IntakeQ Client 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 Client 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).

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/)

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

```
[GET] https://intakeq.com/api/v1/clients?search=123
```

Authentication

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

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

Query Clients

This method returns a list of clients.

```
[GET]
/clients?search=[searchString]&page=[pageNumber]&dateCreatedStart=[yyyy-MM-dd]&dateCreatedEnd=[yyyy-MM-dd]&custom.fieldId=value&dateUpdatedStart=[yyyy-MM-dd]&dateUpdatedEnd=[yyyy-MM-dd]&externalClientId=[yourId]&deletedOnly=[bool]
```

This method accepts the following query string parameters:

Example:

```
https://intakeq.com/api/v1/clients?search=0204&IncludeProfile=true
```

- **search (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 intakes for clients with paul in their names. Alternatively, you can provide the numeric Client ID and this method will return an exact match.
- **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.
- **includeProfile (optional)** – Set this to 'true' to get all the fields in the client profile.
- **dateCreatedStart (optional)** – Use this to get only clients created after a certain date. Date format: yyyy-MM-dd (e.g. 2018-03-23).
- **dateCreatedEnd (optional)** – Use this to get only clients created before a certain date. Date format: yyyy-MM-dd (e.g. 2018-03-25).
- **custom.fieldId** - If you need to find a client using a custom field that was created in the client profile settings. Example: if the custom field ID is "123", and the value that you are searching for is "xxx", then the query string should be *custom.123=xxx.*
    - To get the fieldId for custom fields, you can use the Client search API and inspect the results.They are the same for all clients, so you only need to do this once.

- **dateUpdatedStart (optional)** – Use this to get only clients that have been updated after a certain date. Date format: yyyy-MM-dd (e.g. 2018-03-23).

- **dateUpdatedEnd (optional)** – Use this to get only clients that have been updated before a certain date. Date format: yyyy-MM-dd (e.g. 2018-03-25).
- **externalClientId (optional)** – If you have populated the ExternalClientId when saving the client, you can use it here to retrieve it.
- **deletedOnly (optional)** – Returns only clients that have been deleted. Deleted clients can be retrieved up to 10 days after they have been deleted. Clients 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 first 100 clients, sorted alphabetically by name.

This method returns a JSON object representing an array of clients. When the includeProfile field is not set, or set to false, this is what the response looks like:

```
[
        {
              Name:"test",
              Email:"test@email.com",
              Phone:"(222) 222-2222",
              ClientNumber:"32"
        },
        ...
]
```

When the includeProfile field is set to true, this is what the response looks like:

```
[
        {
                "ClientId": 123,
                "Name":"test name",
                "FirstName":"test",
                "LastName":"test",
                "MiddleName":"test",
                "Email":"test@email.com",
                "Phone":"(222) 222-2222",
                "DateOfBirth":1458526480368, //Unix timestamp in milliseconds
                "MaritalStatus":"Married",
                "Gender":"Female",
                "Tags":["tag1","tag2"],
                "Archived":false,
                "HomePhone":"(222) 222-2222",
                "WorkPhone":"(222) 222-2222",
                "MobilePhone":"(222) 222-2222",
                "Address":"105 Test St, City, State ZipCode, Country",
                "UnitNumber":"X",
                "StreetAddress": "105 Test St",
                "City": "City Test",
                "StateShort": "NY",
                "Country": "USA",
                "PostalCode": "12345",
                "AdditionalInformation":"test",
                "BillingType": 2,
                "PrimaryInsuranceCompany":"test",
                "PrimaryInsurancePolicyNumber":"test",
                "PrimaryInsuranceGroupNumber":"test",
                "PrimaryInsuranceHolderName":"test",
                "PrimaryInsuranceRelationship":"test",
                "PrimaryInsuranceHolderDateOfBirth":1458526480368, //Unix timestamp in mi
```

```
lliseconds,
                "SecondaryInsuranceCompany":"test",
                "SecondaryInsurancePolicyNumber":"test",
                "SecondaryInsuranceGroupNumber":null,
                "SecondaryInsuranceHolderName":null,
                "SecondaryInsuranceRelationship":null,
                "SecondaryInsuranceHolderDateOfBirth":1458526480368, //Unix timestamp in
milliseconds,
                "DateCreated":1458526480368, //Unix timestamp in milliseconds,
                "LastActivityDate":1458526480368, //Unix timestamp in milliseconds,
                "LastUpdateDate":1458526480368, //Unix timestamp in milliseconds,
                "StripeCustomerId": null,
                "SquareCustomerId": null,
                "ExternalClientId":"001",
                "CreditBalance": 0.0,
                "LastActivityName": "Created Appt",
                "ParentClientId": null,
                "LinkedClients": [{
                "ClientNumber": 123,
                        "RelationshipType": 1
                        }, {
                "ClientNumber": 211,
                        "RelationshipType": 3
                        }],
                "CustomFields": [
                        {
                                "FieldId": "abcd",
                                "Text": "Emergency Contact Name",
                                "Value": "Jane Doe"
                        },
                        {
                                "FieldId": "dcba",
                                "Text": "First Visit Date",
                                "Value": "2017-03-15"
                        },
                ]
        }
]
```

