# 图片

## GET

参数：

{

PageNumber: int, =1

PageSize: int, =SERVER\_PAGE\_SIZE

IsOrderByAddTimeDesc bool, =TRUE

LimitLabels:

[

string

]=NULL

}

说明：

返回图片集合。

如果用户未指定页大小，则使用服务器页大小，默认为第1页（首页）。默认采用更新日期逆序排序。如果指定限制标签，这返回所有包含这些标签的图片。

返回：

Head: NextUrl:string

[

{

Id: int,

CreatedTime: string,

Path: string,

Height: string,

Width: string,

Label: [

{

Name: string,

Color: int,

Num: int

}

]

}

]

## GET {$id}

参数：

说明：

返回图片。

返回：

{

Id: int,

CreatedTime: string,

Path: string,

Height: string,

Width: string,

Label: [

{

Name: string,

Color: int,

Num: int

}

]

}

## Post

参数：

{

Paths:[

String

]

ImageFiles:[

file

]

}

说明：

添加图片。

将图片添加到后台图片添加队列中。

本地添加时，只需要指定path即可，远程上传时，只需要指定ImageFiles字段。

获取图片的尺寸和MD5码。添加图片的时候，先扫描MD5码，判断是否已经存在，如果已经存在则指明冲突图片并标记冲突。否则存放到图片库目录下，按照相应的添加日期的年月的子文件夹文件夹，图片名称为MD5码，并添加到数据库。

返回：

203

## Delete {$id}

参数：

说明：

删除图片。

从数据库删除图片，并且从本地删除。如果删除以后文件夹为空，则也删除。如果在缓冲中也存在的话，从缓冲中也删除。

返回：

# 缩略图

## Get {$path}

参数：

说明：

返回图片内容。

如果内存或者固态磁盘缓冲中存在同样的图片，如果宽度等于需要的宽度，则直接取出；如果宽度大于需要的宽度，取出并压缩到指定宽度，输出结束后更新；如果宽度小于需要的宽度或者未命中缓冲，则从磁盘中读取源文件，输出结束以后更新。根据客户端是否判断是否要输出webp格式。输出内容如果需要更新缓冲，则回写。

返回：

图片

# 后台任务

## Get {Add}

参数：

说明：

返回图片添加进度。

返回：

{

CurrentValue float,

Informaiont string,

TaskProgressStatus enum(Running, Finish)

PictureResults:

[

{

Id: int,

Path: string,

Height: string,

Width: string,

Result enum(Pending, Processing, Done, Conflict),

ConflictPictureId int

}

]

}

## Get {Import}

参数：

说明：

返回图片库导入进度。

返回：

{

CurrentValue float,

Informaiont string,

TaskProgressStatus enum(Running, Finish)

LabelResults:

[

{

enum(Pending, Processing, Done, DoNothing, Updated)

}

]

PictureResults:

[

{

Id: int,

CreatedTime: string,

Path: string,

Height: string,

Width: string,

Label: [

{

Name: string,

Color: int,

Num: int

}

]

Result enum(Pending, Processing, Done, Conflict),

ConflictPictureId int

}

]

}

## Get {Output}

参数：

说明：

返回图片库导出进度。

返回：

{

CurrentValue: float,

Informaiont: string,

TaskProgressStatus: enum(Running, Finish),

ResultUrl: string // 完成以后才有

}

# 图片库

## Get

参数：

{

Ids:

[

int

] =NULL

}

说明：

导出图片库。

如果未指定ID，则默认导出所有的图片。导出任务交给后台处理。

选择所有的图片以及其包含的标签和加入包含版本号，导出为json。格式为：

{

Pictures:

{

CreatedTime: string,

Path: string,

Height: string,

Width: string,

MD5: string,

PriHash: string(Base64),

Labels:

[

string

]

}

Labels:

