

## ExpenseWatch.com API Specification V1

### Overview

The Expensewatch api is implemented in a RESTful manner, depending on http verbs and parameters to determine the action performed. Efforts have been made to develop in as standard a way possible, and to be platform agnostic.

### Authentication

Authentication is session based. A request is made to our authentication service, and either an error is returned, or a session token that should be used with further communication with our API. Session tokens expire after 20 minutes of inactivity.

Verb	URL	Parameters
PUT	https://ssl.expensewatch.com/api/v1/authenticate	usr      User name pwd      Password

Example	
PUT	https://ssl.expensewatch.com/api/v1/authenticate/?usr=bob@abc.com&pwd=itsasecret

Responses	
200	4460891_64AE9B164F48190C10063F717CE911544AFEF103
403	Authentication Failure. Please check your credentials and try again.

After you've acquired a valid session token, you should include that on all subsequent calls. You can include the token in the query string or add it to the request header.

To include it in the query string, just add the "t" parameter with the token as its value.

Example	
GET	https://ssl.expensewatch.com/api/v1/invoice/export/333?t={session token}

To add it to the header, add the parameter "exwtoken" to the header, and set its value to the session token.

If the token is invalid or missing, the API will return with a status code of 403, "Authorization Token Missing".

### Content Type

The api will return either xml or json, depending on the requested content type. Please specify "text/xml" or "text/json" in your request. XML is currently the default return type, but that is subject to change, so please specify your desired return type.

## Status Codes

The Expensewatch will return several different status codes depending on what happened during the request.

Code	
200	OK
400	Bad request. Either parameters were expected that weren't provided, or the server was unable to understand the request
403	Unauthorized request. Either the session token is missing or invalid, or the username and password failed to authenticate.
404	Resource not found. This could mean that the url is invalid, or the resource you requested (such as <code>api/v1/invoice/55555</code> ) does not exist.
500	An internal server error.

## Invoices

Invoices are limited to retrieving and updating.

### Retrieving Invoices

#### Invoice.externaldata

The invoice object has a field named "externaldata" that functions as a hash for API users to store data necessary to an integration. For example, an external accounting system ID could be stored there. Invoices can be retrieved by supplying the value of the name/value pair you have stored in the externaldata hash, and specifying the key in the querystring..

Verb	URL	Parameters
GET	<code>https://ssl.expensewatch.com/api/v1/invoice/{invoiceid}</code>	{invoiceid} = Expensewatch ID

Verb	URL	Parameters				
GET	<code>https://ssl.expensewatch.com/api/v1/invoice/{invoiceid}</code>	<table> <tr> <td>key</td> <td>Name of the key in the ExternalData field that you want to search.</td> </tr> <tr> <td>{invoiceid}</td> <td>value associated with supplied key in the hash</td> </tr> </table>	key	Name of the key in the ExternalData field that you want to search.	{invoiceid}	value associated with supplied key in the hash
key	Name of the key in the ExternalData field that you want to search.					
{invoiceid}	value associated with supplied key in the hash					

Verb	URL	Parameters (optional)									
GET	<code>https://ssl.expensewatch.com/api/v1/invoice/</code>	<table> <tr> <td>exported</td> <td>String</td> <td>Y,N, or P (yes, no, pending).</td> </tr> <tr> <td>paid</td> <td>String</td> <td>Y,N, or P (yes, no, pending).</td> </tr> <tr> <td>reviewed</td> <td>String</td> <td>Y,X, or P (approved, disapproved,</td> </tr> </table>	exported	String	Y,N, or P (yes, no, pending).	paid	String	Y,N, or P (yes, no, pending).	reviewed	String	Y,X, or P (approved, disapproved,
exported	String	Y,N, or P (yes, no, pending).									
paid	String	Y,N, or P (yes, no, pending).									
reviewed	String	Y,X, or P (approved, disapproved,									

						pending).
		Invoicenumber	String			
		Erpid	String			
		vendorname	String			
		invoicedate	Date			
		postingdate	Date			
		modifieddate	Date			
		istereport	Boolean	1 or 0		

A GET request will return one or more invoices. An example of the xml returned follows.

```
<EW>
  <Invoices>
    <InvoiceID>1469401</InvoiceID>
    <InvoiceNumber>1407-JFitzpatrick</InvoiceNumber>
    <InvoiceDate>2010-03-02T08:47:15.427-05:00</InvoiceDate>
    <PostingDate>2010-03-02T08:47:15.427-05:00</PostingDate>
    <ExportStatus>Y</ExportStatus>
    <Reviewed>Y</Reviewed>
    <Credit>>false</Credit>
    <IsTEReport>>true</IsTEReport>
    <CheckNumber></CheckNumber>
    <VendorID>46306</VendorID>
    <VendorERPID>zglozman</VendorERPID>
    <VendorName>Fitzpatrick, James</VendorName>
  </Invoices>
</EW>
```

### Updating Invoices

Currently, only the paid flag, the exported flag, and the ExternalData hash can be updated.

