

Print-Mailing Automation API

v 2.10.0

Division: Post & Parcel Germany

This API is used for creating and managing **Print-Mailing campaigns** as well as automated, trigger-based sending of recipient information through **3rd party Marketing Automation Systems**.

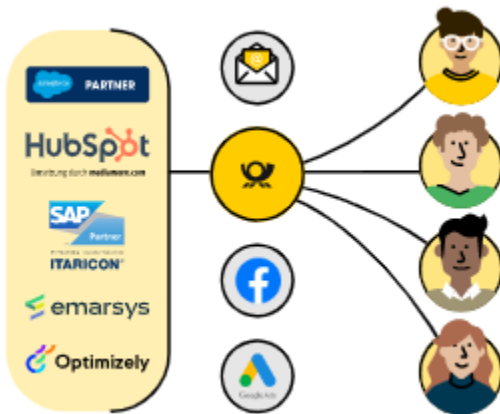
The Print-Mailing API features:

- Creating and managing campaigns
- Designing campaigns with a maximum of **individualization**
- High quality printing
- Best in class delivery by Deutsche Post

Region: Germany

Used for: Shipping, Advertising

Overview



The service Print-Mailing Automation (PMA) of Deutsche Post AG (DPAG) extends the capability of a cross-channel marketing of a partnersystem with a direct mailings advertising channel. It streamlines the creation, design, production, and delivery of personalized, data driven print mailings. This allows a partnersystem user to mix online and offline advertising when addressing their target audience. Print-Mailing Automation is not a marketing automation system. That is, it does not provide the recipients, and it does not handle the events, for which an advertising should be created. These are to be handled by the partnersystem.

Print-Mailing Automation consists of two separate, logical components - a **web service** and a **frontend** (<https://print-mailing.deutschepost.de/planen>). The web services provides a set of APIs, which allows a partnersystem to manage:

- an advertising campaign, e.g. create a campaign with a specific name for a specific period
- variables definitions, i.e. define what recipient information will be send and used in the design individualization in the campaign
- recipient information, as defined in the previous point

The frontend enables the fine tuning of an advertising campaign. Through it, single users are able to select the advertising format, the shipping and production methods as well as visually proof the advertising campaign created via the frontend, and check potentials costs. However, the most important function of the frontend is the **web editor**, which can be used to create the physical design of the mailing, and to define rules, required for the automatic individualization of the advertisement.

This guide will help you start creating, designing and sending print mailings to your customers using your own webpage, mobile app, or marketing automation system by leveraging the power of the Print-Mailing Automation API.

The primary audience are readers with a technical background, who are interested in

- estimating the complexity of integrating the service
- understanding the business and technical requirements and
- integrating the respective APIs to the service.

Actors

In Print-Mailing Automation, we differentiate between three actors:

- **Partnersystem:** A partnersystem is any system that offers Print-Mailing Automation service to its customers as an advertising channel for direct mail by implementing Print-Mailing Automation's APIs. Partner systems can be all systems that control advertising. These are in particular marketing suites or marketing automation systems. But it can also be mailing systems, shops or CRM systems.

- **Customer:** A customer of a partner system, as defined above, operates the partnersystem as the tenant of this system. With the use of Print-Mailing Automation, customers can use the direct mailing advertising channel as an additional advertising channel, whereby they also become the customer of the Print-Mailing Automation web application. A customer has the right to create, change and delete campaigns and variables for their individualization from the partnersystem.
- **Systemuser:** A systemuser is a specific person who is an employee of a company that uses a partnersystem as a customer. The systemuser has the right to log in to the Print-Mailing Automation frontend. There he can edit campaigns for his company by selecting ad formats, specifying production and shipping options and designing the ad.

Audience information

The primary audience of this documentation are readers with a technical background, who are interested in

- estimating the complexity of integrating the service
- understanding the business and technical requirements and
- integrating the respective APIs to the service.

Get Access

The integration of Print-Mailing Automation in your system is a **two-step process**. The first steps is receiving access to the test environment and starting the integration development and testing. Once the development is finished and tests have been successfully conducted, you will be configured on the production environment and will receive your production credentials.

You need to have a **signed contract with DPAG** before the registration process.

How-To request your tenants **test environment**:

1. Please fill the **Einrichtungsauftrag_Automation*.docx**
 - The tenant name you would like to perform your testing with (if not specified, you will be provided one).

- Contact person for technical questions/information exchange (the person will be used for generating example access token, as discussed later in the document. First name, last name, and email address suffice).
 - IP addresses, from which you will be accessing the test environment (your IP address needs to be whitelisted in our firewall for you to gain access to the test environment).
 - To get your **External Customer ID** of the Group Developer Portal, you need to **Register**. After logged-in, the id can be found in the **User Profile**.
2. send it to **it-csp@deutschepost.de**.
 3. Use "[#ma] Configuration of YOUR COMPANY in Print-Mailing Automation Test System" as e-mail subject header.
 4. Our staff will contact you directly.

The configuration process, including the IP whitelisting, may take up to a week. Once this is done, you will receive the following information per email:

- **partnerSystemIdExt**: The numeric id of your partnersystem in our system.
- **partnerSystemCustomerIdExt**: The alphanumeric id identifying your customer, you want to act for.
- **authenticationSecret**: A shared secret for authentication.

Once you have successfully completed the integration and testing, you are eligible to receive access to the **production environment**. For this, repeat the registration process above with email header "[#ma] Configuration of YOUR COMPANY in Print-Mailing Automation Prod System" and the following information in the body:

- Your tenant name, as per your contract. If such does not exist.
- Contact person for generating example token.
- A phone number on which you would like to receive credential information.

Configuration process for production environment may take up to one week. Once this is done, you will receive a password-secured zip file per email. The password for the zip file will be provided in a separate email shortly after.

Environments

We offer two environments which may differ in their version and documentation

- Production environment (PROD)
 - use this [Documentation](#)
- Test Environment (UAT)
 - use this [Documentation for UAT](#)
 - serves as a sandbox
 - *Requests made to the test environment never affect productive systems and are not billed.*

See the servers drop-down menu in the [Reference Docs](#) for the environments URLs.

There are two different technical specifications for both the production environment (PROD) and test environment (UAT). Updates will be available in UAT first. These updates then also appear with a time lag on PROD. You can see from the release notes of both documentations whether the technical specification for the UAT is more up-to-date than for the PROD.

Using the API

This API is built conform to the REST architectural style to provide interoperability.

Exposed API resources can be manipulated using HTTP methods. Requests to a resource URL generate a response with JSON payload containing detailed information about the requested resource. HTTP response codes are used to indicate API errors.

We use a token-based approach to authorize requests coming from your system.

Http requests

Apply the standardized HTTP method semantics to communicate with the Print-Mailing Automation API.

The API offers clearly identifiable **resources** ("things of the Print-Mailing Automation universe"). You can use a resource's **representation** to create, update, delete and execute resources. There are different types of resources:

- **Single resource:** E.g. a systemuser, a campaign and its corresponding properties. A PUT updates properties of an existing single resource identified by an id.
- **List resource:** A (pageable) list of single resources e.g. a list of systemusers. Usually, a list resource doesn't include all properties of the single resources. A POST adds a new single resource to a list resource.
- **Processing resource:** Triggers a usually long running asynchronous process. This might be a price calculation or a campaign order. A POST is used to trigger processing resources.
- **Lookup resource:** Offer permitted (enum) field values for simpler communication with the API. API clients can see which values they can expect and have to include in requests. E.g. values for campaign states (active, paused, ...). Usually, lookup resources include an internationalized label.

POST requests are used to create single resources on a list resource. The semantic for list endpoints is best described as 'please add the enclosed representation to the list resource identified by the URL'. On a successful POST request, the server will create one or multiple new resources and provide their URLs in the response. Successful POST requests will usually generate 201.

PUT requests are used to update an **entire** resources, which already exists and thus can be clearly identified by an id. The semantic is best described as "please put the enclosed representation at the resource mentioned by the URL, replacing any existing resource". On successful PUT requests, the server will replace the entire resource addressed by the URL with the representation passed in the payload (subsequent reads will deliver the same payload). Successful PUT requests will usually generate 200.

