



1. Striiv API	2
1.1 OAuth Authorization in the Striiv API	2
1.2 Striiv Resource Access API	8
1.2.1 API-Get-Activities	8
1.2.2 API-Get-Activity-Stats	11
1.2.3 API-Get-Alltime-AVG-Activity	14
1.2.4 API-Get-Sleep	16
1.3 Striiv Subscriptions API	20
1.4 API Client Libraries	26
1.5 API Open Questions	26
1.6 Striiv Messaging API	26

# Striiv API

## Welcome to the Striiv API

The Striiv API is meant to allow easy access to a user's data. It is a (mostly) RESTful API that uses OAuth 1.0 for authentication.

## OAuth Authorization in the Striiv API

The Striiv API uses The OAuth 1.0 Protocol [RFC 5849](#) to enable partners to access a Striiv user's content on their behalf (and with their authorization).

### Notes

1. The Base URL for both the API and OAuth endpoints is: <https://striiv-api-prod.appspot.com/api/1>
2. HTTPS is required for all operations.
3. We use the Authorization header to pass OAuth parameters.
4. Currently we offer two signature methods: PLAINTEXT and HMAC-SHA1. We recommend you use HMAC-SHA1.
5. The timestamp parameter is in epoch (UTC), and currently **requests with a timestamp older than 5 min are rejected**.
6. Users will be able to revoke access, which will mark the access token as not approved. This will result in requests made with that access token to be denied.

### Steps

1. **Register your application to obtain an API consumer key and secret.**
2. **Obtain a request token and secret**

### Request

**POST /request\_token** (Full URL [https://striiv-api-prod.appspot.com/api/1/request\\_token](https://striiv-api-prod.appspot.com/api/1/request_token))

Authorization header parameters required in alphabetical order:

<b>oauth_callback</b>	Callback URL. This is the URL a user is redirected to after authorizing a consumer app. This parameter is optional.
<b>oauth_consumer_key</b>	Consumer credential obtained in Step 1.
<b>oauth_nonce</b>	Random, unique string generated by consumer to allow the server to verify uniqueness of request and help prevent replay attacks.
<b>oauth_signature</b>	Signature computed as describe in <a href="#">The OAuth 1.0 Protocol Section 3.4: Signature</a> .
<b>oauth_signature_method</b>	Signature method: HMAC-SHA1 or PLAINTEXT
<b>oauth_timestamp</b>	Timestamp in epoch (UTC). Requests older than 5 min will be rejected.
<b>oauth_version</b>	1.0 or 1.0a. This parameter is optional.

### Response

<b>Status Code</b>	200 OK
<b>Headers</b>	Content-Type: text/html; charset=utf-8

<b>Body</b>	oauth_token_secret=<request_token_secret>&oauth_token=<request_token>
-------------	---

### Example

Request
<pre>POST /api/1/request_token HTTP/1.1 Host: striiv-api-prod.appspot.com Authorization: OAuth oauth_consumer_key="4GVX3d72gsq9RkK5",oauth_signature_method="HMAC-SHA1", oauth_timestamp="1415145585",oauth_nonce="Xx6cPU",oauth_version="1.0a", oauth_callback="www.example.org", oauth_signature="vUeLEE4iLC8Z01rTnbDNF5Rcn6A%3D"</pre>
Signature Base String
<pre>POST&amp;https%3A%2F%2Fstriiv-api-prod.appspot.com%2Fapi%2F1%2Frequest_token &amp;oauth_callback%3Dwww.example.org%26oauth_consumer_key%3D4GVX3d72gsq9RkK 5%26oauth_nonce%3DXx6cPU%26oauth_signature_method%3DHMAC-SHA1%26oauth_t imestamp%3D1415145585%26oauth_version%3D1.0a</pre>
Response
<pre>HTTP/1.1 200 OK Content-Type: text/html; charset=utf-8 oauth_token_secret=F8V57vGwLhPGYRpk&amp;oauth_token=SdW6cdWT5UUzDhfK[&amp;oauth_callback_confirmed=true]</pre>

### 3. Redirect user to authorization page

In this step the client application will redirect the user to Striiv so that the user may authorize the client application to access their protected resources.

**Workflow:** If an user is already logged in to Striiv, he or she will be redirected to a page with basic information about the client attempting to gain access, the user is then given the option to either grant or deny a permission. If a user isn't logged in then they will be prompted to log in. If a user was already logged in, and is taken directly to the authorization page, an option is available to log in as a different user (for the case in which the wrong user is already logged in).

#### Request

**REDIRECT** to [https://striiv-api-prod.appspot.com/api/1/authorize?oauth\\_token=<request\\_token>](https://striiv-api-prod.appspot.com/api/1/authorize?oauth_token=<request_token>)

After the user grants access, he or she will be redirected to the callback URL provided in Step 2, along with the oauth token and a verifier passed as query parameters.

Example
<a href="https://mycallbackurl.com?oauth_token=enAjRPRvETXNu3th&amp;oauth_verifier=quB4NhhQgt">https://mycallbackurl.com?oauth_token=enAjRPRvETXNu3th&amp;oauth_verifier=quB4NhhQgt</a>

If no callback URL was provided, the information will be displayed on the Striiv site.

Example
<p><b>Success!</b></p> <p>You have successfully authorized &lt;client app name&gt; to access your data.</p> <p>Request token: SdW6cdWT5UUzDhfK</p> <p>Verifier: ZNYekcpPx2</p> <p>If you wish to revoke access, you can do so at your account page.</p>

If the user disallows access and a callback URL was specified, they will still be redirected to the callback URL, along with an error

message as a query parameter.

### Example

```
https://mycallbackurl.com?error=Access+not+granted+by+user
```

If no callback URL was provided and the user disallows access, a message will be displayed at the Striiv site.

### Example

You have successfully denied MyThirdPartyApp access to your data.

