Custom PHP

This section describes how Custom PHP can be used to increase platform's flexibility.

This article will show:

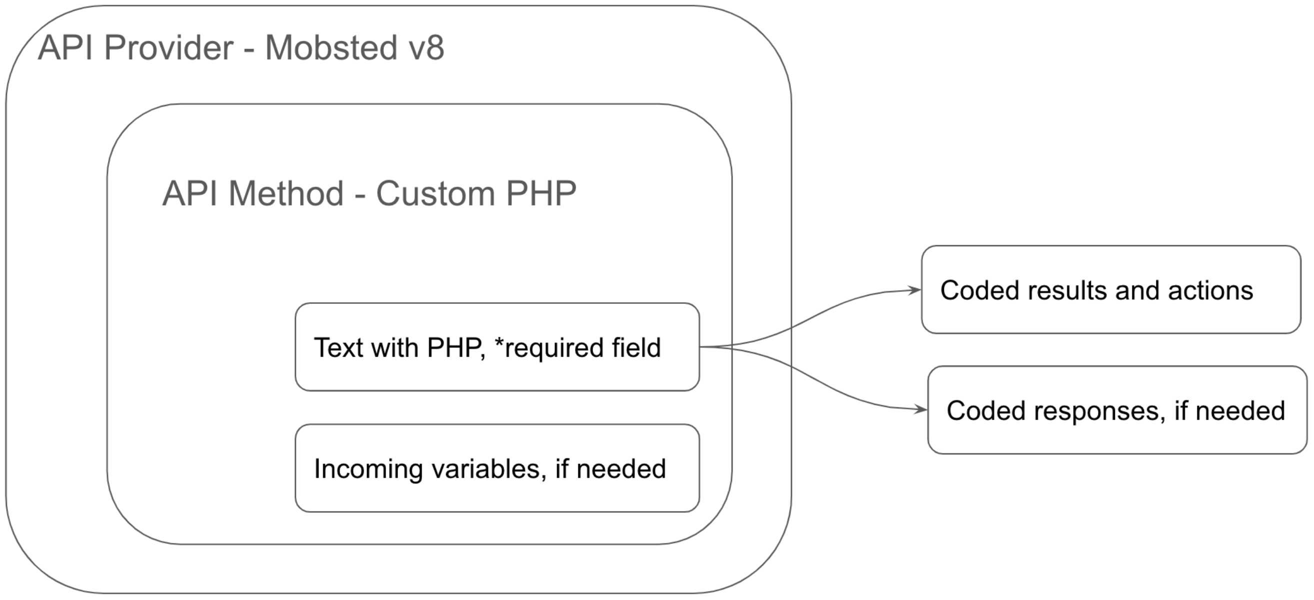
Goals to use Custom PHP Module

There are a lot of different purposes on why you could need to use platform's Custom PHP module. The most common are:

1. Connect directly to external databases without APIs
2. Make complex integrations and logic with any external APIs and Mobsted internal APIs
3. Manage and re-factor data arrays, strings, time/dates
4. Perform complex operations on your app’s backend, instead of front end

Architecture of the Module - "PHP inside API method":

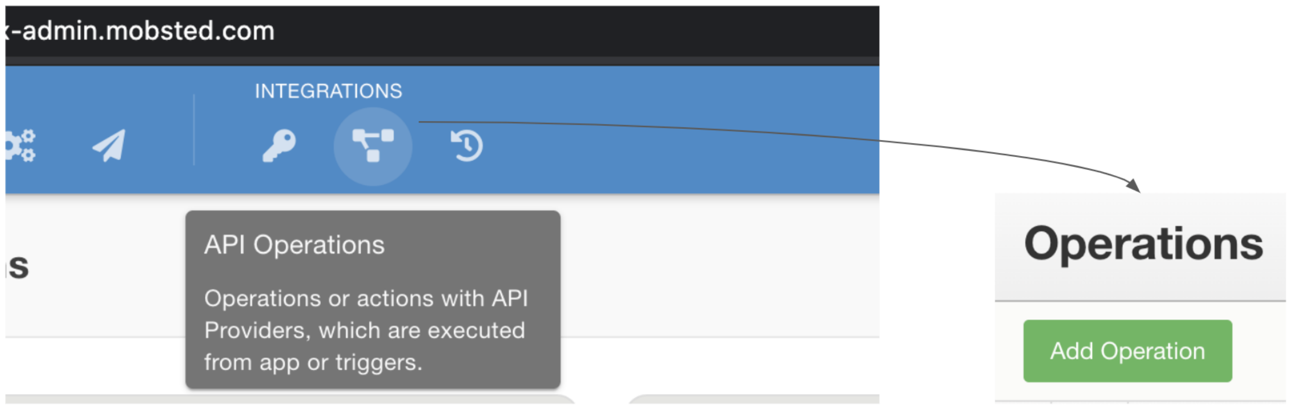
Custom PHP lives inside a API method, which can be called as any other Mobsted API method to perform needed actions.


