E-mail To SMS Admin Rest Web Service

LEKAB Communication Systems AB

Version 7.0.36, 2026-11-02

E-mail To SMS Admin Rest Web Service

Introduction . . . 1
Different authentication methods available for requests . . . 2
1. The `/set` endpoint . . . 3
1.1. GET request example . . . 3
1.2. POST request example . . . 3
1.2.1. Explanation of parameters for /set . . . 3
1.2.2. HTTP response to /set . . . 5
1.2.3. Explanation of response to /set . . . 6
1.2.4. Example Python 3 code . . . 6
2. The `/unset` endpoint . . . 8
2.1. GET request example . . . 8
2.2. DELETE request example . . . 8
2.3. POST request example . . . 8
2.3.1. Explanation of parameters for /unset . . . 8
2.3.2. HTTP response to /unset . . . 9
2.3.3. Explanation of response to /unset . . . 10
2.3.4. Example Python 3 code . . . 10
3. The `/default` endpoint . . . 11
3.1. GET request example . . . 11
3.2. POST request example . . . 11
3.2.1. Explanation of parameters for /default . . . 11
3.2.2. HTTP response to /default . . . 13
3.2.3. Explanation of response to /default . . . 14
3.2.4. Example Python 3 code . . . 14
4. The `/querydomains` endpoint . . . 16
4.1. GET request example . . . 16
4.2. POST request example . . . 16
4.2.1. Explanation of parameters for /querydomains . . . 16
4.2.2. HTTP response to /querydomains . . . 17
4.2.3. Explanation of response to /querydomains . . . 17

Introduction

© 2026 LEKAB Communication Systems AB. Version 7.0.36, 2026-11-02.

This is a Web Service using `HTTP GET` or `HTTP POST` or `HTTP DELETE` requests for administration of the list of email addresses that are allowed or explicitly disallowed to send SMS by sending an email to an email address of the type "46701234567@sms.lekab.com" where the number is the phone number of the SMS recipient and the domain is an agreed upon domain that is forwarded to our special email server with SMS sending capability.

It is possible for a company (e.g. "Big Co") to agree with us that an internal email domain be used, like "46701234567@sms.bigco.com", as long as the internal email server forwards these emails to our special email server and we put the domain on an allowed list and open the appropriate firewalls etc. That part of the administration cannot be accessed from this web service, which is all about which sending email addresses "john.smith@bigco.com" or "sales@stockholm.bigco.com" are allowed to send SMS via this service, with which additional parameters, and who is paying for their SMS traffic.

The intended users are manual or automatic company administrators of SMS traffic. A registered user in our system must have either the company administration role, or a special email to SMS admin role to be allowed to edit the list. The special role is aimed at automatic functionalities that track personnel updates and allow and remove SMS sending privileges. That way the automatic functionality user can only access this list of allowed and disallowed email users, and not log in and make other changes.

An email sender address is allowed to send SMS this way, if it is on the list and not blocked. It is also possible to allow entire email domains (the part after the @) to send SMS. When an email arrives from name@domain, first the table is checked for the full combination "name@domain". If there is such an entry, it is checked if the blocked flag is set. If the email is not blocked, the SMS can be sent, using the parameters from the table. If there is no such entry, the table is checked for "@domain", that is the domain only. If that is not blocked, the SMS can be sent. So an email address can be allowed because it is in the table, or if its domain is in the table. If both exist, the full email address takes precedence.

There are four main endpoints: `/set` and `/unset`, `/default` and `/querydomains`.

Each of the endpoints supports the same function with `GET` and `POST`, but in the `GET` case, parameters are given in the calling `URL` (after a `?` sign, separated by `&` signs) , while in the `POST` case the parameters are given in a json document in the `HTTP POST` request body. `UTF-8` encoding is assumed in all `HTTP` bodies. The `/unset` endpoint also allows `DELETE` calls, with parameters in the `URL`.

`GET`, `POST` and `DELETE` all return responses in the `HTTP` response body as a json document.

The format of the input and output json documents and the input url parameters are described below.

