

DMAPI

- [Basics](#)
- [Account and Handling](#)
- [Domains](#)
- [Contacts](#)
- [Nameservers](#)
- [Modify Zonedata](#)
- [Rights and Roles](#)
- [Other Requests](#)
- [Code-Examples](#)

Basics

Commonalities for all requests

DMAPI-Server URL

This is the service address which has to be used for all requests:

<https://dmapi.joker.com>

This is how a request looks like:

<https://dmapi.joker.com/request/<name-of-request>?<name-of-parameter-1>=<value-of-parameter-1>>

[&<name-of-parameter-2>=<value-of-parameter-2>&auth-sid=<your-session-id>](https://dmapi.joker.com/request/<name-of-request>?<name-of-parameter-1>=<value-of-parameter-1>&<name-of-parameter-2>=<value-of-parameter-2>&auth-sid=<your-session-id>)

Header fields which are returned by most requests

Tracking Id	Unique server-assigned tracking id, assigned to almost all requests
Status-Code	0 if no error occurred, otherwise other than 0
Status-Text	Human readable error description
Result	ACK or NACK (= "Acknowledged" or "Not Acknowledged")
Error	May be returned if (and only if) the request was rejected, in this case reason(s) will be provided. Presence of this line in headers is indicative for that processing didn't take place.
Warning	Indicative of non-fatal processing or validation problems
Proc-Id	Joker.com processing ID
HTTP error codes	200 if everything is OK (request was accepted and processed or queued for processing), otherwise the reason will be provided in Error header lines (or, if this is absent, HTTP error code should be used).

IMPORTANT: Every request (except "login") requires the presence of the **Auth-Sid** variable ("**Session ID**"), which is returned by the "login" request (login). An active session will expire after some inactivity period (default: 1 hour).

In case you are using a browser to access this service, the session id will be set as a cookie, hence need not to be specified as Auth-Sid (unless cookies are not supported or turned off). In any case, Auth-Sid has precedence, if provided.

Requests consist of these parts:

- **"Requires"**: Defines variables (sometimes referred to as 'fields') that **MUST** be present
- **"Accepts"**: Defines variables that **MAY** be present (but not required)
- **"Returns"**: Describes the request's output - header fields and extra data (if any). If "Returns" is omitted, then a standard reply should be expected (Status-Code, Status-Text, Proc-ID etc)

Whois Privacy Services

Privacy services are available for most generic top level domain names. The availability of privacy services is indicated in Joker.com's [price list](#) and [domain search](#).

For requests **"domain-register"** and **"domain-transfer-in-reseller"**, there exists an **additional parameter** "privacy":

- privacy=**basic** (owner contact name and/or organisation left intact, address & emails are masked)
- privacy=**pro** (everything is masked)
- privacy=**none** (same as "no privacy parameter provided")

For **"domain-renew"**, the additional parameter privacy works similar:

- privacy=**basic** (owner contact name and/or organisation left intact, address & emails are masked)
- privacy=**pro** (everything is masked)
- privacy=**none** (explicitly do a renew without privacy services, even if currently privacy is enabled)
- privacy=**keep** (renew with the same level of privacy service which is currently active for the domain) - this now is also **default**; for domains without an existing privacy service subscription, "keep" means "none", so there will be no privacy service ordered. If there is an existing privacy service subscription active, and this privacy service is **not** set to "off", "keep" will also renew this privacy service subscription, together with the domain.

To **order privacy services for existing domains**, the new request "**domain-privacy-order**" can be used:

```
domain=example.com
period=12 (in months, as usual)
expyear=2016 (privacy expiration year, similar to domain renewal request)
privacy=basic|pro
```

If neither period nor expyear is provided, then privacy is ordered for remaining domain lifetime, whatever it is.

Example:

```
https://dmapi.joker.com/request/domain-privacy-order?domain=example.com&privacy=basic&auth-sid=<your-
```

The request "**domain-set-property**" allows to **manage privacy services** for domains which already do have a valid privacy service subscription:

- privacy=basic (activate "basic" privacy service)
- privacy=pro (activate "pro" privacy service)
- privacy=off (deactivate privacy service - **Whois data will be disclosed**)

Samples:

Ordering domain with privacy:

```
https://dmapi.joker.com/request/domain-register?domain=example.com&period=12&owner-c=CCOM-1&billing-c=CCOM-1&admin-c=CCOM-1&tech-c=CCOM-1&ns-list=a.ns.joker.com:b.ns.joker.com&privacy=pro&auth-sid=<your-current-session-id>
```

Renew domain and order privacy:

```
https://dmapi.joker.com/request/domain-register?domain=example.com&period=12&privacy=basic&auth-sid=<your-current-session-id>
```

Request incoming transfer and enable privacy:

```
https://dmapi.joker.com/request/domain-transfer-in-reseller?domain=example.com&transfer-auth-id=zigzag&billing-c=CCOM-1&admin-c=CCOM-1&tech-c=CCOM-1&privacy=pro&auth-sid=<your-current-session-id>
```

Temporarily disable privacy (assuming that it is active):

```
https://dmapl.joker.com/request/domain-set-property?domain=example.com&privacy=off&auth-sid=<your-current-session-id>
```

Fetch real contact data from privacy protected domain:

```
https://dmapl.joker.com/request/query-whois?domain=example.com&internal=1&auth-sid=<your-current-session-id>
```

NOTE: Not retrieved replies will be kept on the server for a period of 30 days, after this time, only the status of specific request will be available (success or failure).

IMPORTANT: Please also note that the registration/renewal period is in **MONTHS, NOT YEARS!**

Account and Handling

login

Login with Username & Password

Requires (mandatory):

<i>username</i>	Joker.com username (email address)
<i>password</i>	Joker.com password

Returns:

<i>Auth-SID</i>	Authenticated Session ID, must be provided with any other request (parameter <i>auth-sid</i>)
<i>List of TLDs</i>	List of domain TLDs which are available to the reseller.

To be able to use the entire API, you must have a reseller account with Joker.com (and use this account to log in).