API Provider - Mobsted v8
API Method - Custom PHP
Text with PHP, *required field
Incoming variables, if needed
Coded results and actions
Coded responses, if needed

This kind of architecture allows the following benefits and flexibility:

- running this code on request from your mobile app’s UI
- running this code as Operation for Filter/Trigger results
- running this code by API call from your outside system

Adding "Custom PHP" module:

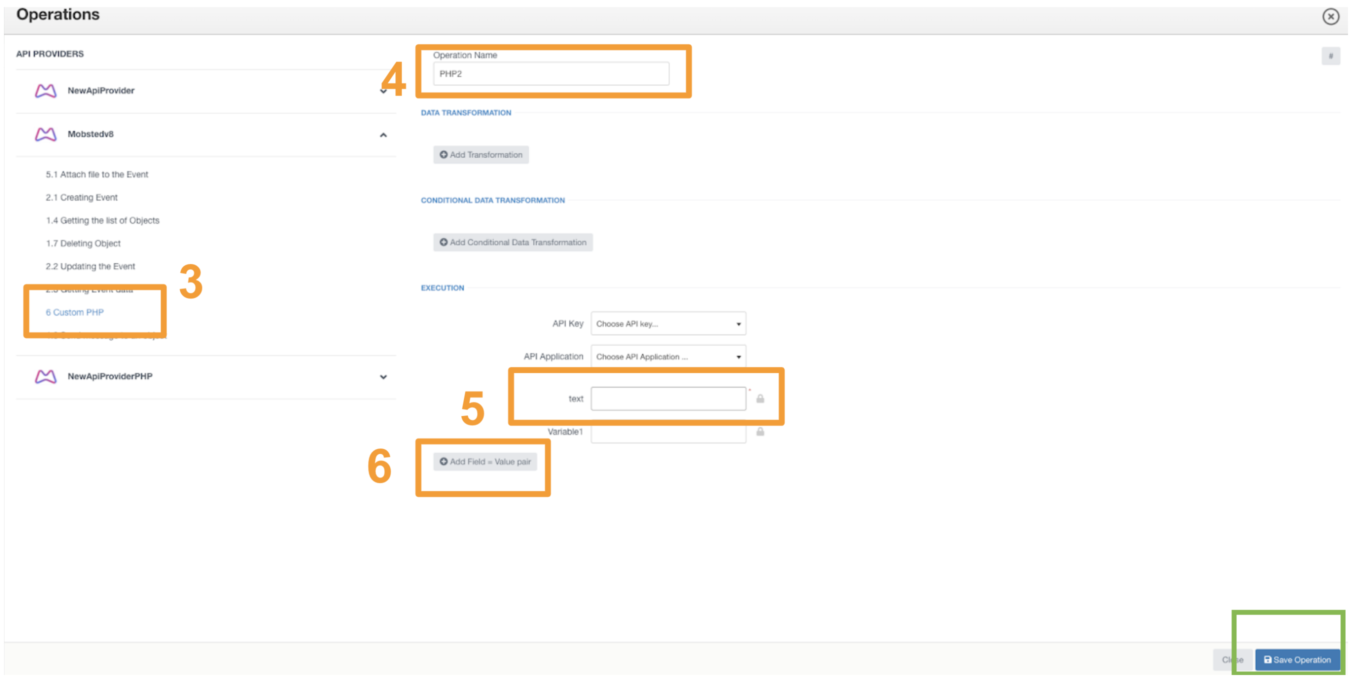


API Operations

Operations or actions with API Providers, which are executed from app or triggers.