GET, DELETE and **OPTIONS** requests follow the standardized HTTP method semantics.

Headers

You can (and sometimes have to) set HTTP headers. This could be the 'Content-Type', 'Accept', 'Accept-Encoding' or 'Authorization' headers, among others. Please only set headers that are necessary.

Caution

Be aware that HTTP headers with empty values are not allowed for security reasons and such requests will be rejected.

Payload and media types

The standard media type for requests and responses is **application/json** with default encoding **UTF-8**. However, other media types like **application/xml** are supported, too. See the concrete resource definition for all accepted media types.

Tip

Set the header '**Accept: application/json**' in your JSON requests in order to get a detailed JSON response in case of an error.

Use the same semantics for **null** and **absent** properties. OpenApi 3.x allows to mark properties as required and as nullable to specify whether properties may be absent (`{}`) or null (`{"example":null}`). If a property is defined to be not required and nullable (see 2nd row in Table below), this rule demands that both cases must be handled in the exact same manner by specification.

The following table shows all combinations and whether the examples are valid:

required	nullable	<code>{}</code>	<code>{"example":null}</code>
true	true	No	Yes
false	true	Yes	Yes
true	false	No	No
false	false	Yes	No

boolean properties must not be null. A boolean is essentially a closed enumeration of two values, true and false.

HTTP response compression

The Print-Mailing Automation API will compress all text-based responses using your favorite compression algorithm.

Tip

Please set a correct **Accept-Encoding** http header for all your request, if applicable!

This increases the API user experience and saves bandwidth.

Date and time formats

Use the date and time formats defined by **RFC 3339**:

- For "date" use strings matching yyyy-MM-dd, for example: 2026-05-28
- For "date-time" use strings matching yyyy-MM-dd'T'HH:mm:ss.SSS'Z', for example 2026-05-28T14:07:17.000Z

Note that the OpenApi format "date-time" corresponds to "date-time" in the RFC and "date" corresponds to "full-date" in RFC. Both are specific profiles, a subset of the international standard ISO 8601.

A zone offset must not be used. Date time values are always interpreted in **UTC** without offsets. Localization of date time should be done by the api client, if necessary.

Status codes and common error structures

Apply the standardized HTTP status code semantics to communicate with the Print-Mailing Automation API.

The OpenApi specification of every resource provides information about specific success and error responses. It is not a complete list to avoid an overload with common sense information. See below the list of most commonly used status codes in the Print-Mailing Automation API:

Status code	Meaning
200 OK	Request completed successfully.
201 Created	Request completed successfully. Resource created.
400 Bad Request	Invalid/not parseable JSON payload. Non existing enum value recognized. Concurrent update of a resource.
401 Unauthorized	Missing access token in the request headers.
403 Forbidden	Access denied because of missing authority. E.g. customer id is not allowed.
404 Not Found	The resource is not found.
422 Unprocessable Entity	The request could be parsed but the contents are invalid. See the business validation error code for further information.
500 Internal Server Error	A generic error indication for an unexpected server execution problem.
503 Service Unavailable	Service is (temporarily) not available.

In case of an error or input data validation failure, the services will respond with the following error representation in the response body (including internationalized messages).

For general errors (e.g. service unavailable):

```
{
  "timestamp": "2026-02-18T16:11:13.456Z",
  "status": 503,
  "correlationId": "3ea93639-3d7f-4a5c-88ee-f6b6d01dffa",
  "error": "Service Unavailable",
  "errors": [{
    "errorCode": "DM_CO_0012",
    "errorMessage": "The requested functionality is currently not available. Try again later.",
    "arguments": [],
    "fieldName": null
  }]
}
```

For specific validation of a sent field:

```
{
  "timestamp": "2026-02-18T16:12:05.352Z",
  "status": 422,
  "correlationId": "08a884be-3aec-4fa5-b647-7f82149144d5",
  "error": "Unprocessable Entity",
  "errors": [{
    "errorCode": "NotBlank",
    "errorMessage": "must not be empty",
    "arguments": [],
    "fieldName": "password"
  }]
}
```

Tip

Notice the unique **errorCode**: It describes a specific error case which is documented in the OpenApi. Let your application handle these specific error cases and don't only look for http status codes!

If you want to get in contact with us because of an error response, please also supply the **correlationId**. This is a unique id which is generated for every request and can help us to trace requests through our backend.

Common representation structures

Most of the resource representations include common fields:

```
{
  "id":87075,
  "createdOn":"2020-04-02T11:48:51.000Z",
  "changedOn":"2020-04-29T13:02:56.000Z",
  "version":44,
  ".....": "....."
}
```

Use this **id** to address the resource. The **version** is a simple counter informing about resource changes.

A **paged list resource** has the following structure:

```
{
  "elements":[
    {
      "id":549,
      "createdOn":"2020-03-04T09:05:11.000Z",
      "changedOn":null,
      "version":1,
      ".....": "....."
    },
    { ... }
  ],
  "page":{
    "size":50,
    "totalElements":13,
    "totalPages":1,
    "number":0
  },
  "links":[
    {
```

```
    "rel": "self",  
  
    "href": "https://print-mailing-api.deutschepost.de/user/customers  
?page=0&size=50&sort=company,asc"  
  }  
]  
}
```

Pagination and filtering list resources

Several list resources of the Print-Mailing Automation API support pagination and filtering to reduce the amount of returned data and improve the API client experience. If no resource fits to the given filtering criteria an empty list is returned. An HTTP 404 is never returned from a list resource.

Request

Meaning

```
?page=0&size=50
```

Get the first page, maximum 50 items per page.

```
?page=0&size=50&company=Dynp
```

Get the first page, maximum 50 items per page. Search in field "company" with value "Dynp".

```
?page=0&size=50&sort=company,asc
```

Get the first page, maximum 50 elements per page. Sort ascending by fieldName "company".

```
?sort=company,asc&sort=type,desc
```

Get the default number of elements per page. Sort ascending by fieldName "company" and descending by "type".

Caution

The maximum number of returned elements per list resource is always **2000**. You **MUST** use paging, if you expect bigger result sets.

List resources also include **link relation fields** for easier pagination:

- **self**: The hyperlink pointing to the same page.
- **prev**: The hyperlink pointing to the previous page.
- **next**: The hyperlink pointing to the next page.

Lookup resources

Lookup resources offer permitted (enum) field values for simpler communication with the API. There are cases where a API client has to send a specific value from a set of permitted values. If you have a frontend application, these kind of values usually can be picked from a drop down menu.

A lookup resource publishes all of this permitted values (which are mostly ids) together with an internationalized label.

```
{
  "Font": [{
    "id": 10,
    "label": "Arial"
  },
  {
    "id": 11,
    "label": "Arial Bold"
  },
  "..."]
}
```

Error code lookup resources

Error code lookup resources offer a complete list of error codes, specific resources may return in an error response. This resource can be helpful during API client development. An error code lookup resource publishes all of this codes with an internationalized label.

```
{
  "CommonBusinessError": [
    {
      "value": "DM_CO_0001",
      "label": "Auswahlwert existiert nicht."
    },
    "...",
  ],
  "RecipientBusinessError": [
    {
      "value": "DM_RC_0001",
      "label": "Die Adressfelder des Print-Mailings sind nicht definiert."
    },
    {
      "value": "DM_RC_0002",
      "label": "Das Print-Mailing existiert nicht."
    },
    "...",
  ]
}
```

Error code lookup resources may include **common business errors**. This type of errors is common to all resources.

Versioning

The API has explicit API versioning within the URL (`/v1/`). However, we apply the following rules to evolve the Print-Mailing Automation API in a backward-compatible way to stay on v1 as long as possible:

- Add only optional, never mandatory fields.

- Never change the semantic of fields (e.g. changing the semantic from customer-ekp to customer-id, as both are different unique customer keys)
- Input fields may have (complex) constraints being validated via server-side business logic. Never change the validation logic to be more restrictive and make sure that all constraints are clearly defined in description.
- Support redirection in case an URL has to change 301 (Moved Permanently).

