

# Using Tidepool Data APIs (Read Only)

**Please do not use Production Tidepool APIs during your development. Only use the integration environment APIs, or your local back-end server environment.**

The Integration environment is for development proof of concept tests only. Do not use it for production data.

To use the Integration environment, you will need to create accounts on the Integration environment (see “Account Creation” below) and upload data via the Integration environment (before logging into the Tidepool Uploader, right click and set server to “Integration”). Or you may use the demo/test account provided below.

Once you have tested your software on the Integration environment, let us know so we can review plans with you to move to our Production environment.

Our data APIs will be changing in the future when we add OAuth2, exact date and nature of the change is TBD. We cannot guarantee backwards compatibility.

The [data model](#) represented in the JSON is likely to remain mostly stable.

If you are developing software that uses these APIs, please send us an email so we know who you are! [info@tidepool.org](mailto:info@tidepool.org). Please note that we are currently updating our APIs and authentication mechanism to use OAuth2. Make sure we know who you are so that we can keep you updated on our progress!

To upload using the [Tidepool Uploader](#) to the integration environment: Upon launching the Tidepool Uploader, control-click (right click) on the login screen and change the environment to Integration. If you'd like to upload data via our APIs, please reach out to us. We are very careful to try to keep data uploaded to our database clean, and want to talk through a few things with you.

Also, to ask questions, be sure to join us in our public #opensource Slack Channel here: <http://public-chat.tidepool.org>

## Endpoint URLs

	Integration ("Int")
Account Creation	<a href="https://int-app.tidepool.org/signup?inviteKey=9bZTtr6CczXx127obTMK">https://int-app.tidepool.org/signup?inviteKey=9bZTtr6CczXx127obTMK</a>
API URL	<a href="https://int-api.tidepool.org/">https://int-api.tidepool.org/</a>
Tidepool for Web URL (to visualize data)	<a href="https://int-app.tidepool.org/">https://int-app.tidepool.org/</a>
Test account to use for reading data	<a href="https://int-app.tidepool.org">https://int-app.tidepool.org</a> user: <a href="mailto:demo+intpublicclinic@tidepool.org">demo+intpublicclinic@tidepool.org</a> pw: eureka-charcoal-longbow

## Application Identity headers

Until Tidepool supports developer API keys, we ask that each application identifies itself with the X-Tidepool-Client-Name and X-Tidepool-Client-Version HTTP headers.

X-Tidepool-Client-Name should be a reverse-DNS-style identifier of the format com.yourcompany.productname, or com.repo.username.project. For example, com.github.pazaan.600SeriesAndroidUploader.

X-Tidepool-Client-Version should be a [semver-compliant](#) application version number.

These headers are *not* included in the examples below for brevity.

## Fetching Tidepool Data

The examples below use CURL to fetch data. Translate these to your favorite language.

### Step 1: Fetch the x-tidepool-session-token and userID

First, get your authorization session token **using your clinic/study master account credentials**:

```
curl -i -X POST -u masteremail@domain.com https://int-api.tidepool.org/auth/login  
Password: [your account password]
```

This will return an HTTP response that looks like this:

```
HTTP/1.1 200 OK  
access-control-allow-headers: authorization, content-type, x-tidepool-session-token  
access-control-allow-methods: GET, POST, PUT  
access-control-allow-origin: *  
access-control-expose-headers: x-tidepool-session-token
```

```
{
  "emailVerified": true,
  "emails": ["demo+intpublicclinic@tidepool.org"],
  "roles": ["clinic"],
  "termsAccepted": "2017-08-16T10:30:56-07:00",
  "userid": "4533925fea",
  "username": "demo+intpublicclinic@tidepool.org"
}
```

## Step 2: Fetch the metadata for the associated users or study subjects

This returns all of the users' metadata associated with the given <your-userid>, including what permissions it has upon those associated users. This step is optional. You may only need to do it if you have not previously stored the user ids of the individual patients or study subjects.

Continuing with our example we would use:

which returns (see [Note 1] at end of the doc for a "pretty" version of this JSON):

```
[{"emailVerified":true,"emails":["demo+jill@tidepool.org"],"termsAccepted":"2017-08-03T12:19:54-07:00","userid":"5d509deb6b","username":"demo+jill@tidepool.org","trustorPermissions":{"view":{}},"profile":{"fullName":"Jill Jellyfish","patient":{"birthday":"2000-01-01","diagnosisDate":"2000-01-01","targetDevices":["omnipod","dexcom"],"targetTimezone":"US/Pacific"}}},{ "userid":"0223d994e9","trustorPermissions":{"custodian":{}},"upload":{}},"view":{}},"profile":{"fullName":"Marissa
```

```
Medpumper", "patient": {"birthday": "2000-01-17", "mrn": "123456-mm", "targetDevices": ["carelink"], "targetTimezone": "US/Pacific"}}}]
```

Note that here we see that **there are two kinds of accounts that the study master account has view access for:**

1. The demo+jill@tidepool.org account, userid 5d509deb6b, is a normal personal account, where Jill has clicked "Share" and shared her account with the master account.
2. The Marissa Medpumper account, userid 0223d994e9, is a custodial account that was created in clinic by the master account. Marissa was not invited or has not claimed her personal account, so there is no email associated with the profile.

