

AddressTwo API Documentation

Action URL: http://api.addresstwo.com/cf/function_api.asp

All actions are sent via POST method to this URL.

Required Fields on ALL posts

txtUserName

txtPassword

(OR *db_id* and *userID* in place of username/password credentials)

cmd (see below for accepted values and respective instructions)

cmd = "auth_user"

To receive a pair of credentials that will never change, pass the username and password of the user and receive two new variables called *db_id* and *userID* that you can store permanently and use for all future API calls. The following input fields are accepted (* denotes a required field):

txtUserName*

txtPassword*

An XML response will include the *db_id* and *userID* variables in <user>...</user>.

cmd = "find_contact"

To search for contacts, insert the value "find_contact" in the cmd field. You can search using 1 or all of the following fields:

txtAccountName // this is the Company Name, if available

txtContactFirstName

txtContactLastName

txtContactEmail

txtCreatedAfter // this is very useful for regular synchronization routines

txtModifiedAfter // this is very useful for regular synchronization routines

txtContactID

An XML response will include a collection of records in <contact>...</contact> with various detail nodes within each object.

cmd = “find_account”

To search for accounts, insert the value “find_account” in the cmd field. You can search using 1 or all of the following fields:

- txtAccountName // this is the Company Name, if available
- txtAccountID
- txtCreatedAfter // this is very useful for regular synchronization routines
- txtModifiedAfter // this is very useful for regular synchronization routines

An XML response will include a collection of records in <account>...</account> with various detail nodes within each object.

cmd = “insert_contact”

To create a new contact/account, insert the value “insert_contact” in the cmd field. The following input fields are accepted (* denotes a required field):

- txtAccountID // If the adding the contact to an existing Account, include txtAccountID. All other txtAccount* fields will be ignored.
- txtAccountName // this is the Company Name, if available
- txtAccountPhone1
- txtAccountPhone2
- txtAccountAddress1
- txtAccountAddress2
- txtAccountCity
- txtAccountState
- txtAccountZip
- txtAccountCounty
- txtAccountWebsite
- txtAccountStatus // this is the customer status, i.e. Customer, Prospect, Lead, etc.
- txtContactFirstName *
- txtContactLastName
- txtContactSalutation // this consists of values such as “Mr.” or “Mrs.”
- txtContactBDay // this is the contact’s birth date, can be full date of birth, or month/day only.
- txtContactPhone1
- txtContactPhone2
- txtContactEmail

cmd = “update_account”

To update an existing account, insert the value “update_account” in the cmd field. The following input fields are accepted (* denotes a required field). Omitted inputs will be ignored in the update statement:

```
txtAccountID * // this is the identifying ID number for the account being updated and can be retrieved in the XML of contact_find
txtAccountName
txtAccountPhone1
txtAccountPhone2
txtAccountAddress1
txtAccountAddress2
txtAccountCity
txtAccountState
txtAccountZip
txtAccountWebsite
txtAccountStatus
txtAccountManager // this must be a valid system username string.
```

cmd = “update_contact”

To update an existing account, insert the value “update_account” in the cmd field. The following input fields are accepted (* denotes a required field). Omitted inputs will be ignored in the update statement:

```
txtContactID * // this is the identifying ID number for the contact being updated and can be retrieved in the XML of contact_find
txtContactFirstName *
txtContactLastName
txtContactSalutation // this consists of values such as “Mr.” or “Mrs.”
txtContactPhone1
txtContactPhone2
txtContactEmail
```

cmd = “insert_history”

To insert a history record in an existing account, insert the value “insert_history” in the cmd field. The following input fields are accepted (* denotes a required field):

```
txtContactID * // this is the identifying ID number for the contact being updated and can be retrieved in the XML of contact_find
txtAccountID * // this is the identifying ID number for the account being updated and can be retrieved in the XML of contact_find
txtResult // this consists of short descriptive phrases such as “Call In” or “Sent Email”
txtMessage // this is the more descriptive text input, up to 255 characters, of the exact event being recorded
```

cmd = “insert_task”