Example:

```
https://dmapl.joker.com/request/login?username=johndoe&password=myspass
```

Login with API Keys

Instead of using your user- and password credentials, you may also use so called '**API keys**' for login. This way, you can create several DMAPl access facilities, which has several advantages:

- you do not need to expose username and password in your scripts

- API keys **can be restricted** to be "read only" (no modifications possible), or to allow 'modifications only' (can not produce costs)
- you may hand API keys to your staff to enable them to do specific things with Joker.com

Create your API keys in '**My Profile**' in section '**Manage Joker.com API access keys**'

Example usage:

```
https://dmapi.joker.com/request/login?api-key=Key_created_in_your_Profile_at_Joker.com
```

The result is the same as for "login"-request, you have to use the provided auth-sid for the subsequent actions.

logout

Returns: Nothing

Used to forcibly close (terminate) a session. The session identified by Auth-SID may not be used anymore to send requests. Normally not required as session will timeout anyway (default: 1 hour).

result-list

Accepts:

<i>pending</i>	get results of requests still in progress
<i>showall</i>	get all results incl. deletes using result-delete
<i>period</i>	get results for specifed period of days (default: 90)
<i>date</i>	get results received on (or before) specified date. When date is specified, parameter "period" will be applied to this date (instead of today) and defaults to 1
<i>offset</i>	start dispalying results from specified position
<i>limit</i>	only get number of results specified by limit
<i>status</i>	get results having specified status (ack/nack/?)
<i>count-only</i>	when set to 1, only count number of records and return single line in format "Records: N"

All following filters may use patterns ("*" and "?")

<i>rtype</i>	get results for requests of specified type (domain-register/etc)
<i>objid</i>	get results for specified object ids (domain names, contacts, hosts)
<i>procid</i>	get results for specified proc-id
<i>svtrid</i>	get results for specifued SvTrId
<i>cltrid</i>	get results for specified ClTrId

Returns:

List of answers from joker.com (one per line):

TimeStamp SvTrId Proc-ID request-type status ClTrId

Where:

TimeStamp: The time when request was made, YYYYMMDDHHMMSS
SvTrId: Tracking-Id associated with this request.
Proc-ID: Proc-Id associated with this request.
request-type: The type of the request.
request-object: The object name (host, domain or contact handle)
status: ack, nack or ?, where ack means that request was completed successfully.
ClTrId: User specified transaction ID, or "-" if nothing was provided by the user

Example:

<https://dmapi.joker.com/request/result-list?auth-sid=20ddb8c3b2ea758dcf9fa4c7f46c0784>

result-retrieve

Accepts:

<i>Proc-ID</i>	One of these must be specified. If both are specified, SvTrId has precedence.
<i>SvTrId</i>	

Returns:

Answer (processing result) associated with specified Tracking/Processing ID.
If detailed information (content) is not available, only status will be returned

Please note:

Since there is no requirement of uniqueness for user-specified transaction ids, it is not possible to use them to retrieve specific results.

Example:

```
https://dmapi.joker.com/request/result-retrieve?proc-id=8181810&auth-sid=<your-current-session-id>
```

result-delete

Accepts:

<i>Proc-ID</i>	One of these must be specified. If both are specified, SvTrId has precedence
<i>SvTrId</i>	

Returns:

A descriptive message (confirmation) in case of success.

This request will delete the content (not the status) of a reply to an asynchronous request. Deleted results will not be listed anymore when using *result-list*.

query-profile

Returns:

Returns reseller profile data in format "key: value". May be used to query account balance

Example:

```
https://dmapi.joker.com/request/query-profile?auth-sid=20ddb8c3b2ea758dcf9fa4c7f46c0784
```

Domains

query-domain-list

Accepts (optional)

<i>pattern</i>	Pattern to match (globbing, like "dom*")
<i>from</i>	Start from this item in list
<i>to</i>	End by this item
<i>showstatus</i>	Returns additional column, showing domain status; may be 0 or 1
<i>showgrants</i>	Returns additional column, showing domain grants; may be 0 or 1
<i>showprivacy</i>	Returns additional columns, showing privacy status; may be 0 or 1 The following columns will be added: privacy-status - shows the currently active privacy status privacy-origin - shows the privacy status, which was bought privacy-expiration - shows the expiration date of privacy

Returns

List of registered domains and their expiration dates (one per line, separated by whitespace). If "showstatus" is present, the the list will be with three columns, the last one showing domain status (like "lock,autorenew" etc - comma separated).

Example:

```
https://dmapi.joker.com/request/query-domain-list?pattern=a*&showstatus=1&auth-sid=<your-current-session-id>
```

domain-register

Requires (mandatory)

<i>domain</i>	Domain name to register
<i>period</i>	Registration period in months (not in years!)
<i>status</i>	Set domain status (only "production" is accepted so far)
<i>owner-c</i>	Owner contact handle
<i>billing-c</i>	Billing contact handle
<i>admin-c</i>	Administrative contact handle
<i>tech-c</i>	Technical contact handle
<i>ns-list</i>	List of name servers, delimited by colon

Accepts (optional)

<i>autorenew</i>	if set to "1", domain will be autorenewed upon expiration
<i>language</i>	3 letter language code for IDN domains
<i>registrar-tag</i>	Registrar tag, also known as "Membership token", currently only needed for .XXX domains
<i>privacy</i>	basic pro none - adds whois privacy subscription to domain order, see also Whois Privacy Services
<i>max-price</i>	maximum price user is willing to pay for a domain registration, renewal or transfer. If provided, it must be a number (fractions allowed) above 0. If not provided and domain has non-standard pricing, or if <i>max-price</i> is lower than actual domain price at the moment of registration, the request will fail. If <i>max-price</i> is higher than the actual domain price, then only the actual price will be deducted

This request allows to **register** a domain.

You must have **registered contacts** (handles) to be able to register a domain.(=> contact-create, query-contact-list)

Important: Please note that the registration period is in **months, not years**. In most cases, this number is a **multiple of 12**.

Example:

```
https://dmapi.joker.com/request/domain-  
register?domain=example.com&period=12&status=production&owner-c=COCO-4711&billing-  
c=COCO-4712&admin-c=COCO-4712&tech-c=COCO-4712&ns-  
list=a.ns.joker.com,b.ns.joker.com,c.ns.joker.com&auth-sid=<your-current-session-  
id>
```

domain-renew

Requires (mandatory)

<i>domain</i>	domain name to renew
<i>period</i>	renewal period in months (not in years!)
<i>expyear</i>	the wanted expiration year (e.g. '2014')

Accepts (optional)

<i>privacy</i>	basic pro none keep - adds whois privacy subscription to domain renewal	
	none: explicitly do a renew without privacy services, even if currently privacy is enabled keep: renew with the same level of privacy service which is currently active for the domain see also Whois Privacy Services	
<i>max-price</i>		maximum price user is willing to pay for a domain registration, renewal or transfer. If provided, it must be a number (fractions allowed) above 0. If not provided and domain has non-standard pricing, or if <i>max-price</i> is lower than actual domain price at the moment of registration, the request will fail. If <i>max-price</i> is higher than the actual domain price, then only the actual price will be deducted

With this request you can renew the domain. Please be aware that all renewals are not refundable.

