Using API to Set Multiple-Option Custom Fields

Oct 03, 2018

HappyFox allows users to create both Ticket & Contact custom fields. These custom fields can be of different types ranging from Text, Number, Date to Multiple-Option. In this article, let us explore how a Multiple-Option custom field can be set via API using PHP.

Sample Multiple-Option custom field

An important thing to remember about multiple-option fields is that they behave similar to lookups/drop-down. When we try to set them via API, we need to pass the “Id” of the lookup values that we have created and not the actual lookup-value itself.
In the sample screen above, if we need to select the following values for Multiple-Option field “Additional Services”

- Refuel
- Thruster Cleanup
- Shield Verification

We need to be using the corresponding Id values in API i.e

Note that the indexing starts from 1 onwards.
So let us consider the scenario where we try to create a ticket via API using a PHP script and we would like to set this multiple-option field “Domain” during creation itself.
Sample php script

```php
<?php
$url = <applicable happyfox url>;
$api_key =
$auth_code =
$input = array(
    "subject" => "Ignition issues with the millenium falcon",
    "text" => "Need to get this piece of junk flying",
    "category" => 1,
    "priority" => 1,
    "email" => "hansolo@rebelalliance.com",
    "name" => "Han Solo",
    "t-cf-1" => "Chewie, we're home!",
    "t-cf-2" => array(
       1, 4, 5
    )
);
$data = json_encode($input);
$headers = array(
    "Content-Type:application/json"
);
$ch = curl_init();
$options = array(
    CURLOPT_URL => $url,
    CURLOPT_HEADER => true,
    CURLOPT_POST => 1,
    CURLOPT_HTTPHEADER => $headers,
    CURLOPT_POSTFIELDS => $data,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_BINARYTRANSFER => true,
    CURLOPT_USERPWD => $api_key . ":" . $auth_code
);
curl_setopt_array($ch, $options);
$result = curl_exec($ch);
if (!curl_errno($ch)) {
    $info = curl_getinfo($ch);
    if ($info['http_code'] == 500) {
        echo "Damn! ERROR 500";
        echo var_dump($info);
    } else {
        echo 'Successful run';
        echo $result;
        echo json_decode($result, true);
    }
}
curl_close($ch);
?>
```

In the above script, we have json-encoded the data that is being used for creating the ticket. Please ensure that when json-encoded data is used, the ContentType matches the type of input data i.e “Content-Type:application/json” in this case.
Also, “Additional Services” custom field is referred to by the internal field-name “t-cf-2”. Note that, the we are passing the lookup-ids to this array instead of the actual values.
On executing this script the following ticket gets created. On the ticket-detail page, we can see the proper display values for “Additional Services” field that we set via API.

Ticket detail page