

# 1. Uploading assets

---

The first step is uploading the assets to the Passworks Platform.

For this example we will try to recreate the "Paw Planet" Coupon.



You can find more information about Passworks assets here <https://github.com/passworks/passworks-api/blob/master/v2/sections/assets.md>

Lets start by uploading the follwoing image to the Platform.

## Uploading the assets using the "Postman" (Chrome extension)

Don't for forget to fill the "Authorization" tab first with your API username and password.

Creating a asset is simply POSTting to the `/v2/assets/` endpoint with the `form-data` encoding.

```
POST https://api.passworks.io/v2/assets/
```

Example of the POSTman plugin setup:

http://api.passworks.git:3000 https://api.passworks X + Immendes (backend\_saas) [eye] [gear]

POST https://api.passworks.io/v2/assets/ Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary

key	value	File	Text
asset[file]	Choose Files icon.png	File	
asset[asset_type]	icon	Text	

Body Cookies Headers (15) Tests Status: 201 Created Time: 1070 ms

In the response body the API will return the URL for the asset and the ID, store the ID we will need it (ID **b9836df9-d810-4d99-ba62-acf1205ea16c**).

Body Cookies Headers (15) Tests Status: 201 Created Time: 1070 ms

Pretty Raw Preview JSON

```

1 {
2   "id": "b9836df9-d810-4d99-ba62-acf1205ea16c",
3   "asset_type": "icon",
4   "created_at": "2016-12-19T16:01:46Z",
5   "asset_url": "https://api.passworks.io/uploads/asset/b9836df9-d810-4d99-ba62-acf1205ea16c/original.png"
6 }

```

The same reply as before but text friendly :-)

```

{
  "id": "b9836df9-d810-4d99-ba62-acf1205ea16c",
  "asset_type": "icon",
  "created_at": "2016-12-19T16:01:46Z",
  "asset_url": "https://api.passworks.io/uploads/asset/b9836df9-d810-4d99-ba62-acf1205ea16c/original.png"
}

```

## Uploading the logo.

Now let's upload the "logo.png" as we did before with the "icon.png" we only need to change the asset type to "logo" and select the right image file.

http://api.passworks.git:3000 https://api.passworks × + Immendes (backend\_saas) ▾ 🔍 ⚙️

POST ▾ https://api.passworks.io/v2/assets/ Params Send ▾ Save ▾

Authorization Headers (1) Body ● Pre-request Script Tests Code

● form-data ● x-www-form-urlencoded ● raw ● binary

✓ asset[file] Choose Files logo.png File ▾ ≡ × Bulk Edit

✓ asset[asset\_type] logo Text ▾ ≡ ×

key value Text ▾

Body Cookies Headers (15) Tests Status: 201 Created Time: 1025 ms

Pretty Raw Preview JSON ▾ ≡ 🔍

```
1 {
2   "id": "3e76b50c-bd47-48e9-b8d6-dd097962aa1b",
3   "asset_type": "logo",
4   "created_at": "2016-12-19T16:20:09Z",
5   "asset_url": "https://api.passworks.io/uploads/asset/3e76b50c-bd47-48e9-b8d6-dd097962aa1b/original.png"
6 }
```

This returned the ID 3e76b50c-bd47-48e9-b8d6-dd097962aa1b for the logo, store this value.

```
{
  "id": "3e76b50c-bd47-48e9-b8d6-dd097962aa1b",
  "asset_type": "logo",
  "created_at": "2016-12-19T16:20:09Z",
  "asset_url": "https://api.passworks.io/uploads/asset/3e76b50c-bd47-48e9-b8d6-dd097962aa1b/original.png"
}
```

## 2. Creating the campaign

So this campaign will have the following:

- Two images: 1 icon that you can't see, and the logo.
- Will have a primary field with an "offer" the "20% discount"
- And two auxiliary fields, one the "customers name", this is not in the provided image but will added it now to make this example a little bit more complex and realistic and also with the expire date "2 weeks".
- Barcode: We will use a qrcode.