"**expyear**" is a safety option which can be used **instead of "period"** to renew domain till specified year (not longer). If you use "period", and by mistake send the request more than once, domain will be renewed again, while with "expyear", it will not be renewed if it's expiration year is greater or equals to specified.

Only one of "period" or "expyear" may be used, but not both.

Please note:

- registration period is in **months, not years!** In most cases, this number is a **multiple of 12**.
- to prevent from unintended/errorneous multiple renewals, **a specific domain name can only be renewed once per 6 hours**. This means, if you want to renew a domain for 2 years, you should use a period of "24" - in case you want to renew using two requests with a period of "12", the 2nd request must not be sent before 6 hours after the 1st one.

Example:

```
https://dmapi.joker.com/request/domain-renew?domain=example.com&period=12&auth-sid=<your-current-session-id>
```

domain-modify

Requires (mandatory)

<i>domain</i>	domain name to modify
---------------	-----------------------

Accepts (optional)

<i>billing-c</i>	domain contact data to modify
<i>admin-c</i>	
<i>tech-c</i>	
<i>ns-list</i>	list of new nameservers delimited by colon ":" (it will replace existing nameservers!)
<i>registrar-tag</i>	Registrar tag, also known as "Membership token", currently used for .XXX domains
<i>dnssec</i>	<p>if specified, allows setting or removal of DNSSEC keys for domain. If not specified, DNSSEC records will not be changed. Value of "0" will remove DNSSEC, value of "1" will add DNSSEC (and ds-N parameters must be provided)</p> <p>ds-1 list of DNSSEC parameter sets (min. 2, max. 6) for DNSSEC enabled domains</p> <p>ds-2 for com/net/org/tv/cc each entry has format:</p> <p>ds-3 tag:alg:digest-type:digest</p> <p>ds-4 for de:</p> <p>ds-5 protocol:alg:flags:pubkey-base64</p> <p>ds-6</p>

With this request you can modify

- contact handles
- nameservers
- DNSSEC parameters (for DNSSEC capable nameservers and TLDs supporting DNSSEC)

Please only specify contact handles which you want to modify, all unspecified handles will be left as is.

For DNSSEC, you will find more information [here](#) .

Example:

1	<code>https://dmap.i.joker.com/request/domain-modify?ns-list=a.my-own-ns.com:b.my-own-ns.com&auth-sid=<your-current-session-id></code>
---	--

Example for adding a key (and enabling DNSSEC):

1	<code>http://dmap.i.joker.com/request/domain-modify?domain=example.de&dnssec=1&ds-1=256:3:5:AwEAA4TaRzEn0zfw9w2qkazHTqOX97XwNzoDHMKgClJbX/jr9YGdCYKpTfcKQbVmpl4EVBjOjNWawOjEgpVivw53yHbtEFLOCSYyJwHuXaWiEv0sESNAoPCXQurp4tbHSLnRpyxyzV3dsh9/qUlkJEXxcEuFPaLG3bz1z0Ude6gMXr</code>
---	---

1	<code>http://dmap.i.joker.com/request/domain-modify?domain=example.com&dnssec=1&ds-1=38698:5:1:AFE249392848D0458B3967BA1EE9BDEA6C3ECB43</code>
---	--

Example for removing DNSSEC information:

1	<code>https://dmap.i.joker.com/request/domain-modify?domain=example.com&dnssec=0</code>
---	---

domain-delete

Requires (mandatory)

<i>domain</i>	domain name to delete
---------------	-----------------------

Accepts (optional)

<i>force</i>	if present and set to '1', 'Y' or 'Yes', the domain will be deleted even if older than 72 hours.
--------------	--

With this request you can delete a domain.

If you delete a domain (**gTLDs only**) within the **first 72 hours** after the registration, **the registration-fee will be refunded** to your account.

To delete a domain which is registered for more than 72 hours, you must specify "force=1", otherwise the request will be rejected.

Please note: The number of refundable deletions of domains per month is limited, and depends on the number of domains in a portfolio, and the related TLD registry.

Example:

```
https://dmapi.joker.com/request/domain-delete?domain=example.com&force=1&auth-sid=<your-current-session-id>
```

domain-owner-change

Requires

<i>domain</i>	domain name to change the owner of
<i>name</i>	(optional) full name (if empty, fname + lname will be used)
<i>fname</i>	first name
<i>lname</i>	last name
<i>title</i>	(optional)
<i>individual</i>	(optional) Y, Yes, N, No
<i>organization</i>	(optional if individual)
<i>email</i>	mailaddress of the contact
<i>address-1</i>	street address
<i>address-2</i>	(optional)
<i>address-3</i>	(optional)
<i>city</i>	
<i>state</i>	(optional)
<i>postal-code</i>	
<i>country</i>	ISO country code (2 letters)
<i>phone</i>	
<i>extension</i>	(optional)
<i>fax</i>	(optional)

<i>lock-opt-out</i>	(optional) yes, no (new IRTP-C: do not apply a 60 days transfer lock to the domain)
---------------------	--

With this request you can change the visible **WHOIS owner** of a domain.

Either "name" or "lname" and "fname" must be provided. Fields **not** marked as optional are mandatory, and must not be empty.

For the revised ICANN IRTP-C (change of registrar) policy, please also visit our [documentation](#).

Example:

1	<a href="https://dmapi.joker.com/request/domain-owner-change?domain=example.com&fname=John&lname=Doe&email=johnd@someisp.com&address-1=Smartroad1&city=Smalltown&postal-code=40122&country=US&phone=+1.422.8001&auth-sid=<your-current-session-id>">https://dmapi.joker.com/request/domain-owner-change?domain=example.com&fname=John&lname=Doe&email=johnd@someisp.com&address-1=Smartroad1&city=Smalltown&postal-code=40122&country=US&phone=+1.422.8001&auth-sid=<your-current-session-id>
---	---

domain-lock

Requires (mandatory)

<i>domain</i>	domain name to lock
---------------	---------------------

With this request you can **lock a domain** to **prevent from fraudulent transfer attempts**.

If a domain is locked, each transfer-request from a foreign registrar will be declined.

Example:

<https://dmapi.joker.com/request/domain-lock?domain=example.com&auth-sid=<your-current-session-id>>

domain-unlock

Requires (mandatory)

<i>domain</i>	domain name to unlock
---------------	-----------------------

With this request you can **unlock a domain**.

