abstracts" : {
  "cn" : "目的 分析患儿消化性溃疡护理及疗效.方法 选取2011年3月至2012年7月
    我院36例消化性溃疡的患儿进行护理.结果 以上36例患儿均临床治愈,住院时间4 d~20
    d,平均8.5 d.其中,经内科保守治疗治愈者20例,转外科手术治疗者16例.结论 婴幼儿多为
    急性、继发性溃疡,年长儿多为慢性、原发性溃疡.通过护理,疗效显著,值得推广.",
  "en" : null
},
"extraInfo" : {
  "date" : null,
  "fundProject" : null,
  "nation" : null,
  "includedRange" : null,
  "auxiliaryAuthors" : null,
  "subjectClassification" : null,
  "type" : null,
  "acquisitionNumber" : null,
  "commentItems" : null,
  "referenceDocumentsCount" : null,
  "bibliographyType" : null,
  "bibTex" : null,
  "factor" : null,

```

```

"lang" : null,
"pics" : null,
"inspectionDate" : null,
"email" : null,
"createDate" : null,
"citationCount" : null,
"titleEntry" : null,
"referenceList" : null,
"comments" : null,
"acquisitionBookNumber" : null,
"modifyDate" : null,
"version" : null,
"referenceDocumentsList" : null,
"tags" : null,
"catagory" : null,
"displayDate" : null,
"clazzId" : null,
"topic" : null,
"dataProvider" : null,
"PMID" : null,
"localPath" : null,
"doi" : "10.3969/j.issn.1673-7555.2012.36.153"
},
"did" :
"CFD54857EB035C025D11E482A5A6428C6301282EFAC985BE53B51C7C105D08A7"
}

```

#### Parameters :

- did - 文献标识ID，可通过该ID直接获取文献内容
- titles - 文献标题，当前包含中文 / 英文标题，依次为第一/二/三标题
- shortTitle - 简短标题
- otherTitle - 其它标题
- topics - 主题词
- keywords - 文献关键词，当前包含中文 / 英文关键词
- authors - 文献作者信息列表
  - names 作者姓名列表
    - last - 姓
    - fore - 名
    - full - 全称
  - organizations - 作者单位列表

- addresses - 作者地址列表
- publishInfo - 出版信息
  - periodicalInfo - 期刊信息
    - name - 期刊名称，当前包含中文 / 英文名称
    - place - 出版地
    - press - 出版社
    - originalPress - 原出版社
    - reprintEdition - 重印版次
    - ISBN/ISSN - ISBN/ISSN号
  - publishYear - 出版年
  - volumeCount - 出版卷数
  - volumeInfo - 出版卷
  - issueInfo - 出版期
  - column - 栏目
  - pageRange - 页码
  - wordsCount - 字数
  - price - 价格
- abstracts - 论文摘要，当前包含中文 / 英文摘要
- extraInfo - 额外信息，主要包含当前需要加入但不明其义字段，这些字段以后可能会有所调整
  - bibliographyType - 题录类型
  - fundProject - 基金项目
  - type - 类型
  - subjectClassification - 学科分类
  - category - 分类
  - topic - 主题
  - titleEntry - 标题项
  - tags - 标签
  - bibTex - BibTex关键字
  - auxiliaryAuthors - 辅助作者
  - commentItems - 评述项
  - factor - 影响因子
  - includedRange - 收录范围
  - comments - 注释
  - pics - 图片

- dataProvider - 数据库提供者
- lang - 语言
- nation - 国别
- citationCount - 引用次数
- referenceDocumentsCount - 引用参考文献数
- referenceDocumentsList - 引用参考文献
- version - 版本
- doi - DOI
- acquisitionNumber - 获取号
- acquisitionBookNumber - 索书号
- displayDate - 显示日期
- date - 日期
- inspectionDate - 查阅日期
- modifyDate - 编辑日期
- createDate - 创建日期
- localPath - 文献路径
- email - Email
- PMID - PMID
- referenceList - 引用列表
- clazzId - ClassID

# 格式相关

## 格式列表

```
GET <baseUrl>/templates
```

Parameters:

- 无参数

服务器返回数据如下：

```
{
  "meta": {
    "count": 21
  },
  "list": [{
    "tid": "a0b923820dcc509a",
    "name": "实验室",
    "type": "SelfDefined",
    "refJournal": "Springer International Publishing",
    "refJournalId": "9d4c2f636f067f89"
  }, ...]
}
```

Parameters:

- meta - 样式列表元数据
  - count - 返回到样式数量
- list - 样式列表，其内容具体意义查考获取格式小节

## 获取格式

```
GET <baseUrl>/templates/{tid}
```

Parameters:

- tid - 样式ID

服务器返回数据格式如下：

```
{
  "tid": "a0b923820dcc509a",
  "name": "实验室",
  "type": "SelfDefined",
  "refJournal": "Springer International Publishing",
  "refJournalId": "9d4c2f636f067f89",
  "descptn": "在学校实验室时投稿所用格式",
  "modifiedTime": 1459330469,
  "createdTime": 1456798459,
  "formats": {
    ...
  }
}
```

Parameters:

- tid - 样式ID，可以通过这个ID获取样式到详细数据
- name - 样式名称，如果为标准格式，则为所属期刊的名称；如果为用户自定义格式，则为所属期刊名称+用户自定义标签
- type - 样式类型，值可为标准格式(Standard) / 用户自定义格式(SelfDefined)
- refJournal - 用户自定义格式才有，该样式参考的标准格式所属期刊的名称
- refJournalId - 用户自定义格式才有，该样式参考的标准格式的ID
- descptn - 关于该样式到描述信息
- modifiedTime - 样式修改时间
- createdTime - 样式创建时间
- formats - 样式具体内容，由客户端进行定义

## 增加格式

```
POST <baseUrl>/templates
{
  "name": "实验室",
  "refJournalId": "9d4c2f636f067f89",
  "descptn": "在学校实验室时投稿所用格式",
  "formats": {
```

```

    ...
  }
}

```

Parameters:

- 参考获取格式小节

如果创建成功，服务器返回状态码201 Created。如果发生错误，服务器返回状态码405 Method Not Allowed，并返回错误信息

## 更新格式

```

PUT <baseUrl>/templates/{id}
{
  "name": "实验室",
  "descptn": "在学校实验室时投稿所用格式",
  "formats": {
    ...
  }
}

```

Parameters:

- 参考获取格式小节

如果更新成功，服务器返回状态码200 OK。如果发生错误，服务器返回状态码405 Method Not Allowed，并返回错误信息