Different authentication methods available for requests

Every request to the service must include authentication, i.e. username and password (or equivalent). For POST requests these can be given in in the corresponding fields in the JSON document which is sent in the (automatically HTTPS = SSL/TLS encrypted) HTTP body. Since GET and DELETE requests cannot have a body, instead username and password can be sent in the U and P url parameters. Note that everything in the URL after the host name is also part of the encrypted request, so url parameters are as safe during transfer as parameters in the body.

We also offer three different alternative ways of supplying these username and password credentials available in all GET, POST and DELETE cases:

1. Username and password can be given as standard Basic authentication, in which the header Authorization should have the value Basic + token, where the token is the Base64 encoding of (a UTF-8 byte array representation of) username:password. Here testuser:testpass will be encoded as dGVzdHVzZXI6dGVzdHBhc3M=
2. Username and password can be given in the HTTP headers, X-Lekab-Userid and X-Lekab-Password, respectively. The values have to be the Base64 encoding of (a UTF-8 byte array representation of) the username or password to allow non-US-ASCII characters. Here testuser will be encoded as dGVzdHVzZXI= and testpass as dGVzdHBhc3M=
3. An API key generated in the Web Portal can be used as a query parameter key, as the value of the header X-API-Key or as the field apikey in the POST request body. The length of the key varies depending on the length of the username (which is contained within the key). TUxVOmRHVnpkSFZ6WlhJOjlKUUczTXU2TVZVU1Exd3Y is a possible example key for the username testuser. The key is independent of the account password.
4. Oauth authentication is available from a separate endpoint By supplying the username and password as Basic authentication to the /token endpoint, in an HTTP POST call of type application/x-www-form-urlencoded containing the body "grant_type=client_credentials", one obtains a token valid for 10 minutes. This token can then be supplied as a Bearer token in the HTTP request, so that the header Authorization shall have the value Bearer + one space + token. More information about this in the **AuthenticationAPI** documentation.

If any of the alternative methods of authentication are used, parameter values pertaining to other authentication methods should be omitted or set to the empty string "".

Chapter 1. The /set endpoint

Used for setting or explicitly blocking an email address in the company as an email to SMS user.

1.1. GET request example

e.g. from web browser or curl. Note that url query parameters must be url-encoded (from UTF-8) if they contain characters that are not allowed in a url. Space can be represented as "+".

```
curl
https://secure.lekab.com/restsms/api/emailtosms/set?U=testuser&P=testpass&EM=john.doe@bigco.com&SD=J+DOE&CM=Y
```

1.2. POST request example

probably from an application

```
https://secure.lekab.com/restsms/api/emailtosms/set
```

With the contents of the HTTP body:

```
{"username":"testuser","password":"testpass",
 "email":"john.doe@bigco.com","sender":"J DOE","checkMobile":true}
```

1.2.1. Explanation of parameters for /set

In the following table, the parameters marked CD are optional and if not given, the C ompany D efault is used. This allows for one-time setting of company default parameters, followed by automatic additions upon staffing changes by just stating the email address (and proper admin authorization).

POST json key	GET query param	json value (strings quoted)	query param value (strings without quotes)	
email	EM	string	string	email sender to be allowed or blocked from SMS sending
blocked	BL	boolean (default false)	T, TRUE, Y or YES	explicitly block this email address
payinguser CD	PU	string	string	The user account that will become owner of sent messages, digits for id or login username

POST json key	GET query param	json value (strings quoted)	query param value (strings without quotes)	
sender CD	SD	string	string	SMS sender id 3 to 11 chars
costCenter CD	CC	string	string	Label for grouping on bill
defaultCountryCode CD	DC	string of digits	string of digits	Replaces leading zero in numbers starting with single zero
maxSms CD	MS	digit string for integer	digit string for integer	max SMS parts per message max 255
checkMobile CD	CM	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Only send if Google library has recipient number format in a possible official mobile number series for the country
includeSubject CD	IS	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Include the email subject in the SMS text
onlyNonDelivery CD	ND	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Only callback failed delivery receipts
origMsg CD	OM	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Include original message with answer callback
splitCallback CD	SC	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Receipts and answers callback to sending email - needs SMTPSPLIT setting in paying user profile
senderLogic CD	SL	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	If your company has an agreement to use several incoming domains, use special company rules determining SMS sender id from incoming domain
username	U	string	string	username of the Email to SMS admin account
password	P	string	string	password of the Email to SMS admin account
apikey	key	string	string	API key of the Email to SMS admin account

The email parameter and allowed domains

Email can be either a single email sender "john.doe@bigco.com" or an entire domain "@bigco.com". For every company there is a list of email domain endings that can be used in the email addresses cleared for email to SMS by that company. In the example, "bigco.com" is probably on the list. In that case "bigco.com", "sales.bigco.com" and even

"longendinginbigco.com" will be allowed for the company, but e.g. "gmail.com" or "smallco.com" will not be allowed. The allowed domains list cannot be edited from this web service, contact our support to add one of your own email domains.

