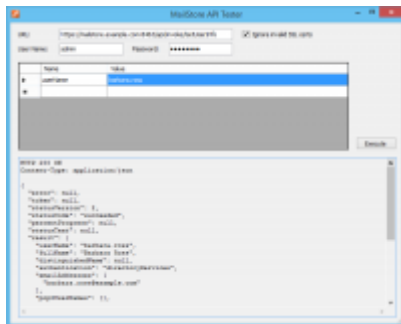


Administration API - Using the API

In this chapter you can find a description of the MailStore Server Administration API. With the Administration API administrative tasks, such as managing users or storage locations, can be remote controlled from a central location. Communication with the Administration API is done using web requests through HTTPS.

For security reasons, the MailStore Server Administration API is deactivated by default and has to be activated using the [MailStore Server Service Configuration](#).

MailStore API Test Application



To help you better understand the samples on this page, we provide a small tool that is able to send commands to the API HTTPS server and display its responses. Simply download the ZIP file provided below and run the included `MailStoreApiTester.exe`.

The download comes with the entire C# source code of the test application.

[Download MailStoreApiTester.zip](#)

Executing Regular API Functions

To execute regular Management API functions, send a HTTP POST request to the following URL:

```
https://<mailstore-server>:8463/api/invoke/<function>
```

With every request to the API, the username and password of a MailStore Administrator must be submitted following the [HTTP Basic Authentication](#) specification as described in [RFC 2617](#)

Example

HTTP Request

```
POST /api/invoke/GetServerInfo HTTP/1.1
Authorization: Basic YWRtaW46UGFzc3cwcmQ=
User-Agent: curl/7.35.0
Host: mailstore.example.com:8463
Accept: */*
Content-Length: 0
```

HTTP Response

```
HTTP/1.1 200 OK
Cache-Control: no-cache,private,no-store,must-revalidate,max-stale=0,post-check=0,pre-check=0
Pragma: no-cache
Content-Length: 251
Content-Type: application/json
Expires: Wed, 25 Jun 2014 14:13:43 GMT
Server: Microsoft-HTTPAPI/2.0
Date: Thu, 26 Jun 2014 14:13:43 GMT
```

```
{
  "error": null,
  "token": null,
  "statusVersion": 2,
  "statusCode": "succeeded",
  "percentProgress": null,
  "statusText": null,
  "result": {
    "version": "9.0.0.9702",
    "machineName": "MAILSTORE"
  },
  "logOutput": null
}
```

To get a list of all available functions, send a POST request to <https://<mailstore-server:8463/api/get-metadata>.

Long Running Processes

Some API functions start a long running process that may take several minutes or even hours to complete. To keep track of the progress a unique token is returned upon initial request, which can be used to periodically poll for status updates.

The overall process is as follows:

1. Send HTTP POST to `https://<mailstore-server>:8463/api/invoke/VerifyStore` in order to execute the function.
2. Check the server's response for the keys `token`, `statusCode` and `statusVersion`.
3. To retrieve status updates, send periodic requests to `https://<mailstore-server>:8463/api/get-status` with the following parameters:

Name	Description
<code>token</code>	The received unique token for the process.
<code>lastKnownStatusVersion</code>	Value of <code>statusVersion</code> from last response.
<code>millisecondsTimeout</code>	Time in milliseconds until the the server stops waiting for a new status update to send. If a new status update is available it will be sent to the client immediately, otherwise the last status is repeated.

4. The process finished when the `statusCode` is not *running*.

Example

Initial HTTP Request

```
POST /api/invoke/VerifyStore HTTP/1.1
Authorization: Basic YWRtaW46UGFzc3cwcmQ=
User-Agent: curl/7.35.0
Host: mailstore.example.com:8463
Accept: */*
Content-Length: 5
Content-Type: application/x-www-form-urlencoded
```

```
id=11
```

Initial HTTP Response

```
HTTP/1.1 200 OK
Cache-Control: no-cache,private,no-store,must-revalidate,max-stale=0,post-check=0,pre-check=0
Pragma: no-cache
Content-Length: 290
Content-Type: application/json
Expires: Wed, 25 Jun 2014 14:15:41 GMT
Server: Microsoft-HTTPAPI/2.0
```

Date: Thu, 26 Jun 2014 14:15:41 GMT

```
{
  "error": null,
  "token": "b3af0ebae4dd755d1a54b9756817baee",
  "statusVersion": 3,
  "statusCode": "running",
  "percentProgress": 0,
  "statusText": null,
  "result": null,
  "logOutput": "Verifying file group #11...\r\nCreating a list of messages to
be verified...\r\n"
}
```

Periodic HTTP Request of Progress

```
POST /api/get-status HTTP/1.1
Authorization: Basic YWRtaW46UGFzc3cwcmQ=
User-Agent: curl/7.35.0
Host: spe.example.com:8474
Accept: */*
Content-Length: 90
Content-Type: application/x-www-form-urlencoded
```

```
token=b3af0ebae4dd755d1a54b9756817baee&lastKnownStatusVersion=3&millisecondsT
imeout=5000
```

HTTP Response to Periodic Progress Requests

```
HTTP/1.1 200 OK
Cache-Control: no-cache,private,no-store,must-revalidate,max-stale=0,post-
check=0,pre-check=0
Pragma: no-cache
Content-Length: 242
Content-Type: application/json
Expires: Wed, 25 Jun 2014 14:08:15 GMT
Server: Microsoft-HTTPAPI/2.0
Date: Thu, 26 Jun 2014 14:08:15 GMT
```

```
{
  "error": null,
  "token": "b3af0ebae4dd755d1a54b9756817baee",
  "statusVersion": 9,
  "statusCode": "running",
```

```
"percentProgress": 0,  
"statusText": null,  
"result": null,  
"logOutput": " 400 messages verified...\r\n"  
}
```

Final HTTP Response

```
HTTP/1.1 200 OK  
Cache-Control: no-cache,private,no-store,must-revalidate,max-stale=0,post-check=0,pre-check=0  
Pragma: no-cache  
Content-Length: 242  
Content-Type: application/json  
Expires: Wed, 25 Jun 2014 15:08:15 GMT  
Server: Microsoft-HTTPAPI/2.0  
Date: Thu, 26 Jun 2014 15:08:15 GMT
```

```
{  
  "error": null,  
  "token": null,  
  "statusVersion": 269,  
  "statusCode": "succeeded",  
  "percentProgress": null,  
  "statusText": null,  
  "result": null,  
  "logOutput": null  
}
```

Error Handling

Additionally to handling protocol specific issues such as HTTP or TCP errors, developers receive detailed information from the API itself in case the request did not succeed.

In case the request was successfully be processed, the response will contain:

```
"error": null
```

Otherwise in the event of a failure the error key will have the following structure

```
"error": {  
  "message": string,  
  "details": string  
}
```

allowing developers to display a meaningful error message and further technical details.