Verb	URL	Parameters (optional)		
PUT	https://ssl.expensewatch.com/api/v1/invoice/{invoiceid}	exported	String	Y,N, or P (yes, no, pending).
		paid	String	Y,N, or P (yes, no, pending).
		key	String	Name of the key in the ExternalData field that you want to search.
		External data	Hash	See "updating externaldata"

If you are attempting to update an invoice identified by a name/value pair in the externaldata hash, then a 400 status code will be returned if the name/value pair specified does not uniquely identify an invoice within your company.

## Invoice Exports

New Invoice exports can be created through the API, and previous exports can be requested as well. The requested content type is ignored for the export, as the content is set in the request parameters, and will always be served as text/html.

### Get an Existing Export

You can request a previously export using the following:

Verb	URL	Parameters (optional)		
GET	https://ssl.expensewatch.com/api/v1/invoice/export/{exportid}	{exportid}	String	EW exportid
		format	String	"xml", "csv", "txt", or "txtold". Defaults to "xml"
		ExportFormat	Int	Detail = 1 Summary = 2 LinearSummary = 3
		ExportType	Int	All = 1 Invoices = 2 TEPayroll = 3 PurchaseOrders = 4
		IncludeHeader	Bool	"true" or "false"

### Create a new Export

To create a new export, a POST action is used with the same parameters, minus the exportid.

Verb	URL	Parameters (optional)		
POST	https://ssl.expensewatch.com/api/v1/invoice/export	format	String	"xml", "csv", "txt", or "txtold". Defaults to "xml"
		ExportFormat	Int	Detail = 1 Summary = 2 LinearSummary = 3
		ExportType	Int	All = 1 Invoices = 2 TEPayroll = 3 PurchaseOrders = 4
		IncludeHeader	Bool	"true" or "false"

## Vendors

Vendors can be retrieved, created, and updated.

### Vendor.externaldata

The vendor object has a field named “externaldata” that functions as a hash for API users to store data necessary to an integration. For example, an external accounting system ID could be stored there. A vendor can be retrieved by supplying the value of the name/value pair you have stored in the externaldata hash, and specifying the key in the querystring.

### Get an Existing Vendor

In the first example below, the vendor is retrieved by the ExpensewatchID. The second request is how you would retrieve a vendor using a key stored in the externaldata hash.

Verb	URL	Parameters
GET	https://ssl.expensewatch.com/api/v1/vendor/{vendorid}	{vendorid} = Expensewatch ID

Verb	URL	Parameters				
GET	https://ssl.expensewatch.com/api/v1/vendor/{vendorid}	<table> <tr> <td>Key</td> <td>Name of the key in the ExternalData field that you want to search.</td> </tr> <tr> <td>{vendorid}</td> <td>value associated with supplied key in the hash</td> </tr> </table>	Key	Name of the key in the ExternalData field that you want to search.	{vendorid}	value associated with supplied key in the hash
Key	Name of the key in the ExternalData field that you want to search.					
{vendorid}	value associated with supplied key in the hash					

A GET response will contain one or more vendors. An XML response would be structured as follows:

```
<EW>
  <Vendors>
    <VendorID>46320</VendorID>
    <externaldata>
      <SerializableHash>
        <hash>
          <item key="intuitid" value="5555" />
          <item key="otherid" value="99999ddddxxxxx" />
        </hash>
      </SerializableHash>
    </externaldata>
    <name>suckit</name>
    <address>42 Wallaby Way</address>
    <address2></address2>
    <city>Sydney</city>
    <state>Au </state>
    <zip>11111</zip>
    <phone>999-999-9999</phone>
    <fax>610-397-0539</fax>
    <email>nmcguire@expensewatch.com; jfitzpatrick@expensewatch.com</email>
    <contact></contact>
    <vendorerpid>abc</vendorerpid>
    <accountnum></accountnum>
    <defaultterm>30</defaultterm>
    <requirereceiving>>false</requirereceiving>
    <requiresingleshiptoforpo>>false</requiresingleshiptoforpo>
    <defaulttermdays>0</defaulttermdays>
```

```

<isccissuer>N</isccissuer>
<currencyid>USD</currencyid>
<active>true</active>
</Vendors>
</EW>

```

### Update an Existing Vendor

If you are attempting to update an invoice identified by a name/value pair in the externaldata hash, then a 400 status code will be returned if the name/value pair specified does not uniquely identify an invoice within your company.

