Caspit Web API

Written By: Dror Saddan, Caspit Software, dror@caspit.biz

Ver: 1.4

Last Update: 18/7/2016

Contents

[First step 2](#_Toc424727091)

[Introduction 3](#_Toc424727092)

 [4](#_Toc424727093)

 [4](#_Toc424727094)

[You need a License to use the API 5](#_Toc424727095)

['Caspit Web' API .Net Project 5](#_Toc424727096)

[Security Token 5](#_Toc424727097)

[Valid Response 6](#_Toc424727098)

[Errors 6](#_Toc424727099)

[Code Example 6](#_Toc424727100)

[Token Expiration 7](#_Toc424727101)

[Foreign Keys 7](#_Toc424727102)

[How to fetch contacts from Caspit 7](#_Toc424727103)

[Contacts 8](#_Toc424727104)

[Expenses 9](#_Toc424727105)

[How to POST a new Contact to Caspit 10](#_Toc424727106)

[How to GET a Contact (לקוח) via his key 10](#_Toc424727107)

[The response - XML Or Json? 11](#_Toc424727108)

[How to GET a Contact (לקוח) via his OsekMorshe number 11](#_Toc424727109)

[Errors 12](#_Toc424727110)

[Infrastructure Code 12](#_Toc424727111)

[Documents 15](#_Toc424727112)

[How to create a document 15](#_Toc424727113)

[The CaspitDocument object (see https://app.caspitweb.biz/ApiHelp/ResourceModel?modelName=CaspitDocument) 24](#_Toc424727114)

[The CaspitDocumentLine Object (see https://app.caspitweb.biz/ApiHelp/ResourceModel?modelName=CaspitDocumentLine) 29](#_Toc424727115)

[The CaspitReceiptLine Object (see https://app.caspitweb.biz/ApiHelp/ResourceModel?modelName=CaspitReceiptLine) 30](#_Toc424727116)

[How to send or print a document 30](#_Toc424727117)

[How to fetch the PDF file 31](#_Toc424727118)

[Products (מוצרים/פריטים) 33](#_Toc424727119)

[How to Get a Product 33](#_Toc424727120)

[Operation Log 34](#_Toc424727121)

[How to GET the log 34](#_Toc424727122)

[The 'Caspit Web' API SDK 35](#_Toc424727123)

[ 35](#_Toc424727124)

[Testing 35](#_Toc424727125)

[FAQ 36](#_Toc424727126)

# First step

In order to get a taste for Caspit's API do the following:

1. Open a browser (Chrome is better than IE for this test).
2. Go to [https://app.caspitweb.biz/api/v1/token?user=demo&pwd=demodemo&osekmorshe=123456789](http://app.caspitweb.biz/api/v1/token?user=demo&pwd=demodemo&osekmorshe=123456789)
3. You will get a TOKEN (e.g 90ddaeb7-51d6-4eb5-85a2-4cdaae401ced).
4. Now fetch contacts via [https://app.caspitweb.biz/api/v1/contacts?token=90ddaeb7-51d6-4eb5-85a2-4cdaae401ced](http://app.caspitweb.biz/api/v1/contacts?token=90ddaeb7-51d6-4eb5-85a2-4cdaae401ced)
5. You will get a list of contacts from Caspit.



*Note: If you perform this demo in Internet Explorer you may get back the results in json format and not in XML format.*

Congratulations! You just performed your first API transaction.

# Introduction

The following document describes the API supported by the Caspit Web ("Caspit") system.

The Caspit Web system is located at [https://app.caspitweb.biz](http://www.caspitweb.biz).

The system has an API based on [the Microsoft ASP.NET Web API](http://www.asp.net/web-api) ("Web API") standard. In short, the Web API standard is based on the idea of **HTTP calls** which allow you to perform [Create, Read Update, Delete](http://en.wikipedia.org/wiki/Create%2C_read%2C_update_and_delete) (CRUD) operations on the entities contained within the system.

The Web API does not use Web Services. Instead it is based on POST, GET and PUT HTTP-calls. Here is how the Http Verbs map into the CRUD operations:

|  |  |
| --- | --- |
| Create | POST |
| Read | GET |
| Update | PUT |
| Delete | DELETE |

In Caspit Web you can:

* Fetch contacts (לקוחות).
* Fetch documents: invoices, receipts, etc.
* Fetch products.
* Fetch expenses and payments for the expenses.
* Create a new Contact (לקוח) and update an existing Contact.
* Create a new Document (חשבונית, חש/קבלה וכו') and update an existing Document.
* Create a new product.
* **Not implemented yet**: Automatically process the Invoice/Kabala: charge the credit card and send the document to the customer.

 Here is the flow of fetching all customers from Caspit Web:

Here is the flow of adding a new Invoice (חשבונית) to Caspit Web:

One very big disadvantage of the Web API approach is that there is no WSDL so you cannot find the API calls and their parameters. This is why you need to carefully follow this documentation.

# You need a License to use the API

Please note that in order to use the API, you need to have the API module enabled in your license record in Caspit. You can verify this in ניהול החשבון שלי בכספית screen (under the API column):



If your license does not include the API your calls to Caspit will be rejected.

# 'Caspit Web' API .Net Project

The way that you communicate with Caspit is by sending the information (e.g. the details of the contact, or the details of the invoice) as URL-Encoded name-value pairs. For example:

In order to make to development easier we supply a .Net project which contains the classes expected by the API. This way you can work with objects instead of single name-pairs. If this sounds confusing, please read on and see the code examples.

# Security Token

Web API does not support Sessions, so each call requires an identification and authorization of the caller by Caspit. One approach would be to require the caller to send its user name and password in each call. In order to minimize the transfer of sensitive information we implemented a token based authorization system.

You start the dialog with Caspit by asking for a token. Once you have the token you need to send it in any further calls. The token is valid for 10 minutes which is by far more than enough time for the simple interactions with Caspit.

Once you added the Contact and the Document discard the token, and ask for a new token in the next interaction with Caspit.

Here is how to get a token.

Send a GET request to [https://app.caspitweb.biz/api/token?user=dsaddan&pwd=a1b2c3&osekmorshe=123456789](http://www.caspitweb.biz/api/token?user=dsaddan&pwd=a1b2c3&osekmorshe=123456789)

You need to send the osekmorshe since there might be several businesses under one account in Caspit.

