NolaPro API Documentation

Noguska LLC

Doc Version 1.8636

Table of Contents

Introduction

The NolaPro API has been created to allow 3rd party applications to more easily put data into and get data out of NolaPro and to ensure that data coming in adheres to the core business rules of NolaPro. The API is based on a REST-style architecture and uses HTTP/HTTPS to manage communication between systems.

To access a resource in NolaPro you will need to construct an appropriate URL and use GET or POST to send data/parameters. The response will come back in XML format.

Example: Reduce the inventory count for NolaPro item 33454

```
URL = https://api.nolapro.com/rest/item/item_decrease
POST Fields:
data[itemid]=33454
data[locationid]=2
data[qty]=5
data[notes]='Waste'
auth[companyid]=1
auth[userid]=3
auth[url]=/rest/item/item_decrease
auth[time]=1197917516
auth[hash]=BKQ065Xg9H92vcdGMxW3ohKGUI8=
```

The URL format is http://domain.com/rest/controller/action where controller is the NolaPro entity or database table name and action is the process or function that you want to perform on that entity. In the example above the entity is item and we want to perform the action item_decrease. You will get a response back that says if there was an error during processing or if the request was successful.

We are continually building additional controllers and actions for new processing or data retrieval that users would like to do from 3rd-party applications.

Any programming language that can send post information via HTTP and can parse XML can be used to interact with the NolaPro API. We have constructed a PHP wrapper class that provides methods to simplify talking to the API.

This document provides details on the current capabilities of the API and also goes over how to construct requests in order to interact properly with the system.

Setting Up the API

The NolaPro API is written in PHP and requires PHP 5 running on Apache in order to function properly. We do not plan to test the application on any other web server programs (IIS, etc.).

Database Connection

To link the API to your NolaPro database open up in a text editor the file ***api/app/config/database.php***. Adjust the values listed below to point to your database. Make these changes for both the $default and $test sections.

```
'host' => 'localhost',
'login' => 'root',
'password' => '',
'database' => 'nolapro',
```

Server Configuration

Turn on the short open tags option in your PHP configuration. The line in the php.ini file should read:

short_open_tag = On

Restart Apache after making any php.ini configuration changes.

You can access the API as a subdirectory off of your NolaPro site (https://mysite.com/nolapro/api), or you can set up a dedicated virtual host entry for it. We strongly encourage adding SSL and only creating a virtual host entry for port 443 (and not one for unencrypted port 80) if you make a dedicated API virtual host. The document root of the API should be ***api/app/webroot***. Below is an example virtual host entry if you choose to use this method. If you are fine accessing the API off of your NolaPro directory, then you do not need to worry about adding the below section to your Apache configuration.

```
NameVirtualHost *:443
<VirtualHost *:443>
    DocumentRoot /var/www/html/api/app/webroot
    ServerName api.mydomain.com
    ErrorLog /var/log/apache2/api.mydomain.com-error_log
    SSLEngine on
    SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
    SSLCertificateFile /etc/apache2/ssl.crt/_.mydomain.com.crt
    SSLCertificateKeyFile /etc/apache2/ssl.key/mydomain.key
    SSLCertificateChainFile /etc/apache2/ssl.crt/sf_issuing.crt
    <Files ~ "\.(cgi|shtml|phtml|php3?)$">
        SSLOptions +StdEnvVars
    </Files>
    SetEnvIf User-Agent ".*MSIE.*" \
             nokeepalive ssl-unclean-shutdown \
             downgrade-1.0 force-response-1.0
    CustomLog /var/log/apache2/api.mydomain.com-access_log common
    <Directory />
            AllowOverride all
            Options -Indexes FollowSymLinks
    </Directory>
    <Location "/">
            Order Deny,Allow
            Deny from all
            Allow from 192.168.1.25
    </Location>
</VirtualHost>
```

In the example above modify the <Location> section as needed to allow access only from certain trusted IPs. The API comes with its own authentication mechanism to prevent unauthorized access, but this additional measure offers even more protection.

NolaPro Setup

In order to use the NolaPro API you will need to activate the API add-on by first purchasing an activation code and then applying the code on the ***Admin -> Setup -> Add-ons & Plug-ins*** page.

After turning the API feature on you will need to generate an API key for one of your NolaPro users. All data entered via the API will be tagged with this user's ID. Go to ***Admin -> Setup -> User Add/Update***. Choose the option *Generate New API Key* for the *API Key Management* select box. Click save. You should now see an API key for this user on the page next to *API Key.*

The next section will go into detail on how to create requests to the API.

Constructing a Proper API Request

Authentication

Every NolaPro request requires that 5 POST or GET fields be set. These fields are used to authenticate the requester and validate that the posted data is from the authenticated user and has not been modified.

1) **auth[url]** – This field should contain the API URL being requested minus the domain name. So for a request to https://apidemo.nolapro.com/rest/gltransvoucher/save the value of auth[url] would be /rest/gltransvoucher/save. For a request to https://demo.nolapro.com/api/rest/gltransvoucher/save auth[url] would also be /rest/gltransvoucher/save
2) **auth[companyid]** – This should be the ID from the gencompany table for the company that the action will be performed on. If you only have one company in NolaPro this will likely be ID 1.
3) **auth[userid]** – This is the ID of the NolaPro user that API requests will be made under. You can get this ID by going to Admin -> Setup -> User Add/Update and bringing up your API user. Look for the label *API User ID*. It can also be found in the genuser table in the database.
4) **auth[time]** – This should be the current time measured in the number of seconds since the Unix Epoch (January 1 1970 00:00:00 GMT) (aka a Unix timestamp).
5) **auth[hash]** – The hash value helps ensure that the data being sent is authentic and from a source with access to the NolaPro user's API key. To get an API key for your user account go to Admin -> Setup -> User Add/Update. Look for the label *API Key Management*. Choose *Generate New API Key* and then save. After saving look for the label *API Key*. This is the secret code you'll need to construct a hash.
   The hash should be built as follows:

   Data String = Concatenation of all post fields except auth[hash] in the form field1value1field2value2...(name=Jack&id=5&color=red would be nameJackid5colorred)

   Hash = base64_encode(HMAC-SHA1 Signature(Data String, API Key))

   You can find an example of calculating the HMAC-SHA1 signature with PHP in the npapi.php file under the hmacsha1() method.

Data

Data that you'd like to send to the API needs to use an array-style naming convention similar to the authentication information (auth[]). All values need to be sent in fields using the name data[]. Information to be saved for a specific model or table name should be in a multi-dimensional format like the following:

```
data[gltransvoucher][voucher]=TSTVCHR
data[gltransvoucher][description]=A Test Voucher from the API Test Page
data[gltransvoucher][wherefrom]=2
```

The field names to pass in are the same as the field names in the database table related to the model. You may want to use MySQL Query Browser or phpMyAdmin as a reference tool to get the field names you'll need when saving data.

You do not need to ever send in data[] fields for companyid, entrydate, createdate, entryuserid or createuserid as these will be filled in for you automatically based on values in auth[].

Info that needs to be sent, but that does not relate to a specific model can be sent without a model prefix simply as:

data[page]=1
data[limit]=25
data[order]=entrydate
data[conditions]=(id > 10) and entrydate < '2008-03-05'
data[fields][1]=entrydate
data[fields][2]=id
data[fields][3]=description
data[fields][4]=voucher

In some cases in order to properly perform an action via the API one or more additional related model elements may need to be submitted at the same time. An example of this is when saving a general ledger voucher (gltransvoucher). The voucher contains generic information about the transaction (description, when it was entered, when it was posted to the GL), but needs related gltransaction records to be submitted at the same time. The gltransaction records contain information about the amount of money going to each general ledger account in the voucher. The data fields for this type of submission would look like:

```
data[gltransvoucher][voucher]=TSTVCHR
data[gltransvoucher][description]=A Test Voucher from the API Test Page
data[gltransvoucher][wherefrom]=2
data[gltransaction][0][glaccountid]=1937
data[gltransaction][0][amount]=78.35
data[gltransaction][1][glaccountid]=1985
data[gltransaction][1][amount]=-78.35
```

The npapi PHP class contains some helper methods for creating these field names a little more easily.

Using the npapi class allows you to do something like the following to construct the field names automatically:

```
$api = new npapi();
$api->set_model('gltransvoucher');
$data = array(    "voucher" => "TSTVCHR",
                  "description" => "A Test Voucher",
                  "wherefrom" => "2");
$api->set_data($data);

$api->set_model('gltransaction');
$data = array(
            array('glaccountid' => 1937,  'amount' => 78.35),
            array('glaccountid' => 1985, 'amount' => -1*78.35)
            );
$api->set_data($data);

$api->action = "gltransvoucher/save";
$api->call_api();...
```

Sending the Request to the API

Most high-level programming languages have ways of sending an HTTP request to a URL and retrieving the contents. In PHP this can easily be accomplished using the curl library. You can see an example of this type of use in npapi.php under the call_api() method. The response back from the API will be an XML document.

Reading the Response

A very useful tool for monitoring the inputs and outputs of the API is NolaPro's API Log. You can access this via the file adminapilog.php in NolaPro or by going to Admin -> Setup -> API Log Viewer. Each transaction that goes through the API will be recorded here (failures and successes). You can click the View link to see the returned XML document and the input parameters. This can be very helpful when debugging. API logging is turned off by default. You can turn it on by going to Admin -> Setup -> System Settings and changing the NolaPro API Logging box to Yes.

