



*MicroMacro Mobile Inc.*

# Pass API User Guide

Version 2.2.3

March 1, 2023

## **Version History**

Version	UpdatedAt	Note
1.0	June 16, 2015	The first Pass API documentation
2.0	February 1, 2018	Rebuild Pass API from the ground up to be simpler, more consistent, and truly RESTful
2.1	December 10, 2018	Add pass issue and installation event callbacks to webhook urls
2.2	Jan 24, 2019	Add get pass status API
2.2.1	July 22, 2019	Correct several typos
2.2.2	May 1, 2022	Add NFC message
2.2.3	March 1, 2023	Add pass status

## **Get Started with the Pass API**

With Pass2U Pass API v2, we've rebuilt this API from the ground up to be simpler, more consistent, and truly RESTful style without binding to any programming language. Retailers, marketing companies, advertisers, third party ticket/coupon providers, or travel agencies, can easily use Pass API to automate the process of creating and updating mobile coupons, membership cards, punch cards, event tickets, and travel passes in existing platforms or systems.

Pass2U uses API keys to authenticate API users. Before starting using Pass API, you have to go to your Pass2U account page to apply for a 30-days free trial one. If you want to use this API key after expiration, please pay at the account page to extend for one year.

# Fundamental Concepts

The primary concept is that passes are created from issued models. All created passes are located at unique download URLs which are in form of `https://www.pass2u.net/d/{passId}`

## Pass API can be used for :

- Create passes with custom data (e.g. ticket/card number, customer name, email, individual discount...).
- Update passes with specified field changes, so the user's pass would be updated automatically via push notification.
- Delete customized attributes or fields of generated passes

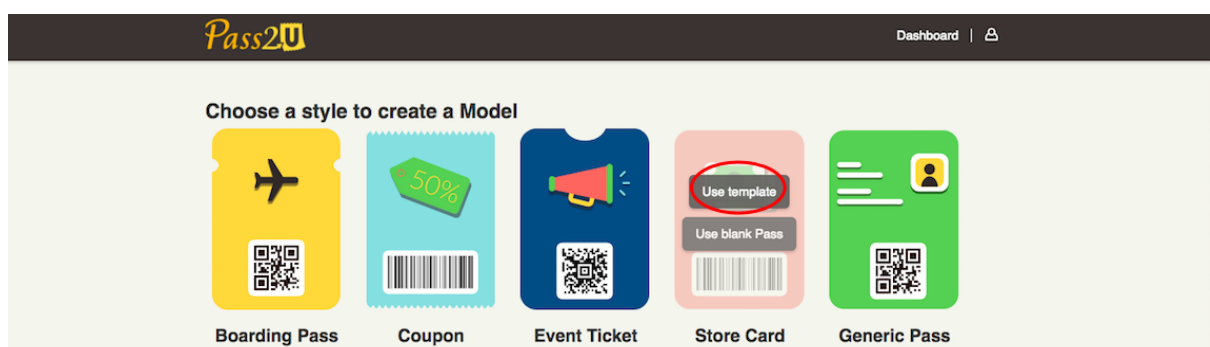
Generated download URLs for passes from API can be distributed through your own channels to customers(e.g. email, SMS, official website, app, facebook...).

## How Does It Work?

1. Login to [Dashboard](#), and click **Create a Model**.

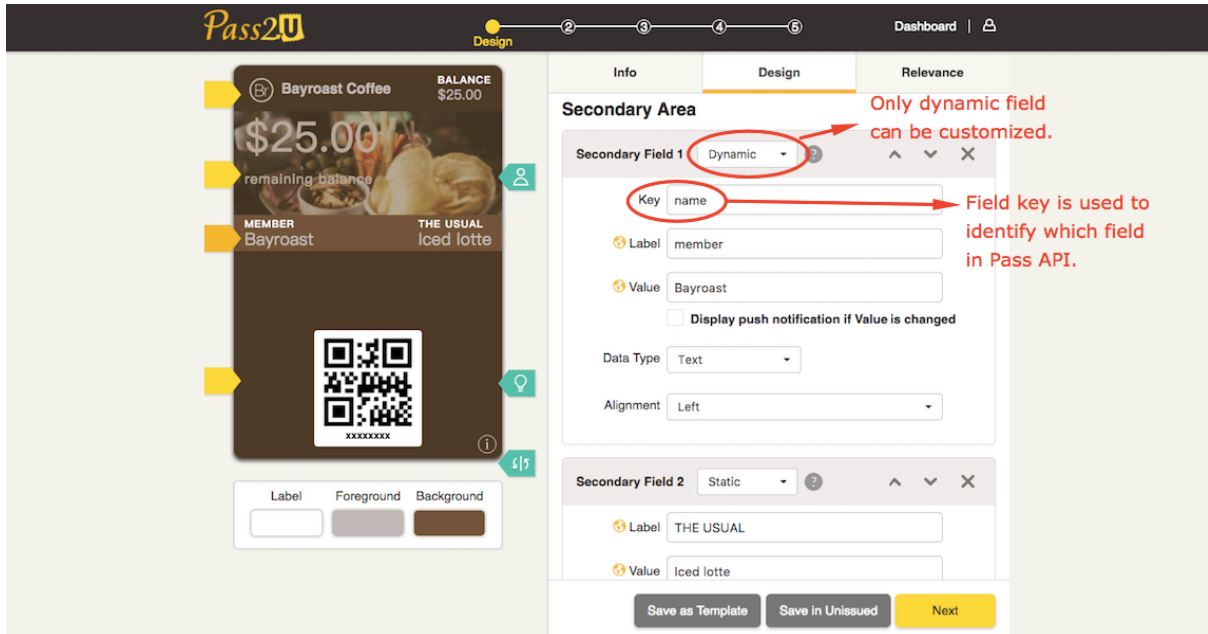


2. Click to select a style you want to distribute to customers. The model's style determines the overall default visual appearance of the passes.