{

Name: string,

Color: int

}

Version: string

}

并且复制所有的图片。压缩为zip，保存到static目录下，名字为output.bnstore。在服务器启动的时候，删除上一次的导出。

返回：

203 Created

## Post

参数：

{

Store: file

}

说明：

导入图片库。

添加到后台线程处理。

解压文件到临时文件夹。

首先尝试导入标签，如果遇到重复的，则更新。

其次导入图片。判断是否已经存在相同图片，如果已经存在则指明冲突图片并标记冲突。否则复制到图片库目录下相应的添加日期的年月子文件夹文件夹，并和标签一起添加到数据库。

返回：

203 Created

# 相似图片

## Get

参数：

说明：

查找相似图片。

添加到后台线程处理。

遍历所有的图片，如果PriHash为空，则读取文件写入PriHash。

遍历每个文件，查询数据库中是否存在和其相似度超过阀值的图片。

返回：

203 Created

## Get {Result}

参数：

说明：

查找相似图片进度结果。

返回：

{

CurrentValue float,

Informaiont string,

TaskProgressStatus enum(Running, Finish)

Groups:

[

[

int // 图片ID

]

]

}

# 标签

## Get

参数：

{

PageNumber: int, =0

PageSize: int, =SERVER\_PAGE\_SIZE

Name: string =NULL

}

说明：

标签列表。

默认为Num逆序排序。如果指定名称，则返回所有标签名称中包含该名称的所有标签。

返回：

Head: NextUrl:string

[

{

Name: string,

Color: int,

Num: int

}

]

## Get {$name}

参数：

说明：

获取指定标签。

返回：

{

Name: string,

Color: int,

Num: int

}

## Post

参数：

{

Name: string,

Color: int

}

说明：

添加新标签。

返回：

标签

## Put

参数：

{

Name: string,

Color: int

}

说明：

修改标签。

返回：

标签

## Delete {$name}

参数：

说明：

删除标签。

返回：

# 截图

## Get

参数：

{

Id: int

}

说明：

返回指定截图。

返回：

{

Id: int,

Path: string,

Height: string,

Width: string,

}

## Post

参数：

{

File: file,

WorkspaceName: string

}

说明：

上传截图。

将截图保存到截图文件夹下，以添加年月保存，并添加到数据库。

返回：

{

Id: int,

Path: string,

Height: string,

Width: string,

}

## Delete {$id}

参数：

说明：

删除截图。

删除截图和文件，如果文件夹为空也删除。

返回：

# 工作区

## Get

参数：

说明：

按照序号排序。返回工作区列表。

返回：

[

{

Name: string,

Index: int

}

]

## Get {$name}

参数：

说明：

返回指定工作区。

返回：

{

Name: string,

Index: int,

Pictures:

[

{

Id: int,

Path: string,

Height: int,

Width: int,

Extension: json {location, scale, isfolked, rotation, transparency}

}

],

ScreenShots:

[

{

Id: int,

Path: string,

Height: int,

Width: int,

Extension: json {location, scale, isfolked, rotation, transparency}

}

]

}

## Post

参数：

{

Name: string,

Index: int

}

说明：

创建工作区。

返回：

## Put

参数：

{

Name: string,

Index: int,

Pictures:

{

Id: int,

Extension: json {location, scale, isfolked, rotation, transparency}

}

ScreenShots:

{

Id: int,

Extension: json {location, scale, isfolked, rotation, transparency}

}

}

说明：

更新工作区。

如果Pictures或者ScreenShots为空，则不更新空内容。

返回：

## Delete {$name}

参数：

说明：

删除工作区，以及其包含的截图。

返回：

# ServerInfo

## Get

参数：

说明：

返回服务器信息和配置信息。

返回：

{

Static: json{/\*计算机系统信息、硬件信息、运行时间等不可更改信息\*/}

Config:json{/\*用户自定义配置信息\*/}

}

## Put

参数：

说明：

设置配置信息。

返回：

{/\*用户自定义配置信息\*/}