API clients should apply the following principles:

- Be conservative with API requests and data passed as input, e.g. avoid to exploit definition deficits like passing megabytes of strings with unspecified maximum length.
- Be tolerant in processing and reading data of API responses.
- Be tolerant with unknown fields in the payload, i.e. ignore new fields **but do not eliminate them from payload if needed for subsequent PUT requests**. Remember, the semantics of PUT.
- Be prepared that enum return parameter may deliver new values; either be agnostic or provide default behavior for unknown values.
- Be prepared to handle HTTP status codes not explicitly specified in endpoint definitions. Note also, that status codes are extensible. Default handling is how you would treat the corresponding 2xx code (see [RFC 7231 Section 6](#)).
- Follow the redirect when the server returns HTTP status code 301 (Moved Permanently).

Authentication

Most of the Print-Mailing Automation API resources are secured using OAuth 2.0.

The first step accessing the Print-Mailing Automation API is to create an access token JWT (json web token) using the `user/authentication` resource.

Add this access token, which usually starts with 'eyJ...', to the `Authorization http header` of every request:

```
Authorization: Bearer eyJJKK78Bhu53KB.....iJKVlQiL
GET https://print-mailing-api.deutschepost.de/resource/4711
```

You can decode (base64) the Print-Mailing Automation JWT to get further information for communicating with the API. A JWT contains the following payload:

```
{
  "sub": "paul",
  "iss": "com.dpdhl.dialogmarketing",
  "fullName": "Paul Watson",
  "customerIds": [ "4" ],
  "exp": 1588272706,
  "locale": "de",
  "iat": 1588269106,
  "userId": 2,
  "authorities": [ "ROLE_PMP_ACCESS", "ROLE_DIS_ACCESS",
"ROLE_DIS_MANAGE_DIALOGPOST" ]
}
```

The most important fields are:

- **customerIds**: Usually the Ids of the customer(s) you have access to. In the most cases, this will be exactly one. You need this id for subsequent requests. Depending on the API you access, this might be you employer. In this case, you have access to several other customers and you are offered a resource which returns the list of accessible customer Ids.
- **exp**: Expiry datetime of the token in UTC (jwt standard field).
- **locale**: Your chosen local for internationalization.
- **userId**: This is your systemuser id.
- **authorities**: An array containing all your access rights.

Every access token has a specific **valid time** (about 30 minutes). An expired access token can never be used again and lets the Print-Mailing Automation refuse the whole request. You have to call the `user/authentication/reauth` resource with a valid token to get a fresh access token with the maximum valid time. We recommend to get a new access token a couple of minutes before the expiration time.

Warning

Handle your access token like a password!

Depending on your systemuser account, you have a specific number of **authorities** ("access rights"). The OpenApi specification of every resource provides information about the authorities you have to own to access a resource.

Login as partnersystem

As a partnersystem you received the following login data from us:

- **partnerSystemIdExt**: The numeric id of your partnersystem in our system.
- **partnerSystemCustomerIdExt**: The alphanumeric id identifying your customer, you want to act for.
- **authenticationSecret**: A shared secret for authentication.

Use the `/user/authentication/partnersystem/credentialsbased` resource to create an access token with these parameters. Your token will include the authority "ROLE_PMP_LONG_TERM_CAMPAIGN_PARTNER_SYSTEM".

Warning

Don't mix up `partnerSystemCustomerIdExt` and `customerId`! They are not the same. You need `partnerSystemCustomerIdExt` for authentication and the `customerId` encoded in the token for most of the subsequent requests.

Example request and response for a partnersystem login:

```
POST
https://print-mailing-api.deutschepost.de/user/authentication/partnersystem/credentialsbased
Accept: application/json
Accept-Encoding: gzip
Content-Type: application/json

{
  "partnerSystemIdExt": "37348",
  "partnerSystemCustomerIdExt": "MyCompany",
  "authenticationSecret": "SHARED-SECRET",
  "locale": "de"
}
```

```
HTTP/1.1 200
{
  "jwtToken": "eyJ0eXiOi.....rJGiHMTcrn_3T9Og"
}
```

Login as partnersystem editor

A partnersystem editor usually is a human systemuser who is registered in your system and should be able to access the Print-Mailing Automation frontend without logging in again. This is achieved as follows:

As a partnersystem you received the following login data from us:

- **partnerSystemIdExt** (equals "masId"): The numeric id of your partnersystem in our system.
- **partnerSystemCustomerIdExt** (equals "masClientId"): The alphanumeric id identifying your customer, you want to act for.
- **ssoTokenKey**: A shared secret used for signing the JWT you generated.

You transfer the relevant data for the systemuser login in a JWT ([RFC 7519](#)) to us. Use a short validity (about 2 minutes) and sign the token with the shared sso token key. Print-Mailing Automation API will validate the JWT and find or create the systemuser.

The JWT you generate, must have the following structure:

```
{
  "firstname": "Connie",
  "lastname": "Baker",
  "email": "connie.baker@dypn.com",
  "username": "connie_baker",
  "masClientId": "DypnCompany",
  "masId": 1000,
  "iss": "issuer",
  "exp": 1589269431,
  "iat": 1589267631
}
```

The most important fields are:

- **firstname**: The systemuser's firstname (max. 50 characters).
- **lastname**: The systemuser's lastname (max. 50 characters).
- **email**: The systemuser's email address (max. 150 characters).
- **username**: The systemuser's username (max. 80 characters). Must be unique for the same partnersystem customer.
- **exp**: Expiry datetime of the token in UTC (jwt standard field).

Tip

There are several libraries for easy JWT creation. Try **Auth0** if you are using JAVA.

If you want to redirect a partnersystem editor to the ma frontend, use the following url in web browser redirect:

<https://print-mailing.deutschepost.de/planen?partnersystem={YOUR-SIGNED-JWT}>.

If you want to skip the dashboard and enter a campaign directly, you need a valid campaign id - use the following url in web browser redirect:

<https://print-mailing.deutschepost.de/planen?partnersystem={YOUR-SIGNED-JWT}&campaignid={VALID-CAMPAIGN-ID}>

Workflow type, authorities and what you can do with the API

Depending on your login type, you have a specific number of **authorities** ("access rights"). The OpenApi specification of every resource provides information about the authorities you have to own to access a resource.

See the following table to get a high level overview of permitted functionality:

The

partnersystem'
s workflow

type

Authority in the access token

Allowed high
level

functionality

TRIGGER_COMPLETE	ROLE_PMP_LONG_TERM_CAMPAIGN_PARTNER_SYSTEM	createCampaign, updateCampaign, createVariableDefinitions, updateVariableDefinitions, createRecipient
TRIGGER_ONLY	ROLE_PMP_LONG_TERM_CAMPAIGN_PARTNER_SYSTEM	createRecipient

API resources and workflow steps

This section walks you through creating a campaign, adding variable definitions and sending your first recipients.

A **campaign** (likewise "longtermcampaign" or "print mailing") comprises all relevant information for a specific print mailing. Print mailings are rendered, printed and sent to your customers.

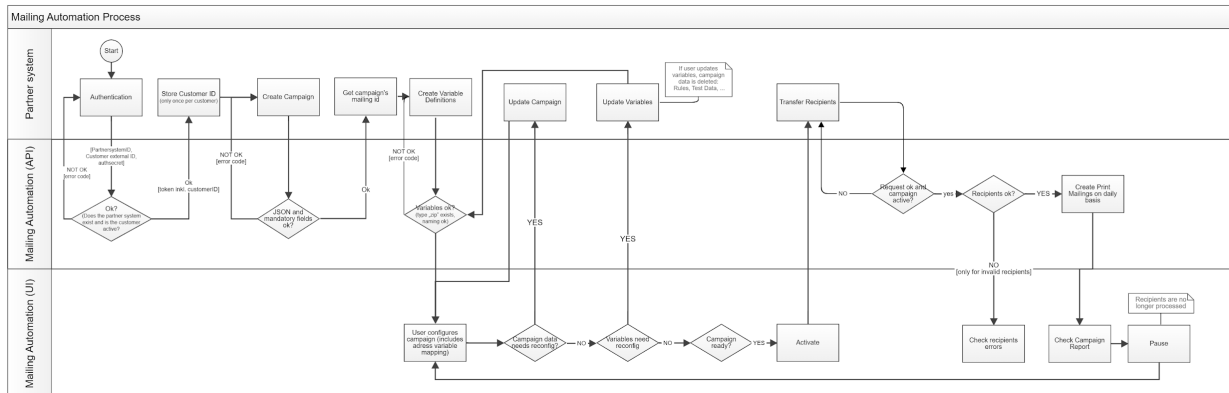
A **mailing** comprises all relevant information needed for the graphical design of a print mailing. Based on a **mailing template** and probably enriched with **variable definitions** you design a personal mail experience for every customer.