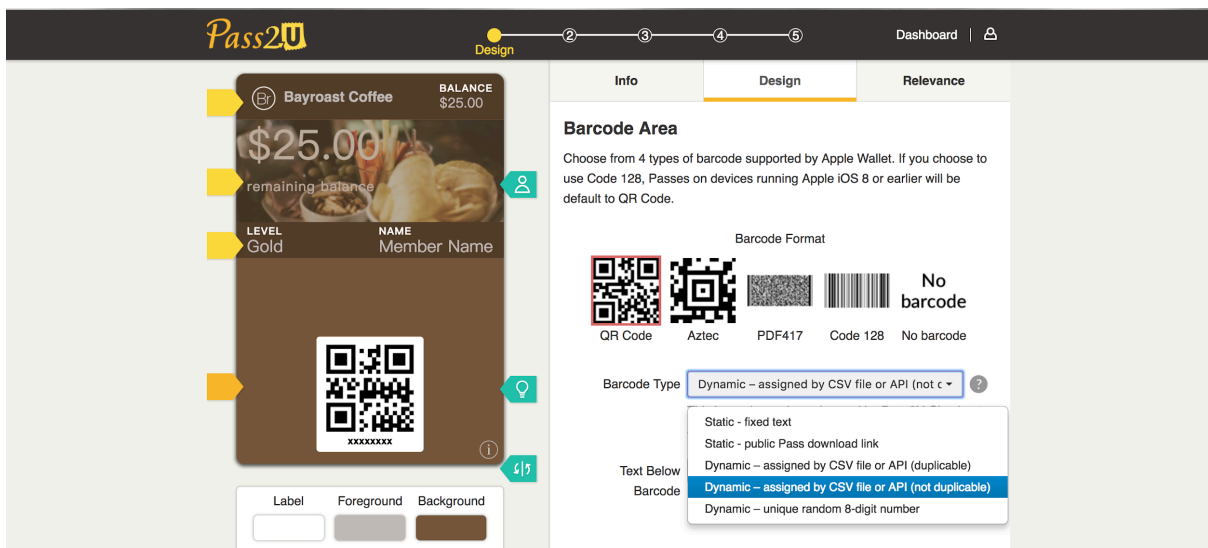


3. On the Model Designer page, you can quickly configure model appearance and the barcode/NFC type. About field type, only **Dynamic** fields (e.g. member name, email, Individual discount, points, balances, and etc.) can be customized

and used in Pass API. In order to identify and customize this field via Pass API, the dynamic field should be named with a unique key. **Fixed** field means all passes of the model use the same data. **Fixed**, **Points**, **Secondary Points**, and **Credits** fields can't be used in the Pass API.

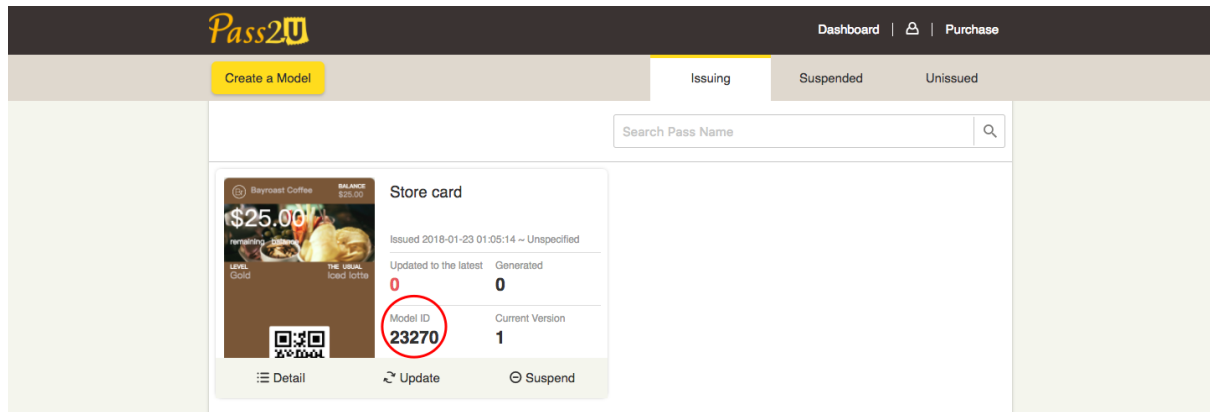


About barcode/NFC data type, **Dynamic – assigned by CSV file or API (duplicable)** or **Dynamic – assigned by CSV file or API (not duplicable)** can be used to specify barcode/NFC messages. **Dynamic – unique random 8-**



**digit number** is used to generate a unique 8-digit number barcode/NFC message for every pass created by Pass2U platform automatically. You needn't specify a barcode/NFC message in Pass2U API.

- After completing model design, please click **Next** until the model is issued. The model appears on the **Issuing** tab in dashboard and you will get the **Model ID** of the model.



- Use the **Model ID** to manipulate the Pass API. After the pass is created, you will get the **passId** in the API response. Please keep it in your system so that passes can be managed individually.