## Valid Response

If Caspit verifies the user name, password and osekmorshe the response will be:

HTTP/1.1 **200 OK**

Pragma: no-cache

Cache-Control: no-cache

Set-Cookie: ASP.NET\_SessionId=t3j1g5npypqb5aejafegwgf5; path=/; HttpOnly, .ASPXAUTH=E1E469BD34A46354; path=/; HttpOnly

Server: Microsoft-IIS/7.5

X-AspNet-Version: 4.0.30319

X-Powered-By: ASP.NET

Date: Tue, 05 Nov 2013 17:43:10 GMT

<string xmlns="https://schemas.microsoft.com/2003/10/Serialization/">4afcca18-d493-4498-af27-9df3fbd00018</string>

## Errors

If the user name and password do not match the response will be 401 Invalid User.

If the user name, password or osekmorshe do not match or they do match but there is no license for the API the response will be:

HTTP/1.1 401 Invalid osekmorshe

Pragma: no-cache

Cache-Control: no-cache

Set-Cookie: ASP.NET\_SessionId=5clei3dadjnrkdk1h1ymurhn; path=/; HttpOnly, .ASPXAUTH=086001F9799F48CB05504DDCA3DA6157DDA87A605E38A817203B7B72CDB8714847A6D179006BD0D33EEA71A620492786A19F5B29625B069506CCA5ED44E98E09BD710E1205F7BF487654D68674B44955632F3E0E2BE7BA26AF98F8BE2A99FC4A; path=/; HttpOnly

Server: Microsoft-IIS/7.5

X-AspNet-Version: 4.0.30319

X-Powered-By: ASP.NET

Date: Tue, 05 Nov 2013 17:40:39 GMT

Invalid osekmorshe = 123456789 or business is locked or no license to use the API.

## Code Example

 private string GetToken()

 {

 // api/token?user=demo&pwd=demodemo&osekmorshe=098765432

 string url = string.Format("{0}/api/token/?user={1}&pwd={2}&osekmorshe={3}", CaspitWebUrl, UserName, Password, OsekMorshe);

 string content = SendGetRequest(url);

 var doc = XDocument.Parse(content);

 var ns = (XNamespace)"https://schemas.microsoft.com/2003/10/Serialization/";

 string token = doc.Element(ns + "string").Value;

 return token;

 }

The code for SendGetRequest is given later in this documentation.

## Token Expiration

The token is valid for 10 minutes since its creation. If you try to use it after it expired you will get the following error:

HTTP/1.1 401 Invalid token

Pragma: no-cache

Cache-Control: no-cache

Date: Tue, 05 Nov 2013 17:59:10 GMT

Set-Cookie: ASP.NET\_SessionId=h2cw0evlta2kfrmj4c3pjhan; path=/; HttpOnly

Server: Microsoft-IIS/7.5

WWW-Authenticate: CaspitToken

X-AspNet-Version: 4.0.30319

X-Powered-By: ASP.NET

Invalid token = 4afcca18-d493-4498-af27-9df3fbd00018

# Foreign Keys

The way that your system and Caspit communicates is via keys. Caspit expects to receive from you a key for entities (חשבונית, לקוח) that you create. When you need the GET (fetch) a Client or Document you do it via your key. Caspit calls your key a foreign key since it keeps its own keys. Note that Caspit's internal keys are not transferred in API calls.

# How to fetch contacts from Caspit

Caspit returns results in an object called CaspitResponse. In order to understand its usage one needs to understand that Caspit returns the records in batches of 50. This is done to prevent an overload on the server. So in order to fetch all the contacts in Caspit you will need to fetch page 0, page 1, … until the last page.

This object (which you can see in the SDK) has the following fields:

|  |  |
| --- | --- |
| **Field name** | **What is it used for** |
| CurrentPage | The current page fetched. 0 based. |
| TotalCount | The total count of records in Caspit. |
| PrevPageUrl | A url that will fetch the previous page. Empty if there is no previous page. |
| NextPageUrl | A url that will fetch the following page. Empty if there is no following page. |
| Results | A collection of CaspitBase objects. A CaspitBase object may be a document (CaspitDocument), a contact (CaspitContact), a product (CaspitProduct), or expense (CaspitExpense). |

Here are two examples:

### Contacts

<CaspitResponse xmlns:i="https://www.w3.org/2001/XMLSchema-instance" xmlns="https://www.caspit.biz/2013/03" slick-uniqueid="3">

 <CurrentPage>0</CurrentPage>

 <NextPageUrl> https://localhost:50003/api/v1/Contacts?page=1&token=7721bbf6-98fa-4ca3-9f82-1dd95f4e7bef</NextPageUrl>

 <PrevPageUrl/>

 <Results>

 <CaspitBase i:type="CaspitContact">

 <Address1/>

 <Address2/>

 <Balance>15835.0000</Balance>

 <BalanceInvoiceIska>15835.0000</BalanceInvoiceIska>

 <BankAcctBankId i:nil="true"/>

 <BankAcctBranch i:nil="true"/>

 <BankAcctNumber i:nil="true"/>

 <BusinessName>אריה מזרחי</BusinessName>

 <ChargeVAT i:nil="true"/>

 <City/>

 <Comments1 i:nil="true"/>

 <Comments2 i:nil="true"/>

 <Comments3 i:nil="true"/>

 <ContactId>189398ed-d69a-4404-b4cf-716a938ea740</ContactId>

 <ContactType>1</ContactType>

 <Country i:nil="true"/>

 <Email i:nil="true"/>

 <Fax i:nil="true"/>

 <InvoiceComment i:nil="true"/>

 <Mobile i:nil="true"/>

 <Name/>

 <Number>1</Number>

 <OpenBalance i:nil="true"/>

 <OsekMorshe>234567777</OsekMorshe>

 <Phone i:nil="true"/>

 <PostalCode/>

 <PriceListNum>1</PriceListNum>

 <TrxCode>ביגוד מקצועי</TrxCode>

 <TrxCodeNumber>3040</TrxCodeNumber>

 </CaspitBase>

 <CaspitBase i:type="CaspitContact">...</CaspitBase>

 ...

