Balance® Rewards API

Version 1.0 – September 24, 2014

Introduction

The Balance® Rewards API enables pre-qualified partners to connect into Walgreens to share individual member health and wellness information to be eligible to receive Balance Rewards points for healthy activity like walking, running, and weight management.

The API is designed to enable partners to seamlessly share individual health and lifestyle activity (Data Sharing), and create or validate user memberships. In order for individuals to receive points for health and wellness activity, they must be a Balance Rewards member and have successfully completed a Walgreens OAuth login or registration.

In order for Balance Rewards members to receive points for data recorded using pre-qualified partners applications, devices, or platform, partners must share full activity information with Walgreens. In no way is Walgreens able to reward activity without supporting aggregate or transaction level information.

Activity Data & Points Rewards Processing:

- Activity sent to Walgreens will be processed and typically appear within the Balance Rewards for healthy choices dashboard within 30 minutes.
- Activity data is visible on Walgreens at https://www.walgreens.com/steps/stepslanding.jsp as well as within our Walgreens iPhone, Android Phone & iPad apps from the Balance Rewards for healthy choices icon on the app home page.
- Points for eligible activities (more info below) will be awarded within 24 hours of when the activity data is received as long as the customer has not reached their daily or monthly maximum point limits.
- Points are viewable at https://www.walgreens.com/youraccount/loyalty/loyalty_landing.jsp as well as within our Walgreens iPhone, Android Phone & iPad apps from the Balance Rewards icon on the app home page.

Currently, Balance Rewards members are eligible to receive the following daily rewards.

- 20 points per logged mile per day; conversion ratio is 2,000 steps per mile
- 20 points per weigh-in, maximum of one award per day
- 20 points per blood pressure measurements, maximum of one award per day
- 20 points per blood glucose measurements, maximum of two awards per day
- 250 points for syncing a compatible health & fitness device/app; up to two devices per month

However, there is a monthly maximum combined award limit of 1,000 points for all eligible activities.

- Once this limit has been reached, customers will see a message in their activity dashboard congratulating them on reaching the monthly maximum.
- This message will go away on the first day of the next month when the customer is eligible to earn points again.
- A customer can continue to log activities and track progress towards goals that have been set even after the monthly award points maximum has been reached.
- Points for device/app syncing are eligible outside of this monthly maximum award limit.

For additional information on the Balance Rewards program, please visit our FAQ page at http://www.walgreens.com/topic/help/steps-balance-rewards.jsp.

What this document covers

This document explains how to:

- Gain access to the Balance Rewards API
- Onboard members to begin posting qualifying activity data
- Post qualifying activity data to the Walgreens
- Unsubscribe users from posting data to Walgreens

Prerequisites

In order for a pre-qualified partner to connect into the Walgreens API, the following would be needed:

1. Walgreens to grant a API key to partner
2. Walgreens to grant an affiliate code (this determines what environments the third party application can point to; for example, Staging and Production)
3. Walgreens to share WAG OAuth flow URL along with URL parameters which is what the call to action link on the third party application portal should be configured to point to
4. Upon a user on the third party portal completing the above mentioned flow, they will be taken back to the callback URL parameter set in the above mentioned URL; the WAG OAuth token will be passed back within the callback URL as a parameter within the callback URL
5. It will be the responsibility of the third party application to persist this token and make sure it doesn't expire (tokens are set to expire every 30 days; token renewal API's will be made available)

Third party applications wishing to post activity data upon successful link of Wag App:

- Third party application would invoke an API passing the API key, affiliate ID, and WAG OAuth key within the API header (if WAG OAuth token is expired then third party is to renew the token prior to calling API; renewal API's will be made available)

Third party applications wishing to unlink an integration from Walgreens will require the following:

- Third party application would invoke an API passing the API key, affiliate ID, and WAG OAuth key within the API header (if WAG OAuth token is expired, then the third party application is to renew the token prior to calling API; renewal API's will be made available)

Instructions on how to get a user account, API key and Affiliate ID are provided below.

Getting Access to the Balance Rewards API

To setup a developer account with Walgreens and obtain the API key and Affiliate ID, developers must:

- Create a user account on the Walgreens Developer Portal (https://developer.walgreens.com/user/register)
- Sign into the Walgreens Developer Portal (https://developer.walgreens.com/user/login)
- Select the "Set Up an Application" button (in the "My Page" tab) and fill out the form (to request an API key). Please choose "Other" and type in "Balance Rewards" into the text box.

Once the API key request has been approved by Walgreens, an automated email will be generated and sent to the requester with a link to view the application's unique API key. The automated email will also include a generic test Affiliate ID. The requester's unique Affiliate ID will be provided to the developer once the requester's application is ready to commence on-boarding testing with the Walgreens API Program team.

