Markdown Version  
  
---

nav\_title: Crowdtwist

article\_title: Crowdtwist

description: "This article outlines the partnership between Braze and Crowdtwist, by way of specially created Braze Data-Transformation templates and Crowdtwists Data Push Objects."

page\_type: partner

search\_tag: Partner

---

# Crowdtwist

> [Crowdtwist][0] is a leading cloud-native customer loyalty solution to empower brands to offer personalized customer experiences. The CrowdTwist solution offers over 100 out-of-the-box engagement paths, providing rapid time-to-value for marketers to develop a more complete view of the customer.

CrowdTwist’s Data Push feature allows for user or event metadata to be passed whenever an update occurs in the CrowdTwist’s platform.

This guide outlines how to integrate CrowdTwist’s User Profile, User Activity, and User Redemption Live Push feeds into your Braze environment. At the time of writing, there are two additional Data Push types available which are not explicitly covered in this documentation, but whose setup would follow the same principles as are laid out below.

[#Live Push User Profile][1]**:** Includes creations of new profiles and updates to existing profiles.

[#Live Push User Activity][2]**:** Includes data on user activity completions.

[#Live Push User Redemption][3]**:** Includes data on user reward redemptions.

By using a Braze Data Transformation template, we can filter out the elements of the Data Push that aren’t relevant to Braze, and assign the values that are needed in Braze in such a way so that they can be leveraged by the available “[destinations](?tab=t.0#heading=h.uhdfh3xyrvy)”.

For example, a Data Push can be used to pass relevant custom events and attributes to Braze, like when a user changes loyalty tier or redeems a reward. It can also be used to log custom attributes in Braze as soon as that data is updated on a member’s user profile, like a user’s points balance.

##Prerequisites

| Requirement | Description |

| --- | --- |

| CrowdTwist account | A [CrowdTwist Account[0]] is required to take advantage of this partnership. |

| Braze Data Transformation Endpoint| This integration will rely on Braze’s [Data Transformation Tool][5]. When a Data Transformation is created, Braze will generate a unique endpoint which can be added as a destination for CrowdTwists Data Push|

{: .reset-td-br-1 .reset-td-br-2 .reset-td-br-3 .reset-td-br-4 role="presentation" }

## Integration

Braze and CrowdTwist have created [Data Transformation templates][6] to help our customers develop their own Data Transformations which leverage the User Profile, User Redemption, and User Activity events.

### Step 1: Create Data Transformation from Crowdtwist Template

Navigate to Data Settings > Data Transformation > Create Transformations > Use a Template > Select “BRAZE <> CROWDTWIST” template of your choice.

You will find four templates - one each for transforming User Profile, User Activity, and User Redemption events, and a master template that uses conditional logic to apply to multiple different Data Push events.

As can be seen from [CrowdTwist’s Data Push documentation][4], Data Push objects contain different meta-data so each requires their own transformation code to create appropriate Braze objects. The Master template illustrates how a single Data Transformation could be set up to accept each of the three types of object and will create an appropriate output with values from each object.

###Step 2: Update and Test Template

Below you will see the annotated templates. The body of these templates is designed as if they were being applied to the /users/track destination. Annotations are marked by the “//” line-start and green text, and can be deleted without affecting the operation of the transformation code.

The transformation is written in Javascript, which builds an object, called “brazecall”. This object is where we create the request body that will be sent to a Braze REST API endpoint. Guidance on the required structures of the requests to these destinations can be found through the links of the “destinations” section.

{% alert note %}

Notice that the “values” of each “key” start with `payload.`. The payload represents the data object that has been received from Crowdtwist. We use Javascript dot notation to choose what piece of data we want to populate the elements of our Braze object. Further information about the schema, or makeup of the objects coming from Crowdtwist can be found here[2]

For example, when we see "external\_id": payload.thirdPartyId”, this means that the Braze external ID is going to be set by the “third\_party\_id” value stored in Crowdtwist.

{% endalert %}

{% alert important %}

The objects sent from Crowdtwist can be used to create users in Braze. By including the “update\_existing\_only” key with the value “false”, if an attribute or event object includes an identifier which does not currently exist in Braze, a user profile will be created with the attributes included in the event or attribute object. If you would prefer that Crowdtwist only update profiles that already exist in Braze, set this attribute as “true” in each attribute or event object.

