

Provisioning SDK

Required

- PHP 7.0 or above
- composer

Installation Steps

- Add provisioning SDK in your project

Invoking Provisioning SDK

- First create a new object of ProvisioningAuthCredentialData with necessary parameters
- ProvisioningAuthCredentialData class needs the following parameters
 - clientId → OAuth2 clientId
 - clientSecret → OAuth2 clientSecret
 - accessToken → OAuth2 accessToken
 - refreshToken → OAuth2 refreshToken
 - code → OAuth2 code
 - url → Provisioning URL / External Plugin Url
 - authTokenRenewCallback → Callback function to receive new credential.
 - instanceIdentifier → Instance Identifier **Needed only for when trying to use ExternalPlugin.**
- Now create a object of Provisioning / ExternalPlugin by passing ProvisioningAuthCredentialData object

NOTE

You can pass parameters using array while initializing ProvisioningAuthCredentialData or set them using set method

NOTE

You need to pass a callback function which takes **ProvisioningAuthCredentialData** as parameter in **authTokenRenewCallback** of **ProvisioningAuthCredentialData** to receive new access token and refresh token if the previous access token and refresh token become invalidated

Example:

```

$provisioningAuthCredentialData = new ProvisioningAuthCredentialData([
    'clientId' => "*****",
    'clientSecret' => "*****",
    'accessToken' => "*****",
    'refreshToken' => "*****",
    'code' => "*****",
    'url' => "*****",
    'authTokenRenewCallback' => function(ProvisioningAuthCredentialData
$authCredentialData){
    -----
    -----
}
]);
$provisioning = new Provisioning($provisioningAuthCredentialData);
or
$externalPlugin = new ExternalPlugin($provisioningAuthCredentialData);

```

Template

- Provisioning template object exposes the following methods
 - listAll(\$max, \$offset, \$search, \$type, \$categories)
 - details(\$templateUuid)
 - typesAndCategory()
 - categoriesByType(\$type)

NOTE

To get all template list, you need to pass **null** value in **max** and **offset** parameters. And **categories** parameter takes a comma separated categories (Without any space before and after comma)

Example:

```

$provisioning = new Provisioning($provisioningAuthCredentialData);
$provisioning->template()->listAll(null, null, "Loyal", "general", "Masum Hasan, New Customer");
$provisioning->template()->details("A6C1083A-5B3D-429E-8FC7-7B0970B5EF39");
$provisioning->template()->typesAndCategory();
$provisioning->template()->categoriesByType("general");

```

Package

- Provisioning package object exposes the following methods
 - listAll() → This methods can take two optional parameters. first one **partnerUuid** and second one **generation**

- details(\$packageUuid)

Example:

```
$provisioning = new Provisioning($provisioningAuthCredentialData);  
$provisioning->package()->listAll();  
$provisioning->package()->details("4367DF22-202C-4ADF-993B-13E605032DC4");
```

Account

- Provisioning account object exposes the following methods
 - signUp(array \$subscriptionData)
 - isEmailExists(\$email)
 - loginByEmailAndPassword(\$email, \$password)

Example:

```
$provisioning = new Provisioning($provisioningAuthCredentialData);  
$provisioning->account()->isEmailExists("masumul@bitmascot.com")
```

NOTE

For signUp method, you have to pass an array with necessary information. Example is given below →

```
[
  "user_details" => [
    "first_name" => "Masumul",
    "last_name" => "Hassan",
    "company" => "Bitmascot pvt ltd",
    "email" => "masumul@bitmascot.com",
    "password" => "123456",
    "phone" => "",
    "web" => "",
    "street_address" => "Suite 910, 9 Yarra St",
    "country" => "AU",
    "state" => "VIC",
    "city" => "Victoria",
    "zip" => "3141",
    "is_email_verified" => true
  ],
  "invoice" => [
    [
      "item_uuid" => "80258DD2-20B2-4B7E-98AF-3BFFD022CD9C",
      "item_type" => "PACKAGE",
      "tiered_uuid" => "80BE2786-54B3-4C22-8AAB-39D254AC09C9",
      "tax_uuid" => ""
    ]
  ],
  "payment" => [
    "card_number" => "4242424242424242",
    "expires" => "1225",
    "cvv" => "999",
    "use_default" => false
  ],
  "is_user_registered" => false,
  "website_type" => "CONTENT" / "E-COMMERCE",
  "default_template_uuid" => "BB72DCA5-2E7D-4479-AFCB-F0023B6870A5"
]
```

- | | |
|-------------|---|
| NOTE | In invoice nested array, item_uuid referring to package_uuid as item_type set to PACKAGE but if we set item_type to TEMPLATE then item_uuid will referring to template_uuid |
| NOTE | If a returning customer wants to use previously used card information, you need to set only "use_default" ⇒ true in payment array |
| NOTE | If you want to enable email verification, then set "is_email_verified" ⇒ false , but remember if you enable email verification for a returning customer then customer will not be able to purchase any item. |

Ticketing

- Provisioning ticketing object exposes the following methods
 - `loginWithTicketsByEmailAndPassword($email, $password)`
 - `newInquiry($name, $email, $subject, $message)`

Example:

```
$provisioning = new Provisioning($provisioningAuthCredentialData);  
$provisioning->ticketing()-  
>loginWithTicketsByEmailAndPassword("masumul@bitmascot.com", "123456")  
$provisioning->ticketing()->newInquiry("Masumul", "masum@bitmascot.com", "Subject",  
"message")
```

Price

- Provisioning price object exposes the following method
 - `getAmountWithTax($tieredUuid, $countryCode)`

Example:

```
$provisioning = new Provisioning($provisioningAuthCredentialData);  
$provisioning->price()->getAmountWithTax($tieredUuid, $countryCode)
```

External Plugin

Wiki

- ExternalPlugin Wiki object exposes the following method
 - `search($search)`
 - `suggestion($search)`

Example:

```
$externalPlugin = new ExternalPlugin($provisioningAuthCredentialData);  
$externalPlugin->wiki()->search("Australia");
```