On-boarding Process

For the purposes of this document, the term “on-boarding” means the process of linking a user of the third party application to a Walgreens Balance Reward account for the same user using a shared security token.

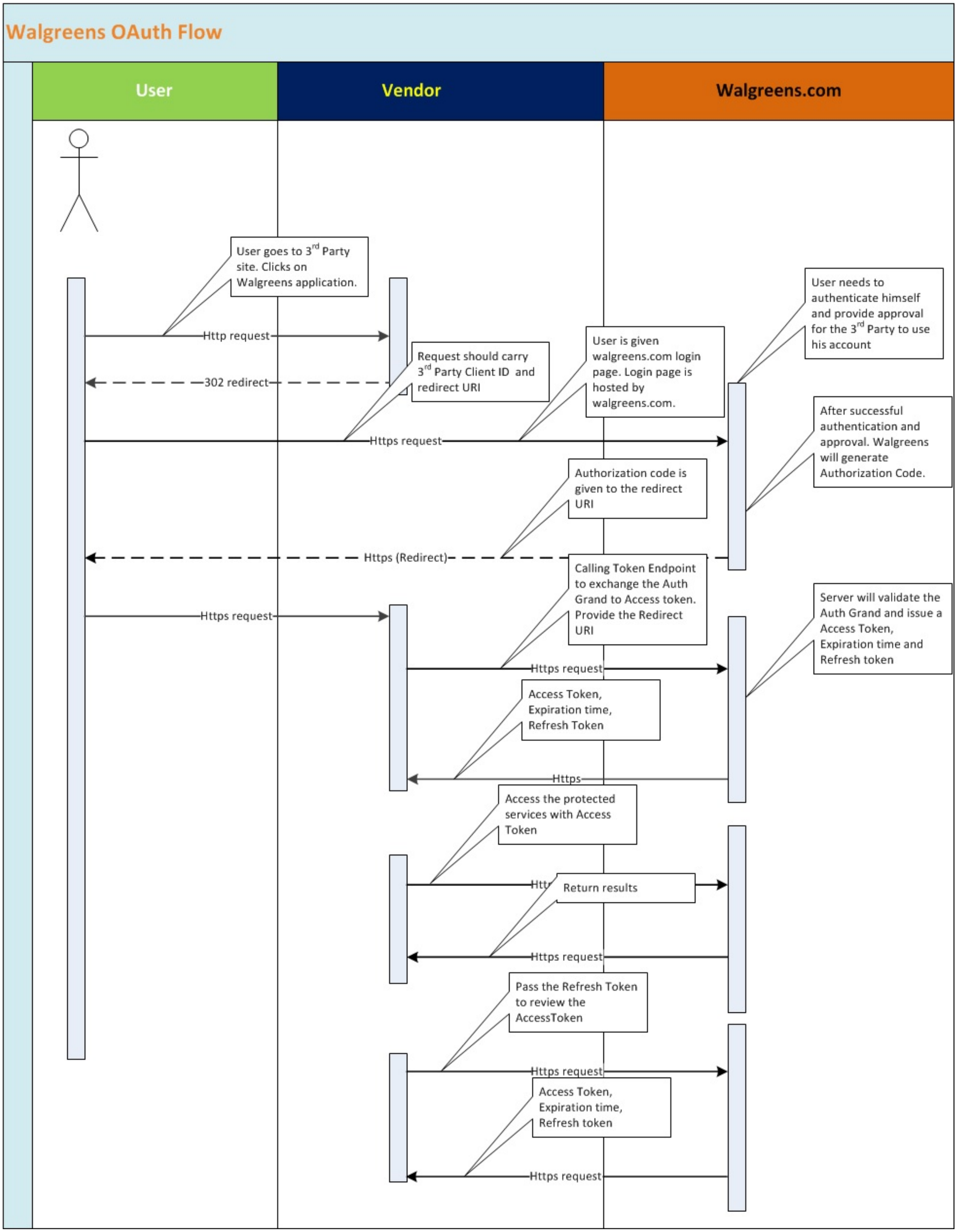
In this process, the user of the third party application must grant permission to authorize their application to share supported activity data with Walgreens. The process of retrieving an OAuth token is summarized by the flow shown below. Walgreens is using the Authorization Code grant type utilizing OAuth 2.0.

Authorization code flow is for the private clients who would request Access token from their webserver. All the traffic from the webserver to Walgreens is expected to go through HTTPS with TLS version preferred to be 1.2. Walgreens would use bearer token type of OAuth 2.0.

As part of the on-boarding process, third party developers will need to provide the redirect_uri, where they expect the Walgreens application to redirect the user to, along with the Authorization code.