 </Results>

 <TotalCount>35</TotalCount>

 <TotalPages>1</TotalPages>

</CaspitResponse>

### Expenses

<CaspitResponse xmlns:i="https://www.w3.org/2001/XMLSchema-instance" xmlns="https://www.caspit.biz/2013/03" slick-uniqueid="3">

 <CurrentPage>0</CurrentPage>

 <NextPageUrl>https://localhost:50003/api/v1/Expenses?page=1&token=7721bbf6-98fa-4ca3-9f82-1dd95f4e7bef&startDate=1900-01-01&endDate=2100-12-31

 </NextPageUrl>

 <PrevPageUrl/>

 <Results>

 <CaspitBase i:type="CaspitExpense">

 <Date>2013-12-19T00:00:00</Date>

 <Details i:nil="true"/>

 <Number>2</Number>

 <OsekMorshe>234567777</OsekMorshe>

 <Payments>

 <CaspitExpensePayment>

 <BankAccountId i:nil="true"/>

 <PaymentDate>2013-12-16T00:00:00</PaymentDate>

 <PaymentDetails i:nil="true"/>

 <PaymentMethodNumber i:nil="true"/>

 <PaymentTypeId i:nil="true"/>

 <TotalPaid>500.0000</TotalPaid>

 </CaspitExpensePayment>

 </Payments>

 <PortionNumber>0</PortionNumber>

 <Reference i:nil="true"/>

 <SupplierId>189398ed-d69a-4404-b4cf-716a938ea740</SupplierId>

 <SupplierName>אריה מזרחי</SupplierName>

 <Total>500.0000</Total>

 <TotalNoVat>423.7300</TotalNoVat>

 <TrxCode>ביגוד מקצועי</TrxCode>

 <TrxCodeNumber>3040</TrxCodeNumber>

 <Vat>76.2700</Vat>

 <VatRate>18.0000</VatRate>

 </CaspitBase>

 <CaspitBase i:type="CaspitExpense">...</CaspitBase>

 ...

 </Results>

 <TotalCount>55</TotalCount>

 <TotalPages>2</TotalPages>

</CaspitResponse>

# How to POST a new Contact to Caspit

In order to create a new Contact you need to send a POST message containing his information:

POST /api/contacts/6800?token=9e52774c-ac03-41cb-8d2f-473188935c94 HTTP/1.1

{"ContactId":"6801","BusinessId":0, "ContactType":1,"Number":null, "BusinessName":"אלדר קיפניס", "OsekMorshe":null, "Name":"גדעון","Address1":"רח' הפרחים 7","Address2":"<שכונת עין הים>","City":"עתלית","PostalCode":null, "Country":null,"Phone":null, "Mobile":null,"Fax":null,"Email":null,"TrxCodeNumber":null, "TrxCode":null,"OpenBalance":null,"Balance":null, "BalanceInvoiceIska":null,"PriceListNum":null, "Comments1":"שלום & ברכה","Comments2":"", "Comments3":"", "InvoiceComment":"\"מספר לקוח שלנו: 823829\"","BankAcctNumber":"1331344", "BankAcctBranch":"829", "BankAcctBankId":10, "ChargeVAT":null}

The Contact information is sent in JSON-format. There is no need to escape or URL-encode special HTML characters (e.g. <, >, &), but you do need to esacpe " (e.g. "\"Your customer ID is 823829\"").

Note that the BusinessId is ignored by the API. It will create the new Contact in the business which the token is related to (when you GET the token you also send the OsekMorshe of the business you want to work with).

# How to GET a Contact (לקוח) via his key

In order to GET a client you will need to know his key. For this example we will assume that you created the Client via a POST so you have his key which is "6801". You also have a valid token.

You send:

GET /api/contacts/6801?token=e0fc8053-9845-42ec-a373-ff1694a4be5b HTTP/1.1

Accept: application/xml

You will receive:

HTTP/1.1 200 OK

Pragma: no-cache

Cache-Control: no-cache

Date: Tue, 05 Nov 2013 18:02:48 GMT

Set-Cookie: ASP.NET\_SessionId=x3b3fukksgnxeutwa44ai50d; path=/; HttpOnly

Server: Microsoft-IIS/7.5

X-AspNet-Version: 4.0.30319

X-Powered-By: ASP.NET

<CaspitContact xmlns:i="https://www.w3.org/2001/XMLSchema-instance" xmlns="https://www.caspit.biz/2013/03"><Address1>רח' הפרחים 7</Address1><Address2>&lt;שכונת עין הים&gt;</Address2><Balance>0.0000</Balance><BalanceInvoiceIska>0.0000</BalanceInvoiceIska><BankAcctBankId>10</BankAcctBankId><BankAcctBranch>829</BankAcctBranch><BankAcctNumber>1331344</BankAcctNumber><BusinessId>817</BusinessId><BusinessName>אלדר קיפניס</BusinessName><ChargeVAT i:nil="true" /><City>עתלית</City><Comments1>שלום &amp; ברכה</Comments1><Comments2></Comments2><Comments3></Comments3><ContactType>1</ContactType><Country i:nil="true" /><Email i:nil="true" /><Fax i:nil="true" /><ContactId>6801</ContactId ><InvoiceComment>"מספר לקוח שלנו: 823829"</InvoiceComment><Mobile i:nil="true" /><Name>גדעון</Name><Number>53</Number><OpenBalance i:nil="true" /><OsekMorshe i:nil="true" /><Phone i:nil="true" /><PostalCode i:nil="true" /><PriceListNum>1</PriceListNum><TrxCode i:nil="true" /><TrxCodeNumber i:nil="true" /></CaspitContact>

The <CaspitContact> XML is simply a serialization of the CaspitContact object (defined in the Caspit Web API SDK). You can load the XML into the object and use it in your code. You can also use the code to understand the semantics (meaning) of each field.

### The response - XML Or Json?

We received XML since we sent Accept: application/xml (this is a feature if the Web API and not if Caspit). You can also get a JSON response. Send:

GET /api/contacts/6801?token=11201f49-f457-472e-a4f3-f13b2aa5a2bb HTTP/1.1

Accept: application/json

And the response is:

HTTP/1.1 200 OK

Pragma: no-cache

X-SourceFiles: =?UTF-8?B?RDpcQ2FzcGl0RGV2XFdlYlxDYXNwaXRXZWJUaWxlc1xDYXNwaXRXZWJUaWxlcy5XZWJcYXBpXGNvbnRhY3RzXDY4MDE=?=

Cache-Control: no-cache

Date: Tue, 05 Nov 2013 20:19:30 GMT

Set-Cookie: ASP.NET\_SessionId=3vox3yxbkeku5dyyakbdjajb; path=/; HttpOnly

Server: Microsoft-IIS/8.0

X-AspNet-Version: 4.0.30319

X-Powered-By: ASP.NET

{"ContactId":"6801","BusinessId":817,"ContactType":1,"Number":53,"BusinessName":"אלדר קיפניס","OsekMorshe":null,"Name":"גדעון","Address1":"רח' הפרחים 7","Address2":"<שכונת עין הים>","City":"עתלית","PostalCode":null,"Country":null,"Phone":null,"Mobile":null,"Fax":null,"Email":null,"TrxCodeNumber":null,"TrxCode":null,"OpenBalance":null,"Balance":0.0000,"BalanceInvoiceIska":0.0000,"PriceListNum":1,"Comments1":"שלום & ברכה","Comments2":"","Comments3":"","InvoiceComment":"\"מספר לקוח שלנו: 823829\"","BankAcctNumber":"1331344","BankAcctBranch":"829","BankAcctBankId":10,"ChargeVAT":null}

# How to GET a Contact (לקוח) via his OsekMorshe number

If you need to GET a Contact that was created via Caspit's GUI then you can use his OsekMorshe number:

GET https://localhost:50003//api/contacts/?token=9e52774c-ac03-41cb-8d2f-473188935c94&osekMorshe=111111118&d=1 HTTP/1.1

Accept: application/xml

Note the d=1 parameter. While has no semantics (= it does nothing), it is a mandatory parameter nonetheless. The API must receive it for technical reasons.