If you want to transfer a domain to a foreign registrar, it has to be unlocked.

Example:

`https://dmapi.joker.com/request/domain-unlock?domain=example.com&auth-sid=<your-current-session-id>`

domain-transfer-in-reseller

Requires (mandatory)

<i>domain</i>	full qualified domain name to transfer to Joker.com
<i>transfer-auth-id</i>	AUTH-ID required for all domains except .eu
<i>owner-c</i>	new owner contact handle to be used for transferred domain
Accepts (optional)	
<i>admin-c</i>	new admin contact handle
<i>tech-c</i>	new tech contact handle
<i>billing-c</i>	new billing contact handle
<i>status</i>	new domain status to be set after transfer (lock, production)
<i>period</i>	renewal period in months, not in years! [not in use, please leave empty]
<i>autorenew</i>	Autorenew flag for the domain (0 or 1); if not set explicitly, defaults to 1
<i>ns-list</i>	list of colon-separated nameservers
<i>privacy</i>	basic pro none - adds whois privacy subscription to domain transfer, see also Whois Privacy Services
<i>max-price</i>	maximum price user is willing to pay for a domain registration, renewal or transfer. If provided, it must be a number (fractions allowed) above 0. If not provided and domain has non-standard pricing, or if <i>max-price</i> is lower than the actual domain price at the moment of registration, the request will fail. If <i>max-price</i> is higher than the actual domain price, then only the actual price will be deducted

With this request you can initiate a transfer of the domain from another registrar to Joker.com.

You have to provide a valid **AUTH-ID** (you have to request this from the **current** provider of the domain).

Please note:

- please make sure, that there is no so-called 'registry lock' active for the domain to be transferred. Please verify, that the current registrar does not block the domain, resp. deactivates the blocking
- the domain to be transferred must have been active for at least 60 days at the current registrar, otherwise the transfer may fail
- the domain to be transferred must not be object of an active ICANN UDRP dispute procedure
- the domain should have a remaining term of least 7 days, to ensure that the transfer succeeds within the time limits (a maximum of 5 days is granted to confirm or object a transfer). In case the domain has less than 7 days of remaining term, a successful transfer cannot be warranted.
- this procedure will trigger email notifications to the owner and the admin contact of the domain to be transferred (this is following mandatory ICANN transfer provisions)

Example:

```
https://dmapi.joker.com/request/domain-transfer-in-reseller?domain=example.com&transfer-auth-id=abc123xyz&owner-c=COCO-4711&admin-c=COCO-4712&tech-c=COCO-4712&billing-c=COCO-4712&auth-sid=<your-current-session-id>
```

domain-set-property

Requires (mandatory)

<i>domain</i>	domain name or pattern
<i>pname</i>	property name
<i>pvalue</i>	property value (may be empty)

With this request you can set a property (flag) for a domain or a set of domains, selected by wildcard pattern.

List of available properties and their effects:

<i>autorenew</i>	0 or 1 If set to 1, the domain will be automatically renewed on expiration (if you have enough funds in your account)
<i>whois-opt-out</i>	0 or 1 currently only used for .tel domains; if set to 1, owner information will not be shown in whois
<i>privacy</i>	off or basic or pro (if available)

If an empty value is provided, then the property will be cleared, i.e. the default will be used.

Example:

```
https://dmapi.joker.com/request/domain-set-property?domain=example.com&pname=autorenew&pvalue=1&auth-sid=<your-current-session-id>
```

domain-get-property

Requires

<i>domain</i>	domain name
<i>pname</i>	property name (same as in domain-set-property)

With this request you can query the value of a specific property set for a domain.

It returns a single line, which looks like:

1	autorenew: 0
---	--------------

Example:

1	<pre>https://dmapi.joker.com/request/domain-get-property?domain=example.com&pname=autorenew&auth-sid=<your-current-session-id></pre>
---	--

domain-transfer-get-auth-id

Requires

<i>domain</i>	domain name to get AUTH-ID for
---------------	--------------------------------

Retrieves domain's **Auth-ID**, which is required when transferring domains to another registrar.

This request is not real-time, i.e. you have to check detailed reply (use "result-retrieve") to get the Auth-ID.

Please note:

Every request will generate a new Auth-ID, thus rendering any previously requested Auth-ID invalid.

Example:

1	<code>https://dmapi.joker.com/request/domain-transfer-get-auth-id?domain=example.com&auth-sid=<your-current-session-id></code>
---	--

domain-transfer-control

This request allows to retrieve status information for all pending transfers. Additionally, transfers can be cancelled, or the FOA email can be re-sent.

Parameters:

action	One of: <i>list, show, resend-foa, cancel</i>
domain	Domain name to control transfer of (ignored when action= <i>list</i>)

action=*list*

When action=list, this request returns a list (columns are separated by tabs) of all currently pending transfers:

<code><domain> <state> <owner-email> <admin-email></code>

where <state> is one of (listed in "natural" transition order):

IN_DB	
PAYMENT_PREPARED	
PAYMENT_ACQUIRED	
PENDING_FOA_SENDING	
	Joker.com could not parse the recipient of the FOA. Manual action from Joker.com is required and happens during office hours, thus in worst case it may take few days (during weekend or holidays).
FOA_BEEN_SENT	
	FOA is sent and Joker.com is waiting for owner's reaction. Owner has 5 days to react, thus the state may last up to 5 days.
FOA_NACK	FOA is rejected
FOA_ACK	FOA is accepted
REGISTRY_REQUEST_SENT	

REGISTRY_REQUEST_RECEIVED_NOW_WAITING

FOA had been accepted, transfer request is sent and the losing registrar has to release the domain. This state may take up to 5 days, we could do nothing to speed it up. While in this state, transfer may be cancelled by using "action=cancel".

REGISTRY_CANCELATION_PENDING

The user has requested cancel of domain transfer by using "action=cancel".

DOMAIN_WITH_JOKER_PENDING_COMPLETE

The domain is actually with Joker.com, but either we haven't noticed yet (batch pending) or manual work is necessary. Please contact Joker.com if this state lasts longer than 1 hour.

PAYMENT_COMMITTED

TRANSFER_PROCESSED_SUCCESSFULLY

action=show

When action=show, the status of a specified domain is returned like:

```
domain: example.com
status: FOA_BEEN_SENT
owner_email: owner@example.com
admin_email: admin@example.com
transfer-id: 123456
```

action=resend-foa

When action=resend-foa, FOA will be resent (only possible in state FOA_BEEN_SENT).

action=cancel

When action=cancel, the (pending) transfer will be cancelled.

