



ORK  ATHEX
Order Record Keeping

Release Notes for
Version Pre-1.1
October 20 2017

- 1 Pre-Release Notes 1
 - 1.1 Purpose and intended audience of this document 1
 - 1.2 Data changes 1
 - 1.2.1 New fields 1
 - 1.2.2 References 1
 - 1.2.3 UI and API Compatibility for the new field 2
 - 1.3 Corrections and Fixes 3
 - 1.3.1 Headers in CSV files 3
 - 1.4 Documentation addenda and fixes 3
 - 1.4.1 Documentation Addendum: Pagination in the REST API 3
 - 1.4.2 Documentation Fix: REST Submission API and JSON data 3
 - 1.4.3 Documentation Addendum: sample cURL call and data for submission 4
 - 1.5 Version 1.1 REST methods 5
 - 1.6 The getSubmissions1_1 method 5
 - 1.7 The getMappings1_1 method 7
 - 1.8 Pagination in REST get methods 10

1 Release Notes

1.1 Purpose and intended audience of this document

This document is intended to accompany the **ORK@ATHEX** Version 1.0 technical specifications.

As part of its effort to enhance its services, AthexGroup has made several improvements and corrections to the v-1.0 implementation of the **ORK@ATHEX** service. As a result, a v-1.1 is going to be released soon.

During the current early release phase, there is no production **ORK@ATHEX** system. Thus, we are co-housing a pre-release of version 1.1 along with the existing 1.0. The co-housed deployment will last for a certain adaptation period, so as to give Athex Members who may have worked with the v-1.0 specs the chance to align to the changes of v-1.1. All new, v-1.1 specific APIs are labeled as such, while the original v-1.0 APIs remain in place.

After the adaptation period is over, obsolete 1.0 conventions, specs, functionality and APIs will be removed, and all APIs labeled as v-1.1 will be renamed to have the version-specific label removed.

This document details the changes, improvements and corrections to the v-1.0 **ORK@ATHEX** service.

1.2 Data changes

1.2.1 New fields

A new “Free Text” field has been added at the end of the mapping entries. Thus, the Short/Long mapping table in v-1.1 becomes as follows (the new field is marked in red):

Field	Values	JSON name	Reference
Entry ID	(system-generated) {INTEGER}	entryId	
Submission ID	(system-generated) {INTEGER}	submissionId	
Entry Status	(system-generated) {ALPHANUM-4}	entryStatus	
Entry Errors	{system-generated} {ALPHANUM-500}	entryErrors	
Report Status	One of 'NEWT', 'UPDT', 'AMND', or 'CANC' {ALPHANUM-4}	reportStatus	[ATHEXORK-1]
Valid From	{DATE} (in YYYYMMDD format)	validFm	[ATHEXORK-3]
Valid To	{DATE} (in YYYYMMDD format)	validTo	[ATHEXORK-4]
Code Type	{INTEGER-1}	codeType	[ATHEXORK-5]
Short Code	{INTEGER}	shortCode	[ATHEXORK-6]
Long Code	{ALPHANUM-50}	longCode	[ATHEXORK-7]
Free Text	{ALPHANUM-100}	freeText	{ATHEXORK-2}
Validity Status	(system-generated) {ALPHANUM-50}	validStatus	
Validity Date	(system-generated) {DATE}	validDate	

The new Free Text field is optional. Its intended use is to let Members supply arbitrary text of their own, in order to be able to distinguish various classes of mappings. Free Text can be used as a filter in the getMappings1_1 REST API call.

1.2.2 References

The field reference table becomes (notice that an ATHEXORK-2 reference existed in v1.0 specs; the new, modified ATHEXORK-2 reference is marked in red):

Reference	Data format	Values
ATHEXORK-1	{ALPHANUM-4}	'NEWT' – New 'UPDT' – Update 'AMND' – Amendment 'CANC' – Cancelation
ATHEXORK-2	{ALPHANUM-100}	User free-text field. Only alphanumerical characters and space are allowed.
ATHEXORK-3	{DATE}	YYYYMMDD
ATHEXORK-4	{DATE}	YYYYMMDD NULL: Valid forever
ATHEXORK-5	{INTEGER-1}	1 - Client Identification Code - LEI 2 - Client Identification Code - National ID 3 - Investment Decision within Firm – National ID 4 - Investment Decision within Firm – Algo ID 5 - Execution within Firm – National ID 6 - Execution within Firm – Algo ID 7 - Non Executing Broker – LEI
ATHEXORK-6	{INTEGER}	Short codes will be of numeric type with values between 0 and 2,147,483,647 ($2^{31}-1$). The values 0, 1, 2 and 3 are reserved for NONE, AGGR, PNAL, and CLIENT respectively. Therefore, valid short codes will be only numbers that are greater than 3 for every code type.
ATHEXORK-7	{ALPHANUM-50}	LEI, National ID or Algorithm ID

1.2.3 UI and API Compatibility for the new field

Data: The new v-1.1 Mappings format is compatible with that of the former v-1.0, in that entries uploaded with the v-1.0 CSV upload mechanism or REST API will have a blank Free Text.