## 4. Obtain an access token and secret

### Request

**POST /access\_token** (Full URL [https://striiv-api-prod.appspot.com/api/1/access\\_token](https://striiv-api-prod.appspot.com/api/1/access_token))

Authorization header parameters required in alphabetical order:

<b>oauth_consumer_key</b>	Consumer credential obtained in Step 1.
<b>oauth_token</b>	Request token obtained in Step 2.
<b>oauth_nonce</b>	Random, unique string generated by consumer to allow the server to verify uniqueness of request and help prevent replay attacks.
<b>oauth_signature</b>	Signature computed as describe in <a href="#">The OAuth 1.0 Protocol Section 3.4: Signature</a> .
<b>oauth_signature_method</b>	Signature method: HMAC-SHA1 or PLAINTEXT
<b>oauth_timestamp</b>	Timestamp in epoch (UTC). Requests older than 5 min will be rejected.
<b>oauth_verifier</b>	Verifier received when user authorized client application.
<b>oauth_version</b>	1.0 or 1.0a. This parameter is optional.

When using HMAC-SHA1 to sign, the `oauth_token_secret` obtained in Step 2 must be used.

### Response

<b>Status Code</b>	200 OK
<b>Headers</b>	Content-Type: text/html; charset=utf-8
<b>Body</b>	<code>oauth_token_secret=&lt;access_token_secret&gt;&amp;oauth_token=&lt;access_token&gt;&amp;striiv_user_id=&lt;id&gt;</code>

### Example

#### Request

```
POST /api/1/access_token HTTP/1.1
Host: striiv-api-prod.appspot.com
Authorization: OAuth
oauth_consumer_key="4GVX3d72gsq9RkK5",oauth_token="rLJqHVHphqzqKzGF",oauth_signature_method="HMAC-SHA1",oauth_timestamp="1415149138",oauth_nonce="DH0bRF",oauth_version="1.0a",oauth_signature="S/1tbhirW8wl8WFikj kRdPrxiYs=",oauth_verifier=s56fGzCq9c
```

#### Signature Base String

```
POST&https%3A%2F%2Fstriiv-api-prod.appspot.com%2Fapi%2F1%2Faccess_token&oauth_consumer_key%3D4GVX3d72gsq9RkK5%26oauth_nonce%3DDH0bRF%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1415149138%26oauth_token%3DrLJqHVHphqzqKzGF%26oauth_verifier%3Ds56fGzCq9c%26oauth_version%3D1.0a
```

### Response

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
oauth_token_secret=g2zNHc2jNVBpWkCb&oauth_token=GdZChqLe8uAV8Cwt&striiv_user_id=4510013087285248
```

## 5. Access a protected resource

### Request

GET <resource\_url>

Authorization header parameters required in alphabetical order:

<b>oauth_consumer_key</b>	Consumer credential obtained in Step 1.
<b>oauth_token</b>	Access token obtained in Step 4.
<b>oauth_nonce</b>	Random, unique string generated by consumer to allow the server to verify uniqueness of request and help prevent replay attacks.
<b>oauth_signature</b>	Signature computed as describe in <a href="#">The OAuth 1.0 Protocol Section 3.4: Signature</a> .
<b>oauth_signature_method</b>	Signature method: HMAC-SHA1 or PLAINTEXT
<b>oauth_timestamp</b>	Timestamp in epoch (UTC). Requests older than 5 min will be rejected.
<b>oauth_version</b>	1.0 or 1.0a. This parameter is optional.

When using HMAC-SHA1 to sign, the `oauth_token_secret` obtained in Step 4 must be used.

### Response

<b>Status Code</b>	200 OK
<b>Headers</b>	Content-Type: application/json
<b>Body</b>	Response in JSON format.

### Example

#### Request

```
GET /api/1/user/-/sleep/start/2010-01-01/end/2014-10-03/json HTTP/1.1
Host: striiv-api-prod.appspot.com
Authorization: OAuth
oauth_consumer_key="4GVX3d72gsq9RkK5",oauth_token="GdZChqLe8uAV8Cwt",oauth_signature_method="HMAC-SHA1",oauth_timestamp="1415150409",oauth_nonce="OOyuDDK",oauth_version="1.0a",oauth_signature="WJI3aHNetQxXzZQWCKPyv+CKrri="
```

#### Signature Base String

```
GET&https%3A%2F%2Fstriiv-api-prod.appspot.com%2Fapi%2F1%2Fuser%2F-%2Fsleep%2Fstart%2F2010-01-01%2Fend%2F2014-10-03%2Fjson&oauth_consumer_key%3D4GVX3d72gsq9RkK5%26oauth_nonce%3DOOyuDDK%26oauth_signature_method%3DHMAC-SHA1%26oauth_timestamp%3D1415150409%26oauth_token%3DGdZChqLe8uAV8Cwt%26oauth_version%3D1.0a
```

#### Response

```
{
  "sleep": [
    {
```

```
"version": "1.0",
"logs": [
  {
    "typeMinutes": [
      2,
      2,
      2
    ],
    "totalMinutes": 6,
    "data": [
      {
        "minutes": 2,
        "type": 1
      },
      {
        "minutes": 2,
        "type": 2
      },
      {
        "minutes": 2,
        "type": 3
      }
    ],
    "startTime": 1412380577
  }
],
"summary": {
  "date": "2014-10-03",
  "totalMinutes": 6,
  "typeMinutes": [
    2,
    2,
    2
  ]
}
},
{
  "version": "1.0",
  "logs": [
    {
      "typeMinutes": [
        1,
        1,
        1
      ],
      "totalMinutes": 3,
      "data": [
        {
          "minutes": 1,
          "type": 1
        },
        {
          "minutes": 1,
          "type": 2
        },
        {
          "minutes": 1,

```

```
        "type": 3
      }
    ],
    "startTime": 1412900186
  }
],
"summary": {
  "date": "2014-10-09",
  "totalMinutes": 3,
  "typeMinutes": [
    1,
    1,
    1
  ]
}
```

