Landing Page API Documentation

# Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comment |
| 0.1 | 2016, March 9th | A. CHABANNES (HF7) | Initial version |
| 0.1.1 | 2016, March 11th | A. CHABANNES (HF7) | Added introduction |
| 0.1.2 | 2016, April 6th | A. CHABANNES (HF7) | Update |
|  |  |  |  |

# Introduction

The goal of this API is to provide an interface between a client-side script, which manages the rendering of the form and client-side validation, and the Marketing system.

It is used to retrieve the form display instructions from the server in the form of a JSON, and to send the submitted information directly to the server.

The display of the form will depend on the set-up that is defined in the Marketing system. This set-up holds the following information:

* **the form lifecycle**: the form can be set to be available for a limited period of time, and out of this timeframe a warning message or a URL redirection can be defined;
* **the list of fields to display according to contact recognition**: for each field, the Marketing defines whether the field is optional or mandatory, and whether the field should be displayed when the contact has been recognized and the information is already found in the database;
* **the post-submission behavior**: what is the type of asset that is consumed and what is the URL of this asset for example;
* other server-side behaviors, such as confirmation emails.

From the client side, the form is currently defined to work in two steps:

* Step 1: the Email field only is requested
* Step 2: the rest of the fields are requested

If the contact is recognized thanks to a cookie or an ID passed as parameter, the API jumps directly to Step 2 and returns the proper JSON.

Even if the contact has been recognized, the form will not be prefilled with their information, with the exception of the email; no other data is to be provided through the JSON results (for obvious security reasons).

# How to use the API

The API has two operations:

* **GetFormJson** : retrieve the form configuration in JSON
* **SubmitForm** : submit the form to the Marketing System

The URL to call is the following:

* DEV : <http://dassault-dev.neolane.net/dsx/lp_api.jssp>
* QAL : [~~http://dassault-test.neolane.net/dsx/lp\_api.jssp~~](http://dassault-test.neolane.net/dsx/lp_api.jssp) [UNAVAILABLE]
* PRD : [~~http://emailing.3ds.com/dsx/lp\_api.jssp~~](http://emailing.3ds.com/dsx/lp_api.jssp) [UNAVAILABLE]

To call one operation, you need to pass:

* the « op » parameter with the operation name as value
* the required parameters for this operation

In order to allow cross-origin requests, you need also to perform the call in **JSON-P** format.

All operations should be performed with the **POST** method.

**April 6th Update**: In order to cancel a contact preloading, you will need to pass the “action” parameter with “clearPreload” as value.

# Operation: GetFormJson

## Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **Type** | **Required** | **Description** |
| Input | lpid | string | Yes | Landing page code. Ex: LDP1234 |
|  | rid | string | No | *Encoded string*. Contact ID. Usually transmitted through a landing page URL used in an email. |
|  | form parameters | string | No | Parameters submitted from the form (there should be only “Email”) |
| Output | result | JSON object |  | Configuration of the form |

## Description

1. Check that the parameters are valid
2. Check and retrieve the form configuration
3. Check whether the contact exists based on their cookies or the “email” and “rid” values
4. Analyze which step should be returned (1 or 2) depending on whether the contact exists
5. If step 2 and if the contact exists, retrieve their information
6. Build JSON based on the known information and the form configuration, and return it

**April 6th Update**: Data may be returned depending on whether the contact has been recognized. However only the values for declared hidden fields will be provided; currently this applies to the email and country fields only.

## Request Example

$.ajax({

url: "http://dassault-dev.neolane.net/dsx/lp\_api.jssp",

method: "POST",

dataType: "jsonp",

data: {

op: "GetFormJson",

lpid: "LDPxxx",

email: "john.doe@company.com"

},

success: function(data){

//console.log(data);

}

});

## Response Examples

#### Error - Landing page does not exist



#### Landing page exists and is available

##### JSON returned for step 1:



**April 6th Update**: As per on-going define, the Step 1 of the form will be returned even if the contact has been recognized. In this case, we will return the known email value in the data object of the result.

##### JSON returned for step 2:



#### Landing page exists and is not available

**April 6th Update**: The marketer can define the lifespan and the maximum capacity of a landing page:

* *Lifespan*: the landing page will not be available before the defined start date or after the defined end date.
* *Capacity*: a marketer can say that they will only accept a maximum amount of submissions of that landing page. When that number is reached, the landing page is no longer available.

In addition to that, the marketer defines the behavior that is expected when the form is not available:

* Redirect to a URL
* Or display a message (HTML)

##### JSON returned to display a message:



##### JSON returned for a redirection:



# Operation: SubmitForm

## Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **Type** | **Required** | **Description** |
| Input | lpid | string | Yes | Landing page code. Ex: LDP1234 |
|  | refcode | string | No | *Encoded string*. Contains the delivery mechanism and the place variables. |
|  | did | string | No | *Encoded string*. Delivery ID. |
|  | rid | string | No | *Encoded string*. Contact ID. Usually transmitted through a landing page URL used in an email. |
|  | form parameters | string | Yes | Parameters submitted from the form |
| Ouput | result | JSON object |  | Configuration of the post-submission behavior |

## Description

1. Check that the parameters are valid
   1. If ok, go on
   2. If ko, return the Error JSON
2. Check whether the contact exists and retrieve their information
3. Create a Marketing activity based on the known information and the submitted parameters
4. Get the post-submission behavior
5. Build JSON and return it

## Request Example

$.ajax({

url: "http://dassault-dev.neolane.net/dsx/lp\_api.jssp",

method: "POST",

dataType: "jsonp",

data: {

op: "SubmitForm",

lpid: “LDPxxx",

email: "john.doe@company.com",

country: "US",

state: "CA",

company: "DOE CORPORATION"

},

success: function(data) {

//console.log(data);

}

});

## Response Examples

#### Error – Server-side field value validation



**April 6th Update**: Error messages are not yet included.

#### Successful submission