The payinguser parameter

By specifying the payinguser, your company acknowledges that email to SMS from the given email address will be paid for by your company. Only an existing active SMS sending user account within your company can be set as the payinguser. The payinguser should have the right to send SMS and an appropriate (or unlimited) quota, or no SMS will be sent. If the user calling this web service is a special limited email to SMS admin user, the payinguser should probably be another account, or the admin user should also be set up for sending SMS. Probably the payinguser will be specified in the company default if many email addresses are sharing a payinguser. If individual SMS sending user accounts are set up for each person, the accounts should be set up first, and this web service called with the appropriate payinguser parameter for each email afterwards. This parameter can be specified either as the user account id "12345" or as the username "bigco-sales".

1.2.2. HTTP response to /set

A successful request will return 200 OK and a json document of the following format. The `result` field will have the value "OK" for a successful request. The `Content-Type` header of the response is `application/json` for all responses.

```
{
  "result" : "OK",
  "email" : "john.doe@bigco.com",
  "blocked" : false,
  "payinguserId" : "12345",
  "payinguserName" : "bigco-sales",
  "sender" : "J DOE",
  "costCenter" : "",
  "defaultCountrycode" : "46",
  "maxSms" : 12,
  "checkMobile" : true,
  "includeSubject" : false,
  "onlyNonDelivery" : false,
  "origMsg" : true,
  "splitCallback" : false,
  "senderLogic" : false,
  "allowedDomainEndings" : [ "bigco.com" ]
}
```

The values of the parameters not given in the call are taken from the company defaults, which can be set either by a company admin in the web portal or by a call to the `/default` endpoint in this web service.

A failed request will have a different HTTP result code from 200 OK, and the result will be

"ERROR". For example, if the wrong login credentials are given, the result code is 401 and the response body is:

```
{"result":"ERROR","error":"Unauthorized"}
```

1.2.3. Explanation of response to /set

json key	json value (strings quoted)	
result	string	OK or ERROR
email	string	Email sender for which email to SMS has been set
blocked	boolean	If true, the sender is blocked from email to SMS
payinguserId	string of digits	user id of SMS sending account
payinguserName	string	login username of SMS sending account
sender	string	The SMS comes "from" this sender id
costCenter	string	For grouping on bill
defaultCountrycode	string of digits	Country code to replace phone number initial zero
maxSms	integer max 255	SMS cannot be longer than this number of SMS parts
checkMobile	boolean	Send only if number is possible mobile number per Google library
includeSubject	boolean	Include email subject in SMS text
onlyNonDelivery	boolean	Callback receipt only if failed SMS delivery
origMsg	boolean	Include original message in answer callback
splitCallback	boolean	Remember sender email and callback receipts and answers to sender
senderLogic	boolean	Only for companies with a deal about multiple domains
allowedDomainEndings	list of strings	Company email domains that can be configured, set by customer service
error	string	Error message (only if result is ERROR)

1.2.4. Example Python 3 code

```
import requests
import json
```

```
import base64

seturl='https://secure.lekab.com/restsms/api/emailtosms/set'
jsondata = {}
jsondata['email']='john.doe@bigco.com'
jsondata['checkMobile']=True

authstringarray="testuser:testpass".encode('utf-8')
authbase64=base64.b64encode(authstringarray).decode('utf-8')
headers={'Authorization':'Basic '+ authbase64}
headers['Content-type']='application/json'

data_json = json.dumps(jsondata) + '\n'

response = requests.post(seturl, data=data_json, headers=headers)

setresp = response.json()
if setresp['result'] == 'OK':
    print("Successfully enabled user " + setresp['email'])
else:
    print("Failed with error message: " + setresp['error'])
```

will output

```
Successfully enabled user john.doe@bigco.com
```

Chapter 2. The `/unset` endpoint

Used for removing an email address from the list of email to SMS users.