```
{
  ]
}
```

## Striiv Resource Access API

The Resource Access API contains a set of REST endpoints that allow read access to users' data. At present endpoints cover activity and sleep data.

### Getting started:

- Striiv API base URL is <https://striiv-api-prod.appspot.com/api/1> (HTTPS is required)
- Our endpoints currently support only **JSON**.
- Access is currently limited to **READ** operations.
- All request must provide [OAuth](#) signature in the *Authorization* header.
- The user ID specified in the URL must match that of the user associated with the OAuth access token. The user ID can also be omitted from the URL, and the user will be that associated with the OAuth access token.

### Rate Limiting

There is currently no limit on the number of requests a client can make.

### Privacy controls

All requests require full OAuth authorization.

### API Unit System

The Striiv API uses a combination of metric and imperial units.

### Timestamps in the API

All date fields in the API requests and responses correspond to a user's timezone. Start times are in epoch time and thus correspond to UTC, not counting leap seconds.

## User Resources Available

The following are currently available:

- [Getting a user's activity data](#)
- [Getting a user's sleep data](#)
- [Getting a user's all time average activity](#)
- [Getting a user's activity stats](#)

## API-Get-Activities

Get a summary of a user's activity for a given day or range of dates in JSON format.

### Where we differ from Fitbit

Unlike Fitbit, we don't have separate endpoints for [Time Series Data](#) and [Activity Data](#). The Striiv Get Activities endpoint returns the same set of properties for a single day than for a range of dates. To get activity data for a single day set the start and end date to the day desired, and the result will be a list with one element: the data for that day.

## Notes

- Access Type: *Read*
- Rate Limited: *No*



- [OAuth](#): required

## Resource URL

GET /api/<api\_version>/user/<user\_id>/activities/start/<start\_date>/end/<end\_date>/<response\_format>

<b>api_version</b>	The API version. Currently 1.
<b>user_id</b>	The Striiv user ID passed when the OAuth access token was obtained, or or "-" (dash) to indicate user currently authenticated via the access token.
<b>start_date</b>	The start date of the interval in the format <b>yyyy-MM-dd</b> . This date is included in the range.
<b>end_date</b>	The end date of the interval in the format <b>yyyy-MM-dd</b> . This date is included in the range.
<b>response_format</b>	The response format. Currently the only supported format is "json" for JSON.

### Request

```
GET /api/1/user/-/activities/start/2014-10-01/end/2014-10-01/json
GET /api/1/user/-/activities/start/2014-10-01/end/2014-10-10/json
GET /api/1/user/1234567/activities/start/2014-11-05/end/2014-11-06/json
```

## Authentication

Via [OAuth](#) access token.

## Response

<b>Status Code</b>	200 OK
<b>Content-Type</b>	<i>application/json</i>

## Response Body Format

Note: Text within <> is a descriptive place holder for a value.

### Example

```
GET /api/1/user/-/activities/start/2014-10-01/end/2014-10-01/json
GET /api/1/user/-/activities/start/2014-10-01/end/2014-10-10/json
GET /api/1/user/1234567/activities/start/2014-11-05/end/2014-11-06/json
```

## JSON Response

```
{ "summary": [
  {
    "distance": <value>,
    "totalSteps": <value>,
    "activitySeconds": <value>,
    "calories" : <value>,
    "walkCount" : <value>,
    "runCount" : <value>,
    "stairCount" : <value>,
    "activityDate" : <value>
  },
  <...>
]
}
```

<b>distance</b>	Total distance traveled that day in miles.
<b>totalSteps</b>	Total number of steps. This is the sum of walking steps, running steps, and stair steps.
<b>activitySeconds</b>	Total duration of all activity in seconds.
<b>calories</b>	Total calories for the day.
<b>walkCount</b>	Total number of walking steps for the day.
<b>runCount</b>	Total number of running steps for the day.
<b>stairCount</b>	Total number of stair steps for the day.
<b>activityDate</b>	Date of the activity in the format <b>yyyy-MM-dd</b> .

## Examples

## JSON Response

```
{
  "summary": [
    {
      "distance": 4.5,
      "totalSteps": 7863,
      "activitySeconds": 21632,
      "calories": 2050.0,
      "walkCount": 3863,
      "runCount": 2000,
      "stairCount": 2000,
      "activityDate": "2014-10-21"
    },
    {
      "distance": 6.7,
      "totalSteps": 12356,
      "activitySeconds": 28830,
      "calories": 2500.0,
      "walkCount": 2302,
      "runCount": 6054,
      "stairCount": 4000,
      "activityDate": "2014-10-22"
    }
  ]
}
```

## API-Get-Activity-Stats

Get user's activity statistics in JSON format. Activity statistics includes *Lifetime* and *Best* activity values, both of which correspond to activity logged by the tracker device.

### Where we differ from Fitbit

Striiv does not currently allow users to manually log activity, so our *Best* and *Lifetime* values only include activity tracked by the device.

## Notes

- Access Type: *Read*
- Rate Limited: *No*
- OAuth: *required*

## Resource URL

GET /api/<api\_version>/user/<user\_id>/activities/<response\_format>

<b>api_version</b>	The API version. Currently 1.
<b>user_id</b>	The Striiv user ID passed when the OAuth access token was obtained, or or "-" (dash) to indicate user currently authenticated via the access token.
<b>response_format</b>	The response format. Currently the only supported format is "json" for JSON.

## Examples

GET /api/1/user/-/activities/json

GET /api/1/user/1234567/activities/json

## Authentication

Via OAuth access token.

## Response

<b>Status Code</b>	200 OK
<b>Content-Type</b>	<i>application/json</i>

## Response Body Format

Note: Text within <> is a descriptive place holder for a value.

## JSON Response