## Errors

If Caspit cannot find the contact it will return the following error message:

HTTP/1.1 404 Contact Not Found

Pragma: no-cache

X-SourceFiles: =?UTF-8?B?RDpcQ2FzcGl0RGV2XWJcYXBpXGNvbnRhY3RzXA==?=

Cache-Control: no-cache

Date: Tue, 05 Nov 2013 19:28:15 GMT

Set-Cookie: ASP.NET\_SessionId=23atvpycn4l1a1f4ifpla1wt; path=/; HttpOnly

Server: Microsoft-IIS/8.0

X-AspNet-Version: 4.0.30319

X-Powered-By: ASP.NET

No Contact with ForeignKey/OsekMorshe = 057103855

If the OsekMorshe number fits more than one Client you will get:

HTTP/1.1 409 Invalid OsekMorshe number

Pragma: no-cache

X-SourceFiles: =?UTF-8?B?RDpcQ2FzcGl0RGV2XFdlYlxDYXNwaXRXZWJUaWxlc1xDYXNwaXRXZWJUaWxlcy5XZWJcYXBpXGNvbnRhY3RzXA==?=

Cache-Control: no-cache

Date: Tue, 05 Nov 2013 20:16:46 GMT

Set-Cookie: ASP.NET\_SessionId=gbd1sks51oiv3owbwr01tgoe; path=/; HttpOnly

Server: Microsoft-IIS/8.0

X-AspNet-Version: 4.0.30319

X-Powered-By: ASP.NET

Invalid OsekMorshe = 5

# Infrastructure Code

Here is the code that you will need to send HTTP messages to Caspit. The code uses the [HttpClient](http://msdn.microsoft.com/en-us/library/system.net.http.httpclient%28v%3Dvs.110%29.aspx) introduced in .Net Framework 4.5. If you are using an earlier .Net Framework or if you are using other frameworks (e.g PHP), you can use other techniques. All you basically need to do is

* Send an HTTP message, get the response,
* Return the object information in case of GET
* Return the new object id in case of POST.
* Throw an error in case of an error

using System;

using System.IO;

using System.Net;

using System.Net.Http;

using System.Net.Http.Formatting;

using System.Net.Http.Headers;

 public static string SendGetRequest(string url, string method = "GET")

 {

 HttpClient client = PrepareHttpClient();

 HttpRequestMessage request = new HttpRequestMessage(new HttpMethod(method), url);

 var taskReult = client.SendAsync(request);

 HttpResponseMessage httpResponse = taskReult.Result;

 Stream st = httpResponse.Content.ReadAsStreamAsync().Result;

 StreamReader reader = new StreamReader(st);

 string content = reader.ReadToEnd();

 //For GET we expect a response of 200 OK

 if (httpResponse.StatusCode == HttpStatusCode.OK)

 {

 return content;

 }

 throw new ApplicationException(content);

 }

 /// <summary>

 /// Post to the server using JSON

 /// </summary>

 /// <typeparam name="T"></typeparam>

 /// <param name="url">The server uri</param>

 /// <param name="e">The object to POST</param>

 /// <returns></returns>

 public static string SendPostRequest<T>(string url, T e, string method = "POST")

 {

 HttpClient client = PrepareHttpClient();

 // Create the JSON formatter.

 MediaTypeFormatter jsonFormatter = new JsonMediaTypeFormatter();

 // Use the JSON formatter to create the content of the request body.

 HttpContent content = new ObjectContent<T>(e, jsonFormatter);

 Stream st = content.ReadAsStreamAsync().Result;

 StreamReader reader = new StreamReader(st);

 string s = reader.ReadToEnd();

 HttpRequestMessage request = new HttpRequestMessage(new HttpMethod(method), url);

 request.Content = content;

 // Send the request.

 var taskResult = client.SendAsync(request);

 //Note: We could simply perform the following line and save some time

 //but then we will not have access to the post content:

 //var taskResult = client.PostAsJsonAsync<T>(url, e);

 HttpResponseMessage httpResponse = taskResult.Result;

 st = httpResponse.Content.ReadAsStreamAsync().Result;

 reader = new StreamReader(st);

 string responseContent = reader.ReadToEnd();

 switch (method)

 {

 case "POST":

 //For POST we expect a response of 201 Created and the id of the new object

 if (httpResponse.StatusCode == HttpStatusCode.Created)

 {

 return responseContent;

 }

 break;

 case "PUT":

 //For POST we expect a response of 200 OK

 if (httpResponse.StatusCode == HttpStatusCode.OK)

 {

 return null;

 }

 break;

 }

 throw new ApplicationException(responseContent);

 }

 public static void SendPutRequest<T>(string url, T e)

 {

 SendPostRequest<T>(url, e, "PUT");

 }

 private static HttpClient PrepareHttpClient()

 {

 var client = new HttpClient { Timeout = new TimeSpan(0, 3, 0) };

 client.DefaultRequestHeaders.Accept.Add(

 new MediaTypeWithQualityHeaderValue("application/xml"));

 return client;

 }

# Documents

You can create the following documents in Caspit:

|  |  |
| --- | --- |
| **Document trx type** | **Value** |
| חשבונית | 1 |
| קבלה | 2 |
| חשבונית/ידנית | 4 |
| חשבונית עסקה | 6 |
| חשבונית/קבלה | 7 |
| חשבונית זיכוי | 9 |
| קבלה לתרומה – ידנית*שם לב שקבלה רגילה לתרומה היא 19* | 11 |
| קבלה ידנית | 12 |
| תעודת החזר כספים | 13 |
| הזמנת עבודה | 14 |
| תעודת משלוח | 15 |
| הצעת מחיר | 16 |
| קבלה לתרומה | 19 |
| חשבון (מחליף את חשבונית בגרסת עמותות) | 30 |
| חשבון/קבלה (מחליף את חשבונית/קבלה בגרסת עמותות) | 31 |
| חשבון זיכוי | 34 |
| הזמנת רכש | 47 |

## How to create a document

In order to create a document, send a POST to the following address: [https://app.caspitweb.biz/v1/Documents/token={value\_of\_token}](http://app.caspitweb.biz/ApiHelp/Api/POST-api-v1-Documents_token)

The body should contain a JSON or XML with the document parameters.