{% endalert %}

###Data Transformation Templates

{% tabs %}

{% tab User Profile Event Template%}

```javascript

let brazecall = {

"attributes": [

{

//Remember that you will need to include an appropriate identifier for your attribute or event object from data available in Crowdtwist. This could be an external ID, braze id, User alias, phone or email address for attribute or event objects

"external\_id": payload.thirdPartyId,

"email": payload.emailAddress,

// \*\*Important\*\* If you want to allow Crowdtwist events to create users in Braze set the value of “\_update\_existing\_only” to false.Otherwise, set this value to TRUE in your event and attribute objects.

"\_update\_existing\_only": false,

"crowdtwist\_loyalty\_points": payload.redeemablePoints,

//In this example, the “tierInfo” object from Crowdtwist is transformed into a Braze Nested Custom Attribute. We use the “\_merge\_objects” value to avoid duplications in a data-point efficient manner.

"tierinfo\_current\_level": payload.tierInfo.currentLevel,

"\_merge\_objects" : true,

"tierInfo" : {

"resetDate": payload.tierInfo.resetDate,

"dateReached":payload.tierInfo.dateReached,

"scoreNeededToReach": payload.tierInfo.scoreNeededToReach,

"nextLevel":{

"minValue":payload.tierInfo.nextLevel.minValue,

"maxValue":payload.tierInfo.nextLevel.maxValue,

"title":payload.tierInfo.nextLevel.title

}

}

}

]

,

//Below we’ll show how both custom attributes and events can be created from a single Crowdtwist User Profile Object.

"events": [

{

"external\_id": payload.thirdPartyId,

"email": payload.emailAddress,

"name": "assignedByEvent",

//Below we can see how a timestamp can be written in your object, which will be a required value for some objects, like the Event Object.

"time": new Date().toISOString(),

"properties": {

"assigned\_by\_event": payload.tierInfo.assignedByEvent,

"date\_assigned": payload.tierInfo.dateAssigned

},

"\_update\_existing\_only": false

}

]

};

// After the /users/track request is assigned to brazecall, you will want to return brazecall to create an output

return brazecall;

```

{% endtab %}

{% tab User Activity Event Template %}  
```javascript

let brazecall = {

"events": [

{

"external\_id": payload.thirdPartyId,

"\_update\_existing\_only": false,

"activityId": payload.activityId,

"name": payload.activityName,

"time": new Date().toISOString(),

"properties": {

"description": payload.description,

"date\_assigned": payload.dateAwarded

},

"\_update\_existing\_only": true

}

]

};

return brazecall;

```  
{%endtab%}  
{%tab Redemption Event Template %}

```javascript

let brazecall = {

"attributes": [

{

"external\_id": payload.thirdPartyId,

//perhaps it is likely that a user redemption event may not have a third party id, in which case you may wish to instead provide the opportunity to include a user alias.

"user\_alias": { "alias\_name" : "crowdtwist\_redemption\_username", "alias\_label" : payload.userName},

"\_update\_existing\_only": false,

"redeemed\_coupon": payload.couponCode,

"total\_points\_redeemed": payload.totalPointsRedeemed

}

]

}

return brazecall;