```

{
  "lifetime": {
    "totalSteps": <value>,
      "distance": <value>,
    "activitySeconds": <value>,
    "calories": <value>,
    "walkCount": <value>,
    "stairCount": <value>,
    "runCount": <value>
  },
  "best": {
    "activitySeconds": <value>,
    "activitySecondsDate": <value>,
    "calories": <value>,
    "caloriesDate": <value>,
    "distance": <value>,
    "distanceDate": <value>,
    "runCount": <value>,
    "runCountDate": <value>,
    "stairCount": <value>,
    "stairCountDate": <value>,
    "totalSteps": <value>,
    "totalStepsDate": <value>,
    "walkCount": <value>,
    "walkCountDate": <value>
  }
}

```

<b>totalSteps</b>	Total number of steps. This is the sum of walking steps, running steps, and stair steps for the day on which the maximum number of total steps was achieved.
<b>distance</b>	Total distance traveled in miles.
<b>activitySeconds</b>	Total duration of all activity in seconds.
<b>calories</b>	Total calories.
<b>walkCount</b>	Total number of walking steps.
<b>runCount</b>	Total number of running steps.
<b>stairCount</b>	Total number of stair steps.

<b>_Date</b>	The date on which that particular best was achieved.
--------------	--

Note that the values of each of property in *best* may have been obtained on different dates, as they correspond to the best/highest for that individual property. Thus, the best totalSteps may not equal the sum of best walkCount, best runCount and best stairCount (which may be from different dates).

## Examples

Examples
<pre>{   "lifetime": {     "totalSteps": 35941,     "distance": 300883.0,     "activitySeconds": 100842,     "calories": 70528.0,     "walkCount": 20941,     "stairCount": 5000,     "runCount": 1000   },   "best": {     "activitySeconds": 18432,     "activitySecondsDate": "2014-08-30",     "calories": 3097.0,     "caloriesDate": "2014-12-25",     "distance": 12706.0,     "distanceDate": "2015-01-13",     "runCount": 13098,     "runCountDate": "2013-10-12",     "stairCount": 5632,     "stairCountDate": "2014-05-20",     "totalSteps": 23405,     "totalStepsDate": "2014-12-25",     "walkCount": 19432,     "walkCountDate": "2014-04-17"   } }</pre>

## API-Get-Alltime-AVG-Activity

Get the daily average of a user's activity over the lifetime of their activity data in JSON format.

## Notes

- Access Type: *Read*
- Rate Limited: *No*
- OAuth: *required*

## Resource URL

GET /api/<api\_version>/user/<user\_id>/allTimeAvg/<response\_format>

<b>api_version</b>	The API version. Currently 1.
<b>user_id</b>	The Striiv user ID passed when the OAuth access token was obtained, or or "-" (dash) to indicate user currently authenticated via the access token.
<b>response_format</b>	The response format. Currently the only supported format is "json" for JSON.

### Examples

```
GET /api/1/user/-/allTimeAvg/json
```

```
GET /api/1/user/1234567/allTimeAvg/json
```

## Authentication

Via [OAuth](#) access token.

## Response

<b>Status Code</b>	200 OK
<b>Content-Type</b>	<i>application/json</i>

## Response Body Format

**Note:** Text within <> is a descriptive place holder for a value.

### JSON Response

```
{
  "average": {
    "totalSteps": <value>,
    "distance": <value>,
  },
  "activitySeconds": <value>,
  "calories": <value>,
  "walkCount": <value>,
  "stairCount": <value>,
  "runCount": <value>
}
```

<b>totalSteps</b>	Average of the daily total number of steps.
<b>distance</b>	Average of the daily total distance (in miles).
<b>activitySeconds</b>	Average of the daily duration of all activity in seconds.
<b>calories</b>	Average of daily total calories.
<b>walkCount</b>	Average of daily total walking steps.
<b>runCount</b>	Average of daily total running steps.
<b>stairCount</b>	Average of daily total stair steps.

## Examples

### JSON Response

```
{
  "average": {
    "totalSteps": 3941,
    "distance": 3883.0,
    "activitySeconds": 3780,
    "calories": 1678.0,
    "walkCount": 3932,
    "stairCount": 2104,
    "runCount": 945
  }
}
```

## API-Get-Sleep

Get a summary of a user's sleep log and sleep summary for a given day or range of dates in JSON format. Each day's summary is of all sleep log entries for that day (including naps).

### Where we differ from Fitbit

Unlike Fitbit, we don't have separate endpoints for [Time Series Data](#) and [Sleep Data](#). The Striiv Get Sleep endpoint returns the same set of properties for a single day than for a range of dates. To get sleep data for a single day set the start and end date to the day desired, and the result will be a list with one element: the data for that day.

## Notes

- Access Type: *Read*
- Rate Limited: *No*
- OAuth: *required*

## Resource URL

GET `/api/<api_version>/user/<user_id>/sleep/start/<start_date>/end/<end_date>/<response_format>`



<b>api_version</b>	The API version. Currently 1.
<b>user_id</b>	The Striiv user ID passed when the OAuth access token was obtained, or or "-" (dash) to indicate user currently authenticated via the access token.
<b>start_date</b>	The start date of the interval in the format <b>yyyy-MM-dd</b> . This date is included in the range.
<b>end_date</b>	The end date of the interval in the format <b>yyyy-MM-dd</b> . This date is included in the range.
<b>response_format</b>	The response format. Currently the only supported format is "json" for JSON.

### Examples

```
GET /api/1/user/-/sleep/start/2014-10-01/end/2014-10-01/json
```

```
GET /api/1/user/-/sleep/start/2014-10-01/end/2014-10-10/json
```

```
GET /api/1/user/1234567/sleep/start/2014-11-05/end/2014-11-06/json
```

## Authentication

Via [OAuth](#) access token.

## Response

<b>Status Code</b>	200 OK
<b>Content-Type</b>	<i>application/json</i>

## Response Body Format

Note: Text within <> is a descriptive place holder for a value.

## JSON Response

```

{
  "sleep": [
    {
      "version": <value>,
      "logs": [
        {
          "typeMinutes": [<value>, <...>],
          "totalMinutes": <value>,
          "data": [
            {
              "minutes": <value>,
              "type": <value>
            },
            <...>
          ],
          "startTime": <value>
        },
        <...>
      ],
      "summary": {
        "date": <value>,
        "totalMinutes": <value>,
        "typeMinutes": [<value>, <...>]
      }
    },
    <...>
  ]
}

```

**Sleep minute types**

Each minute of sleep can be of one of three types, represented as integers:

- 0: Asleep
- 1: Restless
- 2: Awake

<b>sleep</b>	List with one entry of sleep data (logs + summary) per day.
<b>version</b>	Data version. Currently 1.0.

<b>logs</b>	<p>A list containing an entry per sleep instance on that day.</p> <p>Each log contains:</p> <ul style="list-style-type: none"> <li>• <b>typeMinutes</b>: list with an entry per minute type (3), with the total number of minutes of that type for this instance of sleep (stored in the index of the minute type, <code>typeMinutes[minute_type] = total minutes of that type for this sleep instance</code>).</li> <li>• <b>data</b>: list with an entry per each contiguous block of minutes of the same type in this sleep instance. <ul style="list-style-type: none"> <li>• <b>minutes</b>: the number of minutes in this contiguous block (all of which are of the same type).</li> <li>• <b>type</b>: the type of sleep minutes.</li> </ul> </li> <li>• <b>startTime</b>: start time in epoch of this instance of sleep.</li> </ul>
<b>summary</b>	<p>Summary of all sleep instances for that day.</p> <ul style="list-style-type: none"> <li>• <b>date</b>: The date in yyyy-MM-dd format.</li> <li>• <b>typeMinutes</b>: list with an entry per minute type (3), with the total number of minutes of that type for the entire day (stored in the index of the minute type, <code>typeMinutes[minute_type] = total minutes of that type for the day</code>).</li> </ul>

## Examples

### JSON Response

