IntakeQ Notes API

> **Please Note:** The standard PracticeQ subscription includes 10 API requests/minute, max 500/day. Click here (//support.intakeq.com/article/711-api-rate-limits) to learn how to increase your threshold and unlock the power of PracticeQ's flexible API's.

Overview

The IntakeQ API is composed of HTTP endpoints that can be reached through a REST API. Additionally, you can setup a webhook that's fired when a staff member locks a note; giving you the opportunity to download it in realtime and integrate with 3rd party software.

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).

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/notes/[note-id]
```

Authentication

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

```
X-Auth-Key:xxxxxxxxxxxxxxxxxxxxxxxxx
```

Jump to a Method

- Query Treatment Notes (#query-notes)
- Download Note PDF (#download-note)
- Get Full Treatment Note (#full-note)
- Note Locked Webhook (#webhook)

Query Treatment Notes

Use this method to query treatment note summaries. The result set does not contain all the contents of the notes, but only their basic information (id, status, client info).

```
[GET]
/notes/summary?client=[searchString]&clientId=[number]&status=[number]&startDate=[yyyy-MM-dd]&
endDate=[yyyy-MM-dd]&page=[pageNumber]&updatedSince=[yyyy-MM-dd]&deletedOnly=[bool]
```

This method accepts the following query string parameters:

- **client (optional)** – A string used to search the client by name or email. Partial matches will be respected, so a search for "Paul" will return all notes for clients with Paul in their names. Likewise, a search for "paul.smith@gmail.com" will return all notes for that specific client.
- **clientId (optional)** – An integer used to search by the client ID. Only exact matches will be returned.
- **status (optional)** – An integer representing the note lock status. Use 1 to only get locked notes and 2 to get unlocked notes.
- **startDate (optional)** – Return only notes that were created after the specified date. Use the following date format: yyyy-MM-dd (ex.: 2016-08-21)

- **endDate (optional)** – Return only notes that were created before the specified date. Use the following date format: yyyy-MM-dd (ex.: 2016-08-21)
- **page (optional)** – This method returns a maximum of 100 records. Use the page parameter to implement paging from your end. Use 1 for page 1, 2 for page 2, etc.
- **updatedSince (optional)** – Returns only notes that have been changed after a given date.
- **deletedOnly (optional)** – Returns only notes that have been deleted. Deleted notes can be retrieved up to 10 days after they have been deleted. Notes that have been deleted more than 10 days prior cannot be retrieved anymore.

If no query string parameter is passed, the method will return the last 100 notes.

This method returns a JSON object representing an array of treatment notes.

```
[{
   Id: "00000000-0000-0000-0000-000000000000", //GUID
   ClientName: "test",
   ClientEmail: "test@email.com",
   ClientId: 9999,
   Status: "locked",
   Date: 1458526480368, //Unix timestamp,
   NoteName: "Client Follow Up",
   PractitionerEmail: "test@email.com",
   PractitionerName: "FirstName LastName",
   PractitionerId: "12345"
},...
]
```

Let’s look at each property individually:

Field	Explanation
Id	The ID of the treatment note. You can use this to get the full note.
ClientName	The name of the client associated with the note.
ClientEmail	The email of the client associated with the note.
ClientId	The auto-generated number ID assigned to the client.

Status	Possible values: **locked** – the note is in draft state. **unlocked –** the note has been locked.
Date	The date of the note in Unix timestamp.
NoteName	The title of the note. Usually the name of the note form template.
PractitionerEmail	The email of the practitioner associated with the note.
PractitionerId	The auto-generated practitioner ID (string).

> This method returns a maximum of 100 records. If needed, use the page query string parameter to implement paging from your end.

Download Note PDF

Use this method to download the note as a PDF file.

```
[GET]
https://intakeq.com/api/v1/notes/[note-id]/pdf
```

Get Full Treatment Note

Use this method to get a full note in JSON format.

```
[GET] https://intakeq.com/api/v1/notes/[note-id]
```

The full note is very similar to note summary object, except it adds an array of questions.