Returns

Status code is 1000 for successful request, or ≥ 2000 otherwise.

Example

```
https://dmapi.joker.com/request/domain-transfer-control?domain=example.com&action=resend-foa
```

domain-check

Requires (mandatory)

<i>domain</i>	domain name to check
---------------	----------------------

Accepts (optional)

<i>check-price</i>	If provided, specifically check the price for: create, renew, transfer, restore
<i>period</i>	If provided, price will be calculated based on specified period. Period may be specified in years (values from 1 to 10) or in traditional for DMAPI months (≥ 12 , must be a multiple of 12).
<i>language</i>	Relevant only for IDN domains, specifies language as 2 letter (ISO 639.1) or 3 letter (ISO 639.2) code: https://www.loc.gov/standards/iso639-2/php/code_list.php

With this request you can **check if a domain is available for registration, and what type of domain pricing will be applied for different types of orders.**

Returns one or more lines with key-value pairs as follows:

domain-status: <status>

<status> can be one of:

available	regular domain (non-premium) available for registration
premium	premium domain available for registration
unavailable	domain is not available for registration

If <status> is "unavailable", then domain-register request will fail, though domain-transfer-* and domain-renew (if domain is with Joker.com) are possible, of course.

domain-status-reason: <status-reason>

Provides human-readable description why domain is unavailable for registration, like *blocked*, *reserved*, *registered* etc.

May be absent if registry does not provide specific reason. Should not be parsed as value is registry dependent and could be virtually anything.

domain-class: <class>

Specifies price-class. Anything but "standard" is considered "premium", though some registries use "tier-1", "premium", "BBB" etc.

Mostly a hint, values except "standard" should not be interpreted as they have no useful meaning and vary among registries.

domain-price-<type>: <price> <currency> <period>y

Reports domain price, where <type> will be value provided in check-price parameter, i.e. create, renew, transfer, restore.

- <price> is the final price, i.e. amount that will be deducted from the user's account, considering all discounts, promos etc.
- <currency> is a 3-letter currency code, serves only as information.
- <period> is reported in years and suffixed by "y", like "5y", and it may be different from "period" parameter value (depends on registry), for instance when requested period is not available, too low or too high.

The price reported is valid only for specified period, i.e. if price is 100 for 2 years it does not necessarily mean that price for 1 year is 50, thus it only makes sense to check price for the period which will be used verbatim in subsequent request of specified <type>.

domain-price-promo: <start> <end>

<start> and <end> are ISO timestamps in format like *2019-07-01T00:00:00.000Z*

For domains that have promotional pricing, this provides start and end timestamps of promotional period. Usually its mere presence means that promo-pricing is in effect, but to be sure values have to be checked explicitly.

If "domain-status" returned "premium" or "domain-class" returned anything but "standard", then the parameter *max-price* must be present in register/transfer/renew requests.

Full response example (web.blog):

```
domain-status: unavailable
domain-status-reason: blocked
domain-class: standard
domain-price-create: 8.21 USD 1y
domain-price-promo: 2019-09-01T00:00:00.000Z 2019-12-31T23:59:59.000Z
```

Example:

<https://dmapi.joker.com/request/domain-check?domain=example.com&auth-sid=<your-current-session-id>>

Contacts

query-contact-list

Accepts:

<i>pattern</i>	pattern to match (against handle)
<i>from</i>	start from this item in list
<i>to</i>	end by this item in list
<i>tld</i>	limits output to contact handles which may be used with specified toplevel domain (tld), like "com".
<i>extended-format</i>	provides additional information for every contact listed: name & organization. May be "1" or "0", defaults to "0"

Returns:

List of registered contacts (handles), one per line.

When "extended-format" is requested, output columns are separated by tabs ("\t"), and "Columns" header provides column names.

Example:

```
https://dmapi.joker.com/request/query-contact-list?pattern=coco-47*&tld=com&auth-sid=<your-current-session-id>
```

contact-create

Requires:

<i>tld</i>	target TLD where this contact is intended to be used.
<i>name</i>	full name (if empty, fname + lname will be used)
<i>fname</i>	first name (required for .FI contacts)
<i>lname</i>	last name (required for .FI contacts)
<i>title</i>	(optional)

<i>individual</i>	(optional) Y, Yes, N, No
<i>organization</i>	(optional if individual)
<i>email</i>	mailaddress of the contact
<i>address-1</i>	street address
<i>address-2</i>	(optional)
<i>city</i>	
<i>state</i>	(optional)
<i>postal-code</i>	
<i>country</i>	ISO country code (2 letters)
<i>phone</i>	
<i>fax</i>	(optional)
<i>lang</i>	language to use for .EU contacts
<i>app-purpose</i>	required for .US contacts
<i>nexus-category</i>	required for .US contacts
<i>nexus-category-country</i>	required for .US contacts
<i>account-type</i>	required for .UK contacts, if used as owner contact
<i>company-number</i>	required for .UK contacts with specific account types
<i>orgid</i>	required for .SE/.NU contacts
<i>vatid</i>	(optional) for .SE/.NU contacts
<i>x-ficora-type</i>	required for .FI contacts
<i>x-ficora-is-finnish</i>	(yes/no) required for .FI contacts
<i>x-ficora-registernumber</i>	required for .FI contacts, if ficora-type is company
<i>x-ficora-identity</i>	required for .FI contacts, if type is person and finnish
<i>x-ficora-birthdate</i>	required for .FI contacts, if type is persion and not finnish
<i>x-ficora-legalemail</i>	(optional) for .FI contacts

Either "name" or "lname" and "fname" must be provided.

Fields **not** marked '(optional)' are mandatory (and must not be empty).

Parameters "lname" and "fname", if provided, will be converted to "name" (simple concatenation of "fname" and "lname"), because registries support only "name" format. In general, use of "fname" and "lname" is deprecated, and support for these fields will be removed in version 1.2.

"orgid" represents Swedish personal or organisational number and is required for .SE/.NU contacts.

It starts with ISO 3166 Alpha-2 country code in square brackets. If the country code for Sweden is given [SE] a valid Swedish personal or organisational number must be given (6 digits, dash, 4 digits), otherwise 1 to 123 characters can follow.

If organization is empty and the country code for Sweden [SE] is given, orgid must be a personal number, not an organisational number.

"**valid**" is optional for .SE/.NU contacts. It starts with a two letter country code (uppercase), followed by an optional space, followed by a country specific string containing digits 0-9, and letters a-z and A-Z, maximum 64 Chars.