Each response contains an **<rsp>** tag as the root element. **<rsp>** contains an attribute called ***stat*** that is set to either *ok* or *fail*. If ***stat*** is set to *fail* then there will be a child element called **<err>** with attributes of ***code*** and ***msg*** that contain information on the reason for the failure.

<u>Example of a failure return message</u>

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="fail">
        <err code="90010" msg="Invalid parent record ID. A record does not exist
for the given ID: Table = glaccount, Field = id, Value = 193766, companyid = 1 (See
[gltransaction][glaccountid] in data field)" />
</rsp>
```

<u>Example of a success return message</u>

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
        <gltransvoucher id="582" href="/rest/gltransvoucher/view/582">
        </gltransvoucher>
</rsp>
```

PHP 5 offers a very useful tool for parsing XML documents called SimpleXML. You can see an example of parsing the returned document with SimpleXML in the npapi call_api() method.

API Controllers & Actions

To facilitate communication with the NolaPro Business Management System we have structured the API into a series of controllers, which in most cases represent a database table in NolaPro, and actions that can be performed on the controllers. For example, to insert data into the gltransvoucher table (used to store general ledger voucher details) you'll need to call the save action on the gltransvoucher controller. The save action performs several data validation and consistency checks and then adds the info to the database. A success or fail message is sent back in XML format.

All controllers have an action called ***search*** that you can perform to locate records. The ***search*** action has a quick URL format you can use as well as a post version which allows further refinement.

The ***search*** URL format is as follows:

controller/search/field/value/operator

Valid operators are: =,<,<=,>,>=,<>,like (the default operator is = so this can be left off requests using this)

Example:
https://apidemo.nolapro.com/rest/gltransvoucher/search/post2date/2008-01-01
This will return all general ledger vouchers posted to the date Jan 1, 2008.

For the ***search*** POST format you can send additional parameters to narrow the query.

data[recursive] = Whether to return related records to the model being searched. Set to 0 to perform no recursion. (ex. searching gltransvoucher with recursive set to 1 will return gltransaction records associated to gltransvoucher via gltransaction.voucherid)
data[page] = Represents the page number for results if you are using a limit (ex. get the 3rd page of results where 25 results are listed per page; data[page] would be 3 and data[limit] would be 25)
data[limit] = Reduce the number of records returned to this number (ex. to show only the first 10 results data[limit] would be 10; the default value is 5 if this parameter isn't sent)
data[order] = The order in which to sort the results. This should be a list of field names comma separated in the preferred sorting order. (ex. add desc to the field name to do a reverse sort as in data[order] = last_name, first_name, orderid desc)
data[conditions] = Allows you to add conditions on fields to further filter the results. The conditions can be built similarly to an SQL where clause (ex. data[conditions] = (id > 10) and entrydate < '2008-03-05')
data[fields][n] = Where *n* is a positive integer this selector lets you choose which fields you would like to have returned. (ex.
data[fields][1]=entrydate, data[fields][2]=id, data[fields][3]=description, data[fields][4]=voucher)

The URL to perform a POST search would look like:
https://apidemo.nolapro.com/rest/gltransvoucher/search

All controllers also support the ***view*** action which allows you to easily retrieve one record by its ID. Simply construct a URL in the form controller/view/id (ex. gltransvoucher/view/5678)

Most controllers also support a ***save*** action which allows you to insert or update a record for a model. Construct the data[] field names by using the model name and table fields in the form data[model][field1]=value1 data[model][field2]=value2. If the field data[model][id] is not submitted, then an insert will be attempted. If data[model][id] is supplied, an update will run in most cases. As mentioned in the *Constructing a Proper API Request* section you can use the npapi class to simplify setting data fields for submission to the API.

Current Available Controllers & Actions

We are continually adding new controllers and actions to the API. Our current list is below.

The following fields, though listed as required, do not need to be sent when adding new data as they are filled in automatically for you:

entrydate
companyid/gencompanyid
entryuserid

All *date* fields require the format *yyyy-mm-dd*.

All *datetime* fields require the format *yyyy-mm-dd hh:mm:ss*.

accounttype
Controller Actions

- *search*
- *view*
- *save*

Field Info for accounttype				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	description	char(30)	Yes	

arcompany
Controller Actions

- *search*
- *view*
- *save*

Field Info for arcompany				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	imageurl	varchar(150)		
3	cash	int		
4	checking	int		
5	interest	int		
6	discount	int		
7	cost	int		
8	inventory	int		
9	shipliability	int		
10	receivables	int		

11	nextinvoicenum	int
12	servicecharge	decimal(8,2)
13	interestrate	decimal(8,4)
14	baddebt	int
15	default_sellunitid	int
16	default_priceunitid	int
17	default_sell_per_price	int
18	default_invoice_message	text
19	invoiceformat	int
20	labor_sales_glacctid	int
21	hrunitid	int
22	is_employee_labor_taxable	int
23	depositglacctid	int
24	refund	int
25	ccprocessorid	int
26	docfile	text
27	default_salescategory_cashreceipts	int
28	default_salescategory_invoice	int
29	default_salescategory_fulfillment	int
30	default_salescategory_service	int
31	serviceorder_summarize	int
32	tax_on_shipto	int
33	tax_on_freight	int
34	order_base	int
35	b2b_orders	int
36	b2b_service	int
37	tax_how	int
38	donotapply	int
39	restockglid	int
40	restockunitid	int
41	salesterritories	int
42	default_statement	int
43	b2b_commission	int
44	default_discountmsg	text
45	b2b_print_orders	int
46	b2b_pricelist_orders	int
47	default_invoicetermsid	int
48	b2b_allow_signup	int
49	default_customergroupid	int
50	default_b2b_admin_email	varchar(255)
51	invoicecalc	int

52	current_or_order	int
53	default_checkacctid	int
54	nobarcode_on_invoice	int
55	picklist_blockheader	int
56	picklist_blockbarcode	int
57	picklist_sortorder	int
58	pickliststyle	int
59	noboxes_on_invoice	int
60	comparison_levelid	int
61	statement_no_prior_balance	int
62	statement_no_interest_added	int
63	ardate	int
64	noteatbottom	int
65	internal_serviceorder	int
66	internal_fulfillmentorder	int
67	invoicenumber_override	int
68	display_profit_pos	int
69	display_weights_pos	int
70	pos_hide_stats	int
71	pos_hide_fulfillment	int
72	pos_hide_discounts	int
73	pos_default_payment_method	int
74	pos_default_order_type	int
75	checkforcost	int
76	bankdepositautoselect	int
77	show_salesdata	int
78	salesmanBlock	int
79	usecurdateforship	int
80	allowcoinvoice	int
81	hidegl_pos	int
82	showaddl_pos	int
83	revenuerec	text
84	deferredrevenue	int
85	mgmtfee_glaccountid	int
86	posblockchilditemsearch	int
87	latefees	int
88	lawncare	int
89	petdeposit	int
90	petrent	int
91	showpriorbalanceoninvoice	int
92	requireccbillingaddress	int

93	requirecccardcode	int
94	ccpaymentinstructions	text
95	surveyorder	int
96	surveyinvoice	int
97	nocostupdate	int
98	report_on_paid	int
99	maintenance	int
100	commissionmethod	varchar(20)
101	hideoptions	int
102	fulfillmentduedays	int
103	serviceduedays	int
104	hideinvpagenumber	int
105	appendendnotes	int
106	showinvsignature	int
107	defaulttovendorcost	int
108	formsize	int
109	show_discount_on_forms	int
110	default_order_message	text
111	purgeordersdays	int
112	purgeordershow	char(1)

arinvoice

Controller Actions

- *search*
- *view*
- *save*
- *autopay* - *autopay* applies a payment for a customer to any open invoices. An overpayment will be automatically created if the payment is greater than the amounts due.
  Possible Parameters (*required):
  *data[customerid]
  *data[amount] - Must be positive
  *data[type] - 1->Cash, 2->Check, 3->Credit Card, 4->Other
  *data[transactionid] - Check number or other identifier for this payment
  data[date] - Date of the payment; If not set, today's date will be used
  data[bankglaccountid] - GL account for debit side of transaction; If not set, default Cash (if type=1) or Checking (for all other types) GL account will be used
  data[arglaccountid] - GL account for credit side of transaction; If an AR account is set on the invoice, that will be used; If an account isn't passed in, and there isn't one on the invoice, the default receivables account will be used

Field Info for arinvoice				
	Field	**Type**	**Required**	**Relates To**
1	id	int		

2	invoicenumber	varchar(30)	Yes	
3	ponumber	varchar(30)		
4	wherefrom	int	Yes	wherefrom
5	orderid	int		
6	orderbycompanyid	int	Yes	customer
7	shiptocompanyid	int	Yes	shipto
8	status	int		
9	customerbillcode	varchar(20)		
10	shipcost	decimal(19,4)		
11	invoicetotal	decimal(19,4)		
12	invoicetermsid	int	Yes	invoiceterms
13	salesmanid	int		salesman
14	invoicedate	date	Yes	
15	duedate	date	Yes	
16	discountdate	date		
17	discountamount	decimal(19,4)		
18	accruedinterest	decimal(19,4)		
19	datelastinterestcalc	date		
20	gencompanyid	int	Yes	gencompany
21	cancel	int		
22	canceldate	datetime		
23	canceluserid	int		
24	entrydate	datetime	Yes	
25	entryuserid	int	Yes	
26	lastchangedate	timestamp		
27	lastchangeuserid	int		
28	pricelevelid	int	Yes	pricelevel
29	inventorylocationid	int	Yes	inventorylocation
30	payment_method	varchar(12)		
31	cc_type	varchar(20)		
32	cc_owner	varchar(64)		
33	cc_number	varchar(32)		
34	cc_expires	varchar(4)		
35	shipping_method	varchar(32)		
36	currency	varchar(10)		
37	cc_PNREF	varchar(12)		
38	cc_AUTHCODE	varchar(6)		
39	transaction_id	varchar(32)		
40	customers_id	int		
41	currency_value	decimal(14,6)		
42	servicerepid	int		