```
{
  "sleep": [
    {
      "version": "1.0",
      "logs": [
        {
          "typeMinutes": [10, 4, 2],
          "totalMinutes": 16,
          "data": [
            {
              "minutes": 2,
              "type": 2
            },
            {
              "minutes": 2,
              "type": 1
            },
            {
              "minutes": 10,
              "type": 0
            },
            {
              "minutes": 2,
              "type": 1
            }
          ]
        }
      ]
    }
  ]
}
```

```

"startTime": 1412380577
},
{
"typeMinutes": [40, 10, 5],
"totalMinutes": 55,
"data": [
{
"minutes": 5,
"type": 2
},
{
"minutes": 10,
"type": 1
},
{
"minutes": 40,
"type": 0
}
],
"startTime": 1412380577
}
],
"summary": {
"date": "2014-10-03",
"totalMinutes": 71,
"typeMinutes": [50, 14, 7]
}
]
}

```

## Striiv Subscriptions API

- [Quick Start](#)
- [Overview](#)
- [Definitions](#)
- [Discovering Feeds](#)
- [Managing Subscriptions](#)
  - [Configure your account with a subscriber endpoint](#)
  - [Add a subscription](#)
  - [Remove a subscription](#)
  - [List existing subscriptions](#)

- [Receiving Update Notifications](#)
  - [Responding to a Notification](#)
  - [Notification Format](#)
    - [Example Notifications](#)

## Quick Start

To get started quickly:

1. Implement a simple subscriber endpoint as described in [Receiving update notifications](#).
2. **Make sure your subscriber endpoint is accessible from striiv-api-prod.appspot.com servers.**
3. Configure your subscriber endpoints as described in [Configure your account with a subscriber endpoint](#).
4. Authenticate Striiv users on your site as described in the [OAuth Authentication API](#).
5. Add subscriptions as described in [Add a subscription](#).

The following sections describe the subscriptions flow and Subscription API endpoint in more detail.

## Overview

The API allows 3rd party partners to be notified when a Striiv user's data changes. This provides the partner with a user's latest data without the need for polling or a scheduling system.

Subscription notifications are sent to an endpoint you designate. You may also choose to specify multiple endpoints, each receiving change notifications for a subset of members or a subset of data types. These notifications are basically HTTPS POST requests made by Striiv to URLs that you register. These URL's must be accessible to our servers and must be able to handle the incoming HTTPS POST request and process the content of the request (JSON).

**Note:** These notifications just alert you to the fact that certain data has been updated. Your application must then make the appropriate API calls (described in the [Resource Access API](#)) to retrieve the actual data.

## Definitions

<b>Subscriber</b>	A client service endpoint that is notified by Striiv when member data changes. Subscribers are identified by a <b>Subscriber ID</b> .
<b>Subscription</b>	An association between a Striiv member collection and client subscriber that lets us know you're interested in being notified of changes. Subscriptions are identified by a <b>Subscription ID</b> .
<b>Stream</b>	A sequence of changes to member data.
<b>Collection Type</b>	A grouping of related resources. Presently, Striiv supports the following collection types: activities and sleep
<b>Notification</b>	A request sent by Striiv to a subscriber indicating that a Striiv resource has been added, modified, or deleted.

The Striiv Subscriptions API does not attempt to send a notification more than once. Notifications may be missed due to: Striiv API outages, connectivity across the internet, your application being unavailable when Striiv attempts to send the notification, or your application not responding within 3 seconds to a notification request.

## Discovering Feeds

Subscriptions can be created on any user-based collection by appending "/apiSubscriptions" to the end of the URL.

For example, given a user whose version 1 URL is <https://striiv-api-prod.appspot.com/api/1/user/123>, the following are all valid constructs:

- <https://striiv-api-prod.appspot.com/api/1/user/123/apiSubscriptions> - all of user 123's changes
- <https://striiv-api-prod.appspot.com/api/1/user/123/activities/apiSubscriptions> - only user 123's activities data
- <https://striiv-api-prod.appspot.com/api/1/user/123/sleep/apiSubscriptions> - only user 123's sleep data

When you create a subscription on a collection, your subscriber will be notified of changes to any resource within the collection.

## Managing Subscriptions

### Configure your account with a subscriber endpoint

Before you can subscribe, you must provide us with one or more subscriber endpoints. This can be done at the [Register a subscriber site](#) (the form will accept either the consumer ID or the OAuth consumer key) .

A subscriber endpoint is a callback URL where updates are sent. One subscriber endpoint must be designated as the default subscriber. If you specify only one subscriber, it is automatically the default. The default subscriber is used if no subscriber is specified when issuing a subscription request.

You may assign a subscriber ID to your subscribers, which is the ID you will use when issuing a subscription request. If you don't specify a subscriber ID, one will be assigned for you.

Subscriber endpoints have the following parameters:

<b>Consumer ID</b>	<b>required</b>	The consumer's ID in our database (the form will also accept the OAuth consumer key).
<b>Callback URL</b>	<b>required</b>	The full URL to the subscriber endpoint where update notifications will be sent (e.g. " <a href="https://www.example.com/striiv">https://www.example.com/striiv</a> ")
<b>Format</b>	<b>required</b>	The serialization format that update notifications will use. Currently, only JSON is supported.
<b>API version</b>	<b>required</b>	The version of the API used by update notifications.
<b>default=true false</b>	optional	Whether this subscriber should be used as the default on requests where no subscriber is specified. Changing the default subscriber only affects new subscriptions after the change.
<b>Subscriber ID</b>	optional	The ID you will use to refer to this subscriber. If none specified, we will assign one for you (starting with 1, 2, 3...).

### Add a subscription

Add a subscription for the user to get notifications and get a response in the format requested. As explained below, the **<subscription\_id>** provides a way to associate an update notification with a particular user stream in your application.

Access Type: *Read*

#### Resource URL

**POST /api/<api\_version>/user/<user\_id>/<collection\_path>/apiSubscriptions/<subscription\_id>.<response\_format>**

<b>api_version</b>	The API version. Currently 1.
<b>user_id</b>	The Striiv user ID or '-' (user will be implied by the <a href="#">OAuth</a> access token).
<b>collection_path</b>	Collection resource URL; collection to receive notifications from (activities or sleep, currently). If not present subscribing to changes from all collections. Note, that if you have both "all" and specific collection subscriptions, you will get duplicate notifications on that collection's update. Also each subscriber may have only one subscription for a specific user's collection.
<b>subscription_id</b>	Unique ID of the subscription provided by you. Note, that you can have only one subscription with unique ID across the whole set of subscribers and collections. The Striiv servers will pass this ID back to you when you are notified that the user indicated by the parameter in the URL path has new data for you to pull. It will be available in the <subscriptionId> field of the notification. Valid characters: a-z, A-Z, 0-9, -, _
<b>response_format</b>	The response format. Currently the only supported response formats is "json" for responses in JSON.

### Examples