- Walgreens application would always redirect the users coming from the third party application to the developer provided URI.
- For security reasons, even if the third party developer is expected to provide the redirect_uri in the Authorization code request, the Walgreens application will always redirect the user to the URI provided during the on-boarding process.

Service End Points	Description
https://services.walgreens.com/api/oauthtoken/v1 (for production) https://services-qa.walgreens.com/api/oauthtoken/v1 (for testing)	This service is used to generate the Access token for the requested client from the Authorization code. The Refresh token also will be generated and shared to client along with the Access token. This service also renews the Access token based on the Refresh token.
https://services.walgreens.com/api/oauthtoken/delete/v1 (for production) https://services-qa.walgreens.com/api/oauthtoken/delete/v1 (for testing)	This service is used to deactivate or disconnect the Access token based on the Access token.



Walgreens OAuth Flow

User goes to 3rd Party site. Clicks on Walgreens application.

Request should carry 3rd Party Client ID and redirect URI

User is given walgreens.com login page. Login page is hosted by walgreens.com.

User needs to authenticate himself and provide approval for the 3rd Party to use his account

After successful authentication and approval. Walgreens will generate Authorization Code.

Authorization code is given to the redirect URI

Calling Token Endpoint to exchange the Auth Grand to Access token. Provide the Redirect URI

Server will validate the Auth Grand and issue a Access Token, Expiration time and Refresh token

Access Token, Expiration time, Refresh Token

Access the protected services with Access Token

Return results

Pass the Refresh Token to review the AccessToken

Access Token, Expiration time, Refresh token

Request OAuth Authorization Code (valid for only 2 minutes)

The third party client will navigate to the following URL (along with passing parameters outlined below) in order to complete the required Walgreens.com login or registration process.

- https://www.walgreens.com/oauth/authorize.jsp (for production)
- https://m-qa2.walgreens.com/oauth/authorize.jsp (for testing)

Name	**Data Type**	**Max Length**	**Required**	**Description**
response_type	String	4	YES	Value MUST be set to "code".
client_id	String	40	YES	The client identifier which is registered with Walgreens. Also known as the affiliate Id.
redirect_uri	String	200	YES	Redirect URI once the authorization code is generated. The value of this field should not be encoded.
scope	String	50	YES	The scope of the access request and it is separated by space. For the Balance Rewards API, the applicable scope is steps.
transaction_id	Number	16	YES	A unique identifier for each transaction. It's the responsibility of each partner to create a unique ID.
channel	Number	2	Optional	A channel through which a user / service is interacting. Acceptable values: 1 – If the channel is WEB 5 – If the channel is Mobile 6 – If the channel is Tablet
state	String	100	Optional	Any alphanumeric value to maintain the state of the transaction.

Sample URL:

```
https://www.walgreens.com/oauth/authorize.jsp?response_type=code&client_id=abcd123
&redirect_uri=https://www.abcd.com/redirect&scope=steps
&transaction_id=1234567890123456&channel=1&state=abcd1234xyz
```

Response OAuth Authorization Code

Upon successfully logging into Walgreens.com or registering with Walgreens.com, the authorization server validates the incoming parameters, verifies the client registration, and then the authorization server issues an authorization code. The authorization code will be embedded within the redirect URI as a URL parameter.

Name	**Data Type**	**Max Length**	**Required**	**Description**
code	String	100	YES	The authorization code generated by the authorization server. Code is only valid for 2 minutes.
transaction_id	Number	16	YES	The exact value received from the request.
scope	String	50	YES	A scope that is associated to the Authorization Code. For the Balance Rewards API, the applicable scope is steps.
state	String	100	Optional	The exact value which is received from the request.

Sample URL redirect response:

```
GET Response
<<redirect_uri>>?code=SplxlOBeZQQYbYS6WxSbIA&transaction_id=1234567890123456&scope=ste
ps&state=abcd1234xyz
```

Sample Error Response:

```
GET Response
<<redirect_uri>>?error_code=1001&error=invalid_request&error_description=Redirect URI
is invalid&transaction_id=1234567890123456&state=abcd1234xyz
```

Request OAuth Access Token
(Access token expires in 30 days and refresh token expires in 45 days)

The third party client requests the authorization server for access to the token by sending the authorization code, along with other parameters detailed below, which is shared by authorization server using the "application/x-www-form-urlencoded" format. The Access Token request should be a POST method.

Name	**Data Type**	**Max Length**	**Required**	**Description**
grant_type	String	18	YES	Value MUST be set to "authorization_code".
client_id	String	40	YES	The client identifier which is registered with Walgreens. Also known as the Affiliate Id.
client_secret	String	40	YES	The client secret code which is registered with Walgreens. Also known as the API key.
code	String	100	YES	The authorization code received from the authorization server.
redirect_uri	String	200	YES	Redirect URI once the authorization code is generated. The value of this field should not be encoded.
transaction_id	Number	16	YES	A unique identifier for each transaction. It’s the responsibility of each partner to create a unique ID.
channel	Number	2	Optional	A channel through which a user / service is interacting. Acceptable values: 1 – If the channel is WEB

				5 – If the channel is Mobile 6 – If the channel is Tablet
act	String	13	YES	The value should be "getOAuthToken".
state	String	100	YES, but only if "state" was in the part of the initial Authorization Code request	The exact value which is received from the Authorization Code request.