For .fi contacts:

"**x-ficora-type**" is required for .fi contacts. The following values are accepted: privateperson, company, corporation, institution, politicalparty, township, government, publiccommunity

"**x-ficora-registernumber**" is required if x-ficora-type is set to "company".

"**x-ficora-is-finnish**" is always required for .fi contacts: yes = finnish company or person, no = not a finnish person or company

"**x-ficora-identity**" is required for .fi contacts, if x-ficora-type is set to 'privateperson' and x-ficora-is-finnish is set to 'yes'

"**x-ficora-birthdate**" is required for .fi contacts, if x-ficora-type is set to 'privateperson' and x-ficora-is-finnish is set to 'no' in the following date format "YYYY-MM-DD".

Please note:

- Parameters listed here (except "tld") may be used (or are required) in other requests, this is indicated by referring to "Contact fields".
- "lang" must contain two-letter ISO country (language) code, and is only required when creating .EU contacts. The purpose is to specify language to be used in notifications emails, sent from EURid. Please note - this field cannot be modified later, and the default is 'EN' (English)!
- "app-purpose", "nexus-category" and "nexus-category-contry" are required only when creating .US contacts, and cannot be modified later.

Example:

```
https://dmapi.joker.com/request/contact-create?tld=com&name=John  
Doe&email=johnd@someisp.com&address-1=Smartroad 1&city=Smalltown&postal-  
code=40122&country=US&phone=+1.422.8001&auth-sid=<your-current-session-id>
```

contact-modify

Requires:

<i>handle</i>	contact handle to modify.
---------------	---------------------------

Accepts:

Field names exactly like in contact-create request, except that omitted fields won't be modified. That is, if you specify a field, it will be used as a new value, if you omit it, the old value will remain.

The field "tld" is not relevant for this request and will be ignored if present.

contact-delete

Requires:

<i>handle</i>	contact handle to delete
---------------	--------------------------

With this request you can delete previously registered contacts

Example:

```
https://dmapi.joker.com/request/contact-delete?handle=coco-4711&auth-sid=<your-current-session-id>
```

Nameservers

query-ns-list

Accepts:

<i>pattern</i>	pattern to match (against host name, like "ns.dom*")
<i>full</i>	include IPs if non-zero (0 or 1)

Returns:

List of registered name servers, one per line.

If "full" is non-zero, then the list will include IP addresses, IPv4 (2nd column) and IPv6 (3rd column).

Columns will be separated by tab ("t") character. If specific IP is not present (say, there is only IPv4 or IPv6), it will be listed as "-".

Example of list with IPs:

1	ns.example.com	1.2.3.4	-
2	ns6.example.com	-	FE80:0000:0000:0000:0202:B3FF:FE1E:8329

Example:

```
https://dmapi.joker.com/request/query-ns-list?pattern=*my-own-ns*&full=1&auth-sid=<your-current-session-id>
```

ns-create

Requires:

<i>host</i>	Full qualified host name
<i>ip</i>	IPv4 address (must not be from IANA's reserved range)
<i>ipv6</i>	IPv6 address (short notation like fec0::17 is accepted)

With this request a new nameserver can be registered with the registry. This is needed to use a nameserver with glue records.

Either an IPv4 or IPv6 address is required.

ns-modify or host-modify

Requires:

<i>host</i>	full qualified host name
<i>ip</i>	IPv4 address (must not be from IANA's reserved range)
<i>ipv6</i>	IPv6 address (short notation like 'fec0::17' is accepted)

With this request you can modify the IP address of a registered nameserver.

ns-delete or host-delete

Requires:

<i>host</i>	Full qualified host name
-------------	--------------------------

With this request you can delete a registered nameserver.

Modify Zonedata

dns-zone-list

Accepts:

<i>pattern</i>	Pattern to match (globbing, like "dom*")
----------------	--

Returns:

List zones (domains) which are managed and served by Joker.com name servers. Zones are listed one per line.

dns-zone-get

Requires:

<i>domain</i>	Zone (domain) name to fetch data from
---------------	---------------------------------------

Returns:

Returns list of zone records.

The format of zone is as follows (one record per line):

<code><label> <type> <pri> <target> <ttl> <valid-from> <valid-to> <parameters(s)></code>
--

Where:

<i>label</i>	subdomain/redirection label, relative to current zone, or '@' (which means current zone name)
<i>type</i>	record type (A, AAAA, MX, CNAME, URL, MAILFW, TXT, NAPTR, DYNA, DYNAAAA, SRV, CAA)
<i>pri</i>	numeric value, meaningful only for types MX, NAPTR and SRV, must be 0 for all other types

<i>target</i>	record target/value. Must be quoted if contains spaces
<i>ttl</i>	record TTL (time to live) in seconds
<i>valid-from</i>	record is not valid before this time (UNIX timestamp) or 0
<i>valid-to</i>	record is not valid after this time (UNIX timestamp) or 0
<i>parameters</i>	record-specific parameter(s)

All values from "name" to "ttl" are mandatory for every record.

<valid-to> and <valid-from> are not implemented yet, so usually are 0, and may be omitted if there are no other parameters required.

<parameters(s)> are record dependent, used in NAPTR, MAILFW, FRAME and URL records.

<ttl> must be at least 60 for all records except NAPTR and SVC, where it can be 0 (meaning: no caching). For MAILFW/URL/FW/FRAME <ttl> is irrelevant and assumed to be 60 seconds (i.e., any change will be in effect within 60 seconds from zone change).

<pri> preference value for MX records, priority/weight for SRV records and order/preference for NAPTR records. For any other record, the value must be 0.

Examples of zone records:

```
www A 0 127.0.0.1 86400
www AAAA 0 fec0::17 86400
```

```
@ MX 10 mail.example.com. 86400
```

```
redirect URL 0 http://joker.com 86400
```

Sets redirection from redirect.joker.com to http://joker.com (assuming that current zone is "joker.com").

```
frame FRAME 0 http://joker.com 86400 0 0 "Header: Frame-Forward" "head" "title"
"body"
```

Sets frame-based redirection similar to URL redirection.