```  
{%endtab%}  
{%tab Master Template %}  
```javascript

//The master template uses Javascript’s conditional operators to determine what the output of the Data Transformation will be. This example shows how Javascript can be applied to your Transformation to allow for a dynamic range of sources or inputs.

// We open the Transformation with a simple “if” function. We’re simply checking if the value “payload.tierInfo” is present. “tierInfo” is a value which is always populated in the User Profile Live Push object, but is not present in the others.

if (payload.tierInfo) {

let brazecall = {

"attributes": [

{

"external\_id": payload.third\_party\_id,

"email": payload.emailAddress,

"\_update\_existing\_only": false,

"crowdtwist\_loyalty\_points": payload.redeemablePoints,

"tierinfo\_current\_level": payload.tierInfo.currentLevel,

"\_merge\_objects" : true,

"tierInfo" : {

"resetDate": payload.tierInfo.resetDate,

"dateReached":payload.tierInfo.dateReached,

"scoreNeededToReach": payload.tierInfo.scoreNeededToReach,

"nextLevel":{

"minValue":payload.tierInfo.nextLevel.minValue,

"maxValue":payload.tierInfo.nextLevel.maxValue,

"title":payload.tierInfo.nextLevel.title

}

}

}

]

,

"events": [

{

"external\_id": payload.thirdPartyId,

"email": payload.emailAddress,

"name": "assignedByEvent",

"time": new Date().toISOString(),

"properties": {

"assigned\_by\_event": payload.tierInfo.assignedByEvent,

"date\_assigned": payload.tierInfo.dateAssigned

},

"\_update\_existing\_only": false

}

]

};

return brazecall;

//Now we'll use an “else if” operator to change the "brazecall" body if the Object is a User Activity event, by checking if the unique key “activityId” has been populated.

} else if (payload.activityId) {

let brazecall = {

"events": [

{

"external\_id": payload.thirdPartyId,

"\_update\_existing\_only": false,

"activityId": payload.activityId,

"name": payload.activityName,

"time": new Date().toISOString(),

"properties": {

"description": payload.description,

"date\_assigned": payload.dateAwarded

},

"\_update\_existing\_only": true

}

]

};

return brazecall;

//Finally, this conditional statement will trigger if the Data Push Object is a User Redemption event, based on whether the key “rewardId” is populated by a value

} else if (payload.rewardId) {

let brazecall = {

"attributes": [

{

"external\_id": payload.thirdPartyId,

"\_update\_existing\_only": false,

"redeemed\_coupon": payload.couponCode,

"total\_points\_redeemed": payload.totalPointsRedeemed

}

]

}

return brazecall;

} else {

//This error message can be included which will help for troubleshooting in the log if a call should fail. The text in the parentheses can be replaced with anything that might be clearer to your team based on your Data Transformation

throw new Error("No appropriate Identifiers found");

}

```  
{%endtab%}

{% endtabs %}

####Destinations

The templates in this guide have been created as if they are being delivered to the “Track Users” destination, but your template can be designed to send to any of the endpoints listed [here][7], with the support of the associated [[Rest API documentation](https://www.braze.com/docs/api/home)][8].

####Testing

Once you have modified the template to your liking, you must validate that it is operating correctly. Click “Validate” to return a preview of your code’s output and to check if it is an acceptable request for your chosen destination.

![Screenshot of Braze Data transformation UI]({% image\_buster /assets/img/DIRECTORY/IMAGE.png %}){: style="max-width:70%;margin-bottom:15px;border:none;"}

Once you’re happy with the object you see in the “output” field, click “Activate” so that the Data Transformation endpoint will be ready to accept data.   
  
You’ll find your Data Transformation’s webhook URL on the left-hand side panel. Copy this and use it for configuration within CrowdTwist’s Integration Hub.

{% alert important %}

Keep in mind that the Braze Data Transformation endpoints have [a Rate Limit of 1000 requests per minute](https://www.braze.com/docs/user_guide/data/data_transformation/overview/#rate-limits). Consider the speed at which you would like this data made available in Braze, and speak to your Braze Account Manager if you think you’ll require a higher Data Transformation Rate Limit.

{% endalert %}

Data Transformations are a very dynamic tool and can be designed for purposes beyond what has been laid out above with an understanding of Javascript and with the guidance of our REST API documentation. For support or troubleshooting on complex changes to your Data Transformation templates, talk to your Customer Success Manager to learn about the guidance that is available to you.

[0]: https://www.oracle.com/uk/cx/marketing/customer-loyalty/

[1]: <https://docs.oracle.com/en/cloud/saas/marketing/crowdtwist-develop/Developers/PushUserProfile-withTiersv2.html>

[2]: <https://docs.oracle.com/en/cloud/saas/marketing/crowdtwist-develop/Developers/LivePushUserActivity.html>

[3]:<https://docs.oracle.com/en/cloud/saas/marketing/crowdtwist-develop/Developers/LivePushUserRedemption.html>

[4]: <https://docs.oracle.com/en/cloud/saas/marketing/crowdtwist-develop/Developers/DataPush.html>

[5]: <https://www.braze.com/docs/user_guide/data/data_transformation/overview>

[6]: <https://www.braze.com/docs/user_guide/data/data_transformation/creating_a_transformation?redirected=1#step-2-create-a-transformation>

[7]: <https://www.braze.com/docs/user_guide/data/data_transformation/creating_a_transformation/#step-2-create-a-transformation>

[8]: <https://www.braze.com/docs/api/home>