After activating the campaign, you supply us with concrete **recipient data**. Recipients contain delivery address and customization data for each print mailing (e.g. special offers per customer, salutation or date of birth). Recipients are accepted throughout the whole day, however they get rendered only once a day.

Important

A campaign and all dependent information can only be changed as long as the campaign is not active. Creating and updating campaigns and variable definitions as well as sending recipients is a common task of an api user with the role **partnersystem** whereas uploading and design the mailing template, placing variable

definitions on the mailing template and defining the variable definition's type usually is accomplished by **partnersystem editors**. A partnersystem editor typically uses our **frontend** to accomplish these tasks. See **Workflow types** for a high level overview.



Create a new campaign

You need a valid access token for role **ROLE_PMP_LONG_TERM_CAMPAIGN_PARTNER_SYSTEM** to continue (see [this](#) section).

Using the `longtermcampaigns` resource you can list, create and update all campaigns for the customerId your access token is created for.

Warning

Don't mix up **partnerSystemCustomerIdExt** (used during authentication) and **customerId** (which is a property within the access token)!

You need to extract the customerId from the access token and use it in all subsequent requests (it's like a tenant, you are working for).

Example request and response for creating a new campaign:

```
POST
https://print-mailing-api.deutschepost.de/automation/longtermcampaigns
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Authorization: Bearer ACCESS-TOKEN
```

```
{
  "campaignIdExt": "SummerSess2026",
  "campaignName": "Summer session 2026",
  "customerId": "633"
}
```

HTTP/1.1 201

```
{
  "id": 94762,
  "createdOn": "2020-05-05T10:36:51.509Z",
  "changedOn": null,
  "version": 1,
  "campaignType": "LONG_TERM",
  "campaignName": "Summer session 2026",
  "stateId": 110,
  "product": null,
  "sendingReasonId": 10,
  "actions": [
    "EDIT"
  ],
  "requiredActions": [
    "DEFINE_PRODUCT",
    "DEFINE_SENDING_REASON",
    "ESTIMATE_CAMPAIGN",
    "DEFINE_MAILING_TEMPLATE",
    "DEFINE_VARIABLES"
  ],
  "workflowType": "TRIGGER_COMPLETE",
  "hasDummyName": false,
  "campaignIdExt": "SummerSess2026",
  "individualizationId": null,
  "printingProcessId": null,
  "deliveryProductId": null
}
```

Tip

- Check out the `campaignlookups` [lookup resource](#) to get further description of campaign properties.

- Check out the `products` > **resource** > to get the list of available products.

Warning

If the optional property **deliveryCheckSelected** is set to `true`

We check your addresses *with additional costs*

- Amendment of postal details (street, zip, city)
- Undeliverable addresses will be removed.
- Removal of duplicates (if first and last name are available)
- Additionally only for private addresses: > > - Amendment of first name, last name, and house number
 - Removal of relocated / no longer resident people
- Corporate customers' addresses are not checked.
- Deliverable addresses are used even when the recipients are no > longer registered at the address.

You can find more information about the service at DP Direkt **ADDRESSFACTORY**

Important properties:

- **id**: This is the unique campaign id used in subsequent requests. It is recommended to save this id in your system for easy access.
- **workflowType**: The campaign's workflow type which equals to your customer's partnersystem workflow type. Defines which api client role can execute which operations on a campaign and dependant resources.
- **requiredActions**: There are several tasks ("actions") to accomplish before a campaign can be activated. As you continue with completing the campaign, the number of required actions decreases.
- **stateld**: Only campaigns with the state EDIT(110) or PAUSED(125) can be changed.

You may **update** some of the campaign's properties afterwards.

Products

The `products` resource offers the list of all available product types and their products with their corresponding information.

For example, product type *Brief A4* has six products. Each product corresponds to the number of pages and how they are going to be printed.

```
[
  {
    "id": 6,
    "name": "Brief A4",
    "width": 210,
    "length": 297,
    "imageName": "pic_brief_a4.svg",
    "sortOrder": 0,
    "products": [
      {
        "id": 1,
        "productTypeId": 6,
        "name": "Brief A4 - Ein Blatt - Einseitig
bedruckt",
        "imageName": "pic_1_seite_4-0.svg",
        "pageCount": 1,
        "sheetCount": 1,
        "ltcPriceScaleOffsetAddress": "M305",
        "ltcPriceScaleOffsetLimited": "M405",
        "ltcPriceScaleDigital": "M005",
        "otcPriceScaleOffsetAddress": "M105",
        "otcPriceScaleOffsetLimited": "M205",
        "otcPriceScaleDigital": "M505",
        "length": 2290,
        "height": 1140,
        "thickness": 50,
        "weight": 10,
        "pageWithVariables": 1,
        "printWidth": 210,
        "printHeight": 297,
        "coated": false,
        "availableForOtc": true,
        "availableForLtc": true,
        "productTemplateId": "1",
        "duplex": false
      }
    ]
  }
]
```



```

        }, "..."]
    ]
},
{
    "id": 4,
    "name": "Karte im Umschlag",
    "width": 210,
    "length": 105,
    "imageName": "pic_karte_umschlag.svg",
    "sortOrder": 1,
    "products": [
        {
            "id": 8,
            "productTypeId": 4,
            "name": "Karte im Umschlag",
            "imageName": "pic_karte_umschlag.svg",
            "pageCount": 2,
            "sheetCount": 1,
            "ltcPriceScaleOffsetAddress": "M304",
            "ltcPriceScaleOffsetLimited": "M404",
            "ltcPriceScaleDigital": "M004",
            "otcPriceScaleOffsetAddress": "M104",
            "otcPriceScaleOffsetLimited": "M204",
            "otcPriceScaleDigital": "M504",
            "length": 2290,
            "height": 1140,
            "thickness": 50,
            "weight": 12,
            "pageWithVariables": 1,
            "printWidth": 210,
            "printHeight": 105,
            "coated": false,
            "availableForOtc": true,
            "availableForLtc": true,
            "productTemplateId": "8",
            "duplex": false
        }
    ]
}, "..."]
]

```

Tip

Only products with property **availableForLtc** set to `true` are available for Mailing-Automation.

To **create a campaign** with a product, add the optional property **productId** and set the value to the corresponding product id.

Warning

Detailed product information within the campaign in a response to your request is deprecated and will be substituted with the **productId** in the future. Please use this productId to get detailed information through the products resource.

Create a mailing

A **mailing** comprises all relevant information for designing the print mailing. You create *one* mailing per campaign.

Example request and response for creating a new mailing:

POST

`https://print-mailing-api.deutschepost.de/automation/mailings`

Content-Type: `application/json`

Accept: `application/json`

Accept-Encoding: `gzip`

Authorization: `Bearer ACCESS-TOKEN`

```
{
  "customerId": "633",
  "campaignId": 94762
}
```

HTTP/1.1 200

```
{
  "id": 58650,
  "createdOn": "2020-05-05T13:22:07.640Z",
  "changedOn": null,
  "version": 1,
  "campaignId": 94762,
  "senderAddress": null,
```

```
"mailingTemplateType": null
}
```

Important properties:

- **id**: This is the unique mailing id used in subsequent requests. It is recommended to save this id in your system for easy access.
- **senderAddress**: You might want to add a sender address printed on your print mailings afterwards.