Every document in Caspit has a TrxTypeId fields, which indicates wether it is an Invoice, Receipt, Invoice/Receipt,… and a TrxCodeId (סיווג תנועה) which is used to associate the document with a group of similar documents.

For example, suppose that I have two business lines: Line A, and Line B.

I can create two TrxCodes: "51-Line A", and "52Line B". An invoice for a product in Line A will have TrxTypeId=1 (Invoice) and TrxCodeNumber=51, while an Invoice for a product in Line B will have TrxTypeId=1 (Invoice) and TrxCodeNumber=52.

When you print a P&L report (דוח רווח והפסד) there will be two lines under revenues: Line A containing the sum of all invoices with TrxCodeNumber 51, and Line B containing the sum of all the invoices with TrxCodeNumber 52.

Here is an example for an Invoice:



POST https://localhost:50003/api/v1/documents/I-4040?token=09bdd479-f512-4110-8ac1-889f01068513 HTTP/1.1

Content-Type: application/json; charset=utf-8;

Content-Disposition:

Accept: application/xml

{"DocumentLines":[{"Number":1,"ProductId":"ca74f15a-525e-403a-a67d-46b05fb210d2","ProductName":null,"Details":"סדנת יעוץ למנהלים בכירים להשגת המטרות העיסקיות","UnitPrice":12455.93,"Qty":1.00,"CurrencySymbol":"₪","Rate":1.00,"Rebate":100.00,"ExtendedPrice":12355.930000,"ChargeVAT":true,"VATRate":null,"VAT":null},null,null],"DocumentId":"I-4040","TrxTypeId":1,"Number":null,"Date":"2014-12-22T00:00:00","CustomerId":"","CustomerBusinessName":"בית יציקה הלחמי","CustomerOsekMorshe":"829076542","CustomerContactName":"יחיאל הלחמי","CustomerAddress1":"","CustomerAddress2":"","CustomerCity":"אשקלון","CustomerPostalCode":"","CustomerEmail":null,"Details":"","Comments":"","TotalLinesChargeVAT":0.0,"Rebate":0.00,"RountTotal":false,"RebateRounding":null,"TotalBeforeVATChargeVAT":null,"TotalNoChargeVat":null,"Vat":0.0,"Total":null,"VatRate":null,"Payment":null,"TaxDeduction":0.0,"TotalPayment":null,"ReceiptCurrencySymbol":"","ReceiptRate":0.0,"TaxDeductionNIS":0.0,"PaymentNIS":null,"TotalPaymentNIS":null,"ReceiptLines":[null,null],"TrxCode":null,"TrxCodeNumber":3,"Status":null,"DueDate":null,"RowVersion":"AAAAAABQuG4="}

Note the following parameters:

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Value** | **Comments** |
| TrxTypeId (= סוג תנועה) | 1 | קבלה |
| TrxCodeNumber (סיווג תנועה) | 3

|  |  |
| --- | --- |
| **Number** | **Value** |
| 2 | קבלה |
| 3 | הכנסות/מכירות |
| 21 | קניית רכוש קבוע מוכר למעמ |
| 22 | קניית רכוש קבוע לא מוכר למעמ |
| 23 | מכירת רכוש קבוע מוכר למעמ |
| 24 | מכירת רכוש קבוע לא מוכר למעמ |
| 50 | הזמנות רכש |
|  |  |

 | הכנסות/מכירות |

The document contains an array of DocumentLines (=שורות מסמך):

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Value** | **הערות** |
| Number | 1, 2, 3, ... | מספר השורה |
| ProductId | ca74f15a-525e-403a-a67d-46b05fb210d2 | ה-ID של הפריט. לא חובה. משמש לקישור של החשבונית לפריט לצורך ניהול מלאי, דוחות |
| ProductName | "כספית 78245" | שם המוצר |
| Details | "כספית פלדה מחוסמת, מנעול 4 צילינדרים" | תיאור המוצר |
| UnitPrice | 19.90 | מחיר יחידה – זנו המחיר במטבע שבו נמכר הפריט. כלומר, אם הפריט נמכר בש"ח רשום 19.90, אם הפריט נמכר ביורו רשום גם כן 19.90 , המטבע ושער החליפין נמסרים בשדות הבאים |
| Qty | 2.67 | הכמות הנמכרת |
| CurrencySymbol | סמל המטבע נלקח מטבלת המטבעות של כספית (תחזוקה, מטבעות)

|  |  |
| --- | --- |
| **Symbol** | **Currency** |
| ₪ | שקל חדש |
| $ | USD |
| € | Euro |
| $A | דולר אוסטרלי |

 | המטבע |
| Rate | 5.479 | שער החליפין |
| Rebate | 128.98 | ההנחה במטבע של השורה (אם המטבע הוא יורו וההנחה היא של 100 יורו, רשום 100) |
| ChargeVat | True/False | האם לגבות מע"מ על השורה. |

Here is an example for a Receipt:



POST https://localhost:50003/api/v1/documents/R-3009?token=0df968d5-8786-415e-bab7-de840dc4d2c3 HTTP/1.1

Content-Type: application/json; charset=utf-8;

Content-Disposition:

Accept: application/xml

{"DocumentLines":[],"DocumentId":"R-3009", "TrxTypeId":2, "Number":null, "Date":"2014-12-22T00:00:00", "CustomerId":"", "CustomerBusinessName":"בית יציקה הלחמי", "CustomerOsekMorshe":"829076542", "CustomerContactName":"יחיאל הלחמי", "CustomerAddress1":"", "CustomerAddress2":"", "CustomerCity":"אשקלון", "CustomerPostalCode":"", "CustomerEmail":null, "Details":"", "Comments":"","TotalLinesChargeVAT":0.0,"Rebate":0.0,"RountTotal":false, "RebateRounding":null, "TotalBeforeVATChargeVAT":null, "TotalNoChargeVat":null,"Vat":0.0, "Total":null,"VatRate":null,"Payment":null, "TaxDeduction":500.0,"TotalPayment":null,"ReceiptCurrencySymbol":"","ReceiptRate":5.40, "TaxDeductionNIS":2700.00,"PaymentNIS":null, "TotalPaymentNIS":null, "ReceiptLines":[{"Number":1,"PaymentTypeId":1, "AccountNumber":"","BranchOrValidTo":"","BankId":null,"CheckNumber":"", "PaymentDate":null,"Payment":1200.0},{"Number":2,"PaymentTypeId":2,"AccountNumber":"1111111","BranchOrValidTo":"22","BankId":11, "CheckNumber":"30001","PaymentDate":"2014-12-24T00:00:00","Payment":1000.0}],"TrxCode":null, "TrxCodeNumber":2,"Status":null, "DueDate":null, "RowVersion":""}