Let's create the campaign with the following payload:

```

{
  "coupon": {
    "name": "Paw Planet - Helloworld",
    "icon_id": "b9836df9-d810-4d99-ba62-acf1205ea16c",
    "logo_id": "3e76b50c-bd47-48e9-b8d6-dd097962aa1b",
    "foreground_color": "#ffffff",
    "background_color": "#ce8c46",
    "barcode": {
      "format": "qrcode"
    },
    "primary_fields": [
      {
        "key": "offer",
        "label": "Any premium dog food",
        "value": "20% off",
        "behaviour": "fixed"
      }
    ],
    "auxiliary_fields": [
      {
        "key": "username",
        "label": "Name",
        "value": "",
        "behaviour": "dynamic"
      },
      {
        "key": "expires",
        "label": "EXPIRES",
        "value": "2 weeks",
        "behaviour": "fixed"
      }
    ]
  }
}



```

Passworks API as the ability to create a "template" if you provide the "assets" (*iconid*, *logoid*, etc..) inside the create "coupon" call. So we will use this ability in this example.

Read this for more information regarding the fields: <https://github.com/passworks/passworks-api/blob/master/v2/sections/coupon.md#available-fields>

Copy and paste the JSON above (replace the *iconid* and *logoid*) with your own and submit the code, don't forget to change the "endpoint" URL and the encoding to "JSON (application/json)" as showned in the image:

POST https://api.passworks.io/v2/coupons/

http://api.passworks.git:3000 http://api.passworks.ξ X + Immendes (backend\_saas)  

POST http://api.passworks.git:3000/v2/coupons/ Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Code

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☒ JSON (application/json)

```
1 {
2   "coupon": {
3     "name": "Paw Planet - Helloworld",
4     "icon_id": "b9836df9-d810-4d99-ba62-acf1205ea16c",
5     "logo_id": "3e76b50c-bd47-48e9-b8d6-dd097962aa1b",
6     "foreground_color": "#ffffff",
7     "background_color": "#ce8c46",
8     "barcode": {
9       "format": "qrcode"
10    },
11    "code_type": "per_pass",
12    "primary_fields": [
13      {
14        "key": "offer",
15        "label": "Any premium dog food",
16        "value": "20% off",
17        "behaviour": "fixed"
18      }
19    ]
20  }
21 }
```

The API should return something like this, notice that the ID for this campaign is

4e0551fb-1f00-4f6d-962b-59a1c890932e .

```
{
  "id": "4e0551fb-1f00-4f6d-962b-59a1c890932e",
  "name": "Paw Planet - Helloworld",
  "template_id": "5d492013-ef45-45d3-8dc1-001ece418404",
  "certificate_id": "325cbf09-bd19-481b-9c26-d0bdecf7445e",
  "organization_name": "Luis",
  "gwallet_usage": null,
  "gwallet_status": null,
  "gwallet_message": null,
  "barcode": {
    "format": "qrcode",
    "message": "",
    "alt_text": "",
    "alt_text_type": "mirror"
  },
  "barcodes": [],
  "allow_multiple_redeems": false,
  "header_fields": [],
  "primary_fields": [
```

```
{
  "key": "offer",
  "value": "20% off",
  "label": "Any premium dog food",
  "behaviour": "fixed",
  "data_type": "text",
  "required": true
}
],
"secondary_fields": [],
"auxiliary_fields": [
  {
    "key": "username",
    "value": "",
    "label": "Name",
    "behaviour": "dynamic",
    "data_type": "text",
    "required": true
  },
  {
    "key": "expires",
    "value": "2 weeks",
    "label": "EXPIRES",
    "behaviour": "fixed",
    "data_type": "text",
    "required": true
  }
],
"back_fields": [],
"locations": [],
"beacons": [],
"expiration_date": null,
"code_type": "sequencial",
"code_fixed_value": null,
"render_on_short_url": false,
"redeem_count": 0,
"remote_form_url": null,
"og": true,
"og_description": "",
"javascript": null,
"stylesheet": null,
"created_at": "2016-12-19T16:46:32Z",
"updated_at": "2016-12-19T16:46:33Z",
"distributions": [
  {
```

```

    "id": "49fe764c-0235-4b56-8ad8-0c7a32f141fd",
    "distribution_type": "downloadPage",
    "state": "published",
    "page_url": "https://passworks.io/p/_SnkQTLobg",
    "pkpass_url": "https://passworks.io/p/_SnkQTLobg.pkpass",
    "preview_url": "https://passworks.io/p/_SnkQTLobg.png?updated_at=1482165993",
    "page_short_url": "http://pass.yt/d/7L",
    "pkpass_short_url": "http://pass.yt/d/7L.pkpass",
    "preview_short_url": "http://pass.yt/d/7L.png?updated_at=1482165993",
    "created_at": "2016-12-19T16:46:33.156Z",
    "updated_at": "2016-12-19T16:46:33.273Z",
    "published_at": "2016-12-19T16:46:33.278Z"
  }
]
}

