

eCreateEditorSession

Descrizione

Questo metodo può essere utilizzato per inizializzare una sessione di editing all'interno del sistema.

Utilizzo

I parametri richiesti da questa funzione sono:

- **sessionType**: Tipologia della sessione.
- **editorType**: Tipologia di editor associato alla sessione.
- **productId**: Id del prodotto associato alla sessione.
- **productCode**: Codice identificativo del prodotto da usare alternativamente al campo productId.
- **templateId**: Id del template associato alla sessione, **0** se non necessario.
- **projectId**: Id del progetto associato alla sessione, **0** se non necessario.
- **projectName**: Nome di partenza del progetto, **stringa vuota** se non necessario.
- **customerId**: Id del cliente associato alla sessione.
- **customerPasswordHash**: hash in formato SHA1 della password del cliente.
- **shoppingCartUrl**: Url del carrello esterno, **false** se non necessario.
- **shoppingCartExtraParams**: Array di oggetti che rappresentano i parametri extra che verranno inviati al carrello esterno così composti:
 - **paramName**: Nome del parametro extra.
 - **paramValue**: Valore del parametro extra.
- **productEstimationUrl**: Url per il calcolo dei prezzi, **false** se non necessario.
- **productEstimationExtraParams**: Array di oggetti che rappresentano i parametri extra che verranno inviati all'url per il calcolo dei prezzi così composti:
 - **paramName**: Nome del parametro extra.
 - **paramValue**: Valore del parametro extra.
- **customizeLogoutUrl**: Url di logout personalizzato.
- **preFilteredProjects**: Array di interi che rappresentano gli id dei progetti visibile all'interno di questa sessione, se vuoto, tutti i progetti sono visibili.
- **verifyProjectNameConflict**: **true** se si vuole verificare durante il salvataggio del progetto che non ci siano progetti con lo stesso nome per lo stesso utente, **false** in caso contrario.
- **languageCode**: Codice della lingua preimpostata nella sessione, **false** se non necessario, per maggiori informazioni sulle lingue disponibili consultare [bGetAvailableLanguages](#).

In caso di successo ritorna una stringa che rappresenta l'id della sessione.

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_CUSTOMER_PASSWORD**: La password dell'utente non è valida
- **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 sessione

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

- **0**: Sessione per designer.
- **1**: Sessione per cliente.

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



Possibili tipologie di editor

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

- **1**: PhotoEditor.
- **2**: SmartEditor.

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



Utilizzo del carrello esterno

Se si utilizzare un sistema di calcolo prezzi esterno andando a popolare il campo **shoppingCartUrl** è consigliato consultare la guida alla gestione dei dati e della risposta necessaria per il corretto funzionamento alla pagina [Gestione carrello esterno](#).



Utilizzo del calcolo prezzi esterno

Se si utilizzare un carrello esterno andando a popolare il campo **productEstimationUrl** è consigliato consultare la guida alla gestione dei dati e della risposta necessaria per il corretto funzionamento alla pagina [Gestione calcolo prezzi esterno](#).



Utilizzo del logout esterno

Se si utilizza la procedura di logout su url esterno andando a popolare il campo **customizeLogoutUrl** al momento del logout verrà effettuato un redirect all'url impostato nella variabile con l'aggiunta del campo **projectId** contenente l'id del progetto associato alla sessione da cui si proviene (eventuali parametri impostati nell'url verranno mantenuti).

PHP

```
require "vendor/autoload.php"

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

    $availableLanguages = $ezPrintSDK->backOffice->bGetAvailableLanguages();
    $customers = $ezPrintSDK->backOffice->bGetCustomers(false, false, false);
    $products = $ezPrintSDK->backOffice->bGetCatalogProducts(false, $availableLanguages[0]
['languageId'], false, true, 100);

    $sessionHash = $ezPrintSDK->editor->eCreateEditorSession(1, $products[0]['editorType'], $products[0]
['productId'], 0, 0, 'Nome Progetto', $customers[0]['customerId'], sha1("password"), false, array(), false,
array(), false, array(), true, false);
} catch (Exception $e) {
    echo 'Error Code: ', $e->getMessage();
}
```

Node.js

```
try {
    var util = require('util');
    var sha1 = require('sha1');
    var ezPrintSDK = require('ezprint-sdk-wrapper');
    ezPrintSDK.initializeWrapper('http://demo.ezprint.it');

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

    var availableLanguages = ezPrintSDK.backOffice.bGetAvailableLanguages();
    var customers = ezPrintSDK.backOffice.bGetCustomers(false, false, false);
    var products = ezPrintSDK.backOffice.bGetCatalogProducts(false, availableLanguages[0].
languageId, false, true, 100);

    var sessionHash = ezPrintSDK.editor.eCreateEditorSession(1, products[0].editorType, products[0].
productId, 0, 0, 'Nome Progetto', customers[0].customerId, sha1("password"), false, [], false, [], false,
[], true, false);
} 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.customerItem> customers = ezPrintSDK.BackOffice.bGetCustomers
(false, false, false);
    List<ezPrintSDKWrapper.Structures.productItem> products = ezPrintSDK.BackOffice.bGetCatalogProducts
(false, availableLanguages[0].languageId, false, true, 100);

    HashAlgorithm algorithmSHA1 = SHA1.Create();
    StringBuilder sb1 = new StringBuilder();

    foreach (byte b in algorithmSHA1.ComputeHash(Encoding.UTF8.GetBytes("palladio")))
        sb1.Append(b.ToString("x2"));

    string sessionHash = ezPrintSDK.Editor.eCreateEditorSession(1, products[0].editorType, products[0].
productId, 0, 0, "Nome Progetto", customers[0].customerId, sb1.ToString(), false, new List<ezPrintSDKWrapper.
Structures.extraParamItem>(), false, new List<ezPrintSDKWrapper.Structures.extraParamItem>(), false, new
integer[] {}, true, false);

} catch (ezPrintException ex) {
    string ErrorMessage = ex.Message;
}
```