[1. 创建内插图片(articleImages)](#_Toc18472)

[2. 删除内插图片](#_Toc11892)

[3. 创建新文档(articles)](#_Toc30619)

[4. 更新文档](#_Toc28532)

[5. 创建新评论](#_Toc30637)

[6. 读取文档（articles，需要和）](#_Toc4773)

[7. 踩/赞文档（）](#_Toc24067)

# 1. 创建内插图片(articleImages)

**url:** http://domain/wdtp POST

**输入参数：**

Values:{

recordInfo:{

Name:{value:’aaaa’},

hashName:{value:’bbbb’},

articleId:{value: objectId},

},

Method:0,

}

**步骤**

1 检查格式和数据

2 对字段name使用**DOMPurify**进行XSS检测

3 检查articleId是否存在

4 检查article的author是否和当前登录的用户id一致

5 文件是否为允许的图片格式

6 文件的size是否允许

7 从预定义的目录中选择合适的路径，保存图片到folderId，然后见数据插入db

8 recInof添加folderId，authorId，cDate

# 2. 删除内插图片

输入参数？？？

步骤：？？？？

# 3. 创建新文档(articles)

**url:** [http://domain/wd/](http://domain/wd/create) POST

**输入参数（**都是预设参数， 无输入参数，都直接由server生成**）**

Values:{

Method:0

}

实际在server端产生的数据如下：

{

Name:’**新建文档**’,

Status: ‘ **editing** ‘, enum: editing/finish，编辑中，编辑完

AuthorId: objectId, 从登录用户信息中获得

FolderId: default 新建文档默认的目录

cDate: date.now()

}

**步骤**

1 检查是否登录且有创建文档的权限

2 自动生成一个新建文档，并记录到db

3 创建一个新的likeUnLike记录，插入db

# 4. 更新文档

对于用户，其实就是新建文档

**url:** [http://domain/wd/](http://domain/wd/update) POST

**输入参数**

Values:{

“recordInfo”:{

Name:{value:aaaa}

Status: {value: editing/finish } enum: editing/finish，编辑中，编辑完

~~Author: {valued: objectId} 谁创建了目录~~

FolderId: {valued: objectId} ; 放在哪个目录

pureContent:{value: bbbb}

htmlContent: {value: cccc}

category: {value:objectId}

Tags:{value: [‘aaaa’,’bbbb’,’cccc’,’dddd’,’eeee’]}

~~ArticleImages: {value:[objectId1, objectId2 …. objectId5]} optional;0~5 文档内插图片，最多5幅~~

~~ArticleAttachments:[objectId]}~~

~~“curColl”:”articles”~~

},

Method:2

}

1 检查格式和数据

2 对字段field/pureContent/htmlContent，使用**DOMPurify**进行XSS检测

3 检查当前用户是否为field中的author

4 检查status是否为enum中的一员（author字段为required，必定存在）

5 检查folderId

5.1 是否存在

5.2 是否为author所有

6 检查category是否在db中存在（category是required，必定存在）

7 如果tags存在，遍历所有tag，是否在coll tags中存在，不存在，在coll tags中创建新tag，并替换成objectId；如果tag存在，获得objectId，并替换

8 如果articleImages存在，遍历field articleImages，检查每个objectId是否已经在coll articleImages中存在了（image在编辑文档时，直接上传了）

# 5. 创建新评论

articleComments，需要和impeachReasult结合，决定是否可以创建

**url:** <http://domain/wdpl/update> POST

**输入参数**

values:{

“recInfo”:{

authorId:{value:objectId},

articleId:{value:objectId},

content:{value:”aaaa”}

},

~~“curColl”:articleComments~~

}

**步骤**

1. 检查用户是否可以发言（读取impeachReasult）

2. 检查文档是否允许发言（？？默认是可以发言）

3. 对输入的值，使用DOMPurify进行XSS检测

4. 数据存入db的articleComments中

5. comment的objectId加入到articles的articleComments字段中

# 6. 读取文档（articles，需要和）

**url:** [http://domain/wd/](http://domain/wdpl/search) POST

**输入参数**

Values:{

Method:1

}

**步骤：**

1. 根据用户sessionId，已经文档的status（是否为finish），判断是否可以读取文档
2. 同时对表read\_article进行操作，插入一个新的记录
3. 并更新article的字段readNum，自增1

# 7. 踩/赞文档（）

**url:** [http://domain/wdcz](http://domain/wdpl/search) POST

**无输入参数**

values:{

“recorderInfo”:{

articleId: article\_objectId

like: boolean

},

~~“curColl”:articleComments~~

}

**步骤：**

1. 根据用户sessionId，以及文档的status（是否为finish），判断是否可以踩/赞文档。
2. 根据like，写入coll Like\_dislike
3. 写完之后，在coll Like\_dislike\_counter对应counter+1