```

Right now you can't do much with the campaign, so let's create a pass.

### 3. Creating a Pass

---

Let's create a Pass for a user called "Luis" with the current Offer.

Call the following endpoint:

```
POST https://api.passworks.io/v2/coupons/4e0551fb-1f00-4f6d-962b-59a1c890932e/passes
```

And submitting the following data:

```

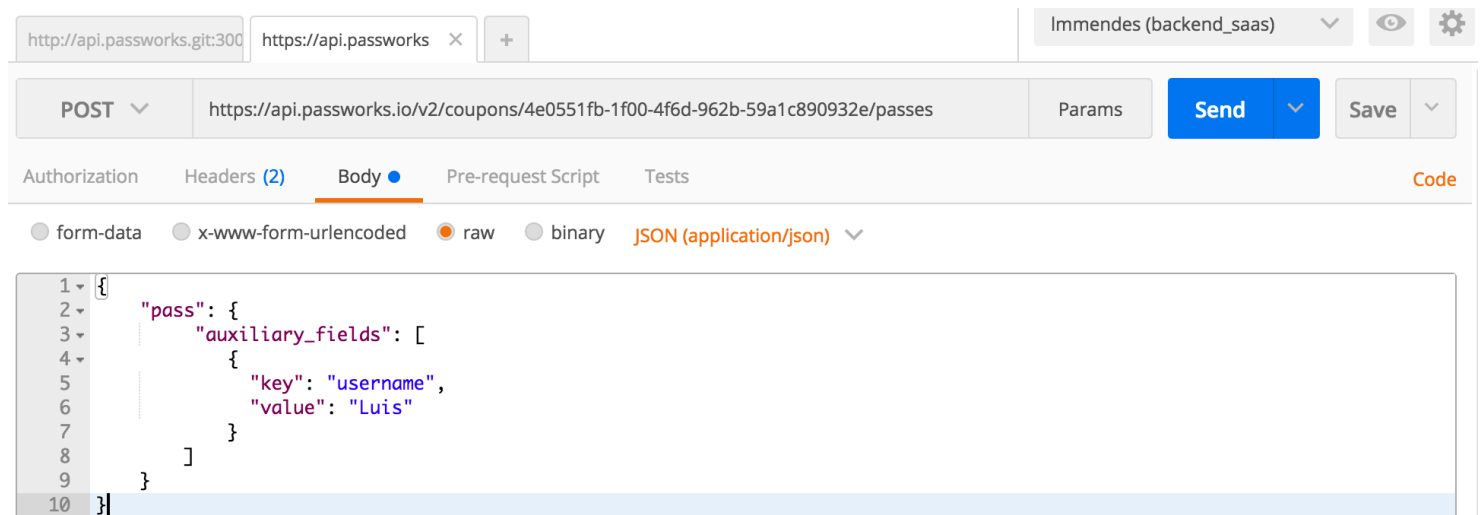
{
  "pass": {
    "auxiliary_fields": [
      {
        "key": "username",
        "value": "Luis"
      }
    ]
  }
}

```

You only need to supply the fields that are marked as dynamic this fields are the ones that change per pass (they are the dynamic / custom fiels).

And you only need to supply the "key" to address the field and the "new value" for that key.