## API Overview

All API URLs referred in the documentation starting with the following endpoint:

<https://api.pass2u.net/v2>

HTTP Method	Path	Function
POST	/models/{modelId}/passes?utm_source={utm_source}	Create pass
PUT	/models/{modelId}/passes/{passId}	Update pass
DELETE	/models/{modelId}/passes/{passId}/attributes/{attributeName}	Reset pass attribute

<b>DELETE</b>	/models/{modelId}/passes/{passId}/fields/{fieldKey}	<a href="#">Reset custom field</a>
<b>DELETE</b>	/models/{modelId}/passes/{passId}/fields/{fieldKey}/attributes/{attributeName}	<a href="#">Reset field attribute</a>
<b>GET</b>	/models/{modelId}/passes/{passId}	<a href="#">Get pkpass file</a>
<b>GET</b>	/models/{modelId}/passes/{passId}/status	<a href="#">Get pass status</a>
<b>POST</b>	/images	<a href="#">Upload pass image</a>

## Authentication

Pass2U uses a 32-characters length of Base64 encoded string as API key to authenticate the request. The API key must be included in all API requests to Pass2U server in header "x-api-key" like the following sample:

```
x-api-key: e6a45207817bb8e9b1f3e67bb696067c
```

API key allows access to your private data in Pass2U. **It should be kept strictly confidential and not shared with others.** If you have to change the API key, please contact Pass2U Support.

## RESTful HTTP Methods

We do our best to use standard HTTP Methods in Pass API:

- Only JSON format is supported as payload of request / response
- Responds with standard HTTP response codes to indicate errors
- Pass2U runs a series of services on AWS
- Pass2U takes PCI DSS (Payment Card Industry Data Security Standard Standard) to protect your data

## API Rate Limits

To help prevent strain on Pass2U's servers, Pass API imposes rate limits per API key. There is a limit of 100 requests per second per API for each API key. If a request

exceeds the limit, Pass2U will return a 429 error along with a message corresponding to which limit was exceeded.

## About API Parameters

All API parameters of requesting payload are listed in the following tables. Please read [Apple Wallet Package Format Reference](#) and [Apple Wallet Pass Programming Guide](#) before using Pass API.

### Top-Level Key attributes

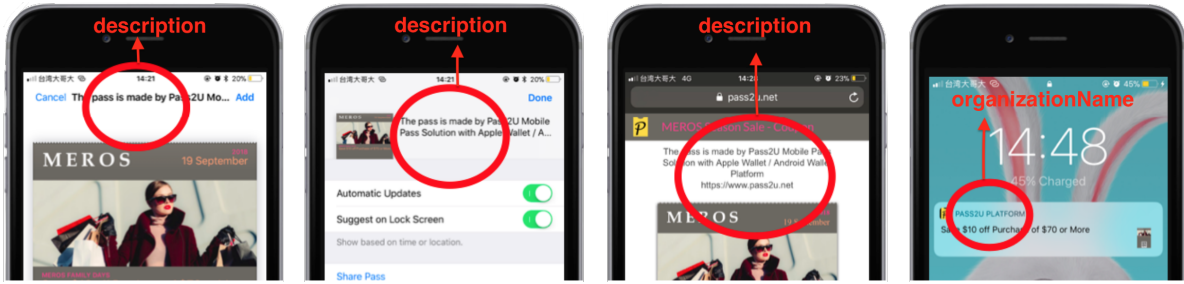
These attributes can be customized and updated when using "[Create pass](#)" or "[Update pass](#)" API. Followings are replacement rules of visual appearance:

- If you don't specify any attributes (e.g. logoText) in request payload, the attributes will be set by model attributes.
- If you specify attributes in request payload, the attributes will be customized for the pass.
- "[Delete pass attribute](#)" API is used to reset the pass attribute to the default attribute of the model.

#### (1) Standard Keys

Name	Type (max length)	Description
description	string (1000)	<b>description</b> is what you describe the campaign about the pass.
organizationName	string (100)	<b>organizationName</b> will be shown on the lock screen while the pass holder is near the specific locations or receives the push notification.
passcode	string (100)	<b>passcode</b> is used to protect the pass download page. If passcode is assigned, pass download page visitors must input correct passcode to get the pass
status	number(1)	<b>status</b> is used to set the pass status. 1 is normal and 2 is not accessible.





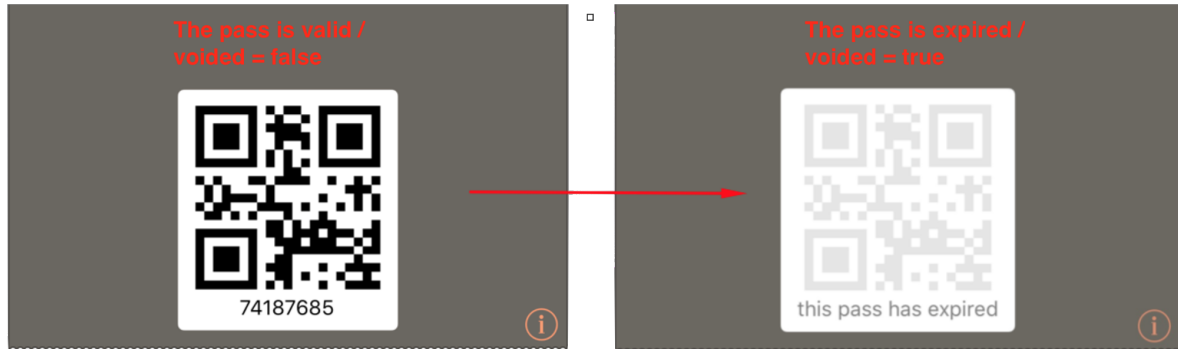
## (2) Associated App Keys

Name	Type	Description
associatedStoreIdentifiers	array of number (11)	A list of iTunes Store item identifiers for the associated iOS apps. Only one item is allowed—the item identifier for an iOS app compatible with the current device. If the app is not installed, clicking the link opens the App Store and shows the app. If the app is already installed, clicking the link launches the app.
appLaunchURL	string (255)	A URL to be passed to the associated iOS app when launched.
associatedPlayIdentifiers	array of string (255)	A list of Google Play identifiers for the associated android apps. Only one item is allowed—the item identifier for an android app compatible with the current device. If the app is not installed, clicking the link opens the Google Play and shows the app. If the app is already installed, clicking the link launches the app.
androidAppLaunchURL	string (255)	A URL to be passed to the associated android app when launched.