1. Press API Operations

2. Press Add



3. Select Custom PHP method (Orange marks 3-6)

4. Name this new Operation - how it will be visible throughout your system. You can have as many of these PHP operations as you need. Just make sure each has a different name.

5. Insert PHP code into TEXT field (*use an outside editing tool, for now*)

6. Optional - add any Variables (key-value pairs), if you want to pass Incoming data to the PHP function.

7. Press - Save operation (GREEN MARK)

Examples of use cases

Connecting external DBs

Connect to supported external databases *(click each for PHP instructions)*:

- Firebird/InterBase
- MongoDB
- Microsoft SQL
- MySQL
- Oracle OCI8
- SQLite
- SQLite3
- SQLSRV

In SaaS, you can perform CRUD, close, connect and query DBs, for example, using MySQL:

```
$link = mysql_connect('example.com:3307', 'mysql_user', 'mysql_password');
if (!$link) {
die('Error: ' . mysql_error());
}

$result = mysql_query('SELECT * WHERE 1 = 1');
if (!$result) {
die('Error: ' . mysql_error());
}

print_r($result);
```

Copy above or download this code for yourself - mysql_custom_php_CRUD.txt

Performing API calls in PHP

Perform any number of external or internal APIs inside your PHP

In SaaS, you can perform 4 functions - ***curl_init, curl_exec, curl_setopt, curl_close****,* for example:

```
$ch = curl_init();
curl_setopt($ch, 10002, "https://apiDomen.xxx/api/sane/url");
curl_setopt($ch, 19913, 1);

$header = array(
'Accept: application/json',
'Content-Type: application/x-www-form-urlencoded',
'Authorization: Basic '. base64_encode("app_key:app_secret")
);
curl_setopt($ch, 10023, $header);
$output = curl_exec($ch);
print($output);
curl_close($ch);
```

Copy above or download this code for yourself - APIcall_custom_php.txt

Manage Data within Custom PHP

Manage and re-factor data arrays, strings, time/dates

You can manage, modify, re-assemble and do whatever you need with data you pass to this “Custom PHP” function with “Incoming Variables” and create any custom processes inside by working with:

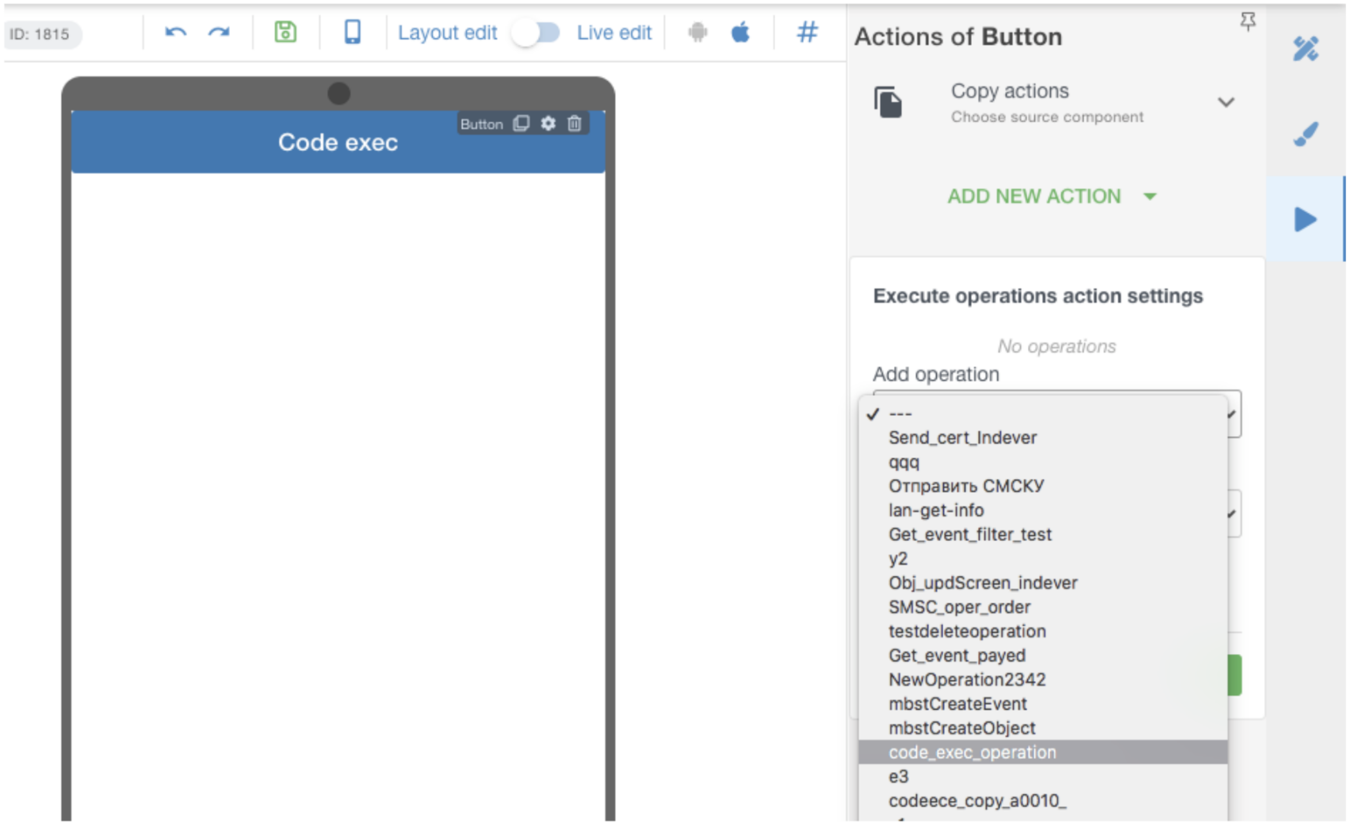
- Arrays - https://www.php.net/manual/en/ref.array.php
- Strings - https://www.php.net/manual/en/ref.strings.php
- Date/Time - https://www.php.net/manual/en/ref.datetime.php

