

# Bearer API

Bearer HTTP :

## Header'

Authorization: Bearer 808e1cc4630440858f5199e4c0a3e706  
Sign: 1c4e379396faee212c676d500ee12a21354d8f68blacbc40b64065cd7dcd50fa

Authorization Bearer , . Sign , . Sign API.

[CreateInvoice](#). Sign :

## Sign CreateInvoice

eshopId::orderId::serviceName::recipientAmount::recipientCurrency::userName::email::successUrl::failUrl::  
backUrl::resultUrl::expireDate::holdMode::preference::signSecretKey

, , , . . signSecretKey , .

[CreateInvoice](#)

## , CreateInvoice

eshopId: 462539  
orderId: myorder  
recipientAmount: 10.00  
recipientCurrency: RUB  
email: e@e.ru

:

## Sign

462539::myorder::::10.00::RUB::::e@e.ru:::::::::21baff51c1a342f3ac059e61e0894583

UTF-8 SHA256 . :

## Sign

1c4e379396faee212c676d500ee12a21354d8f68blacbc40b64065cd7dcd50fa

Sign.

## API

C#, CreateInvoice. , .

## CreateInvoice

```
public static void CreateInvoice() {  
    // Secret Key -> .  
    string eshopSecretKey = "1234abcd";  
  
    //  
    var param = new Dictionary<string, string>
```

```

{
    ["eshopId"] = "499999", //
    ["orderId"] = " ",
    ["recipientAmount"] = "10.00",
    ["recipientCurrency"] = "TST", // , .
    ["email"] = "youremail@gmail.com", // email intellectmoney.
};

// Sign bearer . , CreateInvoice.
param["hash"] = GetCreateInvoiceHash(param, eshopSecretKey);

//
string bearerToken = "808e1cc4630440858f5109e4c0a3e707";
string signSecretKey = "670bcc16e5c9483fa85096c16ce8c413";

// Sign
string signHashString = $"{param["eshopId"]}::{param["orderId"]}::{param["recipientAmount"]}::{param
["recipientCurrency"]}::{param["email"]}::::::::::::::::::{signSecretKey}";
string signHash = GetHashSHA256(signHashString);

//
HttpClientHandler clientHandler = new HttpClientHandler();
using (var content = new FormUrlEncodedContent(param))
{
    using (HttpClient client = new HttpClient(clientHandler))
    {
        client.DefaultRequestHeaders.Add("Authorization", $"Bearer {bearerToken}");
        client.DefaultRequestHeaders.Add("Sign", signHash);

        using (HttpResponseMessage response = client.PostAsync($"https://api.intellectmoney.ru
/merchant/createInvoice", content).Result)
        {
            string responseText = response.Content.ReadAsStringAsync().Result;
            Console.WriteLine(responseText);
            Console.ReadKey();
        }
    }
}

private static string GetCreateInvoiceHash(Dictionary<string, string> param, string eshopSecretKey)
{
    return GetHashMD5($"{param["eshopId"]}::{param["orderId"]}::{param["recipientAmount"]}::{param
["recipientCurrency"]}::{param["email"]}::::::::::::::::::{eshopSecretKey}");
}

private static string GetHashMD5(string source)
{
    byte[] data = Encoding.UTF8.GetBytes(source);
    using (MD5 md5 = new MD5CryptoServiceProvider())
    {
        data = md5.ComputeHash(data);
    }
    return BitConverter.ToString(data).Replace("-", "").ToLower();
}

private static string GetHashSHA256(string source)
{
    byte[] data = Encoding.UTF8.GetBytes(source);
    using (var sha256 = new SHA256CryptoServiceProvider())
    {
        data = sha256.ComputeHash(data);
    }
    return BitConverter.ToString(data).Replace("-", "").ToLower();
}

```