## (3) Expiration Keys

Name	Type	Description
expirationDate	string (30)	The date and time in ISO 8601 format to indicate when the pass expires. The value must be a full date format with hours, minutes, and seconds with a time zone. <b>Example: 2018-10-18T16:53:00+08:00</b>

voided	boolean	Indicates if the pass is void—for example, when a one time use coupon that has been redeemed, this key should be set as true . The default value is false.
--------	---------	--



#### (4) Relevance Keys

Name	Type	Description
maxDistance	number	Maximum distance in meters from a relevant latitude and longitude that the pass will be treated as relevant. This number is compared to the pass's default distance and the smaller value is used.
relevantDate	string (30)	For event tickets and boarding passes; otherwise optional. Date and time in ISO 8601 format to indicate when the pass becomes relevant. For example, the starting time of a movie. The value must be a full date format with hours, minutes, and seconds with a time zone. <b>Example: 2018-10-18T16:53:00+08:00</b>

#### (5) Visual Appearance Keys

Name	Type	Description
backgroundColor	string (18)	Background color of the pass, specified as an CSS-style RGB triple. <b>Example: rgb(23, 187, 82)</b>
foregroundColor	string (18)	Foreground color of the pass, specified as a CSS-style RGB

		triple. <b>Example: rgb(100, 10, 110)</b>
groupingIdentifier	string (50)	For event tickets and boarding passes used to group related passes; otherwise not allowed. Identifier. If a grouping identifier is specified, passes with the same style, pass type identifier, and grouping identifier are displayed in a group. Otherwise, passes are grouped by style and pass type identifier.
labelColor	string (18)	Color of the label text, specified as a CSS-style RGB triple. <b>Example: rgb(255, 255, 255)</b>
logoText	string (35)	The logoText displayed next to the logo on the pass.

## Lower-Level Key attributes

These attributes can be updated by "[Create pass](#)" or "[Update pass](#)" API. The following are replacement rules of visual appearance. For example,

- If you don't specify an image array in JSON, the pass will use the default model image.
- Remember to provide all types of images that you want to display on the pass. If not all images are specified, missing images won't be displayed even if default model images are set in Model Designer.

### (1) Image Dictionary Keys (Array)

The model style controls which images can be used. The table shows the images supported by each model style. For details, see [Apple Wallet Developer Guide](#).

Name	Type	Description
images	an array of JSON objects	An array of image objects. Different model style allows different image types. Please refer to <a href="#">Apple Wallet Developer Guide</a> .
type	string (35)	Image types. Must be one of the following values: <b>icon, logo, strip, background, thumbnail, footer</b> .
hex	string (50)	A hexadecimal hash string responded by <a href="#">Upload Pass</a>

		<a href="#">image</a> API. For example, a hexadecimal hash string could be: 01474ead01d4b4ee6ad3bda2b15013888e74c866.
--	--	--

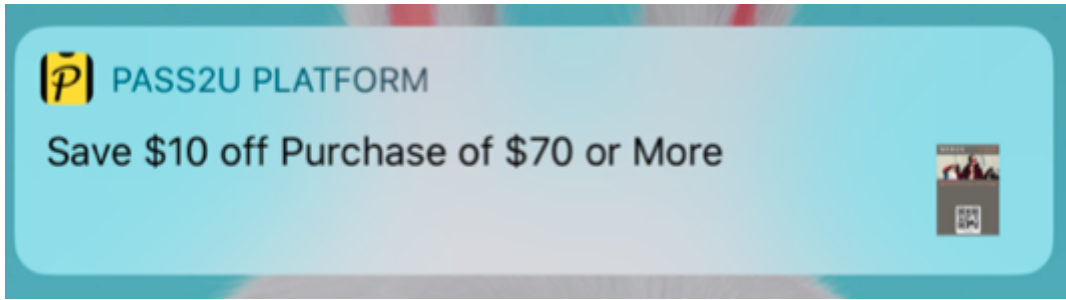
## (2) Pass Structure Dictionary Keys

Name	Type	Description
transitType	string (20)	Required for boarding passes; otherwise not allowed. Type of transit must be one of the following values: <b>PKTransitTypeAir</b> , <b>PKTransitTypeBoat</b> , <b>PKTransitTypeBus</b> , <b>PKTransitTypeGeneric</b> , <b>PKTransitTypeTrain</b> .

## (3) Beacon Dictionary Keys (Array)

You can specify an array of data for up to 10 iBeacons per pass. When the card holder enters within range of a defined iBeacon, a notification will show on the lock screen.

Name	Type	Description
beacons	array of JSON objects	An array of beacon objects. Apple Wallet limits the pass to 10 beacon objects.
major	number (16)	Major identifier of a Bluetooth Low Energy location beacon. It should be a 16-bit unsigned integer.
minor	number (16)	Minor identifier of a Bluetooth Low Energy location beacon. It should be a 16-bit unsigned integer.
proximityUUID	string (36)	Unique identifier of a Bluetooth Low Energy location beacon.
relevantText	string (255)	Text displayed on the lock screen when the pass is currently relevant. For example, a description of the nearby location such as "Store nearby on 1st and Main."