```
POST /api/1/user/123/apiSubscriptions/a1b2z3.json
```

```
POST /api/1/user/-/activities/apiSubscriptions/a1b2z3-activities.json
```

```
POST /api/1/user/123/sleep/apiSubscriptions/a1b2z3-sleep.json
```

#### Tip

If you only require one subscription per user in your application, then you can use your own internal user's ID as the subscription ID. This will allow you to easily associate a change notification with a specific user on your site without any additional lookups. A notification will contain the Striiv user ID ("ownerId") AND the subscription ID ("subscriptionId").

## Authentication

Via [access tokens](#). All authentication header parameters are required.

### Request Headers:

<b>X-Striiv-Subscriber-Id</b>	optional	The ID of the subscriber. This subscriber would receive notifications for this subscription. If not present, the default subscriber is used.
-------------------------------	----------	--

## Response

### Status code:

<b>200 OK</b>	If the given subscriber is already subscribed to this stream.
<b>201 Created</b>	If a new subscription was created in response to your request.
<b>409 Conflict</b>	<ul style="list-style-type: none"> <li>If the given subscriber is already subscribed to this stream using a different subscription ID, OR</li> <li>If the given subscription ID is already used to identify a subscription to a different stream</li> </ul>

### About Conflicts

A subscriber may only subscribe to the same stream once. If a conflict arises trying to add a subscription, the server will respond with an `HTTP 409 Conflict`. The response elements will hold information about the subscription that already exists.

## Response body format

The following elements are returned for all response codes listed above.

<b>subscriberId</b>	ID of subscriber that will receive notifications.
<b>ownerId</b>	ID of the resource owner.
<b>ownerType</b>	Type of the resource owner. Currently "user".
<b>collectionType</b>	Collection whose stream has been subscribed to. When subscribing to all of a user's updates, the value of this element is "user".
<b>subscriptionId</b>	Subscription ID. Subscription IDs are treated as opaque strings up to 50 unicode characters long. <b>In the case of a 409 Conflict response, this is the ID of the subscription already associated with this stream.</b>

Note: Text within `<>` is a descriptive place holder for a value.

### JSON Response

```
{
  "collectionType": <value>,
  "ownerId": <value>,
  "ownerType": <value>,
  "subscriberId": <value>,
  "subscriptionId": <value>
}
```

## Examples

### JSON Response

```
{
  "collectionType": "user",
  "ownerId": "123",
  "ownerType": "user",
  "subscriberId": "abcde123",
  "subscriptionId": "54678912340984"
}
```

## Remove a subscription

Delete a subscription for the user and get a response in the format requested.

Access Type: *Read*

### Resource URL

**DELETE** /api/<api\_version>/user/<user\_id>/<collection\_path>/apiSubscriptions/<subscription\_id>.<response\_format>

<b>api_version</b>	The API version. Currently 1.
<b>user_id</b>	The Striv user ID or '-' (user will be implied by the <a href="#">OAuth</a> access token).
<b>collection_path</b>	Collection to delete subscription from. If not present delete subscription to "user" collection (this will not delete additional subscriptions to individual collections).
<b>subscription_id</b>	The ID of the subscription.
<b>response_format</b>	The response format. Currently, the only supported response format is "json" for responses in JSON.

### Examples

```
DELETE /api/1/user/123/apiSubscriptions/a1b2z3.json
DELETE /api/1/user/-/activities/apiSubscriptions/a1b2z3-activities.json
DELETE /api/1/user/123/sleep/apiSubscriptions/a1b2z3-sleep.json
```

### Authentication

Via [access tokens](#). All authentication header parameters are required.

### Request Headers

<b>X-Striv-Subscriber-Id</b>	optional	The ID of the subscriber. This subscriber would receive notifications for this subscription. If not present, the default subscriber is used.
------------------------------	----------	--

### Response

Status code:

<b>204 No Content</b>	If the given subscription was successfully deleted.
<b>404 Not Found</b>	If the given subscription ID did not exist for this application.

## List existing subscriptions

Get a list of user's subscriptions for your application in the format requested. You can fetch subscriptions for specific collections or the entire list of subscriptions for the user.

Access Type: *Read*



**Resource URL**

GET /api/<api\_version>/user/<user\_id>/<collection\_path>/apiSubscriptions.<response\_format>

<b>api_version</b>	The API version. Currently 1.
<b>user_id</b>	The Striiv user ID or '-' (user will be implied by the OAuth access token).
<b>collection_path</b>	Collection to get subscriptions for (activities or sleep, currently). If not present list will include whole list of subscriptions for the user.
<b>response_format</b>	The response format. Currently, the only supported response format is "json" for responses in JSON.

### Examples

```
GET /api/1/user/123/apiSubscriptions.json
GET /api/1/user-/activities/apiSubscriptions.json
GET /api/1/user/123/sleep/apiSubscriptions.json
```

**Authentication**

Via [access tokens](#). All authentication header parameters are required.

**Request Headers**

None

**Response body format**

Note: Text within <> is a descriptive place holder for a value.

### JSON Response