Sample Request:

```
POST /generateToken HTTP/1.1

Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&client_id=abcd123&client_secret=secretcode
&code=SplxlOBeZQQYbYS6WxSbIA&redirect_uri=https://www.abcd.com/redirect
&transaction_id=1234567890123457&channel=1&act=getOAuthToken&state=abcd1234xyz
```

Response OAuth Access Token

If the authorization token request is valid and authorized, the authorization server issues an access token and refresh token. Response type is JSON.

Name	Data Type	Max Length	Required	Description
access_token	String	100	YES	The access token. Expires in 30 days. The same access_token will be returned if trying retrieve OAuth token for the same person.
refresh_token	String	100	YES	The refresh token. Expires in 45 days. The refresh token will be used to renew the token once the access token is expired.

token_type	String	6	YES	The value is “Bearer” as this is a Bearer token type.
access_token _expires_in	DateTime		YES	The access token expiration period in date time format. Format is “YYYY-MM-DD hh:mm:ss”
refresh_token _expires_in	DateTime		YES	The refresh token expiration period in date time format. Format is “YYYY-MM-DD hh:mm:ss”
scope	String	50	YES	The scope that is associated with the Access token. For the Balance Rewards API, the applicable scope is steps.
transaction_id	Number	16	YES	The exact value received from the request. It’s the responsibility of each partner to create a unique ID.

Sample Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
   "access_token":"2YotnFZFEjr1zCsicMWpAA",
   "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
   "token_type":"Bearer",
   "access_token_expires_in":"2014-05-30 23:05:47",
   "refresh_token_expires_in":"2014-06-14 23:05:47",
   "scope":"steps",
   "transaction_id":"1234567890123457"
}
```

Sample Error Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
   "error_code":"1041",
   "error":"invalid_authcode",
   "error_description":"Authorization code is invalid or expired",
   "transaction_id":"1234567890123457"
}
```

Request Refresh OAuth Access Token

If the authorization token request is valid and authorized, the authorization server issues an access token and refresh token. Response type is JSON.

Name	Data Type	Max Length	Required	Description
grant_type	String	13	YES	Value MUST be set to "refresh_token".
client_id	String	40	YES	The client identifier which is registered with Walgreens. Also known as the affiliate ID.
client_secret	String	40	YES	The client secret code which is registered with Walgreens. Also known as the API key.
refresh_token	String	100	YES	The refresh token issued to the client.
redirect_uri	String	200	YES	Redirect URI once the authorization code is generated. The value of this field should not be encoded.
transaction_id	Number	16	YES	A unique identifier for each transaction. It's the responsibility of each partner to create a unique ID.

channel	Number	2	Optional	A channel through which a user / service is interacting. Acceptable values: 1 – If the channel is WEB 5 – If the channel is Mobile 6 – If the channel is Tablet
act	String	13	YES	The value should be “getOAuthToken”.

Sample Request:

```
POST /generateToken HTTP/1.1

Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&client_id=abcd123&client_secret=secretcode
&refresh_token=tGzv3JOkF0XG5Qx2TlKWIA
&redirect_uri=https://www.abcd.com/redirect
&transaction_id=1234567890123458&channel=1&act=getOAuthToken
```

Response Refresh OAuth Access Token

If the authorization token request is valid and authorized, the authorization server issues an access token and refresh token. Response type is JSON.

Name	Data Type	Max Length	Required	Description
access_token	String	100	YES	The access token.
refresh_token	String	100	YES	The refresh token. This will be used to renew the token once the access token is expired.
token_type	String	6	YES	The value is “Bearer” as this is a Bearer token type.
access_token _expires_in	DateTime		YES	The access token expiration period in DateTime format. Format

				is “YYYY-MM-DD hh:mm:ss”
refresh_token _expires_in	DateTime		YES	The refresh token expiration period in DateTime format. Format is “YYYY-MM-DD hh:mm:ss”
scope	String	50	YES	The scope requested by the client. For the Balance Rewards API, the applicable scope is steps.
transaction_id	Number	16	YES	The exact value received from the request.

