WITHINGS + <yourcompany>

Data API Level 2 [Withings SDK]
Quick Start Guide
24/03/2021
V1.0
Summary

1. Scope & User Experience
2. Create your Withings API environment & request access
3. Withings Device Setup Flow
4. Simplified Architecture
5. Technical Overview
   a. How to use the SDK installation Webview
   b. How to enable Bluetooth LE background sync
   c. How to use the SDK Device Settings Webview
   d. Data / Measure API - Implementation recommendations
   e. Logistics option - Drop-shipment API
   f. Sample Code
6. Timeline Overview
1. Scope & User Experience
1.a. Scope & User Experience - Logistics and Shipment

Option 1
Withings bulk ships devices to your warehouse

Contact sales team

Option 2
Withings drop-ships devices to your members via API

Read the Logistics API doc
Use the createorder API

Option 3
Withings drop-ships devices to your members via user list.

Contact sales team
1.1 Scope & User Experience - Devices unboxing and setup

1. Member Unboxes Devices and reads instructions
2. Member downloads your app by following your instructions
3. Mobile SDK displays setup interface and member sets up the device
4. Withings cloud Notifies yourcloud when new data is recorded
5. <yourapp> app Data Viz
6. Member is able to modify device settings (like Wi-Fi) from SDK Webview
2. Create your Withings API environment & request permissions
Create your Withings API app by following the online documentation here

Enter a valid callback URL for each clientId you create*
Callback url must support POST and HEAD requests and reply in less than 2s.

This callback url will be called by Withings server depending on the features you implement.

Share your clientId with Withings team to get permissions

**WELLNESS CLOUD BASE API URL**
https://wbsapi.withings.net

**HIPAA CLOUD BASE API URL**
https://wbsapi.us.withingsmed.net

*you might need to contact your network administrator to make sure connections from the Withings IP addresses are allowed. Consult the TXT record from ipblock-notify.withings.net to get the range (command: dig +short TXT ipblock-notify.withings.net). Please contact us if your app is hosted on our HIPAA/HDS Cloud.
Withings + <yourcompany> project kickoff

2. Request engineering samples for your team

Withings will provide engineering devices with your target firmware before you start development.

Please contact your sales representative to order your development samples.
3. Open the Withings device setup flow from your native app (Withings SDK)
3. Withings device setup flow from your native app (Withings SDK Webviews)

The **Installation Webviews** are hosted by Withings, minimizing integration and maintenance workload. Once webview is opened by SDK in your native iOS/Android app, most of the technical work is done.
3. Withings device setup flow from your native app (Withings SDK Webviews)

The *Installation Webviews* are hosted by Withings, minimizing integration and maintenance workload. Once webview is opened by SDK in your native iOS/Android app, most of the technical work is done.
3. Withings device setup flow from your native app (Withings SDK Webviews)

Device Data Synchronization options available in Withings SDK

- Direct Wi-Fi synchronization to the Withings Cloud*
- Bluetooth LE synchronization*

*Bluetooth LE Data synchronisation goes silently through the Withings SDK to the Withings cloud. No additional code required.
4. Simplified architecture
4. Simplified architecture

A: Wi-Fi Sync
B: Bluetooth LE Sync*
C: user tokens sharing
D: data fetch
E: data fetch

*Bluetooth LE Data sync is silently bridged through the Withings SDK to the Withings cloud. No code required.
5.a. How to use the SDK Installation Webview / build the Installation URL?
5.a. How to use the SDK Installation Webview / build the Installation URL?

1. <yourapp> requests the secured installation url from <yourcloud>

2. <yourcloud> generates the signed/secured url using your client_secret

3. App opens the install. webview URL

4. Installation Webview
   - Account confirmation
   - Device Setup
   - *device model can be passed as param in url

5. POST
   - Withings Cloud shares user tokens with <yourcloud>
   - Access/refresh tokens

6. Local native notification (success/failure)
   - <yourapp> closes webview
   - <yourapp> starts SDK BLE service

7. Returns Bluetooth advertise_key

8. See iOS documentation here
   - See Android documentation here

Confidential Withings - Do not distribute
5.a. How to use the SDK Installation Webview / build the Installation URL?

Installation URL params
The full documentation is available here.

Installation webview url requires to add encrypted end-users PHIs and PIIs as parameter. Withings will silently create the user account using these params (see encryption protocol)

It also requires to add your developer signature to the request (see signature protocol)

Other params such as language preference for the device/interface and unit display preferences are also required.

About PHI/PIIs
PHIs like weight, height or gender are required because device will use them to function properly. Email provided will be the one Withings will use to create an account on the Withings backend.
5.b. How to enable Bluetooth LE background sync service?

After a successful device setup, the Withings SDK will return a notification following the notification protocol. Please refer to the iOS or Android Background sync documentation to see how to use this notification to enable background bluetooth sync.

*This function should be called if you desire to support Bluetooth sync (For user who don't have Wi-Fi or use a Withings tracker).
5.c How to use the Device Settings Webview / build the Device Settings URL?
5.c. How to use the Device Settings Webview?

- Handle native notification
- Callbacks (success / errors)
- Receive advertise_key
- Close webview
5.d. Data / Measure API - Implementation recommendations
5.d. Data / Measure API - Implementation recommendations

1. Read the online documentation available [here](#).
2. Implement the Data Update Notification service between Withings Cloud and <yourcloud>. (arrow 1.)
3. Based on the data you are fetching, implement the right Data API (arrow 2.). Fetch only new data.
4. Recommended: associate APNS or GCP to trigger background data sync through silent push notification (arrow 2)
5. <yourapp> polls only new data from <yourcloud> by maintaining a local anchor / timestamp (arrow 3)

**Tips**
Avoid using cron to poll data from the Withings Cloud. Instead, implement the Data Update Notification Service.
1. **Subscribe to data update notif. (See doc)**
   - Subscription must be performed once for each program member.
   - Callback url must be listed in withings api portal.
   - Callback url must support HEAD and POST request and reply in less than 4s.

2. **Withings notifies <yourcloud>**
   - Withings notifies your callback url at each new measurement with userid.
   - Withings also includes time range where new measurements are recorded.

3. **<Yourcloud> calls the Data API**
   - <yourcloud> calls Data API using timestamps returned in the notification from step 2.

4. **<yourcloud> records new measures**
   - An APNS / GCP silent notification might also be triggered for mobile background sync.
5.e. Logistics Option: Drop-shipment API (createorder)
As you are implementing Data API Level 2 and Withings SDK, you might be interested in implementing the `createorder API` for drop-shipment to your end user.

Before implementing the `createorder API`, please have a look at the documentation available here.
6. Timeline overview - typical duration 8-10 weeks
Withings + <yourcompany> project kickoff

6. Timeline overview - typical duration 8-10 weeks

- **Scope alignment**
- **2 week tech prototyping**
- **Environment setup**

**API**
- HIPAA API apps creation
- Measure API
- Data Update Notif. API
- Member token renewal
- Dropship API

**SDK**
- Design member journey to sdk webview
- **SDK Init Webview**
  - Authentication & Signed URL generation (server side)
  - Handle Withings Server Callback (store access/refresh tokens)
  - Handle native setup Success/Failure Callbacks
  - Background sync function
- **SDK Device Settings Webview**
  - Display Settings URL
  - Handle Logout / Login
  - Handle expired tokens
  - Handle no network

**End to end testing**

**Progressive rollout**
5.f. Sample code

Please have a look at our Sample code section to help you implement our APIs