Here is the Postman request:



And the API should respond with something like this:

```
{
  "id": "1be36452-c7b0-449a-81a8-2e0e7593dacb",
  "campaign_id": "4e0551fb-1f00-4f6d-962b-59a1c890932e",
  "voided": false,
  "serial_number": "4f2c7e51-390e-4869-8522-9f65ddd12f24",
  "redeem_code": "00000001",
  "redeemed_count": 0,
  "barcode": {
    "format": "qrcode",
    "message": "00000001",
    "alt_text": "00000001",
    "alt_text_type": "mirror"
  },
  "barcodes": [
    {
      "format": "qrcode",
      "message": "00000001",
      "alt_text": "00000001",
      "alt_text_type": "mirror"
    }
  ],
  "expiration_date": null,
  "max_distance": null,
  "relevant_date": null,
  "header_fields": [],
  "primary_fields": [
    {

```

```

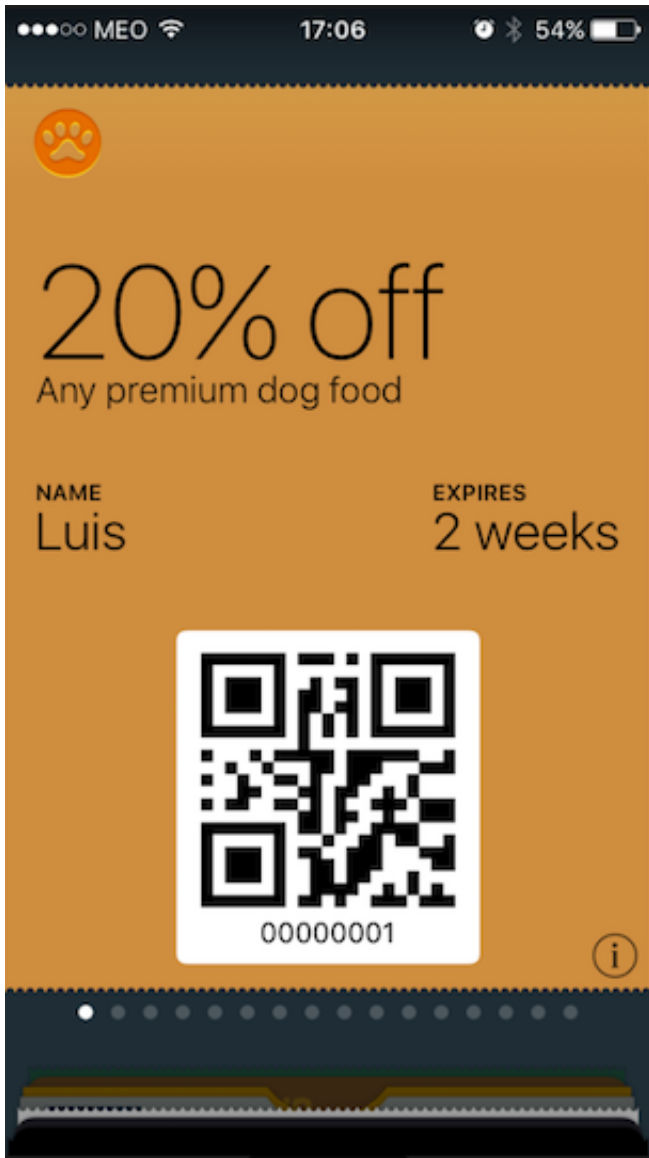
    "key": "offer",
    "value": "20% off",
    "label": "Any premium dog food",
    "behaviour": "fixed"
  }
],
"secondary_fields": [],
"auxiliary_fields": [
  {
    "key": "username",
    "value": "Luis",
    "label": "Name",
    "behaviour": "dynamic"
  },
  {
    "key": "expires",
    "value": "2 weeks",
    "label": "EXPIRES",
    "behaviour": "fixed"
  }
],
"back_fields": [],
"locations": [],
"beacons": [],
"page_url": "https://passworks.io/p/_SnkQTLobg/4KlxD7wUay6-tgLnT6jHLA",
"pkpass_url": "https://passworks.io/p/_SnkQTLobg/4KlxD7wUay6-tgLnT6jHLA.pkpass",
"page_short_url": "https://pass.yt/p/z62",
"pkpass_short_url": "https://pass.yt/p/z62.pkpass",
"gwallet_usage": false,
"gwallet_status": null,
"gwallet_message": null,
"created_at": "2016-12-19T16:56:57Z",
"updated_at": "2016-12-19T16:56:57Z"
}

