

soapgate Q! 5.5

Bug Fixes

Minor fixes and improvements since version 5.0

New Features

Client controlled caching - for smaller data sets (64K limit).

Consider the following scenario: a corporate application for a smart device is accessing a Domino server to withdraw some data from a view. Unless some kind of caching mechanism is in place, multiple devices requesting the data from the same database and view, will cause the Domino server to multiple times generate the same result set.

soapgateQ! 5.5 provides a caching feature to avoid this situation. This feature is entirely controlled by the client. If the request is send with the correct parameters, the server will cache the result-set until a new request comes in stating explicitly not to cache. A request to cache will also force any further request sent by other devices to be put on hold and answered only once the cache is built.

Using this feature, multiple devices can send requests all at the same time, but the server will only process the first request, collect the data, build the cache and only then answer all request from the cache. This resulting in high performance gains.

All view or folder related RESTful or SOAP operations provide the CACHE parameter. The way the cache parameter is to be used in a client application is by creating a string that contains a prefix related to the view accessed and a date time string in reverse format. For instance:

BOOKS.FLAT.201703011600 (Prefix.yyyymmddhhnn)

Depending on how long the cache should remain valid before the next request is processed accessing the most current data, the last n characters of the cache string should be modified accordingly. If the cache should be active for let's say 15 minutes, the last two character should be changed once 15 minutes passed...

BOOKS.FLAT.201703011615 (Prefix.yyyymmddhhnn)

The prefix is required to make the cache string unique and keep it linked to a particular view in a particular Domino database. There are no rules on what prefix to use, just make it unique.

If the server receives a request with a cache string, it will look up if there is a cache entry and return the result set if there is. Otherwise it will lock the cache string until a result set is created and all other requests with the same cache string must wait until the result set for the cache entry is created.

Try the request below against our test server (use two different browser to avoid browser caching). You might not notice much of a difference as the test data-set is pretty small. However, we have this feature implemented in a real world scenario where up to 50 iOS devices hit the server in a 10-15 minute time period to synchronise client records and we have reduced the response time from 2+ minutes down to 15-30 seconds for reading about 10 thousand clients.

```
https://domino.flexdomino.net/SOAPGATEQ_5.NSF/REST4ViewsFolders?openagent&JSONPARAM={
  "jsonparam": {
    "OPERATION": "dbviewentries",
    "CACHE": "BOOKS.FLAT.201703011600",
    "SRVNAME": "flexdomino/flex2domino",
    "DBNAME": "flex/flexdemo.nsf",
    "VIEWNAME": "books.flat"
  }
}
```

Batch processing

The REST4Documents REST service to read and update individual documents now has a new operation called DbBatchProcess. This allows transmitting multiple operation requests in one REST call, thereby improving performance of REST operations. So it is possible to read, amend or create multiple documents in one call.

The DbBatchProcess operation has only one parameter, a list of tags. At the end of the main REST construct to call the DbBatchProcess operation, additional REST constructs for further operations can be added. Additional operations must be part of the REST4Documents list of operations. Each additional call must begin with one of the tags and end with the same tag.

There are no rules towards the format of the tags. However, tags must be unique within the entire message send to the server.

https://domino.flexdomino.net/SOAPGATEQ_5.NSF/REST4Documents?openagent&JSONPARAM={

```
  "jsonparam": {  
    "OPERATION": "dbbatchprocess",  
    "PROCESSTAGS": [{"tag-00001}","tag-00002}","tag-00003}"]  
  }  
}
```

tag-00001}

```
{  
  "jsonparam": {"OPERATION": "dbsavedocfields", "SRVNAME": "server/org", "DBNAME": "requests.nsf", "DOCUNID": "", "FORM": "OnlineCalcRQST", "FIELDS": ["ArrivalDate", "DepartureDate", "CourseStartDate", "Weeks", "CourseUniversalID", "ResidenceUniversalID", "HotelUniversalID", "Supplements", "Visa", "Airport", "DiscountCode"], "TYPES": ["1024", "1024", "1024", "768", "1280", "1280", "1280", "1280", "768", "768", "1280"], "VALUES": ["2017-03-12", "2017-03-25", "2017-03-13", "2", "2F780A45B001394DC1257EEC00561965", "F0A41663A0B724A1C1257EED00432A30", "", "Single", "Yes", "Yes", "CB2017"], "READFIELDS": ["SUMMARY", "PRICE", "TOTALDISCOUNT", "ECOTAXDAYS"], "COMPWF": false, "SAVERPC": "$OnlineCalculatorRQST"}  
}
```

