

eGetProjects

Descrizione

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

Utilizzo

I parametri richiesti da questo metodo sono:

- **projectId**: Id del progetto, se impostato a **false** verranno ritornati tutti i progetti coerenti con i parametri successivi.
- **userId**: Id dell'utente di cui si vogliono ottenere i progetti, se è presente il **projectId** o se impostato a **false** questo campo viene ignorato.
- **productId**: Id del prodotto di cui si vogliono ottenere i progetti associati, se è presente il **projectId** o se impostato a **false** questo campo viene ignorato.
- **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:

- **projectId**: Id del progetto.
- **projectName**: Nome del progetto.
- **productId**: Id del prodotto a cui è associato il progetto.
- **productTemplateId**: Id del template a cui è associato il progetto (per i prodotti realizzati tramite photo editor, il valore è impostato a 0).
- **userId**: Id dell'utente che ha realizzato il progetto.
- **photoGridGroupId**: Id del gruppo di photoGrid a cui è associato il progetto.
- **imageUrl**: url dell'immagine di preview associata al template o al prodotto.
- **type**: Tipologia del progetto (**0** = photoEditor, **1** = smartEditor).
- **locked**: **true** se il progetto è in modalità sola lettura, **false** se non lo è.
- **purchased**: **true** se il progetto è stato acquistato almeno una volta, **false** se non lo è.
- **isPurchasable**: **true** se il progetto è stato salvato almeno una volta è quindi è acquistabile, **false** se non lo è.
- **internalPageCount**: Intero che rappresenta il numero delle pagine del progetto, per i progetti photoEditor rappresenta il numero totale di foto comprensivo di copie.
- **coverPageCount**: Intero che rappresenta il numero delle pagine di copertina del progetto, per i progetti photoEditor è impostato a **0**.
- **dateCreated**: Stringa contenente la data in cui è stato creato il task nel formato **YYYY-MM-DD HH:MM:SS**.
- **dateUpdated**: Stringa contenente la data in cui è stato modificato il task 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_FILTER_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.

PHP

```
require "vendor/autoload.php"

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

    $projectsList = $ezPrintSDK->editor->eGetProjects(false, false, false, 100);

    foreach($projectsList as $project) {
        echo $project['projectId'], ' ', $project['projectName'];
    }
} catch (Exception $e) {
    echo 'Error Code: ', $e->getMessage();
}
```

Node.js

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

    var apiToken = ezPrintSDK.cLoginAPI('lWHm0icsupAEm9fil9en','yCHZ0nKiynlMU0DhpgmG');
    var projectsList = ezPrintSDK.editor.eGetProjects(false, false, false, 100);

    projectsList.forEach(function(project) {
        console.log(project.projectId);
        console.log(project.projectName);
    });
} 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.projectItem> projectsList = ezPrintSDK.Editor.eGetProjects(false,
false, false, 100);

    foreach (ezPrintSDKWrapper.Structures.projectItem project in projectsList) {
        System.Console.WriteLine("Task{0} : {1}", project.projectId, project.projectName);
    }
} catch (ezPrintException ex) {
    string ErrorMessage = ex.Message;
}
```

Descrizione

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

Utilizzo

I parametri richiesti da questo metodo sono:

- **projectId**: Id del progetto, se impostato a **false** verranno ritornati tutti i progetti coerenti con i parametri successivi.
- **userId**: Id dell'utente di cui si vogliono ottenere i progetti, se è presente il **projectId** o se impostato a **false** questo campo viene ignorato.
- **productId**: Id del prodotto di cui si vogliono ottenere i progetti associati, se è presente il **projectId** o se impostato a **false** questo campo viene ignorato.
- **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:

- **projectId**: Id del progetto.
- **projectName**: Nome del progetto.
- **productId**: Id del prodotto a cui è associato il progetto.
- **productTemplateId**: Id del template a cui è associato il progetto (per i prodotti realizzati tramite photo editor, il valore è impostato a 0).
- **userId**: Id dell'utente che ha realizzato il progetto.
- **photoGridGroupId**: Id del gruppo di photoGrid a cui è associato il progetto.
- **imageUrl**: url dell'immagine di preview associata al template o al prodotto.
- **type**: Tipologia del progetto (**0** = photoEditor, **1** = smartEditor).
- **locked**: **true** se il progetto è in modalità sola lettura, **false** se non lo è.
- **purchased**: **true** se il progetto è stato acquistato almeno una volta, **false** se non lo è.
- **variableDataAvailable**: **true** se nel progetto sono presenti dati variabili, **false**

`variableDataAvailable`

- **dateCreated**: Stringa contenente la data in cui è stato creato il task nel formato **YYYY-MM-DD HH:MM:SS**.
- **dateUpdated**: Stringa contenente la data in cui è stato modificato il task 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_FILTER_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.

PHP

```
require "vendor/autoload.php"

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

    $projectsList = $ezPrintSDK->editor->eGetProjects(false, false, false, 100);

    foreach($projectsList as $project) {
        echo $project['projectId'], ' ', $project['projectName'];
    }
} catch (Exception $e) {
    echo 'Error Code: ', $e->getMessage();
}
```

Node.js

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

    var apiToken = ezPrintSDK.cLoginAPI('1WHm0icsupAEm9fil9en','yCHZ0nKiyn1MU0DhpgmG');
    var projectsList = ezPrintSDK.editor.eGetProjects(false, false, false, 100);

    projectsList.forEach(function(project) {
        console.log(project.projectId);
        console.log(project.projectName);
    });
} catch (ex) {
    console.log(ex);
}
```

.NET (C#)

```
try {
    ezPrintSDKWrapper.Client ezPrintSDK = new ezPrintSDKWrapper.Client("http://demo.ezprint.it");
    ezPrintSDK.cLoginAPI("1WHm0icsupAEm9fil9en", "yCHZ0nKiyn1MU0DhpgmG");
    List<ezPrintSDKWrapper.Structures.projectItem> projectsList = ezPrintSDK.Editor.eGetProjects(false,
false, false, 100);

    foreach (ezPrintSDKWrapper.Structures.projectItem project in projectsList) {
        System.Console.WriteLine("Task{0} : {1}", project.projectId, project.projectName);
    }
} catch (ezPrintException ex) {
    string ErrorMessage = ex.Message;
}
```