Note the following parameters:

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Value** | **Comments** |
| TrxTypeId (= סוג תנועה) | 2 | קבלה |
| TrxCodeNumber (סיווג תנועה) | 2 | קבלות |
| ReceiptCurrencySymbol | נלקח מטבלת המטבעות – ראה הסבר בשורת המסמך. | המטבע של חלק הקבלה |

There are no DocumentLines, only ReceiptLines. A receipt line has the following parameters:

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Value** | **הערות** |
| Number | 1, 2, 3, ... | מספר השורה |
| PaymentTypeId =  |

|  |  |
| --- | --- |
| **Number** | **Value** |
| 1 | מזומן |
| 2 | שיק |
| 3 | ויזה לאומי |
| 5 | ישראכרד |
| 6 | מאסטרכרד |
| 7 | אמקס |
| 8 | כרטיס אשראי אחר |
| 9 | העברה בנקאית |
| 10 | הוראת קבע |
| 11 | תשלום מפיקדון |
| 12 | אחר |
| 13 | שובר דואר |
| 14 | אינטרנט |
| 15  | פייפאל |

 | אמצעי התשלום |
| AccountNumber |  | שיק - מספר ח-ן בנקלכ"א –מספר הכרטיס |
| BranckOrValidTo |  | לשיק - מספר הסניףלכ"א – תוקף (01/17 או 0117( |
| BankId |  | לשיק – מספר הבנק |
| CheckNumber |  | לשיק - מספר השיק |
| PaymentDate |  | לשיק -- תאריך הפרעוןלכ"א – תאריך הזיכוי בבנק |
| Payment |  | התשלום – זהו הסכום הנומינלי ששולם. כלומר, אם שולמו 100 ₪, רשום 100. אם שולמו 100 יורו, גם כן רשום 100.המטבע ושער החליפין מועברים בשדות ברמת הקבלה ולא ברמת השורה. |

Here is an example for Invoice/Receipt (חש/קבלה):



מכירות/הכנסות

POST https://localhost:50003/api/v1/documents/I-4049?token=09bdd479-f512-4110-8ac1-889f01068513 HTTP/1.1

Content-Type: application/json; charset=utf-8;

Content-Disposition:

Accept: application/xml

{"DocumentLines":[{"Number":1,"ProductId":"ca74f15a-525e-403a-a67d-46b05fb210d2","ProductName":null,"Details":"","UnitPrice":12455.932,"Qty":1.0,"CurrencySymbol":"₪","Rate":1.0,"Rebate":100.0,"ExtendedPrice":12355.932,"ChargeVAT":true,"VATRate":null,"VAT":null},null,null],"DocumentId":"I-4049","TrxTypeId":7,"Number":null,"Date":"2014-12-22T00:00:00","CustomerId":"","CustomerBusinessName":"בית יציקה הלחמי","CustomerOsekMorshe":"829076542","CustomerContactName":"יחיאל הלחמי","CustomerAddress1":"","CustomerAddress2":"","CustomerCity":"אשקלון","CustomerPostalCode":"","CustomerEmail":null,"Details":"","Comments":"","TotalLinesChargeVAT":0.0,"Rebate":0.00,"RountTotal":false,"RebateRounding":null,"TotalBeforeVATChargeVAT":null,"TotalNoChargeVat":null,"Vat":0.0,"Total":null,"VatRate":null,"Payment":null,"TaxDeduction":500.00,"TotalPayment":null,"ReceiptCurrencySymbol":"","ReceiptRate":5.40,"TaxDeductionNIS":2700.0000,"PaymentNIS":null,"TotalPaymentNIS":null,"ReceiptLines":[{"Number":1,"PaymentTypeId":1,"AccountNumber":"","BranchOrValidTo":"","BankId":null,"CheckNumber":"","PaymentDate":null,"Payment":1200.00},{"Number":2,"PaymentTypeId":2,"AccountNumber":"1111111","BranchOrValidTo":"22","BankId":null,"CheckNumber":"30001","PaymentDate":"2014-12-24T00:00:00","Payment":1000.00}],"TrxCode":null,"TrxCodeNumber":3,"Status":null,"DueDate":null,"RowVersion":""}

## The CaspitDocument object (see <https://app.caspitweb.biz/ApiHelp/ResourceModel?modelName=CaspitDocument>)

Here is a list of the fields of the CaspitDocument. The mandatory columns are used to indicate fields that you must populate to create a new document. The object represents a Caspit document (e.g. חשבונית מס, חש/קבלה, קבלה, תעודת משלוח) An invoice will only have DocumentLines, a receipt will only have ReceiptLines, a חש/קבלה will have both