tag-00001}

tag-00002}

```
{  
  "jsonparam": {"OPERATION": "dbsavedocfields", "SRVNAME": "server/org", "DBNAME": "requests.nsf", "DOCUNID": "", "FORM": "OnlineCalcRQST", "FIELDS": ["ArrivalDate", "DepartureDate", "CourseStartDate", "Weeks", "CourseUniversalID", "ResidenceUniversalID", "HotelUniversalID", "Supplements", "Visa", "Airport", "DiscountCode"], "TYPES": ["1024", "1024", "1024", "768", "1280", "1280", "1280", "1280", "768", "768", "1280"], "VALUES": ["2017-04-14", "2017-04-27", "2017-04-15", "2", "2F780A45B001394DC1257EEC00561965", "F0A41663A0B724A1C1257EED00432A30", "", "Single", "Yes", "Yes", "CB2017"], "READFIELDS": ["SUMMARY", "PRICE", "TOTALDISCOUNT", "ECOTAXDAYS"], "COMPWF": false, "SAVERPC": "$OnlineCalculatorRQST"}  
}
```

tag-00002}

tag-00003}

```
{  
  "jsonparam": {"OPERATION": "dbsavedocfields", "SRVNAME": "server/org", "DBNAME": "requests.nsf", "DOCUNID": "", "FORM": "OnlineCalcRQST", "FIELDS": ["ArrivalDate", "DepartureDate", "CourseStartDate", "Weeks", "CourseUniversalID", "ResidenceUniversalID", "HotelUniversalID", "Supplements", "Visa", "Airport", "DiscountCode"], "TYPES": ["1024", "1024", "1024", "768", "1280", "1280", "1280", "1280", "768", "768", "1280"], "VALUES": ["2017-04-14", "2017-04-27", "2017-04-15", "2", "2F780A45B001394DC1257EEC00561965", "F0A41663A0B724A1C1257EED00432A30", "", "Single", "Yes", "Yes", "CB2017"], "READFIELDS": ["SUMMARY", "PRICE", "TOTALDISCOUNT", "ECOTAXDAYS"], "COMPWF": false, "SAVERPC": "$OnlineCalculatorRQST"}  
}
```

eUniversalID","ResidenceUniversalID","HotelUniversalID","Supplements","Visa","Airport","DiscountCode"],"TYPES":["1024","1024","1024","768","1280","1280","1280","1280","768","768","1280"],"VALUES":["2017-05-10","2017-05-20","2017-05-11",2,"2F780A45B001394DC1257EEC00561965","F0A41663A0B724A1C1257EED00432A30","", "Single","Yes","Yes","CB2017"],"READFIELDS":["SUMMARY","PRICE","TOTALDISCOUNT","ECOTAXDAYS"],"COMPWF":false,"SAVERPC":"\$OnlineCalculatorRQST"}}

{{tag-00003}}

Extended Data Sets

Similar to the batch processing tags, there are a number of predefined tag-names that can be used to separate large data sets (for instance binary data) from the JSON construct for the REST operation. These can only be used for fields of type Richtext (1).

None Encoded Data Block – {[DATA]:tag}

This special tag can be used to transmit for instance longer HTML content to be stored in a Richtext fields. Content stored in None Encoded Data Blocks are not parsed as part of the JSON construct and therefore do not require any encoding to make it JSON parser conformative.

```
https://domino.flexdomino.net/SOAPGATEQ_5.NSF/REST4Documents?openagent&JSONPARAM={
{"jsonparam":{"OPERATION":"dbsavedocfields","SRVNAME":"server/org","DBNAME":"requests.nsf","DOCUNI
D":"","FORM":"HtmlForm","FIELDS":["Body"],"TYPES":["1"],"VALUES":["{[DATA]:{data-
tag_0001}"],"READFIELDS":[""],"COMPWF":false,"SAVERPC":""}}
{{EXTENDED-DATA}}
{data-tag_0001}
any string (for instance HTML)
{data-tag_0001}
```

Base64 Encoded Data Block – {Base64:tag:KEEPRAW}

With this special tag it is possible to transmit files as Base64 encoded data and attach them to a Richtext field. The tag name must be a valid filename (no path). The optional parameter :KEEPRAW allows to keep the Base64 encoded string in addition to the attachment.

```
https://domino.flexdomino.net/SOAPGATEQ_5.NSF/REST4Documents?openagent&JSONPARAM={
{"jsonparam":{"OPERATION":"dbsavedocfields","SRVNAME":"server/org","DBNAME":"requests.nsf","DOCUNI
D":"","FORM":"HtmlForm","FIELDS":["Body"],"TYPES":["1"],"VALUES":["{Base64:image.jpg}"],"READFIELDS":["
"],"COMPWF":false,"SAVERPC":""}}
{{EXTENDED-DATA}}
{image.jpg}
for instance Base64 encoded image data
{image.jpg}
```