Sample Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
   "access_token":"2YotnFZFEjr1zCsicMWpAA",
   "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
   "token_type":"Bearer",
   "access_token_expires_in":"2014-05-30 23:05:47",
   "refresh_token_expires_in":"2014-06-14 23:05:47",
   "scope":"steps",
   "transaction_id":"1234567890123458"
}
```

Sample Error Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
"error_code":"1051",
"error":"invalid_token",
"error_description":"Token is invalid or expired",
"transaction_id":"1234567890123458"
}
```

Unsubscribe users from posting data to Walgreens (Removing OAuth token)

To discontinue and unlink the third party application's user from the Walgreens program, utilize:

- https://services.walgreens.com/api/oauthtoken/delete/v1 (for production)
- https://services-qa.walgreens.com/api/oauthtoken/delete/v1 (for testing)

The format of the request is as follows:

Name	Data Type	Max Length	Required	Description
client_id	String	40	YES	The client identifier which is registered with Walgreens. Also known as the Affiliate ID.
client_secret	String	40	YES	The client secret code which is registered with Walgreens. Also known as the API key.
token	String	100	YES	The access token.
transaction_id	Number	16	YES	A unique identifier for each transaction. It's the responsibility of each partner to create a unique ID.
channel	Number	2	Optional	A channel through which a user / service is interacting. Acceptable values: 1 – If the channel is WEB 5 – If the channel is Mobile 6 – If the channel is Tablet
act	String	15	YES	The value should be "deactivateToken".

Sample Request:

```
POST /deactivateToken HTTP/1.1

Content-Type: application/x-www-form-urlencoded

client_id=abcd123&client_secret=secretcode &token=2YotnFZFEjr1zCsicMWpAA
&transaction_id=1234567890123458&channel=1&act=deactivateToken
```

Response Removing OAuth Token

If the Deactivate request is valid and access token is active then the system will deactivate the access token. The response parameters are:

Name	Data Type	Max Length	Required	Description
status	String	7	YES	The value is “Success”
transaction_id	Number	16	YES	The exact value received from the request.

Sample Response:

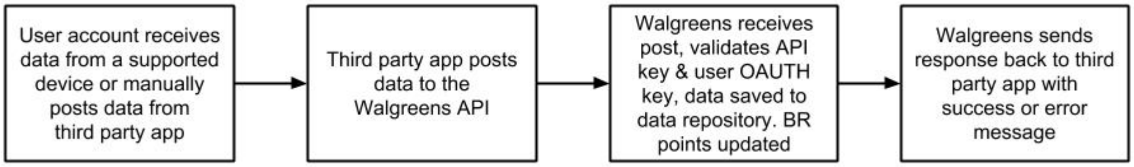
```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
   "status":"success",
   "transaction_id":"1234567890123458"
}
```