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Description | Type | Mandatory for Invoices | Mandatory for Receipts |
| DocumentId | The document's Id. You need to supply a unique Id that you will use to fetch or update the document. | String 0-50 | No | No |
| TrxTypeId | See the enumeration in ApiGlobals.TrxTypes | integer | Yes | Yes |
| Number | מספר המסמך. לדוגמה: 09/400256 לא בשימוש ביצירה/עדכון מאחר שכספית ממספרת את המסמכים באופן אוטומטי ולא ניתן לשנות את המספר של מסמך קיים. כלומר, השדה משמש רק בשליפה של נתונים. | String 0-50 | No | No |
| Date | תאריך המסמך | date | Yes | Yes |
| TrxCode | שמו של סיווג תנועה של המסמך. לדוגמה: "מכירות" | String 0-50 | No | No |
| TrxCodeNumber | מספרו של סיווג תנועה של המסמך. לדוגמה: 3 | integer | Yes | Yes – Must be קבלות |
| Status | See Globals.DocStatus | integer | No | No |
| CustomerId | The document's customer id. | String 0-50 | No | No |
| CustomerBusinessName | Customer's business name | String 0-200 | No | No |
| CustomerOsekMorshe | מספר עוסק מורשה של הלקוח | String 0-20 | No | No |
| CustomerContactName | שם איש הקשר -- ללקוח בכספית יכולים להיות 0, 1 או יותר אנשי קשר | String 0-200 | No | No |
| CustomerAddress1 | שורת כתובת1: שם הרחוב ומספר הבית | String 0-100 | No | No |
| CustomerAddress2 | שורת כתובת2: ת.ד., דירה, שכונה וכו' | String 0-100 | No | No |
| CustomerCity | שם הישוב | String 0-100 | No | No |
| CustomerPostalCode | המיקוד | String 0-20 | No | No |
| CustomerEmail | כתובת המייל של איש הקשר | String 0-100 | No.If you do not send a value Caspit will use the contact's email. | No.If you do not send a value Caspit will use the contact's email. |
| MobilePhone | מספר הטלפון הנייד של איש הקשראם | String 0-25 | NoIf you do not send a value Caspit will use the contact's mobile number | NoIf you do not send a value Caspit will use the contact's mobile number |
| Details | פרטים שלא מוצגים ללקוח | String 0-1000 | No | No |
| Comments | הערות שכן מוצגות ללקוח | String 0-1000 | No | No |
| TotalLinesChargeVAT | סה"כ לפני מע"מ של שורות החשבונית (לאחר הנחת שורה), ולפני הנחות מסמך (=הנחה סתם, המחת עיגול) של הסכום שיש עליו מע"מ לא בשימוש ביצירה/עדכון. מאוכלס בשליפה | decimal | No | No (N/A) |
| Rebate | ההנחה בש"ח שיש במסמך. ההנחה היא מהסכום שיש עליו מעמ. כלומר לא ניתן לתת הנחה מהסכום שאין עליו מע"מ. במקרה כזה תן את ההנחה בשורת המסמך המתאימה. | decimal | No | No (N/A) |
| RountTotal | אם תשלח false | boolean | No | No (N/A) |
| RebateRounding | הנחת עיגול בש"ח ההנחה יכולה להיות שלילית בכדי לעגל את סכום המסמך כלפי מעלה. לא בשימוש ביצירה/עדכון. מאוכלס בשליפה | decimal  | No | No (N/A) |
| TotalBeforeVATChargeVAT | סה"כ לפני מע"מ ואחרי הנחות (=הנחה סתם, המחת עיגול) של הסכום שיש עליו מע"מ לא בשימוש ביצירה/עדכון. מאוכלס בשליפה | decimal  | No | No (N/A) |
| TotalNoChargeVat | סה"כ הסכום בש"ח שאין עליו מע"מ לא בשימוש ביצירה/עדכון. מאוכלס בשליפה | decimal  | No | No (N/A) |
| Vat | המעמ בש"ח שנגבה במסמך לא בשימוש ביצירה/עדכון. מאוכלס בשליפה | decimal  | No | No (N/A) |
| Total | סה"כ הסכום כולל מע"מ שנגבה במסמך לא בשימוש ביצירה/עדכון. מאוכלס בשליפה | decimal  | No | No (N/A) |
| VatRate | אחוז המעמ בחשבונית (17 מסמל 17%) אם תשלח NULL אז השדה יאוכלס בערך המתאים לתאריך המסמך. | decimal  | No | No |
| Payment | סכום התשלום | decimal  | No | Yes |
| TaxDeduction | הניכוי במקור | decimal  | No | Yes |
| TotalPayment | סה"כ שולם. שם לב, זהו הסכום הנומינלי ששולם. כלומר, אם שולמו 40 דולר הערך בשדה יהיה 40 (ולא השווי בש"ח) בכדי לקבל את הסכום ששולם בש"ח השתמש ב-TotalPaymentNIS | decimal  | No (N/A) |  |
| ReceiptCurrencySymbol | המטבע עבור חלק הקבלה הסמלים הם ₪, $, €. יתכן שהמשתמש הוסיף גם מטבעות משלו | string | No (N/A) | Yes |
| ReceiptRate | שער חליפין עבור חלק הקבלה | decimal  | No (N/A) | Yes |
| TaxDeductionNIS | ניכוי במקור בש"ח מחושב לפי הסכום הנומינלי מוכפל בשער החליפין | decimal  | No (N/A) | Noשדה מחושב |
| PaymentNIS | שולם אחרי ניכוי במקור בש"ח מחושב לפי הסכום הנומינלי מוכפל בשער החליפין | decimal  | No (N/A) | Noשדה מחושב |
| TotalPaymentNIS | סה"כ שולם בש"ח מחושב לפי סה"כ התשלום הנומינלי מוכפל בשער החליפין | decimal  | No (N/A) | Noשדה מחושב |
| ReceiptLines |  | Collection of [CaspitReceiptLine](https://app.caspitweb.biz/ApiHelp/ResourceModel?modelName=CaspitReceiptLine) | No (N/A) | Yes |
| DocumentLines | שורות המסמך | Collection of [CaspitDocumentLine](https://app.caspitweb.biz/ApiHelp/ResourceModel?modelName=CaspitDocumentLine) | Yes | No (N/A) |
|  |  |  |  |  |
| DueDate | תאריך תשלום צפוי | date | No | No (N/A) |
| RowVersion | This field is used to identify changes made to the Document by other user When you update a documend send the RowVersion value that you received when you fetched the document | Collection of byte | No | No |
| LinkToPdf | A link to the document's PDF file | String | No | No |

## The CaspitDocumentLine Object (see <https://app.caspitweb.biz/ApiHelp/ResourceModel?modelName=CaspitDocumentLine>)

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Type/Additional information** |
| Number | מספר השורה במסמך. חובה לשלוח את מספר השורה משום שהוא משמש לזיהוי השורה הרלוונטית במסמך שכבר קיים בכספית | int32Required |
| ProductId | המפתח של הפריט. משמש בכספית+מלאי לניהול המלאי ולדוחות מכירה של פריטים. בכספית+ מקשר בין שורות המסמך לטבלת פריטים בגרסאות אחרות אפשר לשלוח NULL. | string0 - 50 |
| ProductName | שם הפריט | string0 - 100 |
| Details | פרטי הפריט או השירות שנמכר | string0 - 1500 |
| UnitPrice | מחיר יחידה | decimalRequired |
| Qty | הכמות שנמכרה | decimalRequired |
| CurrencySymbol | סמל המטבע - הסמלים האפשריים הם הסמלים שטבלת המטבעות של כספית. ראה במסך הראשי, מסך מטבעות. לדוגמה: ₪, $, €, A$ | string0 - 4 |
| Rate | שער החליפין. עבור ש"ח השער הוא 1.0 | decimal |
| Rebate | הנחת שורה בש"ח (או בדולרים וכו'). לא באחוזים. | decimal |
| ExtendedPrice | מחיר כולל של השורה לאחר הנחה ולפני מע"מ | decimal |
| ChargeVAT | האם לגבות מע"מ על השורה | boolean |
| VATRate | אחוז המעמ בשורה (17 מסמל 17%) לא בשימוש ביצירה/עדכון (הערך נלקח מהחשבונית). מאוכלס בשליפה | decimal |
| VAT | המעמ (בש"ח) שיש לגבות על השורה לא בשימוש ביצירה/עדכון. מאוכלס בשליפה | decimal |