2.1. GET request example

e.g. from web browser or curl. Note that url query parameters must be url-encoded (from UTF-8) if they contain characters that are not allowed in a url. Space can be represented as "+".

```
curl
https://secure.lekab.com/restsms/api/emailtosms/unset?U=testuser&P=testpass&EM=john
.doe@bigco.com
```

2.2. DELETE request example

e.g. from curl. Note that url query parameters must be url-encoded (from UTF-8) if they contain characters that are not allowed in a url. Space can be represented as "+".

```
curl -X DELETE
https://secure.lekab.com/restsms/api/emailtosms/unset?U=testuser&P=testpass&EM=john
.doe@bigco.com
```

2.3. POST request example

probably from an application

```
https://secure.lekab.com/restsms/api/emailtosms/unset
```

With the contents of the HTTP body:

```
{"username":"testuser","password":"testpass",
 "email":"john.doe@bigco.com"}
```

2.3.1. Explanation of parameters for /unset

The only parameter apart from authentication is the email address to remove from the list.

POST json key	GET/DELETE query param	json value (strings quoted)	query param value (strings without quotes)	
email	EM	string	string	email sender to be removed from email to SMS list
username	U	string	string	username of the Email to SMS admin account
password	P	string	string	password of the Email to SMS admin account
apikey	key	string	string	API key of the Email to SMS admin account

The email parameter and allowed domains

Email can be either a single email sender "john.doe@bigco.com" or an entire domain "@bigco.com". Unsetting an entire domain means removing one entry in the list which has an empty name and a domain, it does not mean removing e.g. all lines with that domain or anything like that. For every company there is a list of email domain endings that can be used in the email addresses cleared for email to SMS by that company. It is only possible to unset email addresses where the domain ends with one of the endings on that list. In the example, "bigco.com" is probably on the list. In that case "bigco.com", "sales.bigco.com" and even "longendinginbigco.com" will be allowed for the company, but e.g. "gmail.com" or "smallco.com" will not be allowed. The allowed domains list cannot be edited from this web service, contact our support to add one of your own email domains.

How the /set and /default endpoints payinguser parameter affects unsetting

When the email address was set for email to SMS, a paying user was specified. You can only remove email addresses where the paying user belongs to your company. The allowed domains list (see preceding paragraph) should ensure that all your email addresses are bound to paying users in your company, unless our support has previously made special arrangements, e.g. for testing purposes.

2.3.2. HTTP response to /unset

A successful request will return 200 OK and a json document of the following format. The `result` field will have the value "OK" for a successful request. The `Content-Type` header of the response is `application/json` for all responses.

```
{
  "result" : "OK",
  "deletedEmail" : "john.doe@bigco.com"
}
```

A failed request will have a different HTTP result code from 200 OK, and the result will be

"ERROR". For example, if the wrong login credentials are given, the result code is 401 and the response body is:

```
{"result":"ERROR","error":"Unauthorized"}
```

2.3.3. Explanation of response to /unset

json key	json value (strings quoted)	
result	string	OK or ERROR
deletedEmail	string	Email sender removed from email to SMS list
error	string	Error message (only if result is ERROR)
allowedDomainEndings	list of strings	Company email domains that can be configured, set by customer service (only for error when domain is not allowed)

2.3.4. Example Python 3 code

```
import requests
import json
import base64

deleteurl='https://secure.lekab.com/restsms/api/emailtosms/unset?EM=john.doe@bigco.
com'

authstringarray="testuser:testpass".encode('utf-8')
authbase64=base64.b64encode(authstringarray).decode('utf-8')
headers={'Authorization':'Basic '+ authbase64}

response = requests.delete(deleteurl, headers=headers)

deleteresp = response.json()
if deleteresp['result'] == 'OK':
    print("Successfully removed user " + deleteresp['deletedEmail'])
else:
    print("Failed with error message: " + deleteresp['error'])
```

will output

```
Successfully removed user john.doe@bigco.com
```

Chapter 3. The `/default` endpoint

Used for setting the company defaults used when the `/set` endpoint is called

3.1. GET request example

e.g. from web browser or curl. Note that url query parameters must be url-encoded (from UTF-8) if they contain characters that are not allowed in a url. Space can be represented as "+".

```
curl https://secure.lekab.com/restsms/api/emailtosms/default?
   U=testuser&P=testpass&PU=bigco-sales&SD=BIGCO&
   DC=46&MS=12&CM=Y&IS=N&ND=N&OM=Y&SC=N&SL=N
```

