

# eGetSmartTemplates

## Descrizione

Questo metodo può essere utilizzato per ottenere la lista degli smartTemplate presenti all'interno del sistema.

## Utilizzo

I parametri richiesti da questa funzione sono:

- **templateId**: Id del template di cui si vogliono ottenere informazioni, se impostato a **false** verranno ritornati tutti i template presenti.
- **productId**: Id del prodotto di cui si vogliono ottenere i template, **false** se non si vuole applicare questo filtro.
- **languageId**: Id della lingua in cui si vogliono ottenere il nome, la descrizione e i metadata della categoria, per le lingue disponibili vedere il metodo [getAvailableLanguages](#).
- **thumbSize**: dimensione dell'immagine di preview espressa in pixel (la preview è generata in formato quadrato).

In caso di successo ritorna un'array di oggetti così strutturati:

- **templateId**: Id del template.
- **templateCode**: Codice del template.
- **templateName**: Nome del template nella lingua richiesta.
- **templateImage**: Url dell'immagine di preview del template.
- **productId**: Id del prodotto associato al template.
- **colorPaletteType**: Tipologia di palette colore utilizzata nel template.
- **colorPaletteItems**: Array di stringhe che contengono i colori presenti nella palette in valori esadecimali **#FFFFFF**.
- **presetColorId**: Id del preset per i colori impostato nel template, **0** se non presente.
- **presetFontId**: Id del preset per i fonts impostato nel template, **0** se non presente.
- **googleFonts**: Array di interi che contengono gli id dei font google associati a questo template.
- **adobeTypeKit**: Id del kit di adobe associato al template.
- **variableDataAvailable**: **true** se nel progetto sono presenti dati variabile, **false** in caso contrario.
- **experienceType**: Tipologia di interfaccia utente utilizzata nel template.
- **flatWorkingArea**: **true** se sono stati rimossi l'effetto di ombra e il bordo all'area di lavoro, **false** in caso contrario.
- **workingAreaBackgroundColor**: colore di sfondo da applicare all'area di lavoro espresso in valore decimale **#FFFFFF**.
- **fontSizeMin**: valore minimo dello slider per la dimensione del font.
- **fontSizeMax**: valore massimo dello slider per la dimensione del font.
- **fontSizeDefault**: valore di partenza dello slider per la dimensione del font.
- **enable**: **true** se la categoria è attiva, **false** in caso contrario.
- **sortOrder**: Numero che rappresenta l'indice di visualizzazione per questo elemento.
- **dateCreated**: Stringa contenente la data in cui è stato creato il template nel formato **YYYY-MM-DD HH:MM:SS**.
- **dateUpdated**: Stringa contenente la data in cui è stato modificato il template nel formato **YYYY-MM-DD HH:MM:SS**.

## Gestione errori

In caso di errore il metodo genererà un'eccezione che riporterà uno dei seguenti codici di errore:

- **ERROR\_INVALID\_PARAMS**: Uno dei parametri inviati non è valido.
- **ERROR\_INVALID\_TOKEN**: Non è stato effettuato nessun login in precedenza o la sessione è scaduta per inutilizzo.
- **ERROR\_INVALID\_IP**: L'indirizzo ip del server che sta effettuando la chiamata non valido perchè non inserito nella whitelist associata alle credenziali.
- **ERROR\_SERVER\_UNAVAILABLE**: Il server non è disponibile o si è verificato un errore di comunicazione generico.

### ATTENZIONE



#### Possibili tipologie di palette colori

Il campo **colorPaletteType** può avere i seguenti valori:

- **PALETTE**: I colori potranno essere scelti da un palette limitata di valori definita nel template.
- **PICKER**: I colori potranno essere scelti tramite un colorpicker.

In .NET è possibile utilizzare la struttura statica **colorPaletteTypes** presente in Structures.



### Possibili tipologie di interfaccia editor

Il campo **experienceType** può avere i seguenti valori:

- **0:** Interfaccia professionale senza wizard di caricamento foto.
- **1:** Interfaccia professionale con wizard di caricamento foto.
- **2:** Interfaccia smart senza wizard di caricamento foto.
- **3:** Interfaccia smart con wizard di caricamento foto.

In .NET è possibile utilizzare la struttura statica **experienceTypes** presente in Structures.

### PHP

```
require "vendor/autoload.php"

try {
    $ezPrintSDK = new ezPrintSDKWrapper('http://demo.ezprint.it');
    $apiToken = $ezPrintSDK->cLoginAPI('1WHm0icsupAEm9fil9en','yCHZ0nKiyn1MU0DhpgmG');

    $availableLanguages = $ezPrintSDK->backOffice->bGetAvailableLanguages();
    $smartTemplates = $ezPrintSDK->editor->eGetSmartTemplates(false, false, $availableLanguages[0]
['languageId'], 200);

    foreach($smartTemplates as $template) {
        echo $template['templateCode'], ' ', $template['templateName'];
    }
} catch (Exception $e) {
    echo 'Error Code: ', $e->getMessage();
}
```

### Node.js

```
try {
    var util = require('util');
    var ezPrintSDK = require('ezprint-sdk-wrapper');
    ezPrintSDK.initializeWrapper('http://demo.ezprint.it');

    var apiToken = ezPrintSDK.cLoginAPI('1WHm0icsupAEm9fil9en','yCHZ0nKiyn1MU0DhpgmG');

    var availableLanguages = ezPrintSDK.backOffice.bGetAvailableLanguages();
    var smartTemplates = ezPrintSDK.editor.eGetSmartTemplates(false, false, availableLanguages[0].
languageId, 200);

    smartTemplates.forEach(function(template) {
        console.log(template.templateCode);
        console.log(template.templateName);
    });
} catch (ex) {
    console.log(ex);
}
```

## .NET (C#)

```
try {
    ezPrintSDKWrapper.Client ezPrintSDK = new ezPrintSDKWrapper.Client("http://demo.ezprint.it");
    ezPrintSDK.cLoginAPI("lWHm0icsupAEm9fil9en", "yCHZ0nKiynlMU0DhpGmG");

    List<ezPrintSDKWrapper.Structures.languageItem> availableLanguages = ezPrintSDK.BackOffice.
bGetAvailableLanguages();
    List<ezPrintSDKWrapper.Structures.smartTemplateItem> smartTemplates = ezPrintSDK.Editor.
eGetSmartTemplates(false, false, availableLanguages[0].languageId, 200);

    System.Console.WriteLine("Smart Template {0}: {1}", smartTemplates[0].templateCode, smartTemplates
[0].templateName);
} catch (ezPrintException ex) {
    string ErrorMessage = ex.Message;
}
```