To insert a history record in an existing account, insert the value “insert_history” in the cmd field. The following input fields are accepted (* denotes a required field):

txtContactID * // this is the identifying ID number for the contact being updated and can be retrieved in the XML of contact_find
txtAccountID * // this is the identifying ID number for the account being updated and can be retrieved in the XML of contact_find
txtAssignTo // this is the username of the user to whom the task is assigned. If blank, the authenticating user is assigned
txtDueDate // vbShortDate – the date that the task is due. If blank, today’s date is defaulted
txtResult // Text - this consists of short descriptive phrases such as “Call out” or “Send Email”
txtMessage // this is the more descriptive text input, up to 255 characters, of the exact event being recorded
txtSendUserNotice // if set to “yes” an immediate email will be sent to the AssignTo user

cmd = “report_history”

To view a report of history records, insert the value “report_history” in the cmd field. The following input fields are accepted (* denotes a required field):

txtContactID // this is the identifying ID number for a contact and can be retrieved in the XML of contact_find
txtAccountID // this is the identifying ID number for the account and can be retrieved in the XML of contact_find
txtStartDate // if a valid date is sent, results will only include records on or after this date
txtEndDate // if a valid date is sent, results will only include records on or before this date
txtSinceID // if a numeric ID is sent, results will only include records with a higher ID. Valuable for regular synchronization routines.

NOTE: History reports are paginated. A <meta> ... </meta> node will tell you how many pages there are. Paginate by passing *txtPage* variable in your form post with an integer value up to the total number of pages.

cmd = “ebill_charge” ***Action URL: [https:// addresstwo.com/cf/function_api.asp](https://addresstwo.com/cf/function_api.asp)

To charge a credit card, insert the value “ebill_charge” in the cmd field. The following input fields are accepted (* denotes a required field):

txtAccountID * // this is the identifying ID number for the account being updated and can be retrieved in the XML of contact_find
txtCardNumber * // this is the full credit card number without dashes or spaces
txtExpirationM * // the Month value of the expiration date, in 2-digit numeric format
txtExpirationY * // the Year value of the expiration date, in 4-digit numeric format
txtCardFirstName * // the First Name of the Cardholder’s name as displayed on the credit card
txtCardLastName * // the Last Name of the Cardholder’s name as displayed on the credit card
txtAmount * // the amount of the purchase exclusive of any tax or shipping
txtDescription * // the description of the item being purchased
txtCardCode // the card verification code. REQUIRED if the Billing Address is not included.
txtBillingAddress
txtBillingCity

txtBillingState
txtBillingZip
txtContactEmail // if the customer would like an email auto-sent, include this value
txtTax // the amount of sales tax to be added to the txtAmount, if any
txtShipping // the amount of any shipping charges to be added to the txtAmount, if any
txtTransaction // accepts a string value of “yes” or “no”. If yes, then a transaction history entry will be generated.
txtReference // an optional string which can contain invoice number or other reference id.

If included, the following fields will initiate a recurring charge based on the above data:

txtFrequency // accepts a string value of m, q, d, ww, or yyyy. REQUIRED for Recurring Charges
txtStartDate // accepts a date value of the first charge. REQUIRED for Recurring Charges
txtEndDate // accepts a date value for the date when all charges should stop. If blank, charges will recur indefinitely.
txtEBillModify // if the value = “update_recur” and a match is found to the txtReference value, updates details of existing charge

cmd = “list_contactforms”

To return a list of all contact forms, insert the value “list_contactforms” in the cmd field. No input fields are accepted, only an XML result set will be returned including the form names and embed code.

cmd = “find_sku”

To view a list of all SKU’s for integration with e-commerce, insert the value “find_sku” in the cmd field. The following input fields are accepted (* denotes a required field):

txtStatus // accepts a string value of Active, Retired, or Pre-Order and will filter results accordingly
txtSKU // accepts a string value of a valid SKU number to return details for only one sku at a time.

cmd = “auth_contact”

To authenticate a contact against the AddressTwo-hosted logins table for integration with a hosted customer portal or e-commerce application, insert the value “auth_contact” in the cmd field. The following input fields are accepted (* denotes a required field):

txtUserName_Contact * // accepts a string value of the contact’s username
txtPassword_Contact * // accepts a string value of the contact’s password