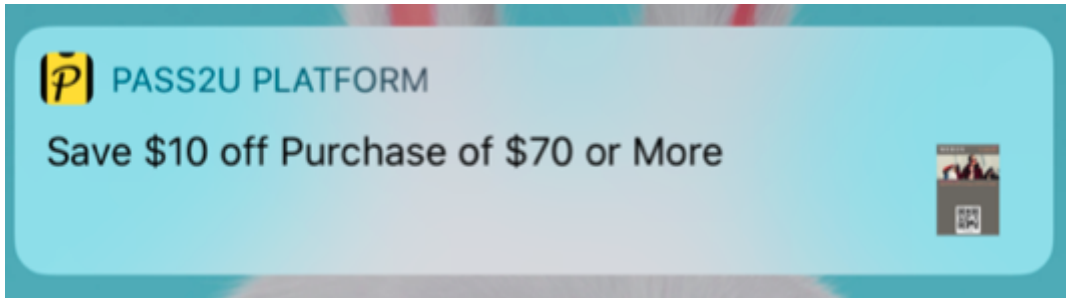


#### (4) Location Dictionary Keys (Array)

You can specify an array of data for up to 10 locations per pass. When the card holder enters within range of a defined location, a notification will show on the lock screen.

Pass Type	Relevant locations
Boarding Pass	With a large radius, on the order of a thousand meters or closer.
Coupon	With a small radius, the current location must be on the order of a hundred meters or closer.
Event Ticket	With a large radius, on the order of a thousand meters or closer.
Store Card	With a small radius, the current location must be on the order of a hundred meters or closer.
Generic	With a small radius, the current location must be on the order of a hundred meters or closer.

Name	Type	Description
locations	array of JSON objects	An array of location objects. Apple Wallet limits the pass to 10 location objects.
altitude	double	Altitude, in meters, of the location.
latitude	double	Latitude, in degrees, of the location.
longitude	double	Longitude, in degrees, of the location.
relevantText	string (255)	Text displayed on the lock screen when the pass is currently relevant. For example, a description of the nearby location such as "Store nearby on 1st and Main."



### (5) Barcode Dictionary Keys

Please note you can't use barcode and nfc key at the same time. If a QR code is required as the alternative identity, please enable "Add QR code as alternative" in Barcode/NFC Data Area in the Model design page.

Name	Type	Description
barcode	an JSON object	A JSON object of barcode.
altText	string (50)	Text displayed under the barcode. For example, a human-readable version of the barcode data in case the barcode can't be scanned.
message	string (255)	Message or payload to be displayed as a barcode.

### (6) NFC Dictionary Keys

Please note you can't use barcode and nfc key at the same time. If a QR code is required as the alternative identity, please enable "Add QR code as alternative" in Barcode/NFC Data Area in the Model design page.

Name	Type	Description
nfc	an JSON object	A JSON object of NFC.
message	string (64)	NFC message.

## Field Dictionary Keys

Every field has its own attributes. Attributes of a field can be given values when using "Create pass" or "Update pass" API. Followings are replacement rules of fields:

- Only **Dynamic** fields can be given values. A field which is configured as **Dynamic** when designing the Model is a **Dynamic** field.

The screenshot shows a configuration window for a field named "Header Field 1". At the top, there is a dropdown menu currently set to "Dynamic", with a help icon (?) to its right. Below this, there are several rows of configuration options, each with a globe icon on the left and a text input field on the right:

- Key**: Input field contains "fi".
- Label**: Input field contains "D".
- Value**: Input field contains "\$".

Below these rows is a checkbox labeled "Display push notification if Value is changed", which is currently unchecked. At the bottom, there are two more dropdown menus:

- Data Type**: Set to "Text".
- Alignment**: Set to "Right".

- If you don't specify custom data to fields in API, the default data of the fields of the model will be used.
- If you don't specify data to attributes of a field in API, the default attributes of the field of the model will be used.
- If you specify custom data for a field in API, API will set the custom data to the field and keep the field data as customized. The field data will not be updated even if updating the model unless you use "**Update pass**" API to update the field data.
- If you want to remove the customized data of the field and reset it to the default field data of the model, please use the "**Delete custom field**" API.
- If you specify an empty value to an attribute of a field (ex: "fields":[ {"key":"field1","label":""} ] ), API will give an empty value to the attribute.

Name	Type	Description
fields	an array of JSON objects	An array of field objects. You can specify an array of field objects that are configured to be <b>Dynamic</b> when designing the model.
key	string (100)	This is a unique identifier for the specified model field that can be defined on Model Designer.
label	string (150)	Label text for the field.
value	string (5000)	Value of the field. It can be a <b>string</b> , <b>ISO 8601 date format as a string</b> , or <b>number</b> .
changeMessage	string (80)	Format string for the alert text that is displayed when the pass is updated. The format string must contain the escape <b>%@</b> , which is replaced by the field's new value. For example, " <b>Get changed to %@</b> ". If you don't specify a change message, the user isn't notified when the field changes.

## Create pass

Create a unique pass from the specified model by customizing attributes and **Dynamic** fields of the model.

### URL Structure

Method	URL
POST	<a href="https://api.pass2u.net/v2/models/{modelId}/passes?utm_source={utm_source}">https://api.pass2u.net/v2/models/{modelId}/passes?utm_source={utm_source}</a>

Name	Type	Required	Description
modelId	number	Yes	The unique ID for the specified model you are using to create the pass. You can find modelId in <a href="#">Dashboard</a> .
utm_source	string	No	The utm_source is the name you give to track the issuing amount of the interested channel. For example, if you have a campaign on social media, on an official website / app, you can use a separate utm_source for each to record the issuing amount automatically in <a href="#">Dashboard</a> .