43	discountapplied	decimal(19,4)		
44	payplan	int		
45	payplan_period	int		
46	payplan_days	int		
47	payplan_next	date		
48	payplan_numtimes	int		
49	sales_categoriesid	int		sales_categories
50	coverage	int		
51	formnumber	varchar(50)		
52	shiptoattn	varchar(100)		
53	old_id	varchar(255)		
54	client_po	varchar(30)		
55	currate	decimal(22,10)		
56	arglaccountid	int	Yes	glaccount
57	leaseid	int		

arinvoicedetail

Controller Actions

- *search*
- *view*
- *save*

Field Info for arinvoicedetail

	Field	Type	Required	Relates To
1	id	int		
2	invoiceid	int	Yes	arinvoice
3	linenumber	int	Yes	
4	itemid	int		item
5	description	varchar(100)		
6	qty	int		
7	qtyunitnameid	int		unitname
8	glaccountid	int	Yes	glaccount
9	taxflag	int		
10	priceach	decimal(19,6)		
11	priceunitnameid	int		unitname
12	qtyunitperpriceunit	decimal(19,6)		
13	totalprice	decimal(19,4)		
14	entrydate	datetime	Yes	
15	entryuserid	int		
16	lastchangedate	timestamp		
17	lastchangeuserid	int		

18	costeach	decimal(19,6)		
19	costglaccountid	int		glaccount
20	main_arinvoicedetailid	int		
21	itemoptionid	int		
22	itemoptiongroupid	int		
23	taxflag2	int		
24	taxflag3	int		
25	taxflagid	int		
26	taxflag2id	int		
27	taxflag3id	int		
28	orig_priceach	decimal(12,5)		
29	discount_amount	decimal(12,5)		
30	discount_percent	decimal(12,5)		
31	old_id	varchar(255)		
32	pricelevelid	int	Yes	pricelevel
33	extended_price	decimal(19,5)		
34	price_override	int		
35	worktypeid	int		
36	worksubtypeid	int		
37	discountid	int		
38	discountpercent	decimal(7,2)		
39	returndetailid	int		
40	returnqty	int		
41	inventorylocationid	int		

arinvoicedetailcost

Controller Actions

- *search*
- *view*
- *save*

Field Info for arinvoicedetailcost				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	invoiceid	int	Yes	arinvoice
3	cost	decimal(19,4)		
4	costglaccountid	int	Yes	glaccount
5	entrydate	datetime		
6	entryuserid	int		
7	lastchangedate	timestamp		
8	lastchangeuserid	int		

9	itemid	int		item
10	locationid	int		inventorylocation
11	arordershipdetailid	int		
12	inventoryglacctid	int	Yes	glaccount

arinvoicedetailnotes

Controller Actions

- *search*
- *view*
- *save*

Field Info for arinvoicedetailnotes

	Field	Type	Required	Relates To
1	id	int		
2	invoiceid	int	Yes	arinvoice
3	linenumber	int	Yes	
4	note	text	Yes	
5	lastchangedate	timestamp		
6	lastchangeuserid	int		

arinvoicenotes

Controller Actions

- *search*
- *view*
- *save*

Field Info for arinvoicenotes

	Field	Type	Required	Relates To
1	invoiceid	int	Yes	arinvoice
2	note	text	Yes	
3	hide	int		
4	lastchangedate	timestamp		
5	lastchangeuserid	int		
6	linenumber	int		
7	source_noteid	int		
8	id	int		

arinvoicepaymentdetail

Controller Actions

- *search*

- *view*
- *save*

Field Info for arinvoicepaymentdetail				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	invoiceid	int	Yes	arinvoice
3	amount	decimal(19,4)		
4	voucherid	int	Yes	gltransvoucher
5	datereceived	date	Yes	
6	paymeth	int	Yes	
7	interest	int		
8	cancel	int		
9	canceldate	datetime		
10	cancel_userid	int		
11	checkid	varchar(100)		
12	bankdepositid	int		
13	bankstatement_enddate	date		
14	old_id	varchar(255)		
15	refund_paymeth	int		
16	cctransactionid	int		
17	currency	varchar(10)		
18	returned	int		
19	feeapbillid	int		
20	feeinvoiceid	int		
21	bankfee_amount	decimal(19,4)		
22	bankfee_glid	int		
23	rnum	varchar(30)		

arinvoicetaxdetail

Controller Actions

- *search*
- *view*
- *save*

Field Info for arinvoicetaxdetail				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	invoiceid	int	Yes	arinvoice
3	taxid	int	Yes	salestax
4	taxamount	decimal(19,4)		

5	taxcap	decimal(19,4)

arorder

Controller Actions

- *search*
- *view*
- *save*

Field Info for arorder

	Field	Type	Required	Relates To
1	id	int		
2	ordernumber	int		
3	onum	varchar(30)		
4	ponumber	varchar(30)		
5	orderbycompanyid	int	Yes	customer
6	shiptocompanyid	int	Yes	shipto
7	status	int		
8	customerbillcode	varchar(20)		
9	companyid	int	Yes	gencompany
10	pricelevelid	int	Yes	pricelevel
11	inventorylocationid	int	Yes	inventorylocation
12	cancel	int		
13	canceldate	datetime		
14	canceluserid	int		
15	entrydate	datetime	Yes	
16	entryuserid	int	Yes	
17	duedate	date	Yes	
18	lastchangedate	timestamp		
19	lastchangeuserid	int		
20	payment_method	varchar(12)		
21	cc_type	varchar(20)		
22	cc_owner	varchar(64)		
23	cc_number	varchar(32)		
24	cc_expires	varchar(7)		
25	shipping_method	varchar(32)		
26	orders_date_finished	timestamp		
27	currency	varchar(10)		
28	cc_PNREF	varchar(12)		
29	cc_AUTHCODE	varchar(6)		
30	transaction_id	varchar(32)		
31	customers_id	int		

32	shipping_cost	decimal(19,4)		
33	order_weight	decimal(12,2)		
34	currency_value	decimal(14,6)		
35	wherefrom	varchar(100)	Yes	wherefrom
36	confirmed_by	varchar(100)		
37	confirmed_date	varchar(100)		
38	confirmed	varchar(100)		
39	paymeth	int		
40	ordertype	int		
41	estimate	decimal(19,4)		
42	deposit	decimal(19,4)		
43	salesmanid	int		salesman
44	sales_categoriesid	int		sales_categories
45	statusid	int		
46	rma_arorder_returnedid	int		
47	rma_orig_arorderid	int		
48	carrierserviceid_preferred	int		carrierservice
49	finalship	int		
50	currate	decimal(22,10)		
51	arglaccountid	int		
52	defaultworktypeid	int		
53	defaultworksubtypeid	int		
54	jobid	int		
55	stageid	int		
56	edi	int		

arorderdetail

Controller Actions

- *search*
- *view*
- *save*

Field Info for arorderdetail				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	orderid	int	Yes	arorder
3	itemid	int		item
4	linenumber	int	Yes	
5	qtyorder	int	Yes	
6	qtyship	int		
7	qtybill	int		

8	glaccountid	int	Yes	glaccount
9	taxflag	int		
10	costeach	decimal(19,6)		
11	priceach	decimal(19,6)		
12	entrydate	datetime	Yes	
13	entryuserid	int	Yes	
14	lastchangedate	timestamp		
15	lastchangeuserid	int		
16	products_tax	decimal(7,4)		
17	description	varchar(255)	Yes	
18	saleunitperpriceunit	decimal(19,6)		
19	costglaccountid	int	Yes	glaccount
20	purchaseunitid	int	Yes	unitname
21	priceunitid	int	Yes	unitname
22	main_arorderdetailid	int		
23	itemoptiongroupid	int		
24	itemoptionid	int		
25	multioption	int		
26	optionindependent	int		
27	inventorylocationid	int	Yes	inventorylocation
28	taxflag2	int		
29	taxflag3	int		
30	taxflagid	int		
31	taxflag2id	int		
32	taxflag3id	int		
33	itemweight	decimal(12,6)		
34	qtyorderprice	int		
35	pricelevelid	int		
36	extended_price	decimal(19,5)		
37	price_override	int		
38	worktypeid	int		
39	worksubtypeid	int		

arordernotes

Controller Actions

- *search*
- *view*
- *save*

Field Info for arordernotes

	Field	Type	Required	Relates To

1	orderid	int	Yes	arorder
2	note	text		
3	lastchangedate	timestamp		
4	lastchangeuserid	int		
5	source_noteid	int		
6	linenumber	int		
7	internal	int		
8	id	int		

arordertax

Controller Actions

- *search*
- *view*
- *save*

Field Info for arordertax

	Field	Type	Required	Relates To
1	id	int		
2	orderid	int	Yes	arorder
3	taxrateid	int	Yes	salestax
4	tax	decimal(19,4)		
5	entrydate	datetime		
6	entryuserid	int		
7	lastchangedate	timestamp		
8	lastchangeuserid	int		
9	taxrate	decimal(12,5)		
10	taxcap	decimal(19,4)		

arorder_deposit

Controller Actions

- *search*
- *view*
- *save*

Field Info for arorder_deposit

	Field	Type	Required	Relates To
1	id	int		
2	orderid	int	Yes	arorder
3	amount	decimal(19,4)		
4	payment_method	int		
5	transaction_id	varchar(32)		