Sample Error Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
   "error_code":"1051",
   "error":"invalid_token",
   "error_description":"Token is invalid or expired",
   "transaction_id":"1234567890123458"
}
```

Posting of Activity Data

Once the on-boarding process has been completed, Walgreens is ready for activity data to be posted whenever new data has been received by the third party application from a supported device or when the user has saved a manual entry via a supported biometric device. The process of posting data is summarized by the following flow.


User account receives data from a supported device or manually posts data from third party app
Third party app posts data to the Walgreens API
Walgreens receives post, validates API key & user OAUTH key, data saved to data repository. BR points updated
Walgreens sends response back to third party app with success or error message

This process is an app-to-app (cloud-to-cloud) transaction and does not present a client-side user interface to the user. The developer has the option to post the data immediately upon receiving the data from the user, or in can be posted in batches on a scheduled basis.

NOTE: Walgreens can support an initial data sync of up to seven (7) days of activity data for which those activities will be awarded points as long as the monthly maximum limit has not been reached.

The following End point shall be used to post the activity data.

- https://services.walgreens.com/api/steps/activity/v1 (for production)
  - Please refer to Appendix D, Section 1 for additional details around API key inclusion
- https://services-qa.walgreens.com/api/steps/activity/v1 (for testing)
  - Please refer to Appendix D, Section 1 for additional details around API key inclusion

The process of posting data is summarized by the following flow:

JSON Request Parameters

Field	Required	Type	Description	Data Range
access_token	YES	String	The access token	100
affiliate_id	YES	String	The Client Id	255 Characters
transaction_id	YES	Number	A unique identifier for each transaction. It's the responsibility of each partner to create a unique ID.	16
date	YES	Date	Date of activity or measurement "yyyy-mm-dd" (Note: event date corresponds to user's time zone) Date of the activity cannot be a future date and cannot be more	N/A

			than 30 days in the past. Please refer to Appendix D, Section 2 for additional details.	
user_device_id	YES	Int	User Device ID of specific device or app. The user_device_id is unique to each third party account that the third party vendor creates when the account gets created.	32-bit Integer
manufacturer_name	YES	String	Name of Third Party Company For example: Lose It, RunKeeper, Fitbit, etc.	255 Characters
device_name	YES	String	Name of Device Third Party App/Device Name For example: Lose It iPhone, Fitbit Flex, Jawbone UP24, etc. Please note that this value will display publicly to the connected customer within the "Device" field on the Walgreens Balance Rewards Activity Dashboard.	255 Characters
data	YES	Array	JSON Array of device data	N/A (see info below)

Data

The JSON Array of Data will have the following format:

Field	**Required**	**Type**	**Description**	**Data Range**
Id	YES	String	A unique identifier for each unique activity for each unique day for each unique user. Please refer to Appendix A for additional details.	60 Characters
type	YES	String	Field Name of data (See Field Name in Values table below)	40 characters

			System may accept activity types besides those outlined in the table below. However, Walgreens will not process the record, nor award any points.	
device_tracked	YES	String	true - data measured from a device false - data manually entered by User. If a user manually enters the data, the device_tracked field value should be 'false'. If the data is synced by a device, the device_tracked field value should be 'true'	true/false
timestamp	YES	DateTime	yyyy-MM-dd hh:mm:ss timestamp of activity Please note that the timestamp corresponds to user's time zone. This indicates the timestamp when the user has either synced the device or manually entered activity in the app.	N/A
value	YES	Various	JSON Array of data values (See Value Definitions Below)	N/A

Note: Difference between the fields of "date" versus "timestamp":

- The "date" field is meant to determine the date in which the activity data request is being sent to Walgreens and not the date of the fitness activity that the end customer completed.
- The "timestamp" field is the time in which the actual fitness activity was completed by the end customer and will correspond to end customer's time zone.

For example, if a vendor is sending Walgreens a end customer's past steps activity data today:

- The "date" field should have today's date.
- The "timestamp" field for each day's "total_steps" activity should have corresponding times at which the activities were performed.

Type definitions and corresponding value definitions (available now):

Field	Additional Field	Required	Type	Description	Units	Data Range
total_steps		YES	Int	Number of Steps	Steps	0-100000
walking	duration	YES	Int	Duration of Exercise	Seconds	0-86400
	distance	YES	Float	Distance Traveled	Miles	0-1400
	steps	Optional	Int	Number of Steps	Steps	0-100000
running	duration	YES	Int	Duration of Exercise	Seconds	0-86400
	distance	YES	Float	Distance Traveled	Miles	0-1400
	steps	Optional	Int	Number of Steps	Steps	0-100000
biking	duration	YES	Int	Duration of Exercise	Seconds	0-86400
	distance	YES	Float	Distance Traveled	Miles	0-1400
weight		YES	Float	Value of Weight	Pounds	0-1500
blood_pressure	systolic	YES	Int	Value of systolic blood pressure	MmHg	0-300
	diastolic	YES	Int	Value of diastolic blood pressure	MmHg	0-300
blood_oxygen_ratio		YES	Float	Value of blood oxygen saturation	Percent	0-100
blood_glucose	meal	YES	String	Meal	N/A	"Breakfast", "Lunch", "Dinner",

						"Snack", "Night"
	meal_relation	YES	String	Relation to meal (Pre or Post)	N/A	"Pre", "Post"
	medicine_relation	Optional	String	Medicine relation (Pre or Post)	N/A	"Pre-Medicine", "Post-Medicine"
	value	YES	Float	Value of blood glucose	Mg/dL	0-1000
sleep (no points awarded)	quality	Optional	String	Quality of sleep	N/A	"Good", "Normal", "Bad"
	duration	YES	Int	Duration	Seconds	0-86400

Notes:

- An individual pedestrian activity should be captured using either the total_steps activity field or the walking field or the running field, not a combination of those fields for the same activity.
- For walking & running activities, the conversation rate is 2,000 steps per 1 mile.
- For biking activity, data is required to include the distance in the form of miles.
- For sleep activity, there are no points currently awarded.

Type definitions and corresponding value definitions (not yet available):

The data activity types, listed below, are not currently available for third party integration at this time, but will be made available in the coming months. If a third party integration attempts to send these data field types in an API call, Walgreens will not reject the data, but it will not display to the end customer in the Walgreens activity dashboard.

Field	Additional Field	Required	Type	Description	Units	Data Range
bmi		Optional	Float	Body Mass Index	N/A	0-100
heart_rate		Optional	Int	Heart Rate	BPM	0-1000
fat_ratio		Optional	Float	Percentage of fat	Percent	0-100
fat_free_mass		Optional	Float	Fat free mass	Pounds	0-1500
fat_mass_weight		Optional	Float	Fat mass	Pounds	0-1500
general_exercise	duration	YES	Int	Duration of Exercise	Seconds	0-86400
	distance	YES	Float	Distance traveled	Miles	0-1400
	steps	Optional	Int	Number of Steps	Steps	0-100000
	type	YES	String	Type	All activities other than Walk, Run, Biking	0-255

Appendix A: Activity Posting Data Types

Activity Posting Data utilizes three separate types: creates, updates, deletes.

- Creates – The system would expect the new activity data that is flowing in for the first time in a particular day with a unique activity id.
  - For example if there is an activity that user has logged for a particular day for the first time, as 1000 steps,
  - The system would expect an id 1a2b3c (unique for this activity for this particular day) with a value of 1000.

- Updates – The system would consider the data as update, if it is has an activity id as 1a2b3c.
  - For example, if the above activity 1a2b3c is updated with 2000 steps, the system would expect the same id 1a2b3c with a value of 2000.
  - Otherwise, it would consider it as a new record, and it would be a create.

- Deletes - The system would delete the existing activities that are already sent.
  - For example if the above activity 1a2b3c sent in the delete, this record will be deleted.