Call from within the App

Calling “Custom PHP” from inside your mobile app to perform needed actions, for example to take care of a custom payment service provider.

This method is accessible just as any other method - from Actions tab of any element in the constructor.

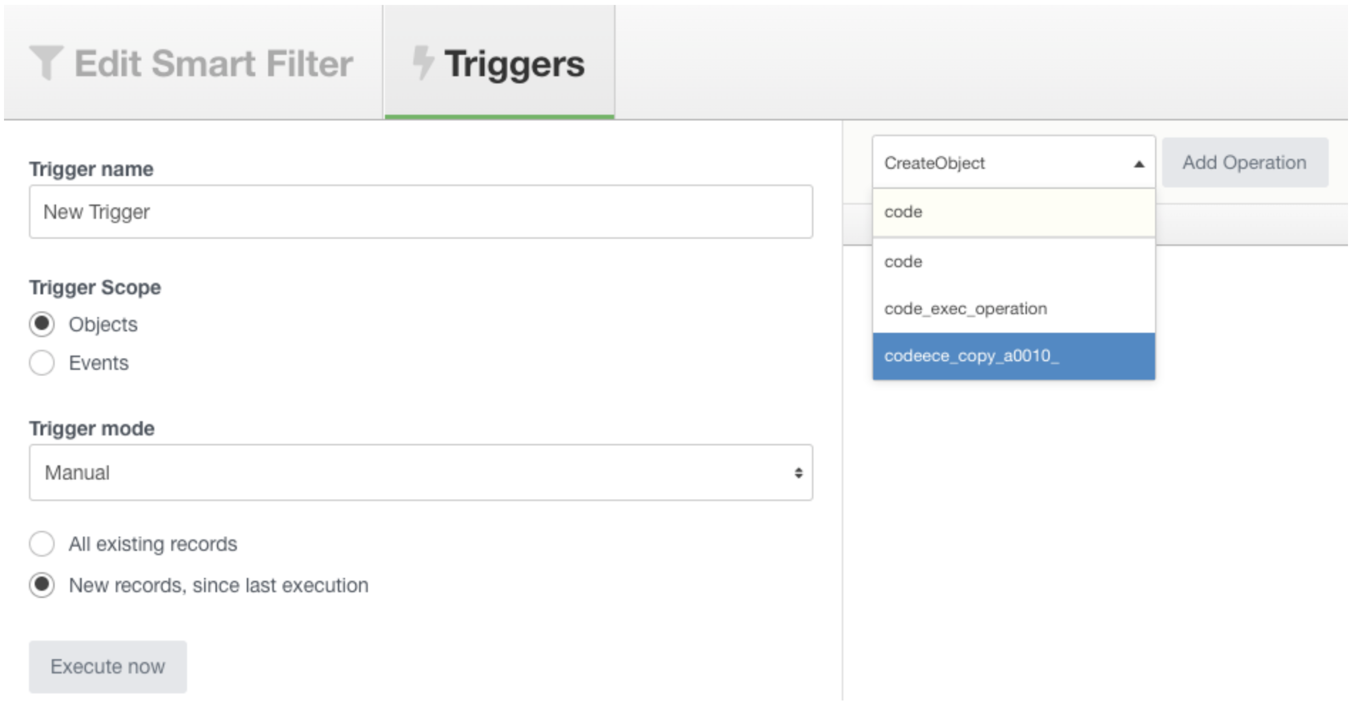
An example of a selection for a button. See which #hashtag# refs are available in session.