Create variable definitions

You can and should define the variable data printed on every print mailing. This is where variable definitions come into place. There are **two types** of variable definitions:

- **Address variable definitions**: The recipient's postal address. To ensure the correct delivery, we have concrete validation rules for the variable values and do an **address variable mapping**. This means that we map a variable definition to a field in a postal address (e.g. street, house number, zip, ...). The Print-Mailing Automation backend implements an algorithm to help you with this and keeps a list of synonyms which probably match *your* variable definition labels so that automatic mapping takes place. However, **a partnersystem editor has to confirm** the mapping before activating the campaign. A partnersystem editor can overrule our guess and set a specific **addressVariableId** which he can retrieve from the `addressvariables` resource. This resource also includes the current list of synonyms.
- **Individual variable definitions**: Anything else you want to print on a print mailing including various data types. Depending on the campaign's individualization type (which a partnersystem editor can choose), the number of variable definitions might be limited.

Special **address line optimizations** take place if you choose to address a **company**. You can choose between a maximum of three address lines for the company's name. Because of address formatting requirements, our API will convert your recipient data according to the following rules:

- companyname_2 and companyname_3 are shifted left, if companyname_1 or companyname_2 are empty.
- If companyname_X contains line breaks (Hex 0A or 0D), the contents are divided and distributed to the other address variables.
- Every address variable content will be truncated to a maximum number of 40 characters.

Company address line optimization examples

Input: companyname_1	Input: companyname_2	Input: companyname_3	Result: companyname_1	Result: companyname_2	Result: companyname_3
Todd's Custom Meats	Todd Carver	and Becky Carver	Todd's Custom Meats	Todd Carver	and Becky Carver
	Todd's Custom Meats		Todd's Custom Meats		
	Nandwani's Tailors	Reginald Milton	Nandwani's Tailors	Reginald Milton	
WASHCOLA UNDRY LLC. Frankling			WASHCOLA UNDRY LLC. Frankling Becks and	friends	

Becks and
friends

WASHCOLA	
UNDRY	
LLC. \n	WASHCOLA Frankling
Frankling	UNDRY Becks and
Becks and	LLC. friends
friends	

Caution

Your recipient data gets **converted**. Keep in mind the rules above if you place company address variables onto a mailing template during the design phase.

Facts about variable definitions

- Variable definition's label must be unique within one mailing.
- Variable definitions are visualized in the ma frontend ordered by **sortOrder** value.
- A variable definition always has a **dataType** (string, date, zip, ...). There always has to be exactly one variable definition of type "zip".
- Only variable definitions of data type "string" can become address variable definitions.
- A variable definition becomes an address variable if an **addressVariableId** is defined. A partner system editor confirmed the address variable mapping if the value **addressVariableMappingConfirmed** is true.
- A variable definition has a **selected** flag. This controls the variable definition's visibility for different campaign individualization types.
- An individual variable definition can have an overprint style which positions the variable definition on a mailing template.
- Role ROLE_PMP_LONG_TERM_CAMPAIGN_PARTNER_SYSTEM can never set overprint style information, address variables and selected flags. You can

(and must) ignore these properties. They are set when designing the mailing template.

Tip

Make sure to have understood the rules above before sending recipients to minimize effort for updating your variable definitions. Use the [addressvariables](#) resource to get information about required address variables and synonyms.

Data types and allowed values

ID	Data Type	Format
10	STRING	Free text to the maximum length of 255 characters. The backslash and control characters are not allowed. CAUTION: Recipient data of type STRING can contain characters from the full UTF-8 character set. However, characters will only be displayed and rendered correctly if a specific character is known to the font used to print the mailing . Otherwise, the character will not be printed or printed as a question mark.
20	INTEGER	A whole number within the range from -2^{31} to $2^{31}-1$. The empty value is not allowed. E.g.: "643", "-12"
30	BOOLEAN	A boolean value. These values are accepted: "true", "false", "0", "1". The empty value is not allowed.

40	DATE	A date in ISO format: yyyy-MM-dd. yyyy stands for the year. MM stands for the month. dd stands for the day. E.g.: "2016-04-14" for April 14, 2016
50	IMAGE	An image file name as used for an asset in the editor. The file name must end with either ".png", ".jpg", ".jpeg", ".tiff", ".tif", ".eps" oder ".psd" The maximum length is 255 characters. Folder names and path delimiters are not allowed. E.g.: "myProduct.jpg"
60	IMAGE_URL	A valid URL up to the maximum length of 255 characters referencing an image. The application will try to download the image from this URL. The <i>path</i> part of the URL must end with either ".png", ".jpg", ".jpeg", ".tiff", ".tif", ".eps" oder ".psd". E.g.: "https://image.yourdomain.com/folder/name.png?param=value"
70	FLOAT	A fraction number with dot as decimal separator. Thousand separators are not allowed. Scientific notation (exponential notation) is allowed. The value range is ± 500000000000 (3 significant decimal digits). The empty value is not allowed. E.g.: "123", "-12.34", "1.237", "123456.7e9", "123.45E-9"
80	ZIP	A postal code. Currently it needs to be a valid German ZIP code of five digits. There must be exactly one

variable with this data type. The value is mandatory.
E.g.: "50670"

90 COUNTRY_CODE A two uppercase letter country code according to the ISO 3166-1-alpha-2 standard. Currently only "DE" is supported. E.g.: "DE"

[Show Full Table](#)

Example request and response for creating new variable definitions

POST

`https://print-mailing-api.deutschepost.de/automation/mailings/58650/variabledefinitions`

Content-Type: application/json

Accept: application/json

Accept-Encoding: gzip

Authorization: Bearer ACCESS-TOKEN

```
{
  "customerId": "633",
  "createVariableDefRequestRepList": [
    {
      "label": "forename",
      "sortOrder": 3,
      "dataTypeId": 10
    },
    {
      "label": "lastname",
      "sortOrder": 4,
      "dataTypeId": 10
    },
    {
      "label": "zip",
      "sortOrder": 7,
```



```
    "dataTypeId": 80
  },
  {
    "label": "salutation",
    "sortOrder": 1,
    "dataTypeId": 10
  },
  {
    "label": "title",
    "sortOrder": 2,
    "dataTypeId": 10
  },
  {
    "label": "street",
    "sortOrder": 5,
    "dataTypeId": 10
  },
  {
    "label": "house_number",
    "sortOrder": 6,
    "dataTypeId": 10
  },
  {
    "label": "city",
    "sortOrder": 8,
    "dataTypeId": 10
  },
  {
    "label": "individual_image_var",
    "sortOrder": 10,
    "dataTypeId": 50
  }
]
}
```

HTTP/1.1 201

```
{
  "elements": [
    {
      "id": 55072,
      "createdOn": "2020-05-05T15:27:21.542Z",
```

```
    "changedOn": null,
    "version": 1,
    "label": "forename",
    "sortOrder": 0,
    "dataType": {
      "id": 10,
      "label": "Text"
    },
    "addressVariableId": null,
    "addressVariableMappingConfirmed": false,
    "selected": false,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  {
    "id": 55075,
    "createdOn": "2020-05-05T15:27:21.543Z",
    "changedOn": "2020-05-05T15:27:21.829Z",
    "version": 2,
    "label": "lastname",
    "sortOrder": 0,
    "dataType": {
      "id": 10,
      "label": "Text"
    },
    "addressVariableId": 5,
    "addressVariableMappingConfirmed": false,
    "selected": false,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  {
    "id": 55076,
```

```
    "createdOn": "2020-05-05T15:27:21.543Z",
    "changedOn": "2020-05-05T15:27:21.826Z",
    "version": 2,
    "label": "zip",
    "sortOrder": 0,
    "dataType": {
      "id": 80,
      "label": "Postleitzahl"
    },
    "addressVariableId": 8,
    "addressVariableMappingConfirmed": false,
    "selected": false,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  {
    "id": 76847,
    "createdOn": "2020-05-05T15:27:22.543Z",
    "changedOn": null,
    "version": 1,
    "label": "individual_image_var",
    "sortOrder": 10,
    "dataType": {
      "id": 50,
      "label": "Bild"
    },
    "addressVariableId": null,
    "addressVariableMappingConfirmed": false,
    "selected": true,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  { ... }
```

```

],
"page": {
  "size": 9,
  "totalElements": 9,
  "totalPages": 1,
  "number": 0
},
"links": []
}