## HTTP Request Header

**x-api-key:** {x-api-key}

**Accept:** application/json

**Content-Type:** application/json

## Example Request Body

Following is an example of the JSON payload used to create a membership card from the specified model:

The membership card has the member's name, date of birth, member level, profile photo, expiration date and relevant locations that the member preferred. In addition, configuring background color to distinguish different member levels that are intended to enhance users' experience. All values provided will replace default values of the model.

```
{
  "expirationDate": "2018-12-31T23:00:15+08:00",
  "foregroundColor": "rgb(51, 50, 46)",
  "backgroundColor": "rgb(237, 219, 21)",
  "labelColor": "rgb(196, 54, 39)",
  "sharingProhibited": true,
  "fields": [
    {
      "key": "name",
      "label": "Name",
      "value": "Fi-Lin, Chen"
    }, {
      "key": "birth",
      "label": "Birth",
      "value": "Dec'10 1999"
    },
    {
      "key": "level",
      "label": "Level",
      "value": "GOLD"
    }
  ],
  "barcode": {
    "message": "1234567890",
  }
}
```

```

        "altText": "1234567890"
    },
    "images": [
        {
            "type": "thumbnail",
            "hex": "d33b9f1426d363340c2946557f173c402cd74014"
        }
    ],
    "locations": [
        {
            "latitude": 25.0413093,
            "longitude": 121.55269329999999,
            "relevantText": "Store nearby on 1st and Main."
        },
        {
            "latitude": 12.0413093,
            "longitude": 181.3226,
            "relevantText": "Take a break with Bistro Coffee"
        }
    ]
}

```

## HTTP Response Header

**Content-Type:** *application/json*

## Example Response Body

If the call is successful, Pass2U returns **passId** and created time.

```

{
    "barcodeMessage": "1234567890",
    "modelId": 1919,
    "passId": "VT-2I77F5ADz",
    "createdTime": "2018-01-25T16:19:36+08:00",
    "expirationDate": "2018-12-31T23:00:15+08:00"
}

```

Name	Description
barcodeMessage	The Information specified to the pass's barcode.
nfcMessage	The Information specified to the pass's NFC message.
modelId	The unique ID of the model this pass created from.

passId	The unique ID of the created pass you should keep. When updating this pass, please provide this passId.
createdTime	The date and time when the pass is created.
expirationDate	The date and time when a pass expires and whether it is still valid. After the expiration date, the pass is automatically voided.

The publicly accessible URL for downloading the pass is:

**`https://www.pass2u.net/d/{passId}`**

## Update pass

Update a generated pass by adding or changing customizable attributes and dynamic fields of the model. A notification message will be sent to mobile devices of the pass holders and shown on lock screens (e.g. latest news, balances, points, special discount, customer information, records...).

### URL Structure

Method	URL
PUT	<b><code>https://api.pass2u.net/v2/models/{modelId}/passes/{passId}</code></b>

Name	Type	Required	Description
modelId	number	Yes	The unique ID of the specified model you want to update the specified pass from. You can find modelId in <a href="#">Dashboard</a> .
passId	string	Yes	The unique ID of the specified pass you want to update.

### HTTP Request Header

***x-api-key:*** {x-api-key}

***Accept:*** application/json

***Content-Type:*** application/json

### Example Request Body

```

{
  "expirationDate": "2019-12-31T23:00:15+08:00",
  "foregroundColor": "rgb(51, 50, 46)",
  "backgroundColor": "rgb(211, 79, 46)",
  "labelColor": "rgb(196, 54, 39)",
  "fields": [
    {
      "key": "level",
      "label": "Level",
      "value": "DIAMON",
      "changeMessage": "You get highest and most rewarding level %@"
    }
  ],
}

```

## HTTP Response Header

**Content-Type:** *application/json*

## Example Response Body

If the call is successful, passId and updated time will be returned.

```

{
  "barcodeMessage": "1234567890",
  "modelId": 1919,
  "passId": "VT-2I77F5ADz",
  "updatedAt": "2018-01-25T16:19:36+08:00",
  "expirationDate": "2018-12-31T23:00:15+08:00"
}

```

Name	Description
barcodeMessage	The information specified to the pass's barcode.
nfcMessage	The Information specified to the pass's NFC message.
modelId	The unique ID of the model this pass created from.
passId	The unique ID of the created pass you should keep. Use this passId to manipulate the pass hereafter.
updatedAt	The date and time when the pass is updated.
expirationDate	The date and time when the pass expires.

## Reset pass attribute

Reset a customized [Top-Level Key](#) or [Lower-Level Key](#) attribute of the specified pass to use the default value of the model (e.g. logo text, colors, relevant date, expiration date...).

### URL Structure

Method	URL
DELETE	<a href="https://api.pass2u.net/v2/models/{modelId}/passes/{passId}/attributes/{attributeName}">https://api.pass2u.net/v2/models/{modelId}/passes/{passId}/attributes/{attributeName}</a>

Name	Type	Required	Description
modelId	number	Yes	The unique ID of the specified model you want to reset the custom attributes of the pass from. You can find modelId in <a href="#">Dashboard</a> .
passId	string	Yes	The unique ID of the specified pass you want to reset the custom attributes from.
attributeName	string	Yes	The unique Top-Level Key or Lower-Level Key attribute of the pass you want to reset.