Call in Filters/Triggers

You can call “Custom PHP” from Filters/Triggers within your Mobsted backend.

When in Triggers - find your “Custom PHP” operation's NAME (BLUE MARK) in a list of operations and select it. It will be performed on Trigger’s conditions.



Trigger name
New Trigger

Trigger Scope
(●) Objects
( ) Events

Trigger mode
Manual

( ) All existing records
(●) New records, since last execution

CreateObject
code
code
code_exec_operation
codeece_copy_a0010_

Calling from outside API

Each “Custom PHP” function is available to an external API call, like any other operation on the platform. Use the following info to make an external API call:

Endpoint - https://XXXXXX-admin.mobsted.com/api/v8/code

Body -

{"operationId": XX,

"applicationId": X,

"screenId": XX,

"objectId": XXXXX,

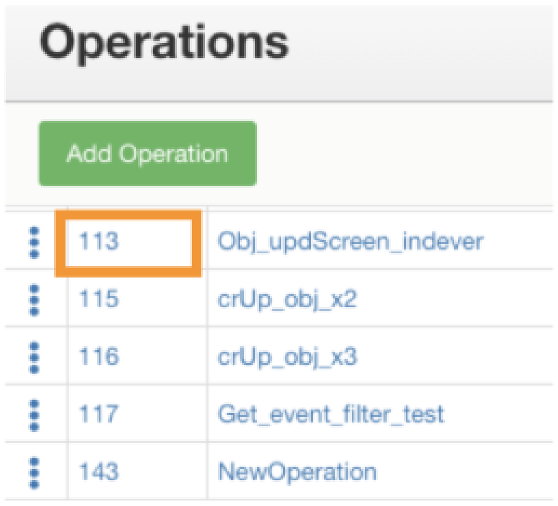
"ExtraParams": {

"variable1": “XXXXX”}} ,

Where required fields are marked in red, and:

Find "**applicationId**" - Check the URL of your app, when in Constructor, for example, XXXX-admin.mobsted.com/applications/edit**/14/**

Find "**operationId**" of your "Custom PHP" in the list of Operations:



113	Obj_updScreen_indever
115	crUp_obj_x2
116	crUp_obj_x3
117	Get_event_filter_test
143	NewOperation

ObjectId is required, as Operation can be connected to #hashtags#, thus referring to some data, so just use any ObjectID, if it is not important.

ExtraParams are to be sent, if your method has key-value pairs predefined, as a nested array, matching your variable key-value pairs.

Working with #Hashtags# data

You can refer to lots of platform's data point in "Custom PHP". You can make calculations, transformations, return results, send results elsewhere, etc.