3.2. POST request example

probably from an application

```
https://secure.lekab.com/restsms/api/emailtosms/default
```

With the contents of the HTTP body:

```
{ "username":"testuser",
  "password":"testpass",
  "payinguser":"bigco-sales",
  "sender" : "BIGCO",
  "defaultCountrycode" : "46",
  "maxSms" : "12",
  "checkMobile" : true,
  "includeSubject" : false,
  "onlyNonDelivery" : false,
  "origMsg" : true,
  "splitCallback" : false,
  "senderLogic" : false}
```

3.2.1. Explanation of parameters for /default

This allows for one-time setting of company default parameters, followed by automatic additions upon staffing changes by just stating the email address (and proper admin authorization).

POST json key	GET query param	json value (strings quoted)	query param value (strings without quotes)	
payinguser	PU	string	string	The user account that will become owner of sent messages, digits for id or login username
sender	SD	string	string	SMS sender id 3 to 11 chars
costCenter	CC	string	string	Label for grouping on bill
defaultCountryCode	DC	string of digits	string of digits	Replaces leading zero in numbers starting with single zero
maxSms	MS	digit string for integer	digit string for integer	max SMS parts per message max 255
checkMobile	CM	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Only send if Google library has recipient number format in a possible official mobile number series for the country
includeSubject	IS	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Include the email subject in the SMS text
onlyNonDelivery	ND	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Only callback failed delivery receipts
origMsg	OM	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Include original message with answer callback
splitCallback	SC	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	Receipts and answers callback to sending email - needs SMTPSPLIT setting in paying user profile
senderLogic	SL	boolean	T, TRUE, Y, YES or F, FALSE, N, NO	If your company has an agreement to use several incoming domains, use special company rules determining SMS sender id from incoming domain
username	U	string	string	username of the Email to SMS admin account
password	P	string	string	password of the Email to SMS admin account
apikey	key	string	string	API key of the Email to SMS admin account

The payinguser parameter

By specifying the payinguser, your company acknowledges that email to SMS from the given email address will be paid for by your company. Only an existing active SMS sending user account within your company can be set as the payinguser. The payinguser should have the right to send SMS and an appropriate (or unlimited) quota, or no SMS will be sent. If the user calling this web service is a special limited email to SMS admin user, the payinguser should probably be another account, or the admin user should also be set up for sending SMS. Probably the payinguser will be specified in the company default if many email addresses are sharing a payinguser. This parameter can be specified either as the user account id "12345" or as the username "bigco-sales".

3.2.2. HTTP response to /default

A successful request will return 200 OK and a json document of the following format. The `result` field will have the value "OK" for a successful request. The `Content-Type` header of the response is `application/json` for all responses.

```
{
  "result" : "OK",
  "companyId" : "14",
  "payinguserId" : "12345",
  "payinguserName" : "bigco-sales",
  "sender" : "BIGCO",
  "costCenter" : "",
  "defaultCountrycode" : "46",
  "maxSms" : 12,
  "checkMobile" : true,
  "includeSubject" : false,
  "onlyNonDelivery" : false,
  "origMsg" : true,
  "splitCallback" : false,
  "senderLogic" : false,
  "allowedDomainEndings" : [ "bigco.com" ]
}
```

The values of the parameters not given in the call are taken from the company defaults, which can be set either by a company admin in the web portal or by a call to the `/default` endpoint in this web service.