6	entrydate	date	Yes	
7	cc_type	varchar(20)		
8	last4ofcard	varchar(4)		
9	voucherid	int		
10	cancel	int		
11	canceldate	date		
12	canceluserid	int		
13	lastchangedate	timestamp		
14	bankdepositid	int		
15	cctransactionid	int		
16	checkid	varchar(20)		
17	bankstatement_enddate	date		
18	checkacctid	int		
19	currency	varchar(10)		
20	returned	int		
21	bankfee_amount	decimal(19,4)		
22	bankfee_glid	int		
23	rnum	varchar(30)		

arstatusoptions

Controller Actions

- *search*
- *view*
- *save*

Field Info for arstatusoptions				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	description	varchar(50)	Yes	
3	sortorder	int		
4	entrydate	datetime	Yes	
5	entryuserid	int		
6	lastchangedate	timestamp		
7	lastchangeuserid	int		
8	cancel	int		
9	canceldate	datetime		
10	canceluserid	int		
11	gencompanyid	int	Yes	gencompany

carrier

Controller Actions

- *search*
- *view*
- *save*

Field Info for carrier				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	customernumber	varchar(20)		
3	trackingurlbase	varchar(150)		
4	trackingurlvarname	varchar(30)		
5	description	varchar(100)	Yes	
6	vendorid	int	Yes	vendor
7	cancel	int		
8	carrier_upcharge	decimal(5,2)		

carrierservice

Controller Actions

- *search*
- *view*
- *save*

Field Info for carrierservice				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	carrierid	int	Yes	carrier
3	description	varchar(30)	Yes	
4	abbreviation	varchar(10)	Yes	
5	sortorder	int		
6	not_on_b2b	int		
7	onquote	int		
8	box1_charge	decimal(19,5)		
9	box_addl	decimal(19,5)		
10	require_shippernum	int		

ccaccount

Controller Actions

- *search*
- *view*
- *save*

Field Info for ccaccount

	Field	Type	Required	Relates To
1	id	int		
2	name	varchar(100)	Yes	
3	ccprocessorid	int	Yes	ccprocessor
4	checkacctid	int		
5	companyid	int	Yes	
6	entrydate	datetime		
7	entryuserid	int		
8	lastchangedate	timestamp		
9	lastchangeuserid	int		
10	cancel	int	Yes	
11	canceldate	datetime		
12	canceluserid	int		

ccaccountoption

Controller Actions

- *search*
- *view*
- *save*

Field Info for ccaccountoption

	Field	Type	Required	Relates To
1	id	int		
2	ccprocessorid	int	Yes	ccprocessor
3	name	varchar(255)	Yes	
4	description	varchar(255)	Yes	
5	fieldname	varchar(50)	Yes	
6	ccvalues	text		
7	ccdefault	varchar(255)		
8	cchold	int		

ccaccountsetting

Controller Actions

- *search*
- *view*
- *save*

Field Info for ccaccountsetting

	Field	Type	Required	Relates To
1	id	int		
2	ccaccountid	int	Yes	ccaccount

3	ccaccountoptionid	int	Yes	ccaccountoption
4	value	varchar(255)	Yes	
5	lastchangedate	timestamp		
6	lastchangeuserid	int		

ccprocessor

Controller Actions

- *search*
- *view*
- *save*

Field Info for ccprocessor

	Field	Type	Required	Relates To
1	id	int		
2	name	varchar(100)	Yes	
3	modulename	varchar(100)	Yes	

checkacct

Controller Actions

- *search*
- *view*
- *save*

Field Info for checkacct

	Field	Type	Required	Relates To
1	id	int		
2	name	varchar(30)	Yes	
3	glaccountid	int		glaccount
4	lastchecknumberused	decimal(20,0)		
5	defaultendorser	varchar(50)		
6	gencompanyid	int	Yes	gencompany
7	ap	int	Yes	
8	pay	int	Yes	
9	cancel	int		
10	canceldate	date		
11	canceluserid	int		
12	lastupdatedate	timestamp		
13	lastupdateuserid	int		
14	destination_routing	varchar(9)		
15	destination_name	varchar(23)		
16	origin_routing	varchar(9)		

17	origin_name	varchar(23)
18	company_name	varchar(16)
19	company_identification	varchar(10)
20	send_prenotes	int
21	accountnumber	varchar(17)
22	fractional_routing	varchar(30)
23	addresscsz	varchar(255)
24	printfullchecks	int
25	logo	varchar(100)
26	xadjust	int
27	yadjust	int
28	pay_type	int
29	currency	varchar(10)
30	depositbankaddress	varchar(200)
31	depositformat	int
32	depositlocation	varchar(30)
33	depositrouting	varchar(30)
34	stubdetails	int
35	stubformat	int
36	accountcode	varchar(50)
37	echeckccaccountid	int
38	siglines	int
39	specialpayroll	int

compositeitemid

Controller Actions

- *search*
- *view*
- *save*

Field Info for compositeitemid				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	itemcodeid	int	Yes	item
3	subitemcodeid	int	Yes	item
4	quantity	int		
5	cancel	int		
6	canceldate	datetime		
7	canceluserid	int		
8	entrydate	datetime		
9	entryuserid	int		

10	lastchangedate	timestamp		
11	lastchangeuserid	int		
12	default_itemvendorid	int		itemvendor
13	mfg	varchar(50)		
14	vordernumber	varchar(30)		
15	reference_notes	text		
16	status	int		
17	displayorder	int		

customer

Controller Actions

- *search*
- *view*
- *save*

Field Info for customer				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	taxexemptid	int		taxexempt
3	creditlimit	int		
4	salesglacctid	int		glaccount
5	salesmanid	int		salesman
6	servicerepid	int		salesman
7	invoicetermsid	int	Yes	invoiceterms
8	quotecommentid	int		quotecomment
9	interest	int		
10	billtoattnname	varchar(30)		
11	quoteattnname	varchar(30)		
12	chargecode	varchar(30)		
13	salestaxnum	varchar(30)		
14	gencompanyid	int	Yes	gencompany
15	cancel	int		
16	canceldate	datetime		
17	canceluserid	int		
18	entrydate	datetime		
19	entryuserid	int		
20	lastchangedate	timestamp		
21	lastchangeuserid	int		
22	companyname	varchar(50)	Yes	
23	address1	varchar(100)		
24	address2	varchar(100)		

25	mailstop	varchar(20)	
26	city	varchar(50)	
27	state	varchar(20)	
28	zip	varchar(15)	
29	country	varchar(20)	
30	phone1	varchar(20)	
31	phone1comment	varchar(20)	
32	phone2	varchar(20)	
33	phone2comment	varchar(20)	
34	phone3	varchar(20)	
35	phone3comment	varchar(20)	
36	phone4	varchar(20)	
37	phone4comment	varchar(20)	
38	email1	varchar(50)	
39	email1comment	varchar(20)	
40	email2	varchar(50)	
41	email2comment	varchar(20)	
42	website	varchar(100)	
43	federalid	varchar(20)	
44	entry_gender	char(1)	
45	entry_firstname	varchar(32)	
46	entry_lastname	varchar(32)	
47	entry_country_id	int	
48	entry_zone_id	int	
49	image	varchar(64)	
50	customers_dob	varchar(8)	
51	customers_password	varchar(40)	
52	customers_promotion_ok	int	
53	customer_price_level	int	pricelevel
54	points_earned	decimal(8,2)	
55	current_sales_process	int	
56	notes	varchar(255)	
57	default_po	varchar(30)	
58	customercode	varchar(30)	
59	preprint_from_box_labels	int	
60	notify_order	int	
61	notify_ship	int	
62	notify_invoice	int	
63	email_order_acknowledgement	int	
64	email_order_ship	int	
65	email_invoice	int	