## The CaspitReceiptLine Object (see <https://app.caspitweb.biz/ApiHelp/ResourceModel?modelName=CaspitReceiptLine>)

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Type/Additional information** |
| Number | מספר השורה במסמך. חובה לשלוח את מספר השורה משום שהוא משמש לזיהוי השורה הרלוונטית במסמך שכבר קיים בכספית | int32Required |
| PaymentTypeId | שיק, מזומן וכו' See the enumeration ApiGlobals.PaymentTypes in the SDK | int32Required |
| AccountNumber | מספר החשבון או כרטיס האשראי | string0 - 50 |
| BranchOrValidTo | מספר הסניף או תוקף הכרטיס | string0 - 50 |
| BankId | הבנק (ה-id נלקח מרשימת הבנקים של בנק ישראל( | int32 |
| CheckNumber | מספר השיק או השובר | string0 - 50 |
| PaymentDate | תאריך השיק או תאריך הזיכוי של כרטיס האשראי | date |
| Payment | סכום התשלום הנומינלי (כלומר, התשלום במטבע של הקבלה) אם לדוגמה התשלום הוא 400 דולר, הערך כאן יהיה 400 והמטבע של הקבלה יהיה $ .סכום התשלום בש"ח מוצג בשדה TotalPaymentNIS ברמת הקבלה. | decimal |

## TrxTypeId and TrxCodeNumber

The TrxTypeId is the type of the document (e.g. חשבונית, קבלה, חש/זיכוי). The TrxCodeNumber is the סיווג of the document (e.g. 3-מכירות/הכנסות or 2-קבלות). Here is table with typical values that are valid 99% of the time:

|  |  |  |
| --- | --- | --- |
| שם המסמך | TrxTypeId | TrxCodeNum |
| חשבונית מס | 1 | 3 |
| חש/קבלה | 7 | 3 |
| קבלה | 2 | 2 |
| חש/זיכוי | 9 | 3 |
| תעודת החזר | 13 | 2 |
| תע/משלוח | 15 | 3 |
| קבלה לתרומה | 19 | 2 |
| חש/עסקה | 6 | 3 |

## How to send or print a document

You cannot print a document via the API since the printing process requires a Web browser. You can only send the document via Email. Caspit will generate a PDF based on the document's details, sign it, and will send the signed document in the mail.



In order to send the document send (a no-standard Verb) EMAIL:

EMAIL https://localhost:50003/api/v1/documents/?token=ae17acfc-76f9-44c6-b9d9-07e11edb35c1&DocumentNumber=98/200002&trxTypeId=7 HTTP/1.1

Content-Type: application/json; charset=utf-8;

Content-Disposition:

Accept: application/xml

{"Subject":"מסמך לבדיקה","Body":"שלום ארי:\r\n\r\nהמסמך מוכן.\r\nנא אשר למשלוח.\r\n\r\nיעקב\r\n","To":"tcc\_12@walla.com","CC":null,"BCC":null}

## How to fetch the PDF file

You can fetch the PDF file that was generated by Caspit via the following call:

GET api/v1/Pdf/{id}?token={token}

The id is the id that you sent to Caspit when you created the document.

The response is a PDF document. The media type is media type application/pdf

# Products (מוצרים/פריטים)

Caspit has the following versions. Not all them support products. If your using a version that does not support products you will be able to send a product to Caspit but the user will not be able to access it via the GUI.

|  |  |
| --- | --- |
| Version | Support for products |
| כספית | No support. There is not product field in the invoice, and there is no product list screen. |
| כספית+ | פריטים = the user can not open the כרטיס פריט. Only one price for the product (=אין מחירונים). |
| כספית+מלאי | Products inc. כרטיס פריט, מחירונים etc. |

Here is a list of the product's fields:

|  |  |
| --- | --- |
| Field Name | Description |

# How to Get a Product

You can get a Product (מוצר/פריט) via its id or via its catalog number (מק"ט):

# Operation Log

It is sometimes very desirable to be able to see what happened to your call inside Caspit. In order to help you to debug the integration you can get the log that Caspit keeps.

You can only fetch the log records that you (= the user that you are using for the API calls) generated.

# How to GET the log

Call GET [https://app.caspitweb.biz/api/ver1/Log?token=xxxx-xxxx-xxx-xxxx&startDate=2014-10-31&endDate=2014-11-25](http://app.caspitweb.biz/api/ver1/Log?token=xxxx-xxxx-xxx-xxxx&startDate=2014-10-31&endDate=2014-11-25).

If you do not specify the start or the end date Caspit will return all the records available.

<CaspitResponse xmlns:i="https://www.w3.org/2001/XMLSchema-instance" xmlns="https://www.caspit.biz/2013/03"><CurrentPage>0</CurrentPage><NextPageUrl>https://localhost:50003/api/v1/Log?page=1&amp;token=05c347e6-652d-44f1-87a2-8059d2c56da2</NextPageUrl><PrevPageUrl></PrevPageUrl><Results><CaspitBase i:type="CaspitLog"><Content>המשתמש נכנס למערכת</Content><DateCreated>2014-10-10T17:23:52.49</DateCreated><DateUpdated>2014-10-10T17:23:52.48</DateUpdated><IP i:nil="true" /><LogId>8669</LogId><LogType>5</LogType><ObjectId>860</ObjectId><ObjectType>70</ObjectType><Source>2</Source><UserName>demo</UserName></CaspitBase> …

# The 'Caspit Web' API SDK

As an API developer you can get from us an SDK that includes:

* "CaspitWeb.Api" - A .Net project (VS 2013) that includes the constants and the classes that you need to send to the API. The project can be added to your own solution, or you can compile into a DLL and reference the DLL in your solution.
* ApiTester - A .Net project (VS 2013) that includes a Windows Forms application that demonstrates how to call the API. The ApiTester uses the CaspitWeb.Api DLL.



The Caspit Web API Tester

## Testing

You can perform testing on the demo account in Caspit Web:

User name: demo, password: demodemo, osekmorshe:  123456789

# FAQ

Q: I get a 405 result when I call the API

A: Are you calling Caspit using https or https? You should use http**s**.

Q: I get a read timeout when calling the API.

A: Try to increase the timeout of the calling mechanism. A reasonable timeout is 10 seconds.

Q: I want to send a document by mail.

A: Use the "EMAIL Document" call. See the online documentation:
<https://app.caspitweb.biz/ApiHelp/Api/EMAIL-api-v1-Documents-id_token_documentNumber_trxTypeId>