```

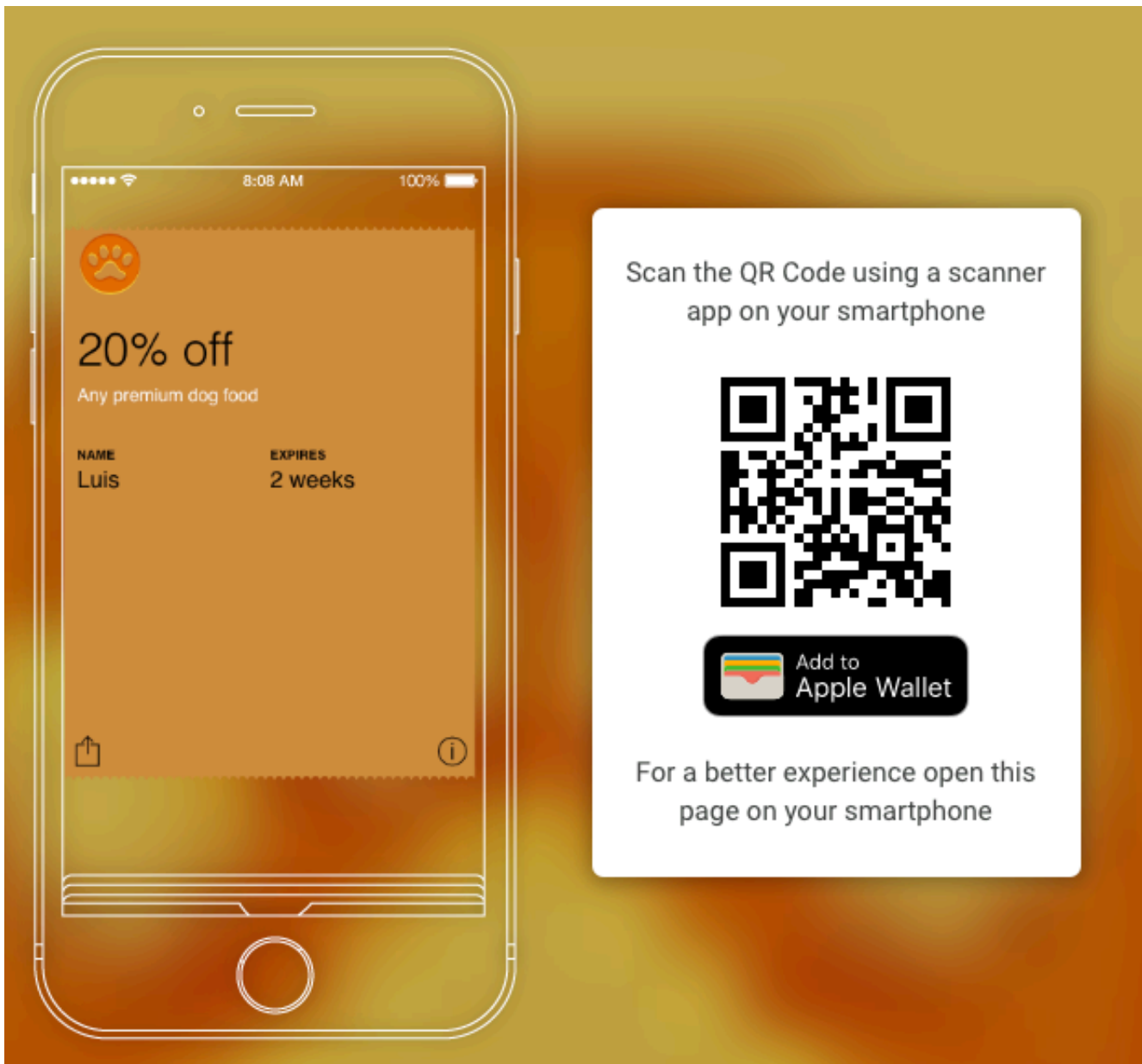
Has you can see the pass was created with the username "Luis" and all other campaign fields where copied from the campaign blueprint.

Copy the "pageurl" and open the pass on your mobile phone or desktop (pageurl = [https://passworks.io/p/\\_SnkQTLobg/4KlxD7wUay6-tgLnT6jHLA](https://passworks.io/p/_SnkQTLobg/4KlxD7wUay6-tgLnT6jHLA)) this is that pass unique URL.