### HTTP Request Header

**x-api-key:** {x-api-key}

**Accept:** application/json

**Content-Type:** application/json

### HTTP Request Body

none

### HTTP Response Header

**Content-Type:** application/json

## HTTP Response Body

Returns an HTTP status of 200 if the call is successful.

none

## Reset custom field

Reset a custom field of the specified pass to use the default field data of the model.

### URL Structure

Method	URL
DELETE	<a href="https://api.pass2u.net/v2/models/{modelId}/passes/{passId}/fields/{fieldKey}">https://api.pass2u.net/v2/models/{modelId}/passes/{passId}/fields/{fieldKey}</a>

Name	Type	Required	Description
modelId	number	Yes	The unique ID of the specified model you want to reset the custom field of the pass from. You can find modelId in <a href="#">Dashboard</a> .
passId	string	Yes	The unique ID of the specified pass you want to reset the field from.
fieldKey	string	Yes	The unique key of the customized field you want to reset. It must be defined in the model.

### HTTP Request Header

**x-api-key:** {x-api-key}

**Accept:** application/json

**Content-Type:** application/json

### HTTP Request Body

none

### HTTP Response Header

**Content-Type:** application/json

## HTTP Response Body

Returns an HTTP status of 200 if the call is successful.

none

## Reset field attribute

Reset an attribute of the customized field to use the default field attribute of the model.

### URL Structure

Method	URL
DELETE	<a href="https://api.pass2u.net/v2/models/{modelId}/passes/{passId}/fields/{fieldKey}/attributes/{attributeName}">https://api.pass2u.net/v2/models/{modelId}/passes/{passId}/fields/{fieldKey}/attributes/{attributeName}</a>

Name	Type	Required	Description
modelId	number	Yes	The unique ID of the specified model you want to reset the custom field attribute of the pass from. You can find modelId in <a href="#">Dashboard</a> .
passId	string	Yes	The unique ID of the specified pass you want to reset the field attributes from.
fieldKey	string	Yes	The unique key of the customized field you want to reset the field attributes from.
attributeName	string	Yes	The name of the field attribute of the specified pass you want to reset.

### HTTP Request Header

**x-api-key:** {x-api-key}

**Accept:** application/json

**Content-Type:** application/json

### HTTP Request Body

none

## HTTP Response Header

**Content-Type:** *application/json*

## HTTP Response Body

Returns an HTTP status of 200 if the call is successful.

none

## Get pkpass file

If you want to distribute a pass via a web page or app other than the Pass2U web page, you can use this API to get the pkpass file directly.

### URL Structure

Method	URL
GET	<a href="https://api.pass2u.net/v2/models/{modelId}/passes/{passId}">https://api.pass2u.net/v2/models/{modelId}/passes/{passId}</a>

Name	Type	Required	Description
modelId	number	Yes	The unique ID of the specified model you want to get the specified pass from.
passId	string	Yes	The unique ID of the specified pass you want to get.

## HTTP Request Header

**x-api-key:** *{x-api-key}*

**Accept:** *application/vnd.apple.pkpass*

**Content-Disposition:** *attachment; filename=pass.pkpass*

## HTTP Response Header

**Content-Type:** *application/vnd.apple.pkpass*



## HTTP Response Body

Returns an HTTP status of 200 if the call is successful.

```
MIME type .pkpass file binary stream.
```

## Get pass status

### URL Structure

Method	URL
GET	<a href="https://api.pass2u.net/v2/models/{modelId}/passes/{passId}/status">https://api.pass2u.net/v2/models/{modelId}/passes/{passId}/status</a>

Name	Type	Required	Description
modelId	number	Yes	The unique ID of the specified model you want to query the specified pass from.
passId	string	Yes	The unique ID of the specified pass you want to query.

### HTTP Request Header

**x-api-key:** {x-api-key}

**Accept:** application/json

### HTTP Request Body

```
none
```

### HTTP Response Header

**Content-Type:** application/json

### HTTP Response Body

Returns an HTTP status of 200 if the call is successful.

```
{  
  "passId": "VT-2I77F5ADz",  
}
```

```

"devices": [
  {
    "brand": "iPhone",
    "model": "XR",
    "installedATime": "2019-12-31T23:00:15+08:00"
  }
],
"redemptions": [
  {
    "campaignId": 2455,
    "campaignName": "NIKE Run",
    "checkoutAccount": "Betty",
    "storeName": "HK No.1",
    "redeemedTime": "2019-12-31T23:00:15+08:00"
  }
],
}

```

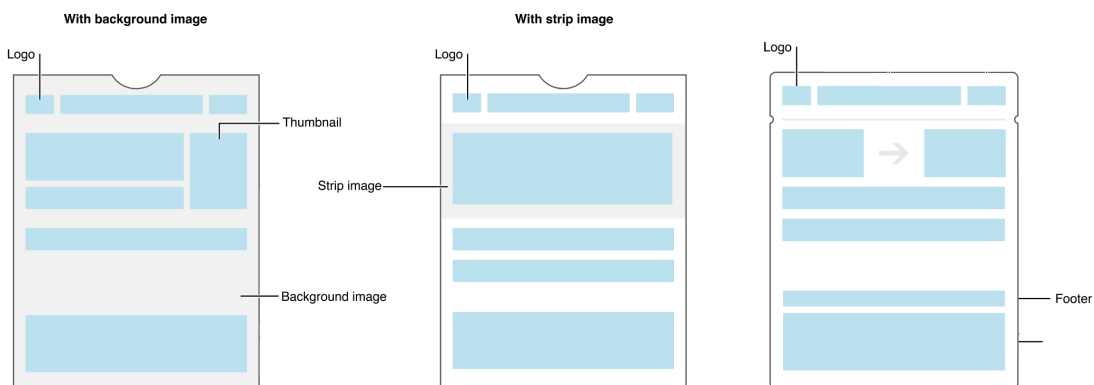
Name	Description
passId	The unique ID of the created pass you query.
devices	An array to indicate which devices ever install the pass.
brand	The brand name of the installed device.
model	The model name of the installed device.
installedTime	The date and time when the pass installed in the device.
redemptions	An array to indicate which one-time redemption campaigns redeem the pass.
campaignId	The ID of the redemption campaign.
campaignName	The name of the redemption campaign.
checkoutAccount	The Pass2U Checkout account name who redeemed the pass.
storeName	The name of the store which Pass2U Checkout account who redeemed the pass belongs to.
redeemedTime	The date and time when the pass is redeemed.