66	groupid	int		igroup
67	onlygroupitems_onb2b	int		
68	taxgroupid	int		taxgroup
69	salescategoryid	int		sales_categories
70	weather_cf	char(1)		
71	weather_zip	varchar(20)		
72	time_zone	varchar(6)		
73	salutation	varchar(20)		
74	middle_name	varchar(50)		
75	name_suffix	varchar(20)		
76	currency	varchar(10)		
77	customertypeid	int		
78	resetkey	varchar(255)		
79	appstatus	char(1)		
80	areaid	int		
81	coinvoicecustomer	int		
82	nocredit	int		
83	nocredit_note	varchar(255)		
84	autobill	int		
85	pricelist_prepaybycc	int		
86	companyname2	varchar(50)		

customersalestax

Controller Actions

- *search*
- *view*
- *save*

Field Info for customersalestax

	Field	Type	Required	Relates To
1	id	int		
2	customerid	int	Yes	
3	salestaxid	int	Yes	

flexfield

Controller Actions

- *search*
- *view*
- *save*

Field Info for flexfield

	Field	Type	Required	Relates To
1	id	int		
2	tablename	varchar(100)	Yes	
3	displayname	text	Yes	
4	fieldtype	enum('T','S','C','D','H','R','A','I')	Yes	
5	sortorder	int		
6	internalname	varchar(100)		
7	defaultvalue	varchar(255)		
8	companyid	int		gencompany
9	readonly	int		
10	donotimport	int		
11	rowabove	int		
12	cellwidth	int		
13	titlelocation	varchar(10)		
14	tbsize	int		
15	linkfield	varchar(50)		
16	required	int		
17	pagelink	varchar(200)		
18	pagelinktitle	varchar(255)		

flexvalue

Controller Actions

- *search*
- *view*
- *save*

Field Info for flexvalue

	Field	Type	Required	Relates To
1	id	int		
2	flexfieldid	int	Yes	flexfield
3	recordid	int	Yes	
4	value	varchar(255)		

gencompany

Controller Actions

- *search*
- *view*
- *save*

Field Info for gencompany

	Field	Type	Required	Relates To

1	id	int	
2	address1	varchar(100)	
3	address2	varchar(100)	
4	city	varchar(50)	
5	state	varchar(20)	
6	zip	varchar(15)	
7	country	varchar(20)	
8	phone1	varchar(20)	
9	phone2	varchar(20)	
10	phone3	varchar(20)	
11	phone4	varchar(20)	
12	email	varchar(50)	
13	ar_email	varchar(50)	
14	order_email	varchar(50)	
15	ap_email	varchar(50)	
16	web	varchar(100)	
17	name	varchar(50)	Yes
18	active	int	
19	menucolor	int	
20	graphicurl	varchar(200)	
21	graphicurl_big	varchar(200)	
22	lastupdateuserid	int	
23	lastupdatedate	timestamp	
24	vat_tax	int	
25	businesstype	varchar(100)	
26	print_only_orders	int	
27	arship_shortform	int	
28	timezoneid	int	
29	currency	varchar(10)	
30	companycode	varchar(50)	
31	isconstruction	int	
32	clientllc	int	
33	defaultbarcodelabelid	int	
34	showautocompleteidboxes	int	
35	hideweather	int	
36	setupfirstrun	int	
37	showsetupmenu	int	
38	weather_cf	char(1)	
39	weather_zip	varchar(20)	

glaccount

Controller Actions

- *search*
- *view*
- *save*
- *get_balance* - get_balance returns the current balance for the given general ledger account id. Parameters can be passed into the *get_balance* action via the short URL form *glaccount/get_balance/id/date/status* where **id** is the ID of the glaccount you want the balance for, **date** is the as-of date in YYYY-MM-DD form (ex. get the balance as of 2008-01-01) and **status** refers to posted vs unposted general ledger entries (0=unposted and 1=posted).
  The POST form of the inputs should be data[id], data[date] and data[status]

Field Info for glaccount				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	name	varchar(20)	Yes	
3	description	varchar(255)		
4	accounttypeid	int	Yes	accounttype
5	companyid	int	Yes	gencompany
6	summaryaccountid	int		
7	lastchangedate	timestamp		
8	lastchangeuserid	int		
9	cancel	int		
10	canceldate	datetime		
11	canceluserid	int		
12	costcenterid	int		gl_costcenter
13	old_id	varchar(20)		
14	accountsubtypeid	varchar(10)		
15	block_journal	int		

gltransaction

Controller Actions

- *search*
- *view*
- *save*

Field Info for gltransaction				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	glaccountid	int		glaccount
3	voucherid	int	Yes	gltransvoucher
4	amount	decimal(19,4)	Yes	
5	currency	varchar(10)		

6	gltr_categoryid	int
7	gltr_costcodeid	int
8	gltr_jobid	int

gltransvoucher

Controller Actions

- *search*
- *view*
- *save*

Field Info for gltransvoucher				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	voucher	varchar(50)	Yes	
3	description	varchar(50)		
4	comments	varchar(50)		
5	wherefrom	int	Yes	
6	status	int		
7	cancel	int		
8	companyid	int	Yes	gencompany
9	standardset	int		
10	entrydate	datetime	Yes	
11	posteddate	datetime		
12	post2date	date		
13	canceldate	datetime		
14	lastchangedate	timestamp		
15	lastchangeuserid	int		
16	entryuserid	int	Yes	
17	canceluserid	int		
18	postuserid	int		
19	bankdepositid	int		
20	referenceid	int		
21	referencetable	varchar(255)		
22	taj	int		

gl_costcenter

Controller Actions

- *search*
- *view*
- *save*

Field Info for gl_costcenter

	Field	Type	Required	Relates To
1	id	int		
2	description	varchar(255)	Yes	
3	companyid	int	Yes	gencompany
4	cancel	int		
5	canceluserid	int		
6	canceldate	datetime		
7	entrydate	date	Yes	
8	entryuserid	int	Yes	

igroup

Controller Actions

- *search*
- *view*
- *save*

Field Info for igroup

	Field	Type	Required	Relates To
1	id	int		
2	groupname	varchar(255)	Yes	
3	groupcode	varchar(20)		
4	createdate	datetime		
5	gencompanyid	int		gencompany
6	createuserid	int		
7	cancel	int		
8	canceldate	datetime		
9	canceluserid	int		
10	lastchangedate	timestamp		
11	lastchangeuserid	int		
12	multiplier	decimal(5,2)		

invcompany

Controller Actions

- *search*
- *view*
- *save*

Field Info for invcompany

	Field	Type	Required	Relates To
1	id	int		

2	cash	int
3	sales	int
4	loss	int
5	cost	int
6	freight	int
7	tax	int
8	custoritemglacct	int
9	locationid	int
10	default_pricelevelid	int
11	weight_units	varchar(10)
12	category_hierarchy	int
13	default_inventoryglacct	int
14	multi_quantity	int
15	costmethod	char(1)
16	addglatpo	int
17	distributetaxfreight	int
18	inventoryvariance	int
19	breakdown_scrap	int
20	costadjustglacctid	int
21	googlebasedevkey	varchar(200)

inventorylocation

Controller Actions

- *search*
- *view*
- *save*

Field Info for inventorylocation				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	companyname	char(35)	Yes	
3	address1	char(100)		
4	address2	char(100)		
5	mailstop	char(20)		
6	city	char(50)		
7	state	char(20)		
8	zip	char(15)		
9	country	char(20)		
10	phone1	char(20)		
11	phone1comment	char(20)		
12	phone2	char(20)		

13	phone2comment	char(20)		
14	phone3	char(20)		
15	phone3comment	char(20)		
16	phone4	char(20)		
17	phone4comment	char(20)		
18	email1	char(50)		
19	email1comment	char(20)		
20	email2	char(50)		
21	email2comment	char(20)		
22	website	char(100)		
23	federalid	char(20)		
24	cancel	int		
25	canceldate	datetime		
26	canceluserid	int		
27	entrydate	datetime		
28	entryuserid	int		
29	lastchangedate	timestamp		
30	lastchangeuserid	int		
31	gencompanyid	int	Yes	gencompany
32	printlocation	int		
33	freighttax	int		
34	argl	int		
35	locationcode	varchar(20)		

invoiceterms

Controller Actions

- *search*
- *view*
- *save*

Field Info for invoiceterms				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	verbal	char(30)	Yes	
3	discountpercent	decimal(6,3)		
4	discountdays	int		
5	discountdayofmonth	int		
6	netduedays	int	Yes	
7	ar	int		
8	ap	int		
9	cancel	int		

10	canceldate	datetime
11	canceluserid	int
12	entrydate	datetime
13	entryuserid	int
14	lastchangedate	timestamp
15	lastchangeuserid	int
16	netduedayfollowing	int
17	discountdayfollowing	int
18	endofmonth	int
19	endofmonthDiscount	int
20	dayofmonthprox	int
21	monthcloseday	int

invpo

Controller Actions

- *search*
- *view*
- *save*

Field Info for invpo				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	vendorid	int	Yes	vendor
3	ponumber	varchar(20)		
4	duedate	date	Yes	
5	locationid	int	Yes	inventorylocation
6	carrierserviceid	int	Yes	carrierservice
7	tracknumber	varchar(50)		
8	contact	varchar(20)		
9	requisition	varchar(20)		
10	ordernumber	varchar(20)		
11	gencompanyid	int	Yes	gencompany
12	complete	int		
13	cancel	int		
14	canceldate	datetime		
15	canceluserid	int		
16	entrydate	datetime	Yes	
17	entryuserid	int	Yes	
18	lastchangedate	timestamp		
19	lastchangeuserid	int		
20	referenceid	int		

21	notes	varchar(255)		
22	initials	varchar(255)		
23	status	int		
24	description	varchar(255)		
25	wherefrom	int		wherefrom
26	salesordernumber	varchar(20)		
27	dropship	int		
28	shiptoid	int		shipto
29	salesorderid	int		arorder
30	paytermsid	int	Yes	invoiceterms
31	ccpurchaseid	int		
32	stageid	int		
33	potype	int		
34	recipient_userid	int		
35	sb	int		
36	main_gltr_jobid	int		
37	totaltax	decimal(19,5)		
38	main_invpoid	int		
39	prior_invpoid	int		
40	retainage	decimal(12,2)		
41	fob	char(2)		
42	chargeback	int		
43	chargeback_invpoid	int		
44	chargeback_percent	decimal(5,2)		
45	chargeback_vendorid	int		
46	pototal	decimal(19,6)		
47	currency	varchar(10)		
48	currate	decimal(22,10)		
49	taxrate	decimal(6,2)		

invpodetail

Controller Actions

- *search*
- *view*
- *save*

Field Info for invpodetail				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	invpoid	int	Yes	invpo
3	itemid	int		item

