Gimbal Camera Web Server Interface Documentation V1.0

Table of Contents.

1 Camera File Interface

1.1 Get File Directory List

This interface is used to obtain the list of file-type directories, making it convenient for users to view file contents by category.

- **URL:** /api/v1/getdirectories
- **Method:** GET

Request Parameters

Parameter	Type	Constraint
media_type	int	0: image, 1: video

Response Parameters

The response parameter format is as follows:

```
{
"code": 200, // status code
"data": {}, // data content
"success": true, // whether the request succeeded
"message": "" // if it fails, the error message
}
```

The data parameters are defined as follows:

Parameter	Type	Constraint
media_type	int	0: image, 1: video

directories	[
	{
	"name": "aa",
	"path": "/yyy/aa"
	},
	{
	"name": "bb",
	"path": "/yyy/bb"
	}
	]
		File directory list

Request Example

Request the directory list of all image files.

```
{
"media_type": 0
}
```

Success Response

Condition: the request parameters are valid and user identity verification passes.

Status code: 200 OK

Response example: the image file directory list returned in the response:

```
{
"code": 200,
"data": {
"media_type": 0,
"directories": [
{
"name": "aa",
"path": "photo/aa"
},
{
"name": "bb",
"path": "photo/bb"
}
]
}
"success": true
}
```

Error Response

Condition: the request data is invalid, e.g., the file type is invalid.

Status code: 400 BAD REQUEST

Response example:

```
{
"code": 400,
"message": "Invalid media type",
"success": false
}
```

1.2 Get the Number of Files in a File Directory

The user uses this interface to obtain the number of files under the specified path.

- **URL:** /api/v1/getmediacount
- **Method:** GET

Request Parameters

Parameter	Type	Constraint
media_type	int	0: image, 1: video
path	String	If it is an empty string, the total number of files of the currently requested type is returned; if it is not empty, the number of files under the specified directory is returned.

Response Parameters

The response parameter format is as follows:

```
{
"code": 200, // status code
"data": {}, // data content
"success": true, // whether the request succeeded
"message": "" // if it fails, the error message
}
```

The data parameters are defined as follows:

Parameter	Type	Constraint
media_type	int	0: image, 1: video
count	int	Number of files
path	int	File directory path

Request Example

Request the number of all images.

```
{
"media_type": 0,
"path": ""
}
```

Request the number of images under the specified path.

```
{
"media_type": 0,
"path": "/photo/aa"
}
```

Success Response

Condition: the request parameters are valid.

Status code: 200 OK

Response example: the number of image files under the 'photo/aa' directory returned in the response:

```
{
"code": 200,
"data": {
"media_type": 0,
"count": 20,
"path": "/photo/aa"
},
"success": true
}
```

Error Response

Condition: the request data is invalid, e.g., the file type is invalid or the file path does not exist.

Status code: 400 BAD REQUEST

Response example:

```
{
"code": 400,
"message": "Invalid media type",
"success": false
}
```

1.3 Get File List

Authorized users are allowed to obtain the file list through this interface.

- **URL:** /api/v1/getmedialist
- **Method:** GET

Request Parameters

Parameter	Type	Constraint
media_type	int	0: image, 1: video
path	String	Empty string: the file list contains all files of the current type. Non-empty string: the file list contains the files under the corresponding path.
start	int	The starting index of the file list
count	int	The number of files in the file list. When start + count is greater than the number of files in the file list, the list from start to the end is returned.

Response Parameters

The response parameter format is as follows:

```
{
"code": 200, // status code
"data": {}, // data content
"success": true, // whether the request succeeded
"message": "" // if it fails, the error message
}
```

The data parameters are defined as follows:

Parameter	Type	Constraint
media_type	int	0: image, 1: video
path	String	The requested path

list	[
	{
	"name": "aa.jpg",
	"url": "http://xxx/yyy/aa.jpg"
	},
	{
	"name": "bb.jpg",
	"url": "http://xxx/yyy/bb.jpg"
	}
	]
		File list

Request Example

Request the image list under the photo/20230630 directory.

```
{
"media_type": 0,
"path": "photo/20230630",
"start": 0,
"count": 10
}
```

Success Response

Condition: the request parameters are valid and user identity verification passes.

Status code: 200 OK

Response example: the image list under the 'photo/20230630' directory returned in the response:

```
{
"code": 200,
"data":{
"media_type": 0,
"path": "photo/20230630",
"list": [
{
"name": "aa.jpg",
"url": "http://xxx/yy/aa.jpg"
},
{
"name": "bb.jpg",
"url": "http://xxx/yy/bb.jpg"
},
...
],
},
"success": true
}
```

Error Response

Condition: the request data is invalid, e.g., the file type is invalid, the file path does not exist, the start index exceeds the maximum value, etc.

Status code: 400 BAD REQUEST

Response example:

```
{
"code": 400,
"message": "path not exist",
"success": false
}
```

Notes

If path is not empty and both start and end are -1, the list of all files of the specified type under path is returned.

1.4 Delete Folders

Authorized users are allowed to delete folders through this interface.

- **URL:** /api/v1/deletemediadirectories
- **Method:** POST

Request Parameters

Parameter	Type	Constraint
media_type	int	0: image, 1: video
paths	String array	Array of folder paths

Response Parameters

The response parameter format is as follows:

```
{
"code": 200, // status code
"data": {}, // data content
"success": true, // whether the request succeeded
"message": "" // if it fails, the error message
}
```

The data parameters are defined as follows:

Parameter	Type	Constraint
media_type	int	0: image, 1: video
success	bool	true: deletion succeeded; false: deletion failed

Request Example

Delete the photo/20230630 directory.

```
{
"media_type": 0,
"paths": ["photo/20230630"]
}
```

Success Response

Condition: the request parameters are valid and user identity verification passes.

Status code: 200 OK

Response example: the result returned after successful deletion:

```
{
"code": 200,
"data":{
"media_type": 0,
"success": true
},
"success": true
}
```

Error Response

Condition: the request data is invalid, e.g., the file type is invalid or the file path does not exist.

Status code: 400 BAD REQUEST

Response example:

```
{
"code": 400,
"message": "path not exist",
"success": false
}
```

1.5 Delete File List

Authorized users are allowed to delete the file list through this interface.

- **URL:** /api/v1/deletemedialist
- **Method:** POST

Request Parameters

Parameter	Type	Constraint
media_type	int	0: image, 1: video
paths	String array	Array of file paths

Response Parameters

The response parameter format is as follows:

```
{
"code": 200, // status code
"data": {}, // data content
"success": true, // whether the request succeeded
"message": "" // if it fails, the error message
}
```

The data parameters are defined as follows:

Parameter	Type	Constraint
media_type	int	0: image, 1: video
success	bool	true: deletion succeeded; false: deletion failed

Request Example

Delete http://xxx/yy/aa.jpg, http://xxx/yy/bb.jpg, http://xxx/yy/cc.jpg.

```
{
"media_type": 0,
"paths": ["http://xxx/yy/aa.jpg","http://xxx/yy/bb.jpg","http://xxx/yy/cc.jpg"]
}
```

Success Response

Condition: the request parameters are valid and user identity verification passes.

Status code: 200 OK

Response example: the result returned after successful deletion:

```
{
"code": 200,
"data":{
"media_type": 0,
"success": true,
},
"success": true
}
```

Error Response

Condition: the request data is invalid, e.g., the file type is invalid or the file path does not exist.

Status code: 400 BAD REQUEST

Response example:

```
{
"code": 400,
"message": "path not exist",
"success": false
}
```