Web UI: The Web UI displays the new “Free Text” field as a new column.

CSV Downloading: downloaded CSV files will contain the new “Free Text” column.

CSV Uploading: there is no separate v-1.1 CSV upload method. CSV files can be uploaded both in 6-column (v-1.0) and 7-columns (v-1.1) format (consult Section 1.3.1 later in this document for information about CSV headers).

REST Downloading: the v-1.0 API methods will not contain the JSON “freeText” field. The v-1.1 getMappings method will return the new “freeText” JSON object member (consult Section 1.5 later for further details).

REST Uploading: there is no separate v-1.1 REST upload method. The same v-1.0 REST method will now accept a “freeText” JSON object member. If “freeText” is absent, it will be left blank (consult Section 1.5 later for further details)

1.3 Corrections and Fixes

1.3.1 Headers in CSV files

V1.0 did not allow CSV headings. V1.1 now optionally accepts headers. Headers are optional, in that a CSV file without headers is accepted. There is no intention of making headers mandatory in future versions. The intention behind enabling headers is to allow headers in CSV files so as to help manual editing and human comprehension of CSV file contents and not to fail machine-generated CSV files that do not contain them.

Headers are only allowed in the first row and must not be repeated anywhere in the CSV file.

Only the predefined headers are accepted. Supplying different headers will result in an attempt to parse the header as data, which will emit an error message.

Both v-1.0 (six columns) and v-1.1 (seven columns) headers are supported.

The recognized header lines are as follows (header text is case-sensitive):

```
"reportStatus";"validFrom";"validTo";"codeType";"shortCode";"longCode"
reportStatus;validFrom;validTo;codeType;shortCode;longCode
"reportStatus";"validFrom";"validTo";"codeType";"shortCode";"longCode";"freeText"
reportStatus;validFrom;validTo;codeType;shortCode;longCode;freeText
```

1.4 Documentation addenda and fixes

1.4.1 Documentation Addendum: Pagination in the REST API

Due to the fact that the REST API allows returning massive volumes of data, it may happen that some REST GET requests return a large number of results. To protect against such requests clogging the Athex Members Portal infrastructure, the REST API supports “pagination”, in that a “page” with a limited number of results is returned and URL pointers for browsing through other “pages” of results (first, next, previous) are supplied.

The Pagination functionality exists since v-1.0, only it was not well documented in the v-1.0 Specifications Document. For v-1.1, the Pagination functionality and variables are documented extensively in Section 1.8 below.

1.4.2 Documentation Fix: REST Submission API and JSON data

The v-1.0 documentation contained an inaccurate URL for the submission REST API method. The correct URL is:

https://t-amp.athexgroup.gr/amp/ork/submit_data

Moreover, the v-1.0 documentation contained an inaccurate syntax for JSON submission data. The correct submission data JSON syntax (including the new “Free Text” field) is as follows:

```
[
  {
    "reportStatus" : reportStatus1 ,
    "validFrom" : validFrom1 ,
    "validTo" : validTo1 ,
    "codeType" : codeType1 ,
    "shortCode" : shortCode1 ,
    "longCode" : longCode1 ,
    "freeText" : freeText1
  },
  {
    "reportStatus" : reportStatus2 ,
    "validFrom" : validFrom2 ,
    "validTo" : validTo2 ,
    "codeType" : codeType2 ,
    "shortCode" : shortCode2 ,
    "longCode" : longCode2 ,
    "freeText" : freeText2
  }
  ...
]
```

Please note that “validTo” and “freeText” fields may be absent, since they are optional.

1.4.3 Documentation Addendum: sample cURL call and data for submission

The following cURL command may be used to submit a data file “data.json” (output from cURL and sample system response are included):

```
$ curl -k -i -u 'user@domain.gr:password' -X POST -H 'Content-Type:
application/json' --data-binary @data.json 'https://t-
amp.athexgroup.gr/amp/ork/submit_data/'

HTTP/1.1 200 OK
Date: Thu, 19 Oct 2017 10:04:40 GMT
Server: Apache
Cache-Control: no-cache
Pragma: no-cache
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

{"submissionId" : "9", "submissionResult" : "Success"}
```

The submission id is returned in the submissionId variable, so it can be used in further inquiries regarding the status of this submission.