```
{
  apiSubscriptions:[
    {
      "collectionType":<value>,
      "ownerId":<value>,
      "ownerType":<value>,
      "subscriberId":<value>,
      "subscriptionId":<value>
    },
    <...>
  ]
}
```

**Examples**

### JSON Response

```
{
  apiSubscriptions:[
    {
      "collectionType": "user",
      "ownerId": "123",
      "ownerType": "user",
      "subscriberId": "abcde123",
      "subscriptionId": "456923042240"
    }
  ]
}
```

## Receiving Update Notifications

### Responding to a Notification

A notification indicates that a resource has been added. You must then fetch the resource to obtain its latest version.

Your server should respond to a subscription notification with HTTP 204 No Content and close the connection within 3-seconds or the connection will be aborted.

### Notification Format

An update notification request to your endpoint will be as follows:

<b>HTTP Method</b>	POST
<b>Content-Type</b>	<i>application/json</i>
<b>Accept</b>	<i>text/plain</i>

### Example Notifications

Example JSON Notification

```
[
  {
    "collectionType": "sleep",
    "date": "2014-11-01",
    "ownerId": "4509835594655",
    "ownerType": "user",
    "subscriptionId": "abcde123"
  },
  {
    "collectionType": "activities",
    "date": "2014-11-01",
    "ownerId": "4509835594655",
    "ownerType": "user",
    "subscriptionId": "fijkl456"
  }
]
```

## API Client Libraries

The Striiv API was designed for easy migration for partners who have previously integrated with Fitbit. To make this transition easier, we have leveraged a tool call [python-fitbit](#) to do testing of our libraries. If you are developing in Python, this tool could help you in your transition.

## API Open Questions

- Would it be useful to add pagination to the resource access endpoints? This would work as follows: a call would include a query parameter "limit" specifying the maximum number of items to be returned in a single response. Each response would include (in the JSON) a parameter called "cursor" if there are elements left to be fetched. If more elements are desired, subsequent calls can include the "cursor" string as an additional query parameter to resume getting elements where the previous call left off.
- Would it be useful to provide support for XML resource format?
- Would it be useful to provide an endpoint to get a user's basic information and timezone?

## Striiv Messaging API

The Striiv Messaging API allows 3rd party partners to send messages and surveys to groups of users.

The base URL is <https://striiv-api-prod.appspot.com>

## Creating a messaging group

This endpoint allows for the creation of a group of users for the purpose of messaging them collectively.

### Resource URL

**POST /api/1/groups**

### Request Body

#### Request Body JSON

```
{
  "group": {
    "name": "My group's name",
    "users": [
      123,
      352,
      345,
      234
    ]
  }
}
```

<b>name</b>	The name for the messaging group.
<b>users</b>	A list of the Striiv user IDs of all the members of the messaging group.

### Response

#### Success

<b>Status</b>	200 OK
<b>Content-type</b>	application/json

#### Respond Body JSON

```
{
  "group": {
    "group_id": 5632763172487168,
    "name": "My group's name"
  }
  "success": true
}
```

#### Failure

Status Code	Content-Type	Response Body
200	application/json	<pre>{   "group": {     "group_id": 5632763172487168,     "name": "My group's name"   }   "errors": [ //List of user's for which there was an error, and that weren't added to the group   {     "user_id": 123,     "error": "Invalid user id format."   },   {     "user_id": dfg123,     "error": "Invalid user."   }   ] }</pre>
400		<pre>"No user list specified.", "Empty user list."</pre>
500	application/json	<pre>{   "message": "Error creating group.",   "success": false }</pre>

## Adding users to a messaging group

### Resource URL

POST /api/1/groups/<group\_id>/users

<b>group_id</b>	The ID of the group to which the users will be added.
-----------------	---

### Request Body

**Request Body JSON**

```
{
  "users": [
    423,
    733,
    909,
    278
  ]
}
```

<b>users</b>	A list of the Striiv user IDs of all the members to be added to the messaging group.
--------------	--

**Response****Success**

<b>Status</b>	200 OK
<b>Content-type</b>	application/json

**Respond Body JSON**

```
{
  "users": [
    423,
    733,
    909,
    278
  ],
  "success": true
}
```

**Failure**

Status Code	Content-Type	Response Body
-------------	--------------	---------------

200	application/json	<pre> {   "users":[ //List of user's successfully added to group     423,     733,     909,     278   ],   "errors": [ //List of user's for which there was an error, and that weren't added to the group     {       "user_id": 123,       "error":"User is already a member of the group."     },     {       "user_id": 123,       "error":"Invalid user id format."     },     {       "user_id": dfg123,       "error":"Invalid user."     }   ] } </pre>
400		<pre> "Group 123456 is not valid", "No user list specified.", "Empty user list." </pre>
500	application/json	<pre> {   "message": "Error adding members to group.",   "success": false } </pre>

## Removing users from a messaging group

### Resource URL

**DELETE** /api/1/groups/<group\_id>/users/<user\_id>[,<user\_id>]

<b>group_id</b>	The ID of the group from which the users will be removed.
<b>&lt;user_ids&gt;</b>	A comma-separated list of user IDs to be removed.

## Response

### Success

<b>Status</b>	200 OK
<b>Content-type</b>	application/json

#### Respond Body JSON

```
{
  "users": [
    423,
    278
  ],
  "success": true
}
```

### Failure

Status Code	Content-Type	Response Body
200	application/json	<pre>{   "users": [ //List of user's successfully removed from group     423   ],   "errors": [ //List of user's for which there was an error, and that weren't     removed from the group     {       "user_id": 278,       "error": "Invalid user."     },     {       "user_id": 123,       "error": "User not in group."     }   ] }</pre>
400		"Group 123456 is not valid"
500	application/json	<pre>{   "message": "Error deleting member(s) from group.",   "success": false }</pre>

## Deleting a messaging group

### Resource URL

**DELETE** /api/1/groups/<group\_id>

<b>group_id</b>	The ID of the group to be removed.
-----------------	------------------------------------

### Response

#### Success

<b>Status</b>	200 OK
<b>Body</b>	"Successfully removed group 12345678."

#### Failure

Status Code	Content-Type	Response Body
400		"Invalid Group: 12345."
500	application/json	{ "message": "Error deleting group.", "success": false }

## Sending messages/surveys to a group

### Resource URL

**POST** /api/1/messages

### Request Body



## Request Body JSON

