



Email Purchase Quick Start

This document describes the command flow to purchase and manage Email services using the eNom API.

Copyright © 2014 eNom, Inc.

No portion of this document may be copied, modified, reproduced, or distributed without the express written permission of eNom Inc..

eNom Email Purchase Quick Start - v1.0
Last modified: December, 2014

Contents

Email Purchase Quick Start	1
Step 1: Purchase and Verify Email Accounts	2
Step 2: Set Auto-Renew for Enom Email	6
Step 3: Create and Verify an Email User	8
Step 4: Set and Verify Administrator Privileges	10
Step 5: Delete a user	12
For More Information	14



Email Purchase Quick Start

eNom provides a customizable and flexible email solution to our global resellers, and we have an API that supports everything you need to configure it - from purchasing email, to managing and deleting users, and everything in between. You can learn more about the features of our email solution by visiting <http://www.enom.com/email/default.aspx>.

If you are already familiar with buying email services using our API, this quick start guide will show what's been updated to support purchasing and managing email services. These changes are specific to the `PurchaseServices` API command, so please take a closer look at [Step 1](#) for additional information.

This document provides quick steps for using the eNom API to implement the purchase and management of our email product. Here's how you can purchase email and manage users:

- [Step 1](#): Purchase and verify an email account
- [Step 2](#): Set Auto-Renew for eNom Email
- [Step 3](#): Create and verify a user
- [Step 4](#): Set and verify administrator privileges
- [Step 5](#): Delete user

NOTE: Some of our API commands used for email have POP3 as part of the command names, and although they're labeled as such, our email product supports both POP3 and IMAP protocols.



Step 1: Purchase and Verify Email Accounts

You can use the `PurchaseServices` command to purchase one or more new email accounts.

Command: `PurchaseServices`

Input Parameters to Note:

QuantityUnit: This parameter has two permitted options: `packs` and `boxes`. `Boxes` defines a single mailbox (email address) and `packs` are units of 10 boxes. The `quantity` parameter value will be multiplied by the number represented by these two values. For example: when `quantity` equals 5 and `quantityunit` equals `packs`, you are purchasing 5 packs of 10 addresses for a total of 50 email addresses.

Quantity: This parameter represents the number of email addresses you wish to purchase. The default value is 10 email addresses. The maximum value allowable is 999.

Capacity: This parameter specifies which storage capacity to purchase. eNom Email is offered in two different storage capacities, 1 GB and 10 GB. The default value is 1 GB.

Billingperiod: This parameter represents the billing period option for the email purchase. The permitted options are `month` and `year`. The default value is set to `year`.

Output Parameters to Note:

BundleID: This parameter is needed for other API commands to manage email users. You can also retrieve this value by calling the `GetPOP3` API command.

Query:

```
http://resellertest.enom.com/interface.asp?
command=purchaseservices&uid=resellid&pw=resellpw
&responsetype=xml&service=pop3&sld=enom-email-example-123
```



Step 1: Purchase and Verify Email Accounts

```
&tld=com&quantity=15&capacity=1&billingperiod=year  
&quantityunit=boxes
```

Output:

```
<interface-response>  
  <Version>3</Version>  
  <orderid>161786106</orderid>  
  <OrderAmount>15</OrderAmount>  
  <Success>True</Success>  
  <Bundles>  
    <ActionType>Create</ActionType>  
    <BundleID>1004279</BundleID>  
    <VasItemID>1004279</VasItemID>  
    <DomainName>enom-email-example-123.com</DomainName>  
    <Capacity>1</Capacity>  
    <Period>1</Period>  
    <BillingPeriod>year</BillingPeriod>  
    <ExpDate>11/25/2015</ExpDate>  
    <Quantity>15</Quantity>  
    <TotalQuantity>15</TotalQuantity>  
  </Bundles>  
  <BundleCount>1</BundleCount>  
  <Command>PURCHASESERVICES</Command>  
  <APIType>API</APIType>  
  <Language>eng</Language>  
  <ErrCount>0</ErrCount>  
  <ResponseCount>0</ResponseCount>  
  <MinPeriod>1</MinPeriod>  
  <MaxPeriod>10</MaxPeriod>  
  <Server>RESELLERTEST</Server>  
  <Site>eNom</Site>  
  <IsLockable>True</IsLockable>  
  <IsRealTimeTLD>True</IsRealTimeTLD>  
  <TimeDifference>+08.00</TimeDifference>  
  <ExecTime>0.340</ExecTime>  
  <Done>>true</Done>  
  <RequestDateTime>11/25/2014 12:47:29 PM</RequestDateTime>  
</interface-response>
```

Step 1: Purchase and Verify Email Accounts

After purchasing email, you can verify that the account was created properly by calling the `GetPop3` command.

Command: `GetPop3`

Input Parameters to Note:

sld: This parameter is the second-level domain name for the domain for which you purchased email services.