The sample file “data.json” has the following contents:

```
[
  {
    "reportStatus" : "NEWT",
    "validFrom" : "20171009",
    "validTo" : "20171009",
    "codeType" : "1",
    "shortCode" : "445566",
    "longCode" : "DRE7863689WEDF98DFG89"
  }, {
    "reportStatus" : "NEWT",
    "validFrom" : "20171009",
    "validTo" : "",
    "codeType" : "2",
    "shortCode" : "3",
    "longCode" : "PAPADDIMIT23023567",
    "freeText" : "customer label 1",
  }
]
```

1.5 Version 1.1 REST methods

Note: the documentation in the current section is new to version 1.1 and is only applicable to v-1.1.

Version Pre-1.1 introduces a different API for the per-existing methods “getSubmissions” and “getMappings”. The v-1.1 methods are thus labeled as “getSubmissions1_1” and “getMappings1_1”, to distinguish them from v-1.0 methods. When v-1.1 is finally deployed, the names of these methods will change to getSubmissions and getMappings, respectively.

1.6 The getSubmissions1_1 method

The getSubmissions1_1 REST API method returns submissions. Without any URL parameters, the getSubmissions1_1 method returns all the submissions that the user can access. The following example using cURL shows two submissions being returned.

For each submission, a JSON object is returned including the following member variables:

- **submissionId**: the submission id
- **submissionTimestamp**: the submission timestamp, in YYYYMMDDHH₂₄mmSS format
- **submissionChannel**: the submission channel; will be 1 for CSV submissions, and 2 for REST
- **submissionUser**: the user who posted the submission (only submissions from users having the same OASIS member code are returned; however, users with the same OASIS member code can view one another’s submissions)
- **submissionStatus**: the submission status (one of RCVD, PART, or ACPT)
- **submissionMappingsURL**: a URL for accessing the entries of this submission via the REST API (please notice that JSON provides for escaping the forward slash in URLs, by prefixing them with a backslash character).

At the end of the list, the Pagination results, parameters and URLs are returned. Please refer to Section 1.8 for details on Pagination.

```
$ curl -k -i -u 'user@domain.gr' -X GET -H 'Content-Type: application/json'
'https://t-amp.athexgroup.gr/amp/ork/getSubmissions1_1/'
HTTP/1.1 200 OK
[HTTP headers removed for brevity]

{
  "submissions": [
    {
      "submissionId": 9
      , "submissionTimestamp": "20171019130440"
      , "submissionChannel": 2
      , "submissionUser": "user1@domain.gr"
      , "submissionStatus": "PART"
      , "submissionMappingsURL": "https://t-amp.athexgroup.gr/amp/ork/getMappings1_1?submissionId=9"
    }
    , {
      "submissionId": 10
      , "submissionTimestamp": "20171019134016"
      , "submissionChannel": 2
      , "submissionUser": "user2@domain.gr"
      , "submissionStatus": "PART"
      , "submissionMappingsURL": "https://t-amp.athexgroup.gr/amp/ork/getMappings1_1?submissionId=10"
    }
  ]
  , "hasMore": false
  , "resultsPerPage": 25
  , "resultsStartOffset": 0
  , "resultsCount": 2
  , "links": [
    {
      "rel": "self"
      , "href": "https://t-amp.athexgroup.gr/amp/ork/getSubmissions1_1?"
    }
    , {
      "rel": "first"
      , "href": "https://t-amp.athexgroup.gr/amp/ork/getSubmissions1_1?"
    }
  ]
}
```

The following URL parameters can be passed to filter the set of submissions returned:

- **submissionId=number** : returns data only for a specific submission with id equal to *number*
- **submissionDateFrom=YYYYMMDD**: returns data for submissions posted on date YYYYMMDD or later
- **submissionDateTo=YYYYMMDD**: returns data for submissions posted on date YYYYMMDD or earlier

Notice how `submissionDateFrom` and `submissionDateTo` can be both specified to denote a time interval; if both these parameters are set to the same date, only submissions of that date will be returned

- `submissionUser=user`: returns only data for submissions posted by user `user`
- `submissionStatus=status` returns only data for submissions with status `status`