4	itemqty	decimal(19,5)		
5	itemprice	decimal(19,6)		
6	detail_description	varchar(255)		
7	inv_glacct	int	Yes	glaccount
8	conversion	decimal(12,6)		
9	itemvendorid	int		
10	purchaseunitid	int		
11	itempriceqty	decimal(19,5)		
12	notes	varchar(500)		
13	gltr_categoryid	int		
14	gltr_costcodeid	int		
15	gltr_jobid	int		
16	taxable	int		
17	detailrow	int		
18	tax	decimal(19,6)		

invponotes

Controller Actions

- *search*
- *view*
- *save*

Field Info for invponotes

	Field	Type	Required	Relates To
1	id	int		
2	invpoid	int	Yes	invpo
3	linenumber	int	Yes	
4	note	text	Yes	
5	entrydate	date	Yes	
6	entryuserid	int	Yes	
7	lastupdatedate	timestamp		
8	lastupdateuserid	int		
9	source_noteid	int		invpostdnotes

invpostdnotes

Controller Actions

- *search*
- *view*
- *save*

Field Info for invpostdnotes

	Field	Type	Required	Relates To
1	id	int		
2	shortnote	varchar(50)	Yes	
3	full_note	text	Yes	
4	sortorder	int		
5	gencompanyid	int	Yes	
6	entrydate	date	Yes	
7	entryuserid	int	Yes	
8	cancel	int		
9	canceldate	date		
10	canceluserid	int		
11	lastupdatedate	timestamp		
12	lastupdateuserid	int		

item

Controller Actions

- *search*
- *view*
- *save*
- *get_activity* - *get_activity* returns detailed transaction info for the given account and time period. Parameters can be passed into the *get_activity* action via the short URL form *glaccount/get_activity/id/begindate/enddate/status* where **id** is the ID of the glaccount you want to get gltransvouchers for, **begindate** is the start of activity date in YYYY-MM-DD form, **enddate** is the final day of activity and **status** refers to posted vs unposted general ledger entries (0=unposted and 1=posted).
  The POST form of the inputs should be data[id], data[begindate], data[enddate] and data[status]
- *get_item_cost* - *get_item_cost* returns the cost of the given item ID based on the inventory location ID.
  The URL short form parameters for this action are *item/get_item_cost/itemid/locationid* where itemid is the ID of the item you want a cost value for and locationid is the inventory location that should be used in determining the cost. (NolaPro records the price paid to vendors when receiving inventory from POs into a specific inventory location in your company; this is used to determine the item cost. This is different from the item's price which is the amount you charge your customers for the product.)
  POST form fields should be *data[itemid] and *data[locationid].
- *get_item_stats* - *get_item_stats* returns the on-hand, committed, available and on-PO quantities for the given item ID based on the inventory location ID.
  The URL short form parameters for this action are *item/get_item_stats/itemid/locationid* where itemid is the ID of the item you want stock levels for and locationid is the inventory location that should be used .
  POST form fields should be *data[itemid] and *data[locationid].
- *item_increase* - *item_increase* increases the on-hand inventory count for the given item ID by the quantity supplied for an inventory location. A cost is also required so that a GL entry can be made to account for the increase in the value of the inventory. An itemtransaction record will

also be created to record the details of this increase.
The URL short form parameters for this action are *item/item_increase/itemid/locationid/qty/cost/offsetglaccountid/notes* where itemid is the ID of the item you want to increase the count for, locationid is the inventory location that should be used, qty is the amount of the increase, cost is the per unit monetary value of the increase, offsetglaccountid is the GL account to use for the credit side of the transaction (the debit will be to the inventory account) and notes can contain additional information about the increase.
POST form fields should be *data[itemid], *data[locationid], *data[qty], *data[cost], data[offsetglaccountid] and data[notes]. (Starred fields are required.)

- *item_decrease* - *item_decrease* decreases the on-hand inventory count for the given item ID by the quantity supplied for an inventory location. A GL entry will be made to account for the decrease in the value of the inventory. An itemtransaction record will also be created to record the details of this decrease.
  The URL short form parameters for this action are *item/item_decrease/itemid/locationid/qty/offsetglaccountid/notes* where itemid is the ID of the item you want to decrease the count for, locationid is the inventory location that should be used, qty is the amount of the decrease, offsetglaccountid is the GL account to use for the debit side of the transaction (the credit will be to the inventory account) and notes can contain additional information about the decrease.
  POST form fields should be *data[itemid], *data[locationid], *data[qty], data[offsetglaccountid] and data[notes]. (Starred fields are required.)

Field Info for item

	Field	Type	Required	Relates To
1	id	int		
2	itemcode	varchar(30)	Yes	
3	compositeitemyesno	int		
4	description	varchar(100)	Yes	
5	categoryid	int	Yes	itemcategory
6	stockunitnameid	int	Yes	unitname
7	priceunitnameid	int	Yes	unitname
8	lbsperpriceunit	int		
9	priceunitsperstockunit	decimal(19,10)	Yes	
10	stockunitsperpriceunit	decimal(19,6)		
11	inventoryglacctid	int	Yes	glaccount
12	salesglacctid	int	Yes	glaccount
13	width	int		
14	length	int		
15	mwt	int		
16	cogs	int		
17	cogseffectivedate	datetime		
18	middlemanprice	int		
19	price	int		
20	priceeffectivedate	datetime		

21	status	int		
22	id_item_useinstead	varchar(100)		
23	invtype	int		
24	catalogdescription	text		
25	products_website	varchar(255)		
26	catalogsheeturl	varchar(200)		
27	graphicurl	varchar(200)		
28	pdffile	varchar(200)		
29	videofile	varchar(200)		
30	soundfile	varchar(200)		
31	companyid	int	Yes	gencompany
32	cancel	int		
33	canceldate	datetime		
34	canceluserid	int		
35	entrydate	datetime		
36	entryuserid	int		
37	lastchangedate	timestamp		
38	lastchangeuserid	int		
39	products_quantity	int		
40	products_viewed	int		
41	products_status	int		
42	products_tax_class_id	int		
43	products_costprice	decimal(8,2)		
44	products_pushup	decimal(4,2)		
45	scart	int		
46	scart_front_page	int		
47	parentitemid	int		
48	scart_minqtydisplay	int		
49	id_printorder	int		
50	taxexempt	int		
51	caliper	decimal(10,4)		
52	upc	varchar(255)		
53	author	varchar(100)		
54	notes	text		
55	b2b_block	int		
56	pos_block	int		
57	gbaseid	varchar(100)		
58	gbasedate	datetime		
59	commissionflat	decimal(19,6)		
60	commissionpercent	decimal(7,2)		

itemcategory

Controller Actions

- *search*
- *view*
- *save*

Field Info for itemcategory				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	companyid	int		gencompany
3	name	varchar(50)	Yes	
4	invtype	int		
5	cancel	int		
6	canceldate	datetime		
7	canceluserid	int		
8	entrydate	datetime		
9	entryuserid	int		
10	lastchangedate	timestamp		
11	lastchangeuserid	int		
12	seasonname1	varchar(50)		
13	seasonname2	varchar(50)		
14	seasonname3	varchar(50)		
15	seasonname4	varchar(50)		
16	seasonbegin1	varchar(25)		
17	seasonbegin2	varchar(25)		
18	seasonbegin3	varchar(25)		
19	seasonbegin4	varchar(25)		
20	seasonend1	varchar(25)		
21	seasonend2	varchar(25)		
22	seasonend3	varchar(25)		
23	seasonend4	varchar(25)		
24	categories_image	varchar(64)		
25	parent_id	int		
26	is_print_order	int		
27	sort_order	int		

itemlocation

Controller Actions

- *search*
- *view*

- *save*

Field Info for itemlocation				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	itemid	int	Yes	item
3	inventorylocationid	int	Yes	inventorylocation
4	onhandqty	int		
5	maxstocklevelseason1	int		
6	minstocklevelseason1	int		
7	orderqtyseason1	int		
8	maxstocklevelseason2	int		
9	minstocklevelseason2	int		
10	orderqtyseason2	int		
11	maxstocklevelseason3	int		
12	minstocklevelseason3	int		
13	orderqtyseason3	int		
14	maxstocklevelseason4	int		
15	minstocklevelseason4	int		
16	orderqtyseason4	int		
17	markupsetid	int		markupset
18	firstcost	decimal(19,6)		
19	midcost	decimal(19,6)		
20	lastcost	decimal(19,6)		
21	wacost	decimal(19,6)		
22	firstqty	int		
23	midqty	int		
24	lastqty	int		
25	waqty	int		
26	cancel	int		
27	canceldate	datetime		
28	canceluserid	int		
29	entrydate	datetime		
30	entryuserid	int		
31	lastchangedate	timestamp		
32	lastchangeuserid	int		
33	bin_shelf	varchar(50)		

itemoptiongroup

Controller Actions

- *search*

- *view*
- *save*

Field Info for itemoptiongroup				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	name	varchar(255)	Yes	
3	description	text	Yes	
4	optiontype	enum('select','radio','checkbox','text')	Yes	
5	allowblank	int		
6	entrydate	datetime		
7	entryuserid	int		
8	lastchangedate	timestamp		
9	lastchangeuserid	int		
10	companyid	int		
11	cancel	int		

itemtransaction

Controller Actions

- *search*
- *view*
- *save*

Field Info for itemtransaction				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	itemid	int	Yes	item
3	wherefrom	int	Yes	
4	referenceid	int		
5	locationid	int	Yes	
6	description	varchar(100)	Yes	
7	quantity	decimal(19,5)		
8	costpereach	decimal(19,6)		
9	trtype	int	Yes	
10	createdate	datetime		
11	createuserid	int		
12	cancel	int		
13	canceluserid	int		
14	canceldate	datetime		
15	id_order_shipment	int		
16	pricepereach	decimal(19,6)		