Output Parameters to Note:

Bundleid: This parameter is needed for other API commands to manage email users.

Query:

```
http://resellertest.enom.com/interface.asp?
uid=resellid&pw=resellpw&command=GetPop3
&sld=enom-email-example-123&tld=com&responsetype=xml
```

Output:

```
<interface-response>
  <EmailCount>0</EmailCount>
  <Paks>
    <Pak>
      <BundleId>1004279</BundleId>
      <QtyPurchased>15</QtyPurchased>
      <QtyAvailable>15</QtyAvailable>
      <ExpDate>
        <![CDATA[ 11/25/2015 ]]>
      </ExpDate>
      <AutoRenew>1</AutoRenew>
      <Quota>1024</Quota>
      <ProdType>2011</ProdType>
      <Version>3</Version>
    </Pak>
  </Paks>
  <Count>1</Count>
  <AdditionalHostAcctLogin/>
  <Successful>True</Successful>
  <Success>True</Success>
  <Command>GETPOP3</Command>
  <APIType>API.NET</APIType>
```

Step 1: Purchase and Verify Email Accounts

```
<Language>eng</Language>
<ErrCount>0</ErrCount>
<ResponseCount>0</ResponseCount>
<MinPeriod>1</MinPeriod>
<MaxPeriod>10</MaxPeriod>
<Server>RESELLERTEST</Server>
<Site>eNom</Site>
<IsLockable/>
<IsRealTimeTLD/>
<TimeDifference>+0.00</TimeDifference>
<ExecTime>0.078</ExecTime>
<Done>true</Done>
<TrackingKey>fea0ee5e-c27e-4975-ba07-alac12e0d-
fa6</TrackingKey>
<RequestDateTime>11/25/2014 12:51:57 PM</RequestDateTime>
</interface-response>
```




Step 2: Set Auto-Renew for Enom Email

You can use the `SetPakRenew` command to specify if you want your email product automatically renewed.

Command: `SetPakRenew`

Input Parameters to Note:

bundleid: This parameter represents the `bundleid` number received from the `PurchaseServices` command's output message.

autopakrenew: This parameter enables/disables the auto renew feature of email. The permitted values are 1 (enabled) and 0 (disabled).

Query:

```
http://resellertest.enom.com/interface.asp?
uid=resellid&pw=resellpw&command=SETPAKRENEW
&AutoPakRenew=1&BundleID=1004279
&sld=enom-email-example-123
&tld=com&responsetype=xml
```

Output:

```
<interface-response>
  <PakUpdated>1004279</PakUpdated>
  <AutoPakRenew>1</AutoPakRenew>
  <success>True</success>
  <Command>SETPAKRENEW</Command>
  <APIType>API.NET</APIType>
```

Step 2: Set Auto-Renew for Enom Email

```
<Language>eng</Language>
<ErrCount>0</ErrCount>
<ResponseCount>0</ResponseCount>
<MinPeriod>1</MinPeriod>
<MaxPeriod>10</MaxPeriod>
<Server>RESELLERTEST</Server>
<Site>eNom</Site>
<IsLockable/>
<IsRealTimeTLD/>
<TimeDifference>+0.00</TimeDifference>
<ExecTime>0.078</ExecTime>
<Done>>true</Done>
<TrackingKey>144b1e36-20bc-4032-9d6e-b3cee407e647</Track-
ingKey>
<RequestDateTime>11/25/2014 2:38:18 PM</RequestDateTime>
</interface-response>
```



Step 3: Create and Verify an Email User

Use the `SetupPop3User` command to create an email user for a previously purchased email account. You can verify if this user was setup correctly by calling the `GetPOP3` API command as you did in [Step 1](#).

Command: `SetupPop3User`

Input Parameters to Note:

Bundleid: This parameter represents the `bundleid` number received from the `PurchaseServices` command's output message.

Query:

```
http://resellertest.enom.com/interface.asp?
uid=resellid&pw=resellpw&command=SETUPPOP3USER
&BundleID=1004279&UserName1=UserOne
&Password1=Abcdefg1&sld1=enom-email-example-123
&tld1=com&ForwardTo1=&responsetype=xml
```

Output:

```
<interface-response>
  <Success>True</Success>
  <AccountsCreated>1</AccountsCreated>
  <AccountsFailed>0</AccountsFailed>
  <Command>SETUPPOP3USER</Command>
  <APIType>API.NET</APIType>
  <Language>eng</Language>
  <ErrCount>0</ErrCount>
```

Step 3: Create and Verify an Email User

```
<ResponseCount>0</ResponseCount>
<MinPeriod>1</MinPeriod>
<MaxPeriod>10</MaxPeriod>
<Server>RESELLERTEST</Server>
<Site>eNom</Site>
<IsLockable/>
<IsRealTimeTLD/>
<TimeDifference>+0.00</TimeDifference>
<ExecTime>4.922</ExecTime>
<Done>>true</Done>
<TrackingKey>0a38040b-4782-4854-9c03-afe2b6f-
d4b8f</TrackingKey>
<RequestDateTime>11/25/2014 12:54:11 PM</RequestDateTime>
</interface-response>
```



Step 4: Set and Verify Administrator Privileges

You can use the `GetPop3` command used in [Step 1](#) and [Step 2](#) to retrieve the list of email administrators for a domain. You should notice the output parameter `admin` returns a value of 1 if the user has already been set up as an administrator, otherwise, the value returned is 0.