Note the following response details:

- Client's Billing Type
  - Unknown = 0
  - SelfPay = 1
  - Insurance = 2
- Linked Client Relationship Type
  - None = 0,
  - Parent = 1
  - Child = 2
  - Spouse = 3
  - Sibling = 4
  - Other = 5
  - Partner = 6

Saving Clients

IntakeQ also allows you to add and update clients using the API. To save a client send a POST request to the URL below:

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

The body of the POST request should contain a JSON object in the following format:

```
{
        "ClientId": 123,
        "Name":"test name",
        "FirstName":"test",
        "LastName":"test",
        "MiddleName":"test",
        "Email":"test@email.com",
        "Phone":"(222) 222-2222",
        "DateOfBirth":1458526480368, //Unix timestamp in milliseconds
        "MaritalStatus":"Married",
        "Gender":"Female",
        "Archived":false,
        "HomePhone":"(222) 222-2222",
        "WorkPhone":"(222) 222-2222",
        "MobilePhone":"(222) 222-2222",
        "Address":"105 Test St, City, State ZipCode, Country",
        "UnitNumber":"123",
        "StreetAddress": "105 Test St",
        "City": "City Test",
        "StateShort": "NY",
        "Country": "USA",
        "PostalCode": "12345",
        "PractitionerId": "123456789", //optional
        "AdditionalInformation":"test",
        "PrimaryInsuranceCompany":"test",
        "PrimaryInsurancePolicyNumber":"test",
        "PrimaryInsuranceGroupNumber":"test",
        "PrimaryInsuranceHolderName":"test",
        "PrimaryInsuranceRelationship":"test",
        "PrimaryInsuranceHolderDateOfBirth":1458526480368, //Unix timestamp in milliseconds,
        "SecondaryInsuranceCompany":"test",
        "SecondaryInsurancePolicyNumber":"test",
        "SecondaryInsuranceGroupNumber":null,
        "SecondaryInsuranceHolderName":null,
        "SecondaryInsuranceRelationship":null,
        "SecondaryInsuranceHolderDateOfBirth":1458526480368, //Unix timestamp in milliseconds,
        "DateCreated":1458526480368, //Unix timestamp,
        "LastActivityDate":1458526480368, //Unix timestamp,
"StripeCustomerId": null,
                "SquareCustomerId": null,
"ExternalClientId":"001",
        "CustomFields": [
                {
                        "FieldId": "abcd",
                        "Text": "Emergency Contact Name",
                        "Value": "Jane Doe"
                },
                {
                        "FieldId": "dcba",
                        "Text": "First Visit Date",
                        "Value": "2017-03-15"
                },
        ]
}
```

There are some important things to note regarding the Client Save endpoint:

- If you inform the ClientId, the API will update an existing client and fail if it doesn't find the client.

- If you don't inform the ClientId, the API will still try to find an existing client based on first name and email or first name and phone number.
- When updating an existing client, make sure to GET the client first, make your changes, and POST the client back.
- You can leave the Address field empty and inform each address component separately (StreetAddress, City, StateShort, PostalCode, Country).
- If you want to save custom fields, you need to populate the FieldId and Value fields (the Text field can be ignored). To get the fieldId for the custom fields, you can use the Client search API and inspect the results. They are the same for all clients, so you only need to do this once or until you change the custom fields in the settings.

Add a Tag to a Client

This method allows you to tag a client.

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

The body of the POST request should contain a JSON object in the following format:

```
{
     ClientId: 123,
     Tag: "text"
}
```

Things to note:

- **If the tag doesn't exist in the system, a new tag will be created.**
- **If the client is already tagged with the provided tag, the endpoint will not perform any action and respond with 200 OK.**

Remove a Tag from a Client

This method allows you untag a client.

```
[DELETE] https://intakeq.com/api/v1/clientTags?clientId=[clientId]&tag=[tag text]
```

Example: to remove the tag "veteran" from client 0234, issue an HTTP DELETE to the URL https://intakeq.com/api/v1/clientTags?clientId=234&tag=veteran (https://intakeq.com/api/v1/clientTags?clientId=234&tag=veteran).

If the tag doesn't exist in the client, the endpoint will still return a 200 OK response.

Query Client Diagnoses

This method returns a list of client diagnoses.

```
[GET] /client/[clientId]/diagnoses
```

This method returns a JSON object representing an array of client diagnoses:

```
[
        {
              Code:"1",
              Description:"Alcohol Disorder",
              Date:"2021-12-08T23:00:00Z",
              EndDate: null,
              NoteId:"82328bc2-3ff8-4ea8..." // Treatment Note
        }
]
```

Still need help? Contact Us (#)

Last updated on October 27, 2025

RELATED ARTICLES

- IntakeQ Questionnaire API (/article/31-intakeq-api)
- IntakeQ Appointments API (/article/204-intakeq-appointments-api)
- IntakeQ Notes API (/article/342-intakeq-notes-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)