For FRAME record, extra parameters are as follows:

- Extra HTTP headers. (newlines may be escaped as \n)
- Extra text placed in <head></head> section of generated HTML.
- Title text (<title></title>) of generated HTML.
- Body text (used for <noframes> section)

```
username MAILFW 0 redirected@example.com 86400 0 0 1
```


This MAILFW records makes redirection of mail sent to username@joker.com (assuming that current zone name is "joker.com") to redirected@example.com, extra parameter ("1") specifies that spam-filtering should be used.

```
naptr NAPTR 10/100 replacement 86400 0 0 "flags" "service" "regex"
```

This NAPTR record has order 10 and preference 100. Only one of "replacement" or "regex" may be specified, if "regex" is specified, "replacement" must be "." (without quotes), if "replacement" is specified, "regex" must be empty string. Quotes are mandatory for all extra parameters.

For details please consult RFC 2915, or look here: <http://de.wikipedia.org/wiki/NAPTR>

```
_ldap._tcp SRV 10/100 ldap.example.com:389 60
```

This SRV record has priority 10, weight 100, target "ldap.example.com" and port "389" with TTL 60 seconds. Please consult RFC 2782 for details.

```
txt TXT 0 "key=value" 86400
```

Quoting of values for TXT records is mandatory.

```
caa CAA 0 issue "letsencrypt.org" 86400
caa CAA 0 issuewild "letsencrypt.org" 86400

caa CAA 0 iodef "mailto:certissues@example.com" 86400
```

For more information about CAA records please check [Wikipedia](#).

```
www CNAME 0 example.com. 86400
```

Two special record types, DYNA and DYNAAAA, are used in case if DynDNS is active. They have same meaning as corresponding "A" and "AAAA" records, except that their targets may be updated using Joker DynDNS service.

To enable/disable DynDNS service, and to define username/password used to access it, the record format is as follows:

```
$dyn dns=yes:username:password
```

username/password may not contain spaces or colon (":") characters. If "no" is specified instead of "yes", DynDNS will be turned off (i.e. DYN* entries will have no effect).

In case if there are any errors, zone modification will not be accepted. All errors will be reported for every erroneous line, so if there are more than one, you will be able to see all detected errors.

dns-zone-put

Requires:

<i>domain</i>	Zone (domain) name to store data to
---------------	-------------------------------------

<i>zone</i>	URL encoded zone data
-------------	-----------------------

Returns:

Replace current zone for provided domain with specified value.

Field "zone" will be parsed as multiline text (so usual line erminators are expected).

Warning: This request will overwrite current zone, i.e. all current ecords will be replaced by new records.

The format of zone is as follows (one record per line):

```
<label> <type> <pri> <target> <ttl> <valid-from> <valid-to> <parameters(s)>
```

Rights and Roles

grants-list

Requires:

<i>domain</i>	Full qualified domain name
---------------	----------------------------

Accepts:

<i>showkey</i>	Show invitation access key
----------------	----------------------------

Get a list of active and pending grants.

Returns lines in the following format (space-separated):

invitation <nr> <scope> <key> domain <domain-name> <role> - - - <invitee-email> <nick-name> grant <nr>
--

Where:

<nr>	record number
<scope>	Grant/Invite id (used to uniquely identify records for revocation)
<key>	Invitation access key, present only when "showkey" is non-zero; it has meaning only for pending invitations, and always is "-" for grants
<domain-name>	domain name (identical to domain in request)
<role>	one of @admin/@billing/@tech/@creator
<inviter-username>	your username (owner of the domain)
<invitee-username>	username of invited user
<invitee-userid>	[leave empty] user-ID of invited user
<nick-name>	name or comment you want to assign to this role

grants-invite

Request parameters:

name	value	mandatory
<i>domain</i>	Full qualified domain name	yes
<i>email</i>	Email of user to be invited to accept a role for 'domain'	yes
<i>role</i>	Role offered (@admin/@billing/@tech/@creator)	yes
<i>client-uid</i>	receiving Joker.com user id as shown on Joker.com - also conforms to reseller-id	no
<i>nick-name</i>	Name or comment for this role/grant for later identification (defaults to email)	no

With this request you can assign 'Roles' (permissions) of domains to other Joker.com users. The special role "creator" is an equivalent to an **internal transfer** of that domain to another Joker.com user. In this case, the domain is **assigned** to the receiving account's portfolio, including all rights and permissions on this domain.

There are two variants to use this functionality: If the **client-uid (Joker.com user account id)** is given, and the client-uid **matches** the **associated user's email address** in our system, the role is **applied immediately** - there will be no request for confirmation on the receiver's side. In case the client-uid is not given, but only the email address, the transaction has to be acknowledged by the receiving user. For this, he will get an email ('invitation'), containing a link and a key. Either the link has to be clicked, or the key has to be applied using Joker.com (=> Transfer). This can also be used for **users which do not yet have a Joker.com account**.

Returns status only (ok or not). Email is sent if request was successful.

grants-revoke

Requires:

<i>domain</i>	full qualified domain name
<i>role</i>	Role to be revoked from domain (@admin/@billing/@tech/@creator)
<i>scope</i>	identifier from 'grants-list' request
<i>type</i>	type of the record to revoke - "grant" or "invitation"

Deletes an issued grant (permission) of a domain, or a pending invitation for a grant.

Returns status only (ok or not). Email is sent if request was successful.

Other Requests

query-object

Accepts:

<i>domain</i>	Domain name	Exactly one of those must be specified. Only objects registered with Joker.com may be queried.
<i>contact</i>	Contact handle	
<i>host</i>	Nameserver	

Returns:

Information about specified object (similar to whois), in format "key: value".

This request is **deprecated** - please use ' query-whois' instead.

query-whois

Accepts:

<i>domain</i>	Domain name	Exactly one of those must be specified. Only objects registered with Joker.com may be queried.
<i>contact</i>	Contact handle	
<i>host</i>	Nameserver	

Returns:

Information about specified object (similar to whois), in the format "key: value".

The difference to the *query-object* request is, that this request reflects actual (live) data in Joker.com database, while *query-object* may show data which not yet up-to-date.

wa-email-list

Lists all domains and owner emails **of domains from your portfolio** which are **pending whois (email) verification**.

For Whois validation, the email address of the owner contact of a newly created, transfered or modified gTLD has to be verified. This verification is done by sending an email to the owner contact, which contains a unique key in a link. When this link is clicked, the related email address is set to "validated".

For full ICANN policy, please see [Whois-Validation](#).

Requires: Nothing

Returns:

```
email-address<TAB>domain-name<TAB>verification-expiration-date
```

```
email-address<TAB>domain-name<TAB>verification-expiration-date
```

Where:

email-address	email address of the domain owner which is pending verification
domain-name	domain name where email is the owner
verification-expiration-date	verification deadline, i.e. if by this date and time (specified in standard ISO format) email is not positively verified, the domain name listed may be deactivated

Example:

```
https://dmapi.joker.com/request/wa-email-list?auth-sid=<your-current-session-id>
```

wa-email-details

Returns email **verification status** and the **associated domain** for provided verification key (whois verification, email validation). If there is more than one domain using the email address to be verified, output will have one line per domain.