Here is an example of filtering results by passing a `submissionId` parameter:

```
$ curl -k -i -u 'user@domain.gr:password' -X GET -H 'Content-Type: application/json' 'https://t-amp.athexgroup.gr/amp/ork/getSubmissions1_1/?submissionId=9'
HTTP/1.1 200 OK
[HTTP headers removed for brevity]

{
  "submissions": [
    {
      "submissionId": 9
      , "submissionTimestamp": "20171019130440"
      , "submissionChannel": 2
      , "submissionUser": "user1@domain.gr"
      , "submissionStatus": "PART"
      , "submissionMappingsURL": "https://t-amp.athexgroup.gr/amp/ork/getMappings1_1?submissionId=9"
    }
  ]
  , "hasMore": false
  , "resultsPerPage": 25
  , "resultsStartOffset": 0
  , "resultsCount": 1
  , "links": [
    {
      "rel": "self"
      , "href": "https://t-amp.athexgroup.gr/amp/ork/getSubmissions1_1?"
    }
    , {
      "rel": "first"
      , "href": "https://t-amp.athexgroup.gr/amp/ork/getSubmissions1_1?"
    }
  ]
}
```

1.7 The `getMappings1_1` method

The `getMappings1_1` method returns mappings stored in **ORK@ATHEX**. Without any URL parameters, the `getMappings1_1` returns all the mappings that the user can access.

For each mapping, a JSON object is returned that contains the following member variables:

- `entryId`: the Entry Id for this mapping; this can be used to uniquely identify the mapping
- `submissionId`: the system-generated Submission Id of the relevant submission
- `entryStatus`: the system-generated Entry Status for this mapping

- **entryErrors**: any system-generated errors that may have occurred during the validation of this mapping; this member variable can be null or missing if no errors have occurred; moreover, it may contain informational messages, so it should not be used as an indication of error
- **reportStatus**: the Report Status that the user has specified when uploading or posting this mapping.
- **memberCode**: the system-generated Member Code that is derived from the user id of the submitting user
- **validFrom**: the first day of the validity period for this mapping
- **validTo**: the last day of the validity period for this mapping, or null, if the mapping is valid forever after its first day of validity
- **codeType**: the Code Type of this mapping
- **shortCode**: the Short Code of this mapping
- **longCode**: the Long Code of this mapping
- **validationDate**: the system-generated last date when this mapping was used by the internal ORK system; this will be null if the mapping was never used by the internal ORK
- **freeText**: the Free Text that the user has specified (may be null)

Moreover, an optional **addSubmissionData** URL parameter can be added to the REST request with a value of true, to instruct that submission data be included in mappings. If **addSubmissionData** is present and is set to true, each mapping returned contains an additional **submissionData** member sub-object, which contains the member variables **submissionId**, **submissionTimestamp**, **submissionChannel**, **submissionUser** and **submissionStatus**, whose meaning and content is identical as the respective member variables outlined in Section 1.6.

An example call for **getMappings** using cURL is:

```
curl -k -i -u 'user@domain.gr:password' -X GET 'https://t-amp.athexgroup.gr/amp/ork/getMappings1_1/'
```

An example output containing just a single mapping is shown below (notice the Pagination parameters in the end of the reply). In most real-world uses, the **mappings** array is expected to contain more elements.

```

HTTP/1.1 200 OK
[HTTP headers removed for brevity]

{
  "mappings": [
    {
      "entryId": 34
      , "submissionId": 10
      , "entryStatus": "FVLD"
      , "entryErrors": "validation error: \u0027AMND\u0027 entry does not match any
appropriate mapping\n"
      , "reportStatus": "AMND"
      , "memberCode": "UU"
      , "validFrom": "20171001"
      , "validTo": "20171020"
      , "codeType": 2
      , "shortCode": 4
      , "longCode": "THEOFILOPO23024551"
      , "validationDate": null
      , "freeText": "user free text"
      , "submissionData": {
        "submissionId": 10
        , "submissionTimestamp": "20171019134016"
        , "submissionChannel": 2
        , "submissionUser": user@domain.gr
        , "submissionStatus": "PART"
      }
    }
  ]
  , "hasMore": false
  , "resultsPerPage": 25
  , "resultsStartOffset": 0
  , "resultsCount": 1
  , "links": [
    {
      "rel": "self"
      , "href": "https://\t-
amp.athexgroup.gr\amp\ork\getMappings1_1?addSubmissionData=true"
    }
    , {
      "rel": "first"
      , "href": "https://\t-
amp.athexgroup.gr\amp\ork\getMappings1_1?addSubmissionData=true"
    }
  ]
}

```

The following URL parameters can be passed to filter the set of submissions returned:

- **entryId=number** : returns data only for a single mapping with Entry Id equal to *number*.
- **submissionId=number** : returns data only for mappings uploaded in a specific submission with Submission Id equal to *number*.