A failed request will have a different HTTP result code from 200 OK, and the result will be "ERROR". For example, if the wrong login credentials are given, the result code is 401 and the response body is:

```
{"result":"ERROR","error":"Unauthorized"}
```

3.2.3. Explanation of response to /default

json key	json value (strings quoted)	
result	string	OK or ERROR
companyId	string of digits	Unique id of your company within the service
payinguserId	string of digits	User id of SMS sending account
payinguserName	string	login username of SMS sending account
sender	string	The SMS comes "from" this sender id
costCenter	string	Label for grouping on bill
defaultCountrycode	string of digits	Country code to replace phone number initial zero
maxSms	integer max 255	SMS cannot be longer than this number of SMS parts
checkMobile	boolean	Send only if number is possible mobile number per Google library
includeSubject	boolean	Include email subject in SMS text
onlyNonDelivery	boolean	Callback receipt only if failed SMS delivery
origMsg	boolean	Include original message in answer callback
splitCallback	boolean	Remember sender email and callback receipts and answers to sender
senderLogic	boolean	Only for companies with a deal about multiple domains
allowedDomainEndings	list of strings	Company email domains that can be configured, set by customer service
error	string	Error message (only if result is ERROR)

3.2.4. Example Python 3 code

```
import requests
import json
import base64

seturl='https://secure.lekab.com/restsms/api/emailtosms/default'
jsondata = {}
jsondata['payinguser']='12345'
jsondata['sender']='BIGCO'
jsondata['defaultCountryCode']='46'
jsondata['maxSms']='12'
```

```
jsondata['checkMobile']=True
jsondata['includeSubject']=False
jsondata['onlyNonDelivery']=False
jsondata['origMsg']=True
jsondata['splitCallback']=False
jsondata['senderLogic']=False

authstringarray="testuser:testpass".encode('utf-8')
authbase64=base64.b64encode(authstringarray).decode('utf-8')
headers={'Authorization':'Basic '+ authbase64}
headers['Content-type']='application/json'

data_json = json.dumps(jsondata) + '\n'

response = requests.post(seturl, data=data_json, headers=headers)

setresp = response.json()
if setresp['result'] == 'OK':
    print("Successfully set company defaults")
else:
    print("Failed with error message: " + setresp['error'])
```

will output

```
Successfully set company defaults
```

Chapter 4. The `/querydomains` endpoint

Used for accessing the list of allowed domain endings for your company.

4.1. GET request example

e.g. from web browser or curl. Note that url query parameters must be url-encoded (from UTF-8) if they contain characters that are not allowed in a url. Space can be represented as "+".

```
curl
https://secure.lekab.com/restsms/api/emailtosms/querydomains?U=testuser&P=testpass
```

4.2. POST request example

probably from an application

```
https://secure.lekab.com/restsms/api/emailtosms/querydomains
```

With the contents of the HTTP body:

```
{"username":"testuser","password":"testpass"}
```

4.2.1. Explanation of parameters for /querydomains

This endpoint needs no other parameters than proper authentication

POST json key	GET query param	json value (strings quoted)	query param value (strings without quotes)	
username	U	string	string	username of the Email to SMS admin account
password	P	string	string	password of the Email to SMS admin account
apikey	key	string	string	API key of the Email to SMS admin account

The email parameter for /set and /unset and allowed domains

Email can be either a single email sender "john.doe@bigco.com" or an entire domain "@bigco.com". For every company there is a list of email domain endings that can be used in the email addresses cleared for email to SMS by that company. In the example, "bigco.com" is probably on the list. In that case "bigco.com", "sales.bigco.com" and even "longendinginbigco.com" will be allowed for the company, but e.g. "gmail.com" or "smallco.com"

will not be allowed. The allowed domains list cannot be edited from this web service, contact our support to add one of your own email domains.

4.2.2. HTTP response to /querydomains

A successful request will return 200 OK and a json document of the following format. The `result` field will have the value "OK" for a successful request. The `Content-Type` header of the response is `application/json` for all responses.

```
{
  "result" : "OK",
  "allowedDomainEndings" : [ "bigco.com" ]
}
```

A failed request will have a different HTTP result code from 200 OK, and the result will be "ERROR". For example, if the wrong login credentials are given, the result code is 401 and the response body is:

```
{"result":"ERROR","error":"Unauthorized"}
```

4.2.3. Explanation of response to /querydomains

json key	**json value (strings quoted)**	
result	string	OK or ERROR
allowedDomainEndings	list of strings	Company email domains that can be configured, set by customer service
error	string	Error message (only if result is ERROR)