```
{
  "messages": [
    {
      "message": {
        "name": "survey name",
        "group": 14234,
        "user": 345678,
        "trigger": "2014-09-22 T 12:12:12",
        "expiration": "2014-09-22 T 12:12:12",
        "icon": 34,
        "vib": 2,
        "text": "small string",
        "response": {
          "type": 2,
          "min": 0,
          "max": 10
        }
      }
    },
    {
      "message": {
        "name": "message name",
        "group": 14234,
        "user": 345678,
        "trigger": "2014-09-22 T 12:12:12",
        "expiration": "2014-09-22 T 12:12:12",
        "icon": 34,
        "vib": 2,
        "text": "small string"
      }
    }
  ]
}
```

<b>name</b>	The name of the message or survey.
<b>group</b>	The group ID that the message or survey is to be sent to.
<b>user</b>	The user ID of the creator of the message.
<b>trigger</b>	The trigger time of the message <b>in UTC</b> .
<b>expiration</b>	The expiration time of the message <b>in UTC</b> .
<b>icon</b>	Integer designation the icon of the message.
<b>vib</b>	Integer designating the type of vibration to accompany the message.
<b>text</b>	The text of the message.
<b>response</b>	Optional parameter, included only for surveys. <ul style="list-style-type: none"> <li>• <b>type</b>: Integer indicating the response type.</li> <li>• <b>min</b>: Integer designating the minimum acceptable value of the response.</li> <li>• <b>max</b>: Integer designating the maximum acceptable value of the response.</li> </ul>

**Response Type**

Currently, the only response type that is checked is 1 for "Yes or No Surveys", which must have a "min" of 0 and a "max" of 1.

## Response

### Success

<b>Status</b>	200 OK
<b>Content-type</b>	application/json

### Respond Body JSON

```
{
  "messages": [
    {
      "message": {
        "broadcast_msg_id": "61e5ead5bfd48172fe725c1359d21bd",
        "name": "survey name",
        "group": 14234,
        "user": 345678,
        "trigger": "2014-09-22 T 12:12:12",
        "expiration": "2014-09-22 T 12:12:12",
        "icon": 34,
        "vib": 2,
        "text": "small string",
        "response": {
          "type": 2,

```

```

        "min":0,
        "max":10
    }
}
},
{
    "message":{
"broadcast_msg_id": "72e5fd48172fe725c1359d21cf",
        "name":"message name",
        "group":14234,
"user": 345678,
        "trigger":"2014-09-22 T 12:12:12",
        "expiration":"2014-09-22 T 12:12:12",
        "icon":34,
        "vib":2,
        "text":"small string"
    }
}
],
"success": true
}

```

<b>broadcast_msg_id</b>	The ID produced for this message/survey, which can be used to retrieve responses to the survey.
-------------------------	---

## Failure

Status Code	Content-Type	Response Body
200	application/json	<pre> {   "messages":[ //List of messages successfully posted     {       "message":{ "broadcast_msg_id": "61e5ead5bfd48172fe725c1359d21bd",         "name":"survey name",         "group":14234, "user": 345678,         "trigger":"2014-09-22 T 12:12:12",         "expiration":"2014-09-22 T 12:12:12", </pre>

```
        "icon":34,
        "vib":2,
        "text":"small string",
        "response":{
            "type":2,
            "min":0,
            "max":10
        }
    }
}
],
"errors": [ //List of messages the weren't posted due to an error
{
"error": {
"message_name": "name of message",
"group": 123dfg,
"reason": "Invalid group ID."
}
},
{
"error": {
"message_name": "name of message",
"group": 345,
"reason": "Group ID not found."
}
},
{
"error": {
"message_name": "name of message",
"group": 345,
"reason": "Invalid user ID: 345678."
}
},
{
"error": {
"message_name": "name of message",
"group": 345,
"reason": "Trigger time has passed."
}
},
},
}
```

		<pre>{   "error": {     "message_name": "name of message",     "group": 345,     "reason": "Responses of type 1 must have a min of 0 and a max of 1."   } }</pre>
400		//Error messages
500	application/json	<pre>{   "message": "Error posting messages.",   "success": false }</pre>

## Getting responses to a survey

### Resource URL

GET /api/1/messages/<broadcast\_msg\_id>/responses[?limit=<limit\_value>][&cursor=<cursor\_value>]

<b>broadcast_msg_id</b>	The survey/message ID whose responses will be fetched.
<b>limit_value</b>	Optional query parameter for pagination. Sets the number of survey responses that will be sent in a single response. Default is 20.
<b>cursor_value</b>	Optional query parameter for pagination. A different one is sent with every response (as long as there are still survey responses to be served), to be passed on subsequent requests to continue survey response retrieval where it last left off.

## Response

### Success

<b>Status</b>	200 OK
<b>Content-type</b>	application/json

## Respond Body JSON

```
{
  "responses": [
    {
      "user": 1234,
      "message_id": 1234,
      "response_type": 2,
      "response_timestamp": "2005-10-30 10:45",
      "response_value": 3
    },
    {
      "user": 2342,
      "message_id": 5678,
      "response_type": 2,
      "response_timestamp": "2005-10-30 10:45",
      "response_value": 8
    }
  ],
  "cursor": "E-ABAOsB8gEScmVzcG9uc2VfdGltZXN0YW1w"
}
```

<b>responses</b>	A list of responses, one per user in the messaging group the broadcast survey was sent to.
<b>user</b>	The user ID of the user that generated the response.
<b>message_id</b>	The ID of the individual message that was sent to the user. Not to be confused with the broadcast_msg_id (which indicates the canonical message).
<b>response_type</b>	The type of response that is associated with the survey.
<b>response_timestamp</b>	Timestamp indicating the time when the response was produced, in UTC.
<b>response_value</b>	The value of the response.
<b>cursor</b>	String that must be passed as a query parameter in the next request to continue getting responses.

**Cursors**

Whenever there are responses left to be served, the response will include a cursor. If no cursor is included, then all responses have been retrieved. The cursor will be different with each response, to indicate the different places where the query left off.

**Failure**

Status Code	Content-Type	Response Body
400		//Error messages
500	application/json	<pre>{   "message": "Error getting responses.",   "success": false }</pre>