17	itemvendorid	int		itemvendor
18	status	int		
19	withdrawn	decimal(12,5)		
20	first_costeach	decimal(19,6)		
21	last_costeach	decimal(19,6)		
22	mid_costeach	decimal(19,6)		
23	wtd_costeach	decimal(19,6)		
24	orig_transid	int		
25	gltr_categoryid	int		
26	gltr_costcodeid	int		
27	gltr_jobid	int		

itemvendor

Controller Actions

- *search*
- *view*
- *save*

Field Info for itemvendor

	Field	Type	Required	Relates To
1	id	int		
2	vendorid	int	Yes	vendor
3	itemid	int	Yes	item
4	vordernumber	varchar(30)		
5	vitemunitnameid	int	Yes	unitname
6	vitemconversion	decimal(12,6)	Yes	
7	vitemcost1	decimal(19,5)		
8	vitemqty1	decimal(19,5)		
9	vitemcost2	decimal(19,5)		
10	vitemqty2	decimal(19,5)		
11	vitemcost3	decimal(19,5)		
12	vitemqty3	decimal(19,5)		
13	vitemcost4	decimal(19,5)		
14	cancel	int		
15	canceldate	datetime		
16	canceluserid	int		
17	entrydate	datetime		
18	entryuserid	int		
19	lastchangedate	timestamp		
20	lastchangeuserid	int		
21	mfg	varchar(50)		

22	is_default	int		

item_group

Controller Actions

- *search*
- *view*
- *save*

Field Info for item_group

	Field	Type	Required	Relates To
1	itemid	int	Yes	item
2	groupid	int	Yes	igroup
3	price	decimal(15,6)		

markupset

Controller Actions

- *search*
- *view*
- *save*

Field Info for markupset

	Field	Type	Required	Relates To
1	id	int		
2	description	char(35)	Yes	
3	costbased	int		
4	companyid	int		
5	cancel	int		
6	canceldate	datetime		
7	canceluserid	int		
8	entrydate	datetime		
9	entryuserid	int		
10	lastchangedate	timestamp		
11	lastchangeuserid	int		

osccompany

Controller Actions

- *search*
- *view*
- *save*

Field Info for osccompany				
	Field	Type	Required	Relates To
1	id	int		gencompany
2	default_pricelevelid	int		
3	locationid	int		
4	cost	int		glaccount
5	sales	int		glaccount
6	default_sellunitid	int		unitname
7	default_priceunitid	int		unitname
8	checking	int		glaccount
9	depositglacctid	int		glaccount
10	default_checkacctid	int		checkacct
11	default_salescategory_fulfillment	int		sales_categories
12	default_invoicetermsid	int		invoiceterms
13	oscdbhost	varchar(50)		
14	oscdbname	varchar(50)		
15	oscdbpw	varchar(50)		
16	oscdbuser	varchar(50)		
17	osclogin	varchar(50)		
18	oscpw	varchar(50)		
19	oscurl	varchar(255)		

pricelevel

Controller Actions

- *search*
- *view*
- *save*

Field Info for pricelevel				
	Field	Type	Required	Relates To
1	id	int		
2	description	char(30)	Yes	
3	companyid	int		gencompany
4	cancel	int		
5	canceldate	datetime		
6	canceluserid	int		
7	entrydate	datetime		
8	entryuserid	int		
9	lastchangedate	timestamp		
10	lastchangeuserid	int		

priceperpriceunit

Controller Actions

- *search*
- *view*
- *save*

Field Info for priceperpriceunit				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	itemid	int	Yes	item
3	itemlocationid	int	Yes	inventorylocation
4	pricelevelid	int	Yes	pricelevel
5	price	decimal(19,5)		
6	cancel	int		
7	canceldate	datetime		
8	canceluserid	int		
9	entrydate	datetime		
10	entryuserid	int		
11	lastchangedate	timestamp		
12	lastchangeuserid	int		

quotecomment

Controller Actions

- *search*
- *view*
- *save*

Field Info for quotecomment				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	comments	char(100)	Yes	
3	cancel	int		
4	canceldate	datetime		
5	canceluserid	int		
6	entrydate	datetime		
7	entryuserid	int		
8	lastchangedate	timestamp		
9	lastchangeuserid	int		

salesman

Controller Actions

- *search*
- *view*
- *save*

Field Info for salesman				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	payrollid	int		
3	commissionrate	decimal(6,3)		
4	commissionbase	int		
5	servicerep	int		
6	salesman	int		
7	gencompanyid	int	Yes	
8	cancel	int		
9	canceldate	datetime		
10	canceluserid	int		
11	entrydate	datetime	Yes	
12	entryuserid	int	Yes	
13	lastchangedate	timestamp		
14	lastchangeuserid	int		
15	name	varchar(50)	Yes	
16	address1	varchar(100)		
17	address2	varchar(100)		
18	mailstop	varchar(20)		
19	city	varchar(50)		
20	state	varchar(20)		
21	zip	varchar(15)		
22	country	varchar(20)		
23	phone1	varchar(20)		
24	phone1comment	varchar(20)		
25	phone2	varchar(20)		
26	phone2comment	varchar(20)		
27	phone3	varchar(20)		
28	phone3comment	varchar(20)		
29	phone4	varchar(20)		
30	phone4comment	varchar(20)		
31	email1	varchar(50)		
32	email2	varchar(50)		
33	email1comment	varchar(20)		
34	email2comment	varchar(20)		

35	federalid	varchar(20)
36	commissionrate2	decimal(6,3)
37	vendorid	int
38	blockchange	int

salestax

Controller Actions

- *search*
- *view*
- *save*

Field Info for salestax				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	taxname	char(30)	Yes	
3	taxrate	decimal(7,4)		
4	taxbase	int		
5	glacctid	int	Yes	glaccount
6	cancel	int		
7	canceldate	datetime		
8	canceluserid	int		
9	entrydate	datetime		
10	entryuserid	int		
11	lastchangedate	timestamp		
12	lastchangeuserid	int		
13	taxstate	varchar(5)		
14	glacctid_taxpaid	int		glaccount
15	capamount	decimal(14,6)		
16	gencompanyid	int		gencompany

sales_categories

Controller Actions

- *search*
- *view*
- *save*

Field Info for sales_categories				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	description	varchar(50)	Yes	
3	entrydate	datetime		

4	entryuserid	int		
5	cancel	int		
6	canceldate	datetime		
7	canceluserid	int		
8	lastupdatedate	timestamp		
9	gencompanyid	int		gencompany
10	allow_b2b_display	int		

search_cat

Controller Actions

- *search*
- *view*
- *save*

Field Info for search_cat

	Field	Type	Required	Relates To
1	id	int		
2	name	varchar(100)	Yes	
3	level	int	Yes	
4	cancel	int		
5	categories_image	varchar(64)		
6	companyid	int	Yes	gencompany
7	parentid	int		

search_tree

Controller Actions

- *search*
- *view*
- *save*

Field Info for search_tree

	Field	Type	Required	Relates To
1	id	int		
2	itemid	int	Yes	item
3	companyid	int	Yes	gencompany
4	searchcatid	int	Yes	search_cat

service_workclass

Controller Actions

- *search*

- *view*
- *save*

Field Info for service_workclass

	Field	Type	Required	Relates To
1	id	int		
2	name	varchar(50)	Yes	
3	gencompanyid	int		gencompany
4	entrydate	date		
5	entryuserid	int		
6	cancel	int		
7	canceldate	datetime		
8	canceluserid	int		
9	lastchangedate	timestamp		
10	lastchangeuserid	int		

shipto

Controller Actions

- *search*
- *view*
- *save*

Field Info for shipto

	Field	Type	Required	Relates To
1	id	int		
2	defaultshipvia	int		carrierservice
3	cancel	int		
4	canceldate	datetime		
5	canceluserid	int		
6	entrydate	datetime		
7	entryuserid	int		
8	lastchangedate	timestamp		
9	lastchangeuserid	int		
10	address1	varchar(100)		
11	address2	varchar(100)		
12	mailstop	varchar(20)		
13	city	varchar(50)		
14	state	varchar(20)		
15	zip	varchar(15)		
16	country	varchar(20)		
17	phone1	varchar(20)		