## Preview of the created pass inside the Apple Wallet



Preview of the created when link opened inside a desktop browser



## 4. Updating the campaign

---

Updating a campaign only changes the passes created after the update, if you wish to also reflect the campaign changes in all existing passes simply call the MERGE endpoint to "push" the changes to all passes. See point 4.1

Let's update the campaign by changing the "Expire" ( `key: expires` ) value from "2 Weeks" to "1 Week".

Since the `key: expires` is defined as having a `behaviour: "fixed"` this means that you can change this value in one API call for all the passes via "campaign update".

Updating a campaign consists in calling the "campaign" endpoint with a PATCH HTTP method and sending the

fields that you wish changed (note that in the URL the `4e0551fb-1f00-4f6d-962b-59a1c890932e` is the campaign ID).

PATCH `https://api.passworks.io/v2/coupons/4e0551fb-1f00-4f6d-962b-59a1c890932e`

PATCH `https://api.passworks.io/v2/coupons/4e0551fb-1f00-4f6d-962b-59a1c890932e` Params **Send** Save

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary **JSON (application/json)**

```
1 {
2   "coupon": {
3     "auxiliary_fields": [
4       {
5         "key": "expires",
6         "value": "1 weeks"
7       }
8     ]
9   }
10 }
```

Text version of the payload sent (for copy and past):

```
{
  "coupon": {
    "auxiliary_fields": [
      {
        "key": "expires",
        "value": "1 weeks"
      }
    ]
  }
}
```

The API response

```
{
  "id": "4e0551fb-1f00-4f6d-962b-59a1c890932e",
  "name": "Paw Planet - Helloworld",
  "template_id": "5d492013-ef45-45d3-8dc1-001ece418404",
  "certificate_id": "325cbf09-bd19-481b-9c26-d0bdecf7445e",
  "organization_name": "Luis",
  "gwallet_usage": null,
  "gwallet_status": null,
  "gwallet_message": null,
  "barcode": {
    "format": "qrcode",
  }
}
```

```
"message": "",
"alt_text": "",
"alt_text_type": "mirror"
},
"barcodes": [],
"allow_multiple_redeems": false,
"header_fields": [],
"primary_fields": [
  {
    "key": "offer",
    "value": "20% off",
    "label": "Any premium dog food",
    "behaviour": "fixed",
    "data_type": "text",
    "required": true
  }
],
"secondary_fields": [],
"auxiliary_fields": [
  {
    "key": "username",
    "value": "",
    "label": "Name",
    "behaviour": "dynamic",
    "data_type": "text",
    "required": true
  },
  {
    "key": "expires",
    "value": "1 weeks",
    "label": "EXPIRES",
    "behaviour": "fixed",
    "data_type": "text",
    "required": true
  }
],
"back_fields": [],
"locations": [],
"beacons": [],
"expiration_date": null,
"code_type": "sequencial",
"code_fixed_value": null,
"render_on_short_url": false,
"redeem_count": 0,
"remote_form_url": null,
```

```

"og": true,
"og_description": "",
"javascript": null,
"stylesheet": null,
"created_at": "2016-12-19T16:46:32Z",
"updated_at": "2016-12-20T09:33:13Z",
"distributions": [
  {
    "id": "49fe764c-0235-4b56-8ad8-0c7a32f141fd",
    "distribution_type": "downloadPage",
    "state": "published",
    "page_url": "https://passworks.io/p/_SnkQTLobg",
    "pkpass_url": "https://passworks.io/p/_SnkQTLobg.pkpass",
    "preview_url": "https://passworks.io/p/_SnkQTLobg.png?updated_at=1482226393",
    "page_short_url": "https://pass.yt/d/7L",
    "pkpass_short_url": "https://pass.yt/d/7L.pkpass",
    "preview_short_url": "https://pass.yt/d/7L.png?updated_at=1482226393",
    "created_at": "2016-12-19T16:46:33.156Z",
    "updated_at": "2016-12-19T16:46:33.273Z",
    "published_at": "2016-12-19T16:46:33.278Z"
  }
]
}

```

Notice that the campaign as updated and only the `expires` field inside the campaign changed from `2 Weeks` to `1 Week`.