Verb	URL	Parameters (optional)	
PUT	https://ssl.expensewatch.com/api/v1/vendor/{vendorid}	Name	String
		Address	String
		Address2	String
		City	String
		State	String
		Zip	String
		Phone	String
		Fax	String
		Email	String
		Contact	String
		VendorerpId	String
		Accountnum	String
		Defaultterm	String
		Requirereceiving	Bool 1 or 0
		Requiresingleshiptoforpo	Bool 1 or 0
		Defaultermdays	Int
		Isccissuer	String
		Allowpriceediting	Bool 1 or 0
		Currencyid	String
		Active	Bool 1 or 0
		Externaldata	Hash See "Updating ExternalData"
		Key {vendorid}	String Externaldata key
			String EW or hash identifier

### Create a New Vendor

Verb	URL	Parameters	
POST	https://ssl.expensewatch.com/api/v1/vendor/	<b>Required</b>	
		Name	String
		Address	String
		City	String

		State	String	
		Zip	String	
		VendorerpId	string	
		<b>Optional</b>		
		Phone	String	
		Fax	String	
		Email	String	
		Contact	String	
		Address2	String	
		Accountnum	String	
		Defaultterm	String	
		Requirereceiving	Bool	1 or 0
		Requiresingleshi	Bool	1 or 0
		ptoforpo		
		Defaultermdays	Int	
		Isscissuer	String	
		Allowpriceediting	Bool	1 or 0
		Currencyid	String	
		Active	Bool	1 or 0
		Externaldata	Hash	See "Updating ExternalData"

### Updating ExternalData

When updating the externaldata hash, you can supply one or more name value pairs, and these will either be inserted into the hash, or update values if the key already exists.

The pattern of parameters to provide is as follows:

```
ed=keyname
ed_value=keyvalue
ed_1=keyname1
ed_1_value=keyvalue1
ed_2=keyname2
ed_2_value=keyvalue2
```

So, any number of key/value pairs can be supplied by using "ed\_X" as the parameter to supply the key, and "ed\_X\_value" as the parameter to supply the value. The first key/value pair can simply use "ed" and "ed\_value" for simplicity.

## Attachment Exports

Attachments (files associated with transactions) can be listed and downloaded through the API. Currently, the API supports attachments – usually images or PDF files – associated with invoices. The first step to downloading the attachments for an invoice is to request a list of the attachments for a given InvoiceID. Note: an authentication token must be supplied for all attachment requests.

### Get a List of the Attachments for an Invoice

You can request information about the attachments for an invoice using the following:

Verb	URL	Parameters (required)			
GET	<a href="https://ssl.expensewatch.com/api/v1/attachment">https://ssl.expensewatch.com/api/v1/attachment</a>	<table border="0"> <tr> <td>invoiceid</td> <td>Int</td> <td>The InvoiceID of the invoice.</td> </tr> </table>	invoiceid	Int	The InvoiceID of the invoice.
invoiceid	Int	The InvoiceID of the invoice.			

An example of the request for an invoice with an InvoiceID of 30000 would be:

<https://ssl.expensewatch.com/api/v1/attachment?invoiceid=30000>

An XML response would be structured as follows:

```
<EW>
  <Attachment>
    <AttachmentID>17083254</AttachmentID>
    <OriginalFileName>Capture.PNG</OriginalFileName>
    <MimeType>image/png</MimeType>
    <DateCreated>2015-05-15T11:46:04.723-04:00</DateCreated>
  </Attachment>
  <Attachment>
    <AttachmentID>17083255</AttachmentID>
    <OriginalFileName>show-printable.pdf</OriginalFileName>
    <MimeType>application/pdf</MimeType>
    <DateCreated>2015-05-15T11:46:54.603-04:00</DateCreated>
  </Attachment>
</EW>
```

In the example request above, there are two attachments: a PNG image and a PDF file. The name of the file (as uploaded), the MIME type, and the date that the file was uploaded is also included. Invoices that do not have attachments will not return any data.

### Download an Attachment

Once the AttachmentID(s) have been obtained from the previous call, the actual file data can be downloaded using the following:

Verb	URL	Parameters (required)			
GET	<a href="https://ssl.expensewatch.com/api/v1/attachment/{attachmentid}">https://ssl.expensewatch.com/api/v1/attachment/{attachmentid}</a>	<table border="0"> <tr> <td>attachmentid</td> <td>Int</td> <td>The AttachmentID for the attachment to download.</td> </tr> </table>	attachmentid	Int	The AttachmentID for the attachment to download.
attachmentid	Int	The AttachmentID for the attachment to download.			

An example of the request for an attachment with an AttachmentID of 17083255 would be:  
<https://ssl.expensewatch.com/api/v1/attachment/17083255>

The response from the server will include a Content-Type header set to the MIME type of the attachment, a Content-Disposition header set with the original filename of the attachment, and the binary stream of the file data.