```
{
  "creates": [
    {
      "user_device_id": "9167332",
      "date": "2014-05-05",
      "access_token": "7f001e60a8601b2a873b6fcd6b62d45ba47",
      "affiliate_id": "loseit",
      "transaction_id": "904986923560501",
      "manufacturer_name": "Lose IT",
      "device_name": "Lose IT",
       "data": [

        {
          "id": "a323802078a5082fadd3442c032b56c9",
          "type": "total_steps",
          "value": 1999,
          "device_tracked": "true",
          "timestamp": "2014-03-18 02:10:10"
        },


        {
          "id": "a323802078a5082fadd3442c032b56c1",
          "type": "walking",
          "device_tracked": "false",
          "timestamp": "2014-03-18 02:10:10",
               "value": {
                     "steps": 1231,
                     "distance": 2.3,
                    "duration": 7320
                     },
           },

     {
          "id": "d323402073a5014faade442c042a56c3",
          "type": "sleep",
          "device_tracked": "true",
          "timestamp": "2014-03-18 02:10:10",
               "value": {
 "sleep_quality": "Normal",
                             "duration": 28800

                            },
         }
      ]
    }
  ],
  "updates": [
    {
      "user_device_id": "9167332",
      "date": "2013-01-05 ",
      "access_token": "2YotnFZFEjr1zCsicMWpAA",
      "affiliate_id": "12345676",
      "transaction_id": "1234567890123456",
      "manufacturer_name": "Fitbit",
      "device_name": "Fitbit Force",
      "data": [
        {
          "id": "a323802078a5082fadd3442c032b56c4",
          "type": "running",
          "device_tracked": "false",
          "timestamp": "2014-03-18 02:10:10",

          "value": {
            "steps": 1231,
            "distance": 2.3,
            "duration": 7320

          },


        },
        {
          "id": "c32340207ba5012faddb442ca32a56c7",
          "type": "walking",
          "device_tracked": "true",
          "timestamp": "2014-03-18 02:10:10",
          "value": {
            "steps": 2402,
            "distance": 6.9,
            "duration ": 7500

          },
        }
      ]
    }
  ],
  "deletes": [
    {
      "user_device_id": "9167332",
      "date": "2013-01-05",
      "access_token": "2YotnFZFEjr1zCsicMWpAA",
      "affiliate_id": "12345676",
      "transaction_id": "1234567890123456",
      "manufacturer_name": "Fitbit",
      "device_name": "Fitbit Force",
      "data": [
        {
          "id": "b323802478a5082f4dde4d44a3ca56c1",
          "type": "running",
          "device_tracked": "false",
          "timestamp": "2014-03-18 02:10:10",
          "value": {
            "steps": 1231,
            "distance": 2.3,
            "duration": 7320
          },

        }
      ]
    }
  ]
}
```

Example JSON requests:

```
Example of Total Steps data:
{
   "id":"a6e3502053df0131569d1231392d2ff1",
   "type":"total_steps",
   "value":13583,
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10"
}
```

```
Example of Walking:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"walking",
   "device_tracked":"false",
   "timestamp":"2014-03-18 02:10:10",
   "value":{
      "duration":4343,
      "distance":3.24,
      "steps":"7343"
   }
}
```

```
Example of Running:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"running",
   "device_tracked":"false",
   "timestamp":"2014-03-18 02:10:10",
   "value":{
      "duration":4343,
      "distance":3.24,
      "steps":"7343"
   }
}
```

```
Example of Biking:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"biking",
   "device_tracked":"false",
   "timestamp":"2014-03-18 02:10:10",
   "value":{
      "duration":4343,
      "distance":3.24
   }
}
```

```
Example of Weight:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"weight",
   "value":145.32,
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10"
}
```

```
Example of Blood Pressure:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"blood_pressure",
   "value":{
      "systolic":123,
      "diastolic":82
   },
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10"
}
```