## Upload pass image

When creating or updating the pass, you may want to replace the default model image with a new image. The model style defines which types of images can be assigned. Please refer to [Apple Wallet Developer Guide](#) to learn more about image types for

different styles.

Pass Type	Supported images
Boarding Pass	Icon, Logo, Footer
Coupon	Icon, Logo, Strip
Eventticket - Layout 1	Icon, Logo, Background, Thumbnail
Eventticket - Layout 2	Icon, Logo, Strip
Store Card	Icon, Logo, Strip
Generic	Icon, Logo, Thumbnail



## URL Structure

Method	URL
POST	<a href="https://api.pass2u.net/v2/images">https://api.pass2u.net/v2/images</a>

## HTTP Request Header

**x-api-key:** {x-api-key}

**Accept:** application/json

**Content-Type:** image/png

## HTTP Request Body

The image binary data.

## Upload limitations

- Format supported: png
- Maximum file size is: 4MB
- Maximum image dimensions are: 1024 px x 1024 px

## Recommended Image dimensions

Image Type	Width	Height
Icon	58	58
Logo	320	100
Strip	750	288
Thumbnail	180	180
Background	360	440
Footer	572	30

## HTTP Response Header

**Content-Type:** image/png

**Content-Disposition:** attachment; filename={imageName}.png

## Example Response Body

```
{  
  "hex": "89b9a6edb1132bae969711230ef63e3fb367f34e"  
}
```

Name	Description
hex	The unique hex for the image you may use if the pass have to be customized the image.

## Error Codes

HTTP Status Code	Error Code	Retry
400	Bad Request Exception	No
403	Access Denied Exception	No
404	Not Found Exception	No
405	Bad Method	No
405	HTTP Not Acceptable	No
409	Conflict Exception	No
415	HTTP Unsupported Type	No
429	Too Many Requests Exception	Yes
503	Service Unavailable Exception	Yes
504	Endpoint Request Timed-out Exception	Yes

## Sample Error Response

```
{
  "message": "error"
}
```

## Callback to Webhook Url

If you generated a pass which is not created by API or a pass is installed in a user device, you can get the notification by HTTP POST when the relevant webhook urls are specified. You can assign pass generating and installation webhook urls in step 3 “Configure Model Issuing Settings” in the Model issuing process.

Pass2U Issue Settings

Dashboard | Barcode Store | User

If not specified, the pass can be generated unlimitedly till the Model is stopped issuing.

**Expiration restriction**  
Set expiration date on new passes, and the barcodes on passes will be marked by grey color and become invalid at the time. If not specified, the passes won't be expired.

**Get public download link**  
After issuing, you can get a sharable link and share it on social medias or put the link on your official website, or advertising banners for users to download passes.

**Set up required fields to customized passes**  
Select the required fields needed to be filled in while generating the passes by Public download link. It will collect the data to generate customized passes.

姓名 ✓

**Webhooks**  
You can enter any URL you'd like to have to receive the notifications about events that happen to the passes.

Pass generated

Pass installed

Previous Next

Or edit “Issuing Settings” after the model is issued

Pass2U

Dashboard | Barcode Store | User

Info Report Analytics

**Change the issuing settings?**

Allow to get new passes

**Webhooks**  
You can enter any URL to receive the notifications about events of the passes.

Pass generated

Pass installed

Cancel Save

Suspend Generate Pass Issue Setting Send by Email Send by SMS

## Pass Generated Event

### URL Structure

Method	URL
POST	Your pass generating webhook url (HTTPS is required) and will be called by Pass2U

Name	Type	Description
modelId	number	The unique ID of the specified model which the generated pass belongs to. You can find modelId in <a href="#">Dashboard</a> .
passId	string	The unique ID of the generated pass.
distributedAt	string	The date and time when pass is generated in ISO 8601 format.

### HTTP Request Header

**Content-Type:** application/json

### Example Request Body

```
{
  "passId": "fjnUGLFqLst9",
  "distributedAt": "2018-12-10T09:41:50+08:00",
  "modelId": 2126
}
```

## Pass Installed Event

### URL Structure

Method	URL
POST	Your pass installation webhook url (HTTPS is required) and will be called by Pass2U

Name	Type	Description
modelId	number	The unique ID of the specified model which the installed pass belongs to. You can find modelId in <a href="#">Dashboard</a> .
passId	string	The unique ID of the installed pass.
installedAt	string	The date and time when pass is installed in ISO 8601 format.

## HTTP Request Header

**Content-Type:** *application/json*

## Example Request Body

```
{
  "passId": "fjnUGLFqLst9",
  "installedAt": "2018-12-10T09:41:50+08:00",
  "modelId": 2126
}
```