18	phone1comment	varchar(20)		
19	phone2	varchar(20)		
20	phone2comment	varchar(20)		
21	phone3	varchar(20)		
22	phone3comment	varchar(20)		
23	phone4	varchar(20)		
24	phone4comment	varchar(20)		
25	email1	varchar(50)		
26	email2	varchar(50)		
27	email1comment	varchar(20)		
28	email2comment	varchar(20)		
29	customerid	int	Yes	customer
30	shipname	varchar(50)	Yes	
31	entry_firstname	varchar(32)		
32	entry_lastname	varchar(32)		
33	entry_country_id	int		
34	entry_zone_id	int		
35	dummy_record	int		
36	taxexemptid	int		taxexempt
37	salestaxnum	varchar(30)		
38	weather_cf	char(1)		
39	weather_zip	varchar(20)		
40	time_zone	varchar(6)		
41	is_default	int		
42	shipper_number	varchar(50)		
43	shipname2	varchar(50)		

shiptosalestax

Controller Actions

- *search*
- *view*
- *save*

Field Info for shiptosalestax				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	shiptoid	int	Yes	
3	salestaxid	int	Yes	

taxexempt

Controller Actions

- *search*
- *view*
- *save*

Field Info for taxexempt				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	exemptname	char(30)	Yes	
3	cancel	int		
4	canceldate	datetime		
5	canceluserid	int		
6	entrydate	datetime		
7	entryuserid	int		
8	lastchangedate	timestamp		
9	lastchangeuserid	int		

<u>taxgroup</u>
Controller Actions

- *search*
- *view*
- *save*

Field Info for taxgroup				
	Field	**Type**	**Required**	**Relates To**
1	id	int		
2	groupcode	varchar(20)	Yes	
3	createdate	datetime		
4	gencompanyid	int		gencompany
5	createuserid	int		
6	cancel	int		
7	canceldate	datetime		
8	canceluserid	int		
9	lastchangedate	timestamp		
10	lastchangeuserid	int		

<u>unitname</u>
Controller Actions

- *search*
- *view*
- *save*

Field Info for unitname				
	Field	Type	Required	Relates To
1	id	int		
2	unitname	char(50)	Yes	
3	cancel	int		
4	canceldate	datetime		
5	canceluserid	int		
6	entrydate	datetime		
7	entryuserid	int		
8	lastchangedate	timestamp		
9	lastchangeuserid	int		

vendor

Controller Actions

- *search*
- *view*
- *save*

Field Info for vendor				
	Field	Type	Required	Relates To
1	name	varchar(30)		
2	id	int		
3	orderfromname	varchar(30)		
4	paytermsid	int	Yes	invoiceterms
5	paynone	int		
6	defaultglacctid	int		glaccount
7	defaultbilldescription	varchar(50)		
8	customeraccount	varchar(20)		
9	gencompanyid	int	Yes	gencompany
10	cancel	int		
11	canceldate	datetime		
12	canceluserid	int		
13	entrydate	datetime		
14	entryuserid	int		
15	lastchangedate	timestamp		
16	lastchangeuserid	int		
17	companyname	varchar(100)	Yes	
18	address1	varchar(100)		
19	address2	varchar(100)		
20	mailstop	varchar(20)		
21	city	varchar(50)		

22	state	varchar(20)	
23	zip	varchar(15)	
24	country	varchar(20)	
25	phone1	varchar(20)	
26	phone1comment	varchar(20)	
27	phone2	varchar(20)	
28	phone2comment	varchar(20)	
29	phone3	varchar(20)	
30	phone3comment	varchar(20)	
31	phone4	varchar(20)	
32	phone4comment	varchar(20)	
33	email1	varchar(50)	
34	email1comment	varchar(20)	
35	email2	varchar(50)	
36	email2comment	varchar(20)	
37	website	varchar(100)	
38	federalid	varchar(20)	
39	orderfrom_companyname	varchar(100)	
40	orderfrom_address1	varchar(100)	
41	orderfrom_address2	varchar(100)	
42	orderfrom_mailstop	varchar(20)	
43	orderfrom_city	varchar(50)	
44	orderfrom_state	varchar(20)	
45	orderfrom_zip	varchar(15)	
46	orderfrom_country	varchar(20)	
47	image	varchar(64)	
48	scart	int	
49	workclassid	int	service_workclass
50	default_cost_per_hour	decimal(12,6)	
51	vendor_code	varchar(50)	
52	vendor_lead_time	int	
53	carrierserviceid	int	
54	noreturns	int	
55	percent_commission	decimal(8,4)	
56	flat_commission	decimal(12,4)	
57	parent_vendorid	int	
58	bankrouting	varchar(255)	
59	bankaccountnum	varchar(255)	
60	taxgroupid	int	taxgroup
61	vendorgroupid	int	
62	vendorcode	varchar(30)	

63	salutation	varchar(20)		
64	first_name	varchar(50)		
65	middle_name	varchar(50)		
66	name_suffix	varchar(20)		
67	vendortypeid	int		
68	default_retainer	decimal(6,3)		
69	currency	varchar(10)		
70	gets1099	int		

vendorsalestax

Controller Actions

- *search*
- *view*
- *save*

Field Info for vendorsalestax

	Field	Type	Required	Relates To
1	id	int		
2	vendorid	int	Yes	
3	salestaxid	int	Yes	

wherefrom

Controller Actions

- *search*
- *view*
- *save*

Field Info for wherefrom

	Field	Type	Required	Relates To
1	id	int		
2	module	varchar(10)	Yes	
3	name	varchar(100)		

Code Examples

Invoice Add PHP Example

```
$api = new npapi();
// Main Invoice Info
$api->set_table('arinvoice');
$data = array(    "invoicenumber" => '5569-LR'.date("is"),
                  "orderbycompanyid" => 18,     // Customer ID
                  "shiptocompanyid" => 9,       // Ship To ID
                  "wherefrom" => 2,             // This should be 2
                  "ponumber" => '34568R',
                  "pricelevelid" => 1,
                  "inventorylocationid" => 1,
                  "duedate"=>date("Y-m-d"),
                  "invoicedate"=>date("Y-m-d"),
                  "discountdate"=>date("Y-m-d"),
                  "shipcost" => 15.33,
                  "invoicetermsid" => 1,
                  "salesmanid" => 1,
                  "sales_categories" => 1,
                  "arglaccountid" => 866
);
$api->set_data($data);

// Invoice Details
$api->set_table('arinvoicedetail');
$data = array(
            array('linenumber' => 1, 'qty' => 5, 'glaccountid' => 1026,
'costglaccountid' => 1015, 'pricelevelid' => 1, 'qtyunitnameid' => 1, 'description'
=> "Joe Bean's Candy", 'priceach' => 2.21, 'costeach' => 1.99, 'taxflag' => 1,
'taxflagid' => 1),
            array('linenumber' => 2, 'qty' => 4, 'glaccountid' => 1026,
'costglaccountid' => 1015, 'pricelevelid' => 1, 'itemid' => 19, 'priceach' => 6.25,
'costeach' => 4.23, 'taxflag' => 1, 'taxflagid' => 1),
            array('linenumber' => 3, 'qty' => 3, 'glaccountid' => 1026,
'costglaccountid' => 1015, 'pricelevelid' => 1, 'itemid' => 32, 'priceach' => 4.99,
'costeach' => 2.99, 'taxflag' => 1, 'taxflagid' => 1),
            );
$api->set_data($data);

// Invoice Taxes
$api->set_table('arinvoicetaxdetail');
$data = array(
            array('taxid' => 2, 'taxamount' => 3.98)
            );
$api->set_data($data);

// Invoice Costs
$api->set_table('arinvoicedetailcost');
$data = array(
            array('cost' => 16.52, 'inventoryglacctid' => 1025, 'costglaccountid'
=> 1030, 'itemid' => 19, 'locationid' => 1),
            array('cost' => 4.21, 'inventoryglacctid' => 1025, 'costglaccountid' =>
1014),
            );
$api->set_data($data);
```

```
// General Invoice Note
$api->set_table('arinvoicenotes');
$data = array(
            array('note' => "General note for the entire invoice")
            );
$api->set_data($data);

// Detail Invoice Note
$api->set_table('arinvoicedetailnotes');
$data = array(
            array('linenumber' => 2, 'note' => "Specific note for line 2"),
            array('linenumber' => 3, 'note' => "Some additional info about line 3")
            );
$api->set_data($data);
$api->action = "arinvoice/save";
$api->display_xml = 1;
$api->debug = 0;
$api->create_xml_object = 0;
$buffer = $api->call_api();
```

Invoice Autopay PHP Example

```
$api = new npapi();
$data = array(    'customerid' => 15,
                  'amount' => 700,
                  'type' => 1,
                  'transactionid' => '834543',
                  'date' => '2012-12-14',
                  'bankglaccountid' => 862);
$api->set_data($data);

$api->action = "arinvoice/autopay";
$api->display_xml = 1;
$api->debug = 0;
$api->create_xml_object = 0;
$buffer = $api->call_api();
```

Purchase Order PHP Example

```
$api->set_table('invpo');
$data = array(    'paytermsid' => 1,
                  'carrierserviceid' => 1,
                  'vendorid' => 13,
                  'locationid' => 1,
                  'duedate' => '2013-07-15',
                  'notes' => 'Please ship all together',
            );
$api->set_data($data);

$api->set_table('invpodetail');
$data = array(
            array('itemid' => 3143, 'purchaseunitid' => 2, 'itemqty' => 4,
'itemprice' => 50.35, 'inv_glacct' => 899),
            array('itemid' => 57, 'purchaseunitid' => 2, 'itemqty' => 120,
'itemprice' => 25, 'inv_glacct' => 900),
            array('detail_description' => 'Non-Inventory Item Request', 'itemprice'
=> 12.35, 'notes' => "We need this one as soon as possible.", 'inv_glacct' => 899),
                  );
```

```
$api->set_data($data);

$api->action = "invpo/save";
$api->display_xml = 1;
$api->debug = 0;
$api->create_xml_object = 0;
$buffer = $api->call_api();
```