- **entryStatus=status1[,status2...]** : filters the results so that only mapping for the specified subset of Entry Status values are returned (notice that the comma character may have to be escaped depending on the client software)
- **reportStatus=status1[,status2...]** : filters the results so that only mapping for the specified subset of Report Status values are returned (notice that the comma character may have to be escaped depending on the client software)
(Notice that there is no filter on Member Code, since every user can only view mappings for the Member Code associated with that user's account)
- **validFrom=YYYYMMDD** : filters the returned results so that only mappings valid at date *YYYYMMDD* or earlier are returned ¹
- **validTo=YYYYMMDD|forever** : filters the returned results so that only mappings not having expired before *YYYYMMDD* are returned ²; the value **forever** limits the results so that only mappings that never expire are returned
- **codeType=number** : filters the returned results so that only mappings with Code Type equal to *number* are returned
- **shortCode=number** : filters the returned results so that only mappings with Short Code equal to *number* are returned
- **longCode=longCode** : filters the returned results so that only mappings with Long Code equal to *longCode* are returned
- **validationDateFrom=YYYYMMDD|never** : filters the returned results so that only mappings used by the internal ORK on or after *YYYYMMDD* are returned; the value **never** limits the results so that only mappings never used by the internal ORK are returned
- **validationDateTo=YYYYMMDD** : filters the returned results so that only mappings used by the internal ORK on or before *YYYYMMDD* are returned
- **freeText=text** : filters the returned results so that only mappings with Free Text equal to *text* are returned (notice: this tests only equality, and full text search is not supported)

1.8 [Pagination in REST get methods](#)

As mentioned before, the REST methods do not return the full set of results at once; instead, they return “pages” of results. Each page has an offset, which is the number of results before the first result in this page. The offset of the first page is thus zero.

¹ Notice that this includes *more mappings* than those having a Valid From value *exactly equal* to *YYYYMMDD*. Another way of expressing this constraint is that **validFrom** limits the results to the mappings whose validity period started on or before *YYYYMMDD*. However, please notice that this filter alone does not guarantee that the mappings returned are still valid at *YYYYMMDD*: a combination of **validFrom** and **validTo** is required for this purpose. Finally, “valid” and “validity” here does not mean that the returned mappings are or were actually valid; they may have failed validation or be superseded by updates, so more accurate filtering is required if the actual valid mappings are to be retrieved.

² Notice that this includes *more mappings* than those having a Valid To value *exactly equal* to *YYYYMMDD*. Another way of expressing this constraint is that **validTo** limits the results to the mappings whose validity period ends on or after *YYYYMMDD*, including these mappings that have an indefinite (blank) end date. However, please notice this filter alone does not guarantee that the mapping is valid at *YYYYMMDD*: a combination of **validFrom** and **validTo** is required for this purpose. Moreover, “valid” and “validity” here does not mean that the returned mappings are or were actually valid; they may have failed validation or be superseded by updates, so more accurate filtering is required if the actual valid mappings are to be retrieved.

Pagination works by producing pages of identical size. Thus, Pagination in `getSubmissions1_1` and `getMappings1_1` is controlled by two parameters:

- `resultsPerPage=number`: this parameter defines the number of results per page
- `resultsStartOffset=number`: this parameter indicates the starting offset (as defined above)

There is an implementation upper limit to `resultsPerPage`, so that no excessive stress is placed on the AMP infrastructure by abusive values of this parameter.

At the end of each reply, a `hasMore` flag, along with the pagination parameters are provided, including links for the same page, the first page, and for the next page (if a next page exists). For example:

```

. . .
,"hasMore":true
,"resultsPerPage":3
,"resultsStartOffset":0
,"resultsCount":3
,"links":[
{
"rel":"self"
,"href":"https://t-
amp.athexgroup.gr/amp/ork/getMappings1_1?resultsPerPage=3"
}
,{
"rel":"first"
,"href":"https://t-
amp.athexgroup.gr/amp/ork/getMappings1_1?resultsPerPage=3"
}
,{
"rel":"next"
,"href":"https://t-
amp.athexgroup.gr/amp/ork/getMappings1_1?resultsPerPage=3\u0026resultSta
rtOffset=3"
}

```