You can then use the `ModifyPop3` command to promote a user to become an email administrator for a domain. An email administrator can update passwords for any email account for a domain. You would use the `GetPOP3` command once again to verify the `isadmin` parameter's value has been set to 1.

NOTE: You can also use the `ModifyPOP3` command to update an email user's password by using the `Password1` input parameter with the new password.

Command: `ModifyPop3`

Input Parameters to Note:

Username1: The user name of the email user to be promoted or demoted to/from email administrator.

Bundleid: This parameter represents the `bundleid` number received from the `PurchaseServices` command's output message.

Emailcount: This is the number of user accounts on this domain to update. If you choose to update two user accounts using one call, you would send 2 for `emailcount`, and the appropriate `username1` and `username2` values.

Query:

```
http://resellertest.enom.com/interface.asp?  
uid=resellid&pw=resellpw&command=ModifyPop3
```

Step 4: Set and Verify Administrator Privileges

```
&UserName1=UserOne&IsAdmin1=1&BundleID=1004279
&sld=enom-email-example-123&tld=com&responsetype=xml
&EmailCount=1
```

Output:

```
<interface-response>
  <Success>True</Success>
  <AccountsModified>1</AccountsModified>
  <AccountsFailed>0</AccountsFailed>
  <Version>3</Version>
  <Command>MODIFYPOP3</Command>
  <APIType>API.NET</APIType>
  <Language>eng</Language>
  <ErrCount>0</ErrCount>
  <ResponseCount>0</ResponseCount>
  <MinPeriod>1</MinPeriod>
  <MaxPeriod>10</MaxPeriod>
  <Server>RESELLERTEST</Server>
  <Site>eNom</Site>
  <IsLockable/>
  <IsRealTimeTLD/>
  <TimeDifference>+0.00</TimeDifference>
  <ExecTime>1.984</ExecTime>
  <Done>true</Done>
  <TrackingKey>9c5e4492-d281-47bb-a163-a7814c159124</Track-
ingKey>
  <RequestDateTime>11/25/2014 2:25:51 PM</RequestDateTime>
</interface-response>
```



Step 5: Delete a user

You can use the `DeletePop3` command to delete an email user. Keep in mind that once the email user is deleted it cannot be recovered, and all email objects associated with that user object (emails, calendar, contacts) will be permanently deleted. You can use the `GetPOP3` command again to verify that the user object has been deleted.

Command: `DeletePop3`

Input Parameters to Note:

Username1: The user name of the email user to be deleted.

Sld: This parameter is the second-level domain name for the domain associated with the user name to be deleted.

Query:

```
http://resellertest.enom.com/interface.asp?
uid=resellid&pw=resellpw&command>DeletePop3
&UserName1=UserOne&sld=enom-email-example-123
&tld=com&responsetype=xml
```

Output:

```
<interface-response>
  <AccountsDeleted>1</AccountsDeleted>
  <success>True</success>
  <Command>DELETEPOP3</Command>
  <APIType>API.NET</APIType>
  <Language>eng</Language>
  <ErrCount>0</ErrCount>
  <ResponseCount>0</ResponseCount>
  <MinPeriod>1</MinPeriod>
```

Step 5: Delete a user

```
<MaxPeriod>10</MaxPeriod>
<Server>RESELLERTEST</Server>
<Site>eNom</Site>
<IsLockable/>
<IsRealTimeTLD/>
<TimeDifference>+0.00</TimeDifference>
<ExecTime>2.578</ExecTime>
<Done>>true</Done>
<TrackingKey>998faee9-05c6-42df-9b48-9c82e388b43b</Track-
ingKey>
<RequestDateTime>11/25/2014 2:38:50 PM</RequestDateTime>
</interface-response>
```




For More Information

For more advanced control and functionality over email services, see the following in our API Command catalog:

- `GetCatchAll` - Get the forwarding address for emails addressed to nonexistent mailboxes for a domain name.
- `SetCatchAll` - Set a forwarding address for any emails addressed to nonexistent mailboxes for a domain name.
- `GetPopForwarding` - Retrieve the email forwarding record, either POP or our email forwarding, for a specified mailbox name.
- `SetPopForwarding` - Create, change, or delete an email forwarding address for a domain name.
- `GetPopExpirations` - Retrieve a list of email expiration dates, including the domains they are associated with.