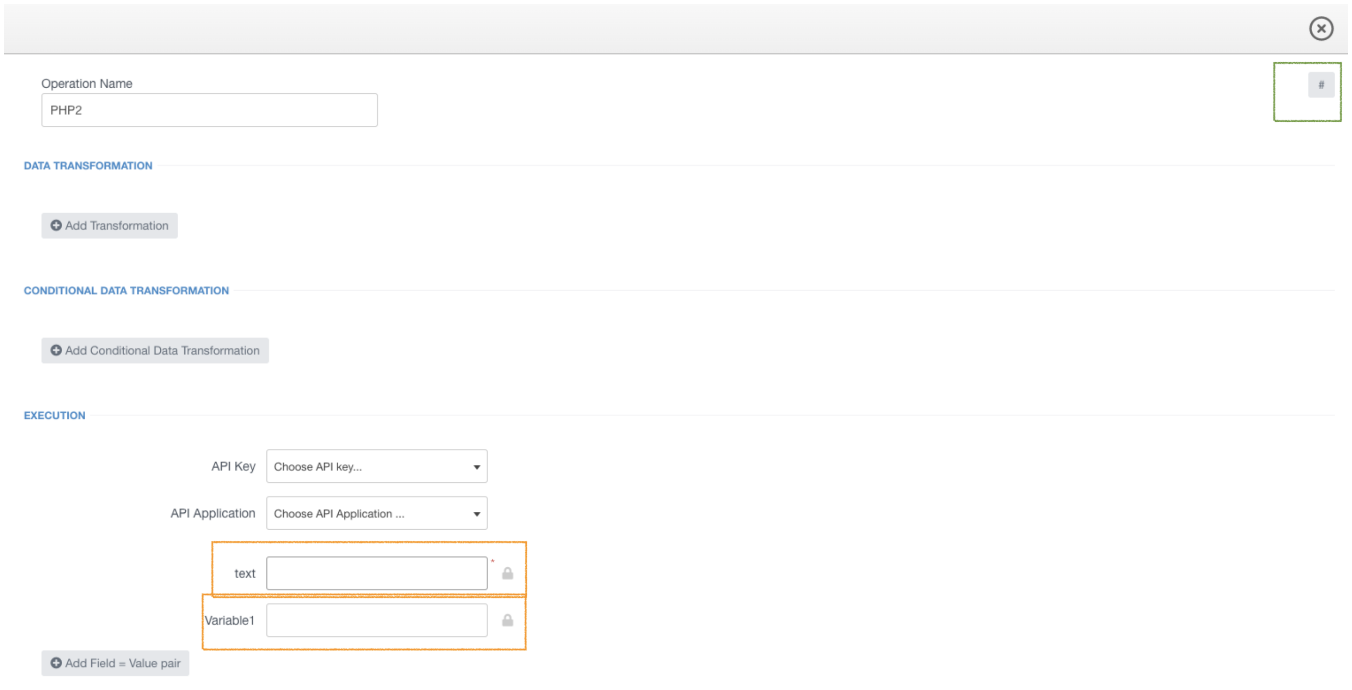
As this code just a normal API call, you can perform it from inside of your APP, or from your outside system, receive results back and do what ever you need with the results.

#Hashtags# available in APP's front end (on a mobile user device), meaning you need to launch the Custom PHP call from a user's/object's perspective.

- Object,
- Screen,
- Application,
- Tenant,
- EventFilter,
- ObjectFilter,
- Operation,
- Event,
- User,
- Variable,
- Backendname,
- System

#Hashtags# available in Triggers -

- Object,
- Application,
- System,
- Tenant,
- Event


Operation Name

PHP2

DATA TRANSFORMATION

CONDITIONAL DATA TRANSFORMATION

EXECUTION

API Key: Choose API key...

API Application: Choose API Application ...

text

Variable1

The #hashtag# generator is marked GREEN

The Custom PHP operation will accept #hashtags# in two places, marked ORANGE:

- PHP code itself (in Text field, declare with “echo”)
- in Variable to insert relevant value in runtime

Modifying PHP on a Fly

You can modify PHP code you need to perform on a fly, without saving it on the platform. So every new execution can have a different source code for it.

Use the default “Custom PHP” method, which needs no creation or naming, as it already exists in Mobsted; and call it externally with a code, which must be performed. The following construct allows this:

Endpoint - https://XXXXXX-admin.mobsted.com/api/v8/code/exec

Body - {"text":”*paste any PHP code here*”}

,where "XXXXXX" is your Mobsted account ID and "Body" contains the code to be performed for each time.

It is important that this way PHP acts OUTSIDE of you app’s environment. Thus you can not access any #hashtags#, but you can modify your PHP code on a go.

- Filters for Objects, Events, Table Lists
- View and Manage Sub-Accounts
- Create Sub-Accounts Manually

- Send Push to Android and iOS
- Create Sub-Accounts by API