```
Example of Blood Oxygen Saturation:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"blood_oxygen_ratio",
   "value":98,
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10"
}
```

```
Example of blood_glucose
{
   "id":"a6e3502053df0131569d123a392xxxxx",
   "type":"blood_glucose",
   "device_tracked":"true",
   "timestamp":"1393394400",
   "value":{
      "meal":"Breakfast",
      "meal_relation":"Pre",
      "medicine_relation":"Pre-Medicine",
      "value":15.2
   }
}
```

```
Example of Sleep:
{
   "id":"a6e3502053df0131569d123a392xxxxx",
   "type":"sleep",
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10",
   "value":{
      "quality":"Normal",
      "duration":34552
   }
}
```

```
Example of BMI:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"bmi",
   "value":23.43,
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10"
}
```

```
Example of Heart Rate:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"heart_rate",
   "value":64,
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10"
}
```

```
Example of Fat Ratio:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"fat_ratio",
   "value":23.43,
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10"
}
```

```
Example of Fat Free Mass:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":" fat_free_mass",
   "value":23.43,
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10"
}
```

```
Example of Fat Mass Weight:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"fat_mass_weight",
   "value":23.43,
   "device_tracked":"true",
   "timestamp":"2014-03-18 02:10:10"
}
```

```
Example of General Exercise:
{
   "id":"a6e3502053df0131569d1231392xxxxx",
   "type":"general_exercise",
   "device_tracked":"false",
   "timestamp":"2014-03-18 02:10:10",
   "value":{
      "duration":4343,
      "distance":3.24,
      "steps":"7343",
      "type":"(Activity Name)"
   }
}
```

Sample Success Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
   "success_code":"2000",
   "transaction_id":"1234567890123457"
}
```

Sample Error Response when token expired:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
   "error_code":"1051",
   "error":" invalid_token ",
   "error_description":"Token code is invalid or expired",
   "transaction_id":"1234567890123457"
}
```

Sample Error Response when processing activities failed:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
   "error_code":"1081",
   "error":"process_activity_error ",
   "error_description":"Error processing one or more activities",
   "transaction_id":"1234567890123457",
   "activities":[
      {
         "id":"b623802078a5012fadde442c032a56z1"
      },
      {
         "id":"b623802078a5012fadde442c032a56x2"
      }
   ]
}
```

Appendix B: API Error Codes

Error Code	**Error Type**	**Description**
1001	invalid_request	Any missing parameters in the request. Invalid response_type. Invalid grant_type. Invalid value in the request params.
1011	unauthorized_client	The client_id is invalid or not registered. The redirect_uri is invalid.
1021	access_denied	When the Authorization Code already exists for the user
1031	invalid_scope	The requested scope is invalid or not authorized to that client.
1041	invalid_authcode	The authorization code which is in the request is invalid or expired.
1051	invalid_token	The access token or refresh token which is in the request is invalid or expired.
1061	temporarily_unavailable	The authorization server is currently unable to handle the request due to a temporary overloading or maintenance of the server.
1071	application_error	Unable register user in Walgreens.com (Light or Balance Rewards registration) due to some internal issues.
1081	process_activity_error	Token validation success but unable process one or more activities. Only failed activity ids will be send in the response object

API Success Codes

Success Code	**Success Type**	**Description**
2000	Success	Token Validation success

Appendix C: Error Responses - Common for All Services:

- Error Response when apiKey(Client_secret) is not provided:

```
{
"errCode":"403",
"apiKey":"",
"errMsg":"Key doesn't Exists"
}
```

- Sample Error Response when apiKey(Client_secret) provided is invalid that doesn’t exist:

```
{
"errCode":"403",
"apiKey":"<APIKEYVALUEPROVIDED>",
"errMsg":"Key doesn't Exists"
}
```

- Error Response when apiKey(Client_secret) provided is valid and doesn’t have access to the API (End Point) accessed:

```
{"err":"403",
"errMsg":"Unauthorized Access"}
```

Appendix D

Section 1- API Key & End Point Information

The Walgreens issued API key is valid for access in both our Testing and Production environments. However, it is the API end point that determines which environment the request is going to be target.

Examples:

For an activity data push to production, the end point should look like:
https://services.walgreens.com/api/steps/activity/v1?apiKey=45a353c32682e997748948def6ccdf34

For activity data push to staging, the end point should look like:
https://services-qa.walgreens.com/api/steps/activity/v1?apiKey=45a353c32682e997748948def6ccdf34

Section 2 – Data Specification

The Activity Date cannot be more than 30 days old and should not be more than one day in the future. There is some buffer on the future day in order to to accommodate differences in time zones.

However, even though the system can accept a date one day in the future, if the date is more than one day ahead, it will be converted to the current date.