```
{
   Id: "00000000-0000-0000-0000-000000000000", //GUID
   ClientName: "test",
   ClientEmail: "test@email.com",
   ClientId: 9999,
   Status: "locked",
   Date: 1458526480368, //Unix timestamp,
   NoteName: "Client Follow Up",
   PractitionerEmail: "test@email.com",
   PractitionerName: "FirstName LastName",
   PractitionerId: "12345"
   Questions: [...], //See Question structure below
   AppointmentId: "xxxxxxxx" //only if note is associated with appointment
}
```

The Question Object

Because we support several question types, the question structure is a little more complex. We use a common structure for all questions, but certain properties will be populated and others will be empty, depending on the question type. This makes things simpler when you're deserializing the JSON response using static languages.

Here's an example of a question array:

```
[
   {
       Id: "xxxx-1",
       Text: "Full name",
       Answer: "Dexter Morgan",
       QuestionType: "OpenQuestion",
       Rows: [],
       ColumnNames: [],
       OfficeUse: false,
       OfficeNote: null,
   },
   {
       Id: "xxxx-2",
       Text: "Address",
       Answer: "134 Silk Road",
       QuestionType: "OpenQuestion",
       Rows: [],
       ColumnNames: [],
       OfficeUse: false,
       OfficeNote: null,
   },
   {
       Id: "o5iF-1",
       Text: "Please indicate if you or any family members have ever had any of the following problems. Specify who.",
       Answer: "High Cholesterol",
       QuestionType: "MultipleChoice",
       Rows: [],
       ColumnNames: []
   }, {
       Id: "4dHk-1",
       Text: "Please list your top major health concerns in order of importance, and indicate date of diagnosis (where relevant):",
       Answer: null,
       QuestionType: "Matrix",
       Rows: [
           {
               Text: "1",
               Answers: ["High Cholesterol", "Feb 2016"]
           },
           {
               Text: "2",
               Answers: ["", ""]
```

```
            }
        ],
        ColumnNames: ["Concern", "Date"],
        OfficeUse: false,
        OfficeNote: "Test note from practitioner.",
    }, {
        Id: "efiE-1",
        Text: "Please upload your insurance card.",
        Answer: null,
        QuestionType: "Attachment",
        Attachments: [
                {
                    Id: "352420dfnkdfsd9321j294k",
                    Url: "https://intakeq.com/api/v1/attachments/352420dfnkdfsd9321j294k",
                    ContentType: "image/png",
                    FileName: "insurance.png"
                }
        ],
        Rows: [],
        ColumnNames: [],
        OfficeUse: false,
        OfficeNote: null,
    },
```

Notice that you can use the QuestionType field to figure out which fields will be populated (e.g. the Matrix question type will have Rows and Columns). Or you can use the question Id field to get specific questions.

For questions of the type "Attachment", an array of attachments will be provided (as shown above), and you can use the URL property to download the file. It's important to note that the API key must be included in the header of the download request as well.

Note Locked Webhook

If you populate the Note Webhook URL in the API settings page, we will send the following JSON object in a POST message every time a note is locked by a staff member:

```
{
   NoteId: "00000000-0000-0000-0000-000000000000",
   Type: "Note Locked",
   ClientId: 123
}
```

Once you have the Note ID, you can use it to get the full note (#full-note) or download the PDF (#download-note). The Client ID allows you to use the client API (//support.intakeq.com/article/251-intakeq-client-api) to get the full client object.

Still need help? Contact Us (#)

Last updated on March 20, 2025

RELATED ARTICLES

- IntakeQ Questionnaire API (/article/31-intakeq-api)
- IntakeQ Client API (/article/251-intakeq-client-api)
- IntakeQ Appointments API (/article/204-intakeq-appointments-api)
- IntakeQ Invoice API (/article/385-intakeq-invoice-api)
- IntakeQ Claims API (/article/568-intakeq-claims-api)
- IntakeQ Files API (/article/430-intakeq-files-api)