```

Important properties

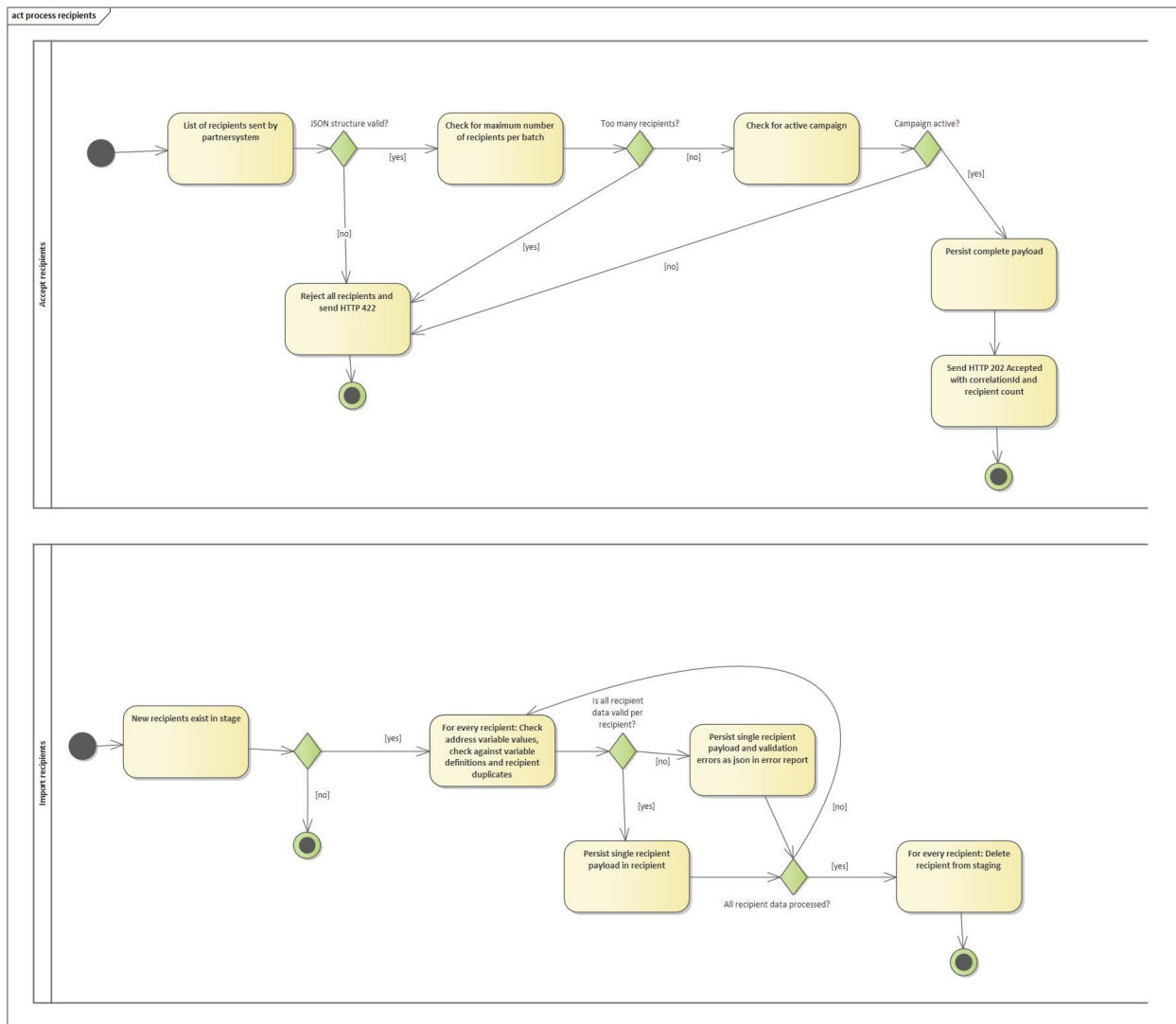
- **label:** This is the unique (per mailing) name you will use to send concrete recipient data for later.
- **addressVariableId:** Link to a field in a postal address. Automatic address mapping succeeded for the two variables "lastname" and "zip".
- **dataTypeId:** The variable definition's data type ID. See [this table](#) or `VariableDefDataType` of the `mailinglookups` resource for available values.
- **addressVariableMappingConfirmed:** Is false as long as a partnersystem editor did not confirm the address variable mapping.
- **sortOrder:** A simple numeric order, if you want to present the variable definitions in a specific order. Set it to 0 if the order is irrelevant.
- **selected, x, y, font, fontColor, fontSize, spanHeight:** You can (and must) ignore these properties. They are set when designing the mailing template.

Transfer recipients

After activating the campaign, you supply us with concrete **recipient data** matching the previously defined variable definitions. Please note that recipient data will only be accepted if the campaign has been activated. Once activated, the recipients will be accepted throughout the whole day, however they get rendered only once a day.

You need a valid access token for role `ROLE_PMP_LONG_TERM_CAMPAIGN_PARTNER_SYSTEM` to continue (see [this section](#)).

Depending on the number of print mailings you want to create, there might be several thousands of recipients you want to send to the ma API. To keep the transmission fast and secure we distinguish between accepting and importing recipients:



Accepting recipients validates the http payload you send us, verifies the number of recipients and checks for an active campaign state.

Caution

The maximum number of recipients per batch is 100. The maximum number of recipient variable data per recipient is 100. **We recommend to send recipients in few http requests and big batches rather than sending a lot of small requests.**

Importing recipients loads and validates every recipient against your variable definitions and creates an error report, if necessary. Every recipient has to contain a variable value for each of the previously defined variable definitions.

Example request and response for adding one recipient:

```
POST
https://print-mailing-api.deutschepost.de/automation/recipients
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Authorization: Bearer ACCESS-TOKEN
```

```
{
  "campaignId": 94762,
  "customerId": "633",
  "recipients": [
    {
      "recipientData": [
        {
          "label": "forename",
          "value": "Chris"
        },
        {
          "label": "zip",
          "value": "48309"
        },
        {
          "label": "street",
          "value": "Baker Street"
        },
        {
          "label": "city",
          "value": "Dover"
        },
        {
          "label": "individual_image_var",
          "value": ""
        }
      ],
      "recipientIdExt": null
    }
  ]
}
```

```
}
]
}

HTTP/1.1 202
{
  "correlationId": "2b350272-8bff-4db5-be59-15fa5c277da1"
}
```

Important properties:

- **recipientIdExt**: This is an id from your system which you can use to correlate reported errors to your recipients. It can be null. A former automatic unique check on this id is **not** done anymore.
- **correlationId**: This is a unique id per batch which can be used to find a specific recipient batch in the error report.
- **label**: The label needs to match the one provided with the variable definition.
- **value**: The value needs to match the format of the data type of your variable definition. The value may be an empty string if it is not an address field and required for building the address.

If the imported recipients contain errors, you can download the recipient's error report (see [Requesting a campaign's recipients overview report](#)).

To obtain an overview of all recipients delivered to a specific campaign, see [Requesting a campaign's recipients overview report](#).

Fetch recipients report and errors

After you have provided us the recipients. You can obtain an overview report of the delivered recipients for a specific campaign, and also a recipient error report for a specific day.

You need a valid access token for role `ROLE_PMP_LONG_TERM_CAMPAIGN_PARTNER_SYSTEM` to continue (see [this](#) section).

Important: The recipients will be accepted throughout the whole day, however they get rendered only once a day.

Requesting a campaign's recipients overview report

This resource delivers for a specific campaign an overview of all imported recipients, number of recipients delivered, number of recipients with errors, and the list of recipients imported per day.

