

IntakeQ Files API

Please Note: The standard PracticeQ subscription includes 10 API requests/minute, max 500/day. Click here (#) to learn how to increase your threshold and unlock the power of PracticeQ's flexible API's.

Overview

The IntakeQ Files API is composed of HTTP endpoints that can be reached through a REST API.

For the Intake Forms API, please refer to this article ([//support.intakeq.com/article/31-intakeq-api](https://support.intakeq.com/article/31-intakeq-api)).

Getting Started

To get started, you first need to enable API access (navigate to More > Settings > Integrations > Developer API). That's also where you'll find your API key, used to authenticate your HTTP calls.

Only the main account owner has access to the API tab.

Never expose your API key in plain text (config files, source control, etc.).

Consider setting an IP Allow-List as explained here ([//support.intakeq.com/article/200-restricting-access-using-ip-address#api-ip-restriction](https://support.intakeq.com/article/200-restricting-access-using-ip-address#api-ip-restriction)).

Base URL

All endpoints are located under the following URL: <https://intakeq.com/api/v1/>
[\(https://intakeq.com/api/v1/\)](https://intakeq.com/api/v1/)

An example of a typical API call would look like this:

```
[GET] https://intakeq.com/api/v1/files?clientId=123
```

Authentication

Every HTTP request needs to contain your API key embedded in an authentication header named X-Auth-Key.

```
X-Auth-Key:xxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Get Client's Files

This method returns a list of files.

```
[GET] https://intakeq.com/api/v1/files?clientId=123
```

This method accepts the following query string parameters:

- clientId (mandatory) – An integer used to identify the client

If clientId parameter is not passed, the method will return a bad request error.

This method returns a JSON object representing an array of files. This is what the response looks like:

```
[  
  {  
    "Id": "xxx",  
    "DateCreated": 1618939932507, //Unix timestamp  
    "Size": 17432,  
    "FileName": "text.txt",  
    "ContentType": "text/plain",  
    "FolderId": null  
  },  
  {  
    "Id": "yyy",  
    "DateCreated": 1619063201638, //Unix timestamp  
    "Size": 21091,  
    "FileName": "insurance card.png",  
    "ContentType": "image/png"  
    "FolderId": "xxxxx"  
  }  
]
```

Get Folders

This method returns a list of folders

```
[GET] https://intakeq.com/api/v1/folders
```

This method returns a JSON object representing an array of files. This is what the response looks like:

```
[  
  {  
    "Id": "xxx",  
    "Name": "Folder1"  
  },  
  {  
    "Id": "yyy",  
    "Name": "Folder2"  
  }  
]
```

Download File

This method allows you to download a file.

```
[GET] https://intakeq.com/api/v1/files/[fileId]
```

- fileId (mandatory) – A string used to identify the file

Upload File

This method allows you to upload a file

```
[POST] https://intakeq.com/api/v1/files/[clientId]
```

- clientId (mandatory) – A string used to identify the client. The file will be associated with that client.

This request needs to contain a file stream and the headers properly set. Below is an example of how to do it.

```
public async Task UploadFile(string clientId, byte[] fileData, string fileName, string contentType)
{
    using (HttpClient client = new HttpClient())
    {
        client.DefaultRequestHeaders.Add("X-Auth-Key", _apiKey);

        using (var content = new MultipartFormDataContent())
        {
            using (var streamContent = new StreamContent(new MemoryStream(fileData)))
            {
                streamContent.Headers.Add("Content-Type", "application/octet-stream");
                streamContent.Headers.Add("Content-Disposition", "form-data; name=\"file\"; filename=\"" + fileName + "\"");
                streamContent.Headers.ContentType.MediaType = contentType;

                content.Add(streamContent, "file", fileName);
            }

            HttpResponseMessage response = await client.PostAsync($"https://intakeq.com/api/v1/files/{clientId}", content);
        }
    }
}
```

Delete File

This method allows you to delete a file

```
[DELETE] https://intakeq.com/api/v1/files/[ fileId ]
```

- fileId (mandatory) – A string used to identify the file