### Step 3: Fetch device data for the user

```
curl -s -X GET -H "x-tidepool-session-token: <your-session-token>" -H  
"Content-Type: application/json"  
"https://int-api.tidepool.org/data/<subject-userid>"
```

Note that you are using the session token for the master account login. It has permissions to see data for the user or study subject account.

Optionally, you may narrow the query by specifying a type, subtype and/or date query parameter to the API URL(see [github](#) comments for more detail):

- type (optional) : The Tidepool data type to search for. Only data with a type field matching the specified type param will be returned. It can be a single type (e.g. "?type=smbg") or a comma-separated list (e.g. "?type=smbg,cbg"). If is a comma-separated list, then data matching any of the specified types will be returned.
- subType (optional) : The Tidepool data subType to search for. Only data with a subType field matching the specified subType param will be returned. It can be single subType (e.g. "?subType=normal") or a comma-separated list (e.g. "?subType=normal,square"). If is a comma-separated list, then data matching any of the specified subTypes will be returned.
- startDate (optional) : Only data with 'time' field equal to or greater than start date will be returned. Must be in ISO date/time format (e.g. "?startDate=2015-10-10T15:00:00.000Z").
- endDate (optional) : Only data with 'time' field less than to or equal to end date will be returned. Must be in ISO date/time format (e.g. "?endDate=2015-10-10T15:00:00.000Z").

Here is the [Tidepool Data Model](#) documentation for the data that will be returned by this step.

For example, to fetch just the pump settings for Jill Jellyfish (a patient at the Integration Demo Clinic test account), use:

```
curl -s -X GET -H "x-tidepool-session-token: <your-session-token>" -H
"Content-Type: application/json"
"https://int-api.tidepool.org/data/5d509deb6b?type=pumpSettings"
```

Note that all glucose data in Tidepool is stored in mmol/L. To convert to mg/dL, multiply by 18.01559!

See [Note 2] below about Dexcom CGM Data.

### (Optional) Step 4: Fetch the notes for the user or study subject

```
curl -s -X GET -H "x-tidepool-session-token: <your-session-token>" -H
"Content-Type: application/json"
"https://int-api.tidepool.org/message/notes/<subject-userid>"
```

Optionally specify starttime and endtime in ISO date/time format (similar to fetching data, see above) (e.g.

"?starttime=2015-10-10T15:00:00.000Z&endtime=2015-10-11T15:00:00.000Z").

### For Reference

- [Tidepool Data Model](#)
- [Command line data tools](#) for fetching data and storing as XLS. This may be helpful to you as you write your code, or if you wish to see the data in XLS form.

---

### Note 1: "Pretty JSON" version of the output from step 2:

```
[
  {
    "emailVerified": true,
    "emails": [
      "demo+jill@tidepool.org"
    ],
    "profile": {
      "fullName": "Jill Jellyfish",
      "patient": {
        "birthday": "2000-01-01",
        "diagnosisDate": "2000-01-01",
        "targetDevices": [
          "omnipod",
```

```

        "dexcom"
      ],
      "targetTimezone": "US/Pacific"
    }
  },
  "termsAccepted": "2017-08-03T12:19:54-07:00",
  "trustorPermissions": {
    "view": {}
  },
  "userid": "5d509deb6b",
  "username": "demo+jill@tidepool.org"
},
{
  "profile": {
    "fullName": "Marissa Medpumper",
    "patient": {
      "birthday": "2000-01-17",
      "mrn": "123456-mm",
      "targetDevices": [
        "carelink"
      ],
      "targetTimezone": "US/Pacific"
    }
  },
  "trustorPermissions": {
    "custodian": {},
    "upload": {},
    "view": {}
  },
  "userid": "0223d994e9"
}
]

```

---

## Note 2: Dexcom CGM Data

In November 2017, Tidepool added the ability for users to connect to their Dexcom account (via OAuth2) and connect their Tidepool account to their Dexcom account as a mechanism to fetch Dexcom data. This caused the possibility of duplicate or overlapping data if the user is also using (or has used) the Tidepool Mobile or the Tidepool Uploader to upload their Dexcom data.

If the standard GET data Tidepool API is used, then **only Dexcom API** CGM data will be returned **within** this time range and any other CGM data from any other source (Tidepool Uploader, HealthKit via Tidepool Mobile, etc) will **only** be returned **outside** of this time range.

So, you have three time ranges that apply here:

Start Time	End Time	CGM Data Returned
Beginning of Time	Dexcom API Earliest Data Time	All CGM data
Dexcom API Earliest Data Time	Dexcom API Latest Data Time	<b>Only Dexcom API CGM data</b>
Dexcom API Latest Data Time	End of Time	All CGM data

However, if you use the standard GET data Tidepool API, but add the "?dexcom=true" request flag, then it will return all CGM data from all sources regardless of the Dexcom API state.