Example request and response for retrieving a campaign's recipient report overview:

GET

<https://print-mailing-api.deutschepost.de/automation/recipientreport/overview?campaignId=94762&customerId=633>

Accept: application/json

Accept-Encoding: gzip

Authorization: Bearer ACCESS-TOKEN

```
{
  "totalAmount": 6,
  "totalAmountSent": 0,
  "totalAmountErrors": 6,
  "entryList": {
    "elements": [
      {
        "reportDate": "2021-07-13",
        "totalAmountPerDay": 6,
        "totalAmountSentPerDay": 0,
        "totalAmountErrorsPerDay": 6,
        "deliveryCheckReportPerDay": null,
        "historical": false
      }
    ],
    "page": {
      "size": 20,
      "totalElements": 1,
      "totalPages": 1,
      "number": 0
    },
    "links": []
  }
}
```


For additional sorting or pagination parameter, please refer to [Pagination and filtering list resources](#).

Requesting recipients error report

This resource delivers a report of all incorrect recipients in CSV format with ISO-8859-15 encoding for all recipients imported on a specific day.

Example request and response for retrieving a recipient import error report:

GET

```
https://print-mailing-api.deutschepost.de/automation/recipientreport/detail?campaignId=94762CUSTOMERID=633&REPORTDATE=2020-05-08
ACCEPT: TEXT/CSV
```

AUTHORIZATION: BEARER ACCESS-TOKEN

HTTP/1.1 200

CORRELATION IDExternal

ID;forename;city;salutation;lastname;zip;street;title;house_number;Error Message

b55e5486-bdf4-405d-be36-751799620d03;;Mark;Dover;;;High Row Str;;;DM_CO_0003 zip: The **value** must be 5 to 5 characters in length.

Important

The parameter `reportDate` must be in ISO format: yyyy-MM-dd.

After transferring the recipients your print mailings are printed and sent to your customers.

Pause a campaign and update variable definitions

You can modify a subset of the campaign's or variable definition's properties as long as a campaign has the state EDIT(110) or PAUSED(125). Only a campaign editor is authorized to set the campaign's state (e.g. using the ma frontend).

Tip

Remember the semantics of [http PUT requests](#) while updating resources!

Update campaign

As a partner system you are only allowed to change the campaign name using the `longtermcampaigns` resource.

Update variable definitions

You can add, modify or delete variable definitions. For this, the variable definition's **id** plays an important role. If the request includes an **id**, the variable definition gets modified. If an **id** is missing, a new variable definition is created. If no variable definition is present at all, it gets deleted.

Tip

Keep in mind the rules for variable definitions: see section [create variable definitions](#).

You need your **mailing id**. You can get it as follows:

GET

```
https://print-mailing-api.deutschepost.de/automation/mailings?campaignId=94762&customerId=633
```

Accept: application/json

Accept-Encoding: gzip

Authorization: Bearer ACCESS-TOKEN

You need your **variable definition ids**. You can get it as follows:

GET

```
https://print-mailing-api.deutschepost.de/automation/mailings/58650/variabledefinitions?customerId=633
```

Accept: application/json

Accept-Encoding: gzip

Authorization: Bearer ACCESS-TOKEN

Example request which modifies variable definitions as follows:

- Rename variable definition with **label** "city", **id** "55073" to "town".
- Remove variable definition with **label** "individual_image_var", **id** "76847".
- Add new variable definition with **label** "discount".

PUT

<https://print-mailing-api.deutschepost.de/automation/mailings/58650/variabledefinitions>

Content-Type: application/json

Accept: application/json

Accept-Encoding: gzip

Authorization: Bearer ACCESS-TOKEN

```
{
  "customerId": "633",
  "updateVariableDefRequestRepList": [
    {
      "id": 55072,
      "label": "forename",
      "sortOrder": 3,
      "dataTypeId": 10
    },
    {
      "id": 55073,
      "label": "town",
      "sortOrder": 8,
      "dataTypeId": 10
    },
    {
      "id": 55074,
      "label": "salutation",
      "sortOrder": 1,
      "dataTypeId": 10
    },
    {
      "id": 55075,
      "label": "lastname",
      "sortOrder": 4,
      "dataTypeId": 10
    },
    {
      "id": 55076,
      "label": "zip",
      "sortOrder": 7,
      "dataTypeId": 80
    },
  ],
}
```

```
{
  "id": 55077,
  "label": "street",
  "sortOrder": 5,
  "dataTypeId": 10
},
{
  "id": 55078,
  "label": "title",
  "sortOrder": 2,
  "dataTypeId": 10
},
{
  "id": 55079,
  "label": "house_number",
  "sortOrder": 6,
  "dataTypeId": 10
},
{
  "label": "discount",
  "sortOrder": 10,
  "dataTypeId": 70
}
]
```

HTTP/1.1 200

```
{
  "elements": [
    {
      "id": 55072,
      "createdOn": "2020-05-06T15:08:24.000Z",
      "changedOn": "2020-07-15T13:46:13.000Z",
      "version": 3,
      "label": "forename",
      "sortOrder": 3,
      "dataType": {
        "id": 10,
        "label": "Text"
      },
      "addressVariableId": 4,

```

```
    "addressVariableMappingConfirmed": false,
    "selected": false,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  {
    "id": 55073,
    "createdOn": "2020-05-06T15:08:24.000Z",
    "changedOn": "2020-07-15T13:57:24.970Z",
    "version": 5,
    "label": "town",
    "sortOrder": 8,
    "dataType": {
      "id": 10,
      "label": "Text"
    },
    "addressVariableId": 9,
    "addressVariableMappingConfirmed": false,
    "selected": false,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  {
    "id": 55074,
    "createdOn": "2020-05-06T15:08:24.000Z",
    "changedOn": "2020-07-15T13:46:13.000Z",
    "version": 4,
    "label": "salutation",
    "sortOrder": 1,
    "dataType": {
      "id": 10,
      "label": "Text"
    },
  },
```

```
    "addressVariableId": 2,  
    "addressVariableMappingConfirmed": false,  
    "selected": false,  
    "x": null,  
    "y": null,  
    "font": null,  
    "fontColor": null,  
    "fontSize": null,  
    "spanHeight": null  
  },  
  {  
    "id": 55075,  
    "createdOn": "2020-05-06T15:08:24.000Z",  
    "changedOn": "2020-07-15T13:46:13.000Z",  
    "version": 4,  
    "label": "lastname",  
    "sortOrder": 4,  
    "dataType": {  
      "id": 10,  
      "label": "Text"  
    },  
    "addressVariableId": 5,  
    "addressVariableMappingConfirmed": false,  
    "selected": false,  
    "x": null,  
    "y": null,  
    "font": null,  
    "fontColor": null,  
    "fontSize": null,  
    "spanHeight": null  
  },  
  {  
    "id": 55076,  
    "createdOn": "2020-05-06T15:08:24.000Z",  
    "changedOn": "2020-07-15T13:46:13.000Z",  
    "version": 4,  
    "label": "zip",  
    "sortOrder": 7,  
    "dataType": {  
      "id": 80,  
      "label": "Postleitzahl"
```

```
    },
    "addressVariableId": 8,
    "addressVariableMappingConfirmed": false,
    "selected": false,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  {
    "id": 55077,
    "createdOn": "2020-05-06T15:08:24.000Z",
    "changedOn": "2020-07-15T13:46:13.000Z",
    "version": 4,
    "label": "street",
    "sortOrder": 5,
    "dataType": {
      "id": 10,
      "label": "Text"
    },
    "addressVariableId": 6,
    "addressVariableMappingConfirmed": false,
    "selected": false,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  {
    "id": 55078,
    "createdOn": "2020-05-06T15:08:24.000Z",
    "changedOn": "2020-07-15T13:46:13.000Z",
    "version": 4,
    "label": "title",
    "sortOrder": 2,
    "dataType": {
      "id": 10,
```

```
      "label": "Text"
    },
    "addressVariableId": 3,
    "addressVariableMappingConfirmed": false,
    "selected": false,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  {
    "id": 55079,
    "createdOn": "2020-05-06T15:08:24.000Z",
    "changedOn": "2020-07-15T13:46:13.000Z",
    "version": 4,
    "label": "house_number",
    "sortOrder": 6,
    "dataType": {
      "id": 10,
      "label": "Text"
    },
    "addressVariableId": 7,
    "addressVariableMappingConfirmed": false,
    "selected": false,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  },
  {
    "id": 76848,
    "createdOn": "2020-07-15T13:57:24.963Z",
    "changedOn": "2020-07-15T13:57:25.254Z",
    "version": 2,
    "label": "discount",
    "sortOrder": 10,
    "dataType": {
```



```

      "id": 70,
      "label": "Fließkommazahl"
    },
    "addressVariableId": null,
    "addressVariableMappingConfirmed": false,
    "selected": true,
    "x": null,
    "y": null,
    "font": null,
    "fontColor": null,
    "fontSize": null,
    "spanHeight": null
  }
],
"page": {
  "size": 9,
  "totalElements": 9,
  "totalPages": 1,
  "number": 0
},
"links": []
}

```

As a partner system you are only allowed to set the required properties **label**, **sortOrder** and **dataTypeld**. After modifying variable definitions a campaign editor has to confirm the address variable mapping again and can start working with new variable definitions.