If you create a new pass by following the `step 3` you will see that the new generated pass is created with the `expires` with the updated value. And that the Pass in your iPhone didn't change, this is because campaign updated only change new passes created after the campaign update.

Let's see how to update also the already existing passes.

## 4.1 Pushing the campaign changes to all passes

Pushing the campaign changes requires you to call the `merge` campaign endpoint, this will iterate through all passes merge the campaign changes with them and send the update version to the Apple Wallet.

You don't need to send any payload (body content) when calling this endpoint.

```
POST https://api.passworks.io/v2/coupons/4e0551fb-1f00-4f6d-962b-59a1c890932e/merge
```

POST

https://api.passworks.io/v2/coupons/4e0551fb-1f00-4f6d-962b-59a1c890932e/merge

Params

Send

Save

Authorization

Headers (2)

Body

Pre-request Script

Tests

Code

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

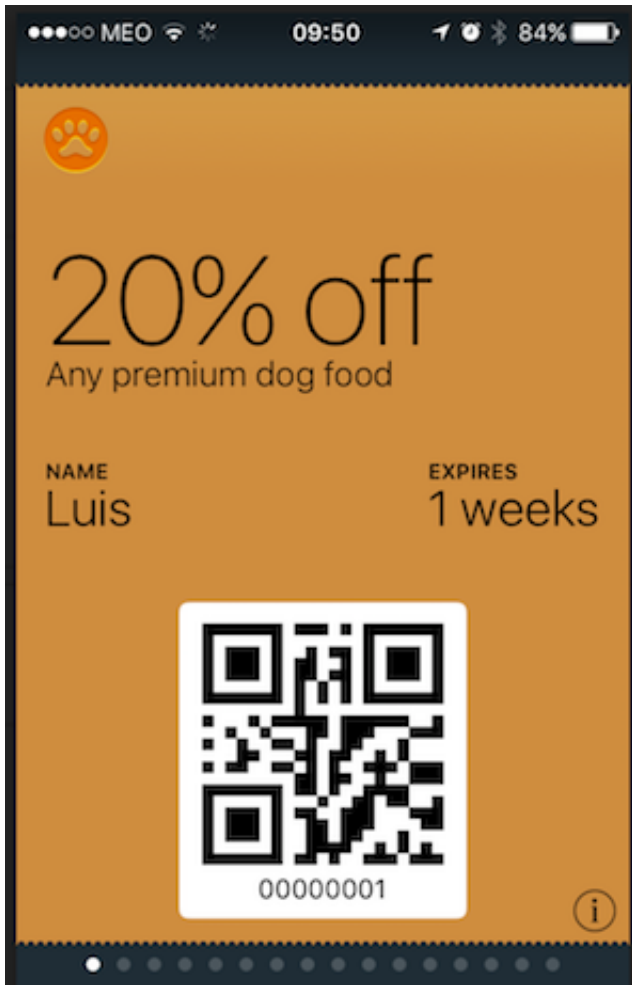
1

The API will respond with the number of passes that will be updated, you can ignore this message is just for information proposes.

```
{
  "message": "Started merging 1 passes for campaign 'Paw Planet - Helloworld'."
}
```

NOTE: Updating a campaign can take from some 5 second to one or more hours depending the size of your >campaign, but each time a pass is update the changes are sent to the users Wallet.

Open your iPhone and search for the installed pass you will see that it was instantly updated:

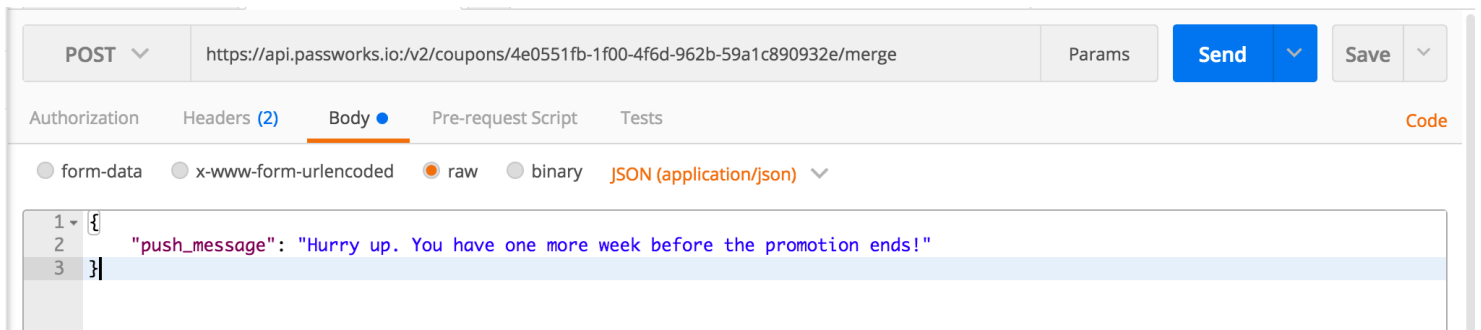


#### 4.1.1 Pushing the campaign changes to all passes with notification to the user

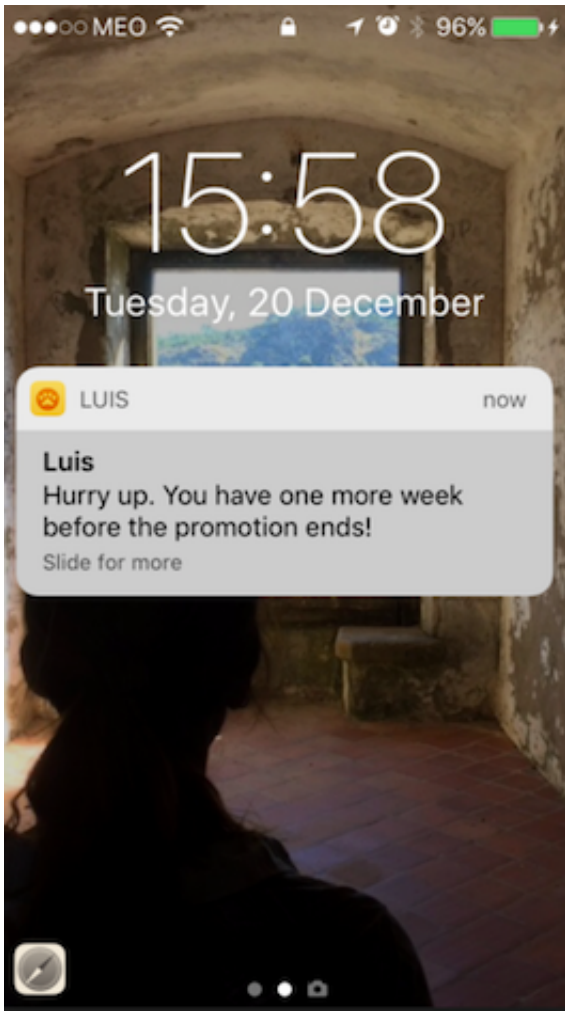
When updating (merging) passes you can send to you existing users a notification message when they receive their updated passe.

For that just call the `merge` endpoint as shown before but this time send `push_message` attribute in the payload.

```
POST https://api.passworks.io/v2/coupons/4e0551fb-1f00-4f6d-962b-59a1c890932e/merge
```



```
{  
  "push_message": "Hurry up. You have one more week before the promotion ends!"  
}
```



Please note that the "LUIS" that appears inside next to the "ICON" and underneath it is my current company name "Luís" and not the person that belongs to the "pass". You can update the campaign by changing the "organization\_name" that will change the "Luis" text to whatever you wish.

## 5. Sending push notifications to all users of the campaign

You can send a "push message" to all of your campaign users by calling the "push" endpoint.

```
POST https://api.passworks.io/v2/coupons/4e0551fb-1f00-4f6d-962b-59a1c890932e/push
```

POST

https://api.passworks.io/v2/coupons/4e0551fb-1f00-4f6d-962b-59a1c890932e/push

Params

Send

Save

AuthorizationHeaders (2)BodyPre-request ScriptTestsCode

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

```
1 {  
2   "push_message": "Hello from Paw Planet! This is a the push message :-)"  
3 }
```

```
{  
  "push_message": "Hello from Paw Planet! This is a the push message :-)"  
}
```

Example of the message on the iPhone lockscreen