See the response codes of `variabledefinitions` resource for possible error cases.

FAQs

How long does it take until the IP addresses are enabled for the UAT environment?

The activation of your IP addresses on the firewall should be completed within 5 working days after your order. After successful activation, you will receive feedback and can test the communication connection.

Support

Please contact us if you recognize errors, want to improve the documentation or have questions! For technical support during the integration, please contact us at it-csp@deutschepost.de with a mail subject "[#ma] YOUR COMPANY SHORT DESCRIPTION".

Support regarding the integration process and any further communication in this context is to be requested exclusively via e-mail to the address it-csp@deutschepost.de.

Since direct inquiries to individual employees (e.g. to their personal e-mail address within Deutsche Post) may not be processed in a timely manner, please refrain from doing so and use the above service address instead.

It is important to ensure that the subject line clearly refers to the product "Print-Mailing Automation" with the label [#ma], as the correct internal assignment and thus the response time of the ticket may be delayed otherwise.

After receiving and assigning the ticket, the sender will get a reply from the ticket system with a dedicated ticket number in the subject. This subject incl. ticket number is to be kept for all inquiries in connection with this request.

Your business contact is Mathias Rochlitz (Mathias.Rochlitz@deutschepost.de)

Glossary

- **Partnersystem:** A partnersystem is any system that offers Print-Mailing Automation service to its customers.
- **Customer:** A customer of a partner system operates the partnersystem as the tenant of this system.

Reference Docs

[Download API Spec](#)

[the developer - Website](#)

[Send email to the developer](#)

Servers

<https://print-mailing-api.deutschepost.de> - Production

[environmenthttps://uat.print-mailing-api-test.deutschepost.de](https://uat.print-mailing-api-test.deutschepost.de) - Test environment

Errorcodelookups

The Error Code Lookup Resource

Endpoints

GET

`/automation/campaignerrorcodelookups`

Retrieve all error code lookups for this service.

GET

`/user/customererrorcodelookups`

Retrieve all error code lookups for this service.

GET

`/automation/mailingerrorcodelookups`

Retrieve all error code lookups for this service.

GET

`/automation/productmanagementerrorcodelookups`

Retrieve all error code lookups for this service.

GET

`/automation/recipienterrorcodelookups`

Retrieve all error code lookups for this service.

Longtermcampaigns

Endpoints

GET

/automation/longtermcampaigns/{id}

Get a campaign.

PUT

/automation/longtermcampaigns/{id}

Updates a long-term-campaign.

DELETE

/automation/longtermcampaigns/{id}

Deletes a long-term-campaign.

GET

/automation/longtermcampaigns

Get long-term-campaigns of a customer.

POST

/automation/longtermcampaigns

Creates a long-term-campaign.

Lookups

Endpoints

GET

/automation/campaignlookups

Retrieve all lookups for this service.

GET

/automation/maillinglookups

Retrieve all lookups for this service.

GET

/automation/productmanagementlookups

Retrieve all lookups for this service.

GET

/automation/recipientlookups

Retrieve all lookups for this service.

Partnersystem

The Partner System Resource

Endpoints

GET

/user/partnersystems

Get partner systems.

GET

/user/partnersystems/{id}/logoimage

Retrieve asset and return content

GET

/user/partnersystems/{id}/addressvariablemappingimage

Retrieve asset and return content

Risearch

The RISE Search Resource

Endpoints

GET

/user/risearch

Search customers in RISE according to the provided filters.

Authentication

Get Access Tokens For Print-Mailing Automation Services

Endpoints

POST

/user/authentication/postshoplogin

Retrieve a dm access token for a registered user from the Post Shop.

POST

/user/authentication/partnersystem/tokenbased

Retrieve a dm access token for a registered user from a partner system using token based authentication.

POST

/user/authentication/partnersystem/credentialbased

Retrieve a dm access token for a registered user from a partner system using credential based authentication.

POST

/user/authentication/businesslogin

Retrieve a jwt for a business login.

GET

/user/authentication/reauth

Creates a new jwt with the maximum period of validity for the currently logged in user. You must call this method with a valid token in the authentication header

Partnersystemcustomer

The Partner System Customers Resource

Endpoints

PUT

/user/partnersystemcustomers/{id}

Update a partner system customer.

GET

/user/partnersystemcustomers

Get all existing partner system customers.

POST

/user/partnersystemcustomers

Add a new partner system customer.

Systemuser

The Systemuser Resource

Endpoints

GET

`/user/systemusers`

Get system users according to the provided pageable values and filters.

GET

`/user/systemusers/{id}`

Retrieve profiledata for a systemuser.

Account

The Account Resource

Endpoints

GET

`/user/account`

Returns the account information of the user authenticated via JWT.

GET

`/user/account/customers`

Returns all customer addresses of the logged in user or only the requested address.

GET

/user/account/customers/{id}

Returns the customer address.

Customer

The Customer Resource

Endpoints

GET

/user/customers

Get customers according to the provided pageable values and filters.

POST

/user/customers

Creates a customer using the unique dpdhl customermanagement (RISE) id.

Variabledefinitions

The Variable Definition Resource

Endpoints

GET

/automation/mailings/{id}/variabledefinitions

Gets a collection of variable definitions

PUT

/automation/mailings/{id}/variabledefinitions

Updates variable definitions.

POST

/automation/mailings/{id}/variabledefinitions

Create variable definitions for a mailing. This function only supports long-term campaigns.

Addressvariables

The Address Variable Resource

Endpoints

GET

/automation/mailings/addressvariables

Returns a list of address variables with their name and sort order

Mailings

The Mailing Resource

Endpoints

PUT

/automation/mailings/{id}

Adds or removes a sender address to the mailing.

GET

/automation/mailings

Returns the list of mailings for the campaign or if not present returns mailings for the logged in customer.

POST

/automation/mailings

Creates a new mailing.

Documents

Resource For Downloading Static Files.

Endpoints

GET

/automation/documents/{id}/download

Download a static document (ZIP, PDF, etc.)

Products

The Product Resource

Endpoints

GET

/automation/products

Retrieve product types and products.

GET

/automation/products/{id}

Gets a product.

GET

/automation/products/{id}/template

Downloads the zip template.

GET

/automation/products/{id}/layout

Get the product layout information

Recipients

The Recipient Resource

Endpoints

POST

/automation/recipients

Accept recipients of a campaign.

GET

/automation/recipients/template/excel

Downloads the recipient address template as an Excel file.

GET

/automation/recipients/template/csv

Downloads the recipient address template as an CSV file.

Recipientreport

The Recipient Report Resource

Endpoints

GET

/automation/recipientreport/overview

Returns recipient report data.

[GET](#)

/automation/recipientreport/detail

Gets the recipient error report of a day as CSV file in ISO-8859-15 encoding.

Downloads

By downloading any file, you accept our **Terms of Use**

[Open API Specification File](#)

YAML

[Einrichtungsauftrag_Automation_2022_12_0.docx](#)

DOCX - 16.85 KB