For full ICANN policy, please see [Whois-Validation](#).

Requires:

```
key Verification key sent to owner by verification email
```

Returns:

```
status<TAB>email<TAB>domain
```

Where:

status	is 'verification', if 'key' is a valid verification key
email	email address 'key' has been sent to
domain	related domain

Sample:

```
https://dmapi.joker.com/request/wa-email-details?key=<key-to-query>&auth-sid=<your-current-session-id>
```

wa-email-validate

Triggers a new email verification. In case the original verification email got lost or did not succeed otherwise, this request will send a new email. The original key will be **automatically included**.

This request returns special response header "Result", which will contain "ACK" if the validation request has been sent successfully, or "NACK" otherwise, e.g. if the email address is not in "verification pending" status, or already verified (whois verification, email validation).

For full ICANN policy, please see [Whois-Validation](#).

Requires:

email Email address to send validation request to. This must be domain owner's email address.

Returns:

Result: ACK validation request has been sent
Result: NACK no validation request has been sent for this email

Sample:

```
https://dmapi.joker.com/request/wa-email-validate?email=<email-of-owner>&auth-sid=<your-current-session-id>
```

wa-email-verify

Set an email address to 'validated' by using the correct key, and voting with 'yes'.

This request returns a special response header "Result", which will contain "ACK" if the verification has been accepted or "NACK" otherwise, e.g. in case the key is not correct (whois verification, email validation).

This is usefull e.g. in case you want to implement your own individual landing page for your customers. The URL for this landing page can be set in Joker.com's 'Reseller Settings'.

For full ICANN policy, please see [Whois-Validation](#).

Requires:

key	Verification key sent to owner by verification email
answer	Answer to verification request - "yes" or "no"

If the answer is "yes", the verification status is set to "verified" and email is confirmed as valid, if the answer is "no", then verification status is set to "invalid". If there is no answer before verification deadline (normally 15 days), the status is also set to "invalid".

Returns:

Result: ACK	validation request has been successful
Result: NACK	validation request has been rejected or failed

Sample:

```
https://dmapi.joker.com/request/wa-email-verify?key=<key-to-use>&answer=[yes|no]&auth-sid=<your-current-session-id>
```

domain-privacy-order

This request adds a subscription to Whois Privacy Services to an existing domain name.

A summary of all privacy related DMAPI commands is also available at DMAPI Whois Privacy Services.

Requires (mandatory):

<i>domain</i>	domain name to order privacy service for
<i>privacy</i>	basic pro

Accepts (optional):

<i>period</i>	renewal period in months (not in years!)
<i>expyear</i>	the wanted expiration year (e.g. '2016')

"**expyear**" is a safety option which can be used **instead of "period"** to order privacy until the specified year (not longer).

If you use "period", and by mistake send the request more than once, the order would be executed again, while with "expyear", it will not be renewed if it's expiration year is greater or equals to the specified one.

Only one of "period" or "expyear" may be used, but not both.

If neither period nor expyear is provided, then privacy is ordered for the remaining domain lifetime, whatever it is.

Important: Please note that registration period is in **months, not years!** In most cases, this number is a **multiple of 12.**

Example:

```
https://dmapi.joker.com/request/domain-privacy-order?domain=example.com&privacy=pro&auth-sid=<your-current-session-id>
```

This would order privacy service "pro" for the same period as the current term of the domain "example.com".

cor-verify

Authorizes a 'Change of Registrant' (COR) according to the ICANN IRTP-C policy as of 2016-12-01.

Please find the full documentation of the process at Joker.com [here](#), the original ICANN policy is available [here](#).

This request can be used if you did not setup a verification 'push' URL with our provided module (see link above), or you want to use the standard email procedure.

This request returns special response header "Result", which will contain "ACK" if the authorization request has been sent successfully, or "NACK" otherwise, e.g. if the key sent does not match the transaction.

Requires:

key	authorization key (provided in COR confirmation request)
answer	answer to confirm the COR - "yes" or "no"

If the answer is "yes", the change-of-registrant will be executed, if in case of "no", it will be rejected. Please also note the new flag 'lock-opt-out' in 'domain-owner-change' if you do **not** want to lock the domain after the COR for 60 days.

Returns:

Result: ACK	authorization key has been accepted
Result: NACK	authorization key has not been accepted

Sample:

```
https://dmapi.joker.com/request/cor-verify?key=<authorization key sent by Joker.com>&answer=yes&auth-sid=<your-current-session-id>
```

query-price-list

Returns

List of domain prices per action and year (separated by tab) in the currency of the reseller account. The prices already include the reseller discount and also vat if it applies.

Example:

```
https://dmapi.joker.com/request/query-price-list?auth-sid=<your-current-session-id>
```

Code-Examples

Perl

List domains of customer

```
#!/usr/bin/perl
use LWP::UserAgent;
use Data::Dumper;

my $dmapiURL = "https://dmapi.joker.com/request";

my $ua = LWP::UserAgent-> new;

my $req = HTTP::Request-> new(GET => $dmapiURL.'/login?username='.shift().'&password='.shift());
my $res = $ua-> request($req);

unless ($res-> is_success) {
    print "Failed: ", $res-> status_line, "\n";
    exit -1;
}

my $auth=$res-> as_string;
my $output;

if ($auth =~ /Auth-Sid:s*(w+)/ms) {
    $req=HTTP::Request-> new(GET => $dmapiURL.'/query-domain-list?auth-sid='.$1);
    $res = $ua-> request($req);
    if ($res-> is_success) {
        $output=$res-> content;
    }

    else{
        print "Failed: ", $res-> status_line, "\n";
    }
}
```

```

exit -2;
}
}

else {
print "Failed: Got no auth-id from DMAPi:n";
print $res-> content;
exit -3;
}

my @lines=split ("n",$output);
my $month=now + ("2M");
my %list;

my $p=0;
for my $line(@lines) {
#Skip first line
next unless $p or $line =~ /^s*/;
unless ($p) { $p=1; next};
my ($fqdn,$exp)=split ("[ t]+",$line);
$list{$fqdn}=$exp;
}

for (sort keys %list) {
print "$_n";
}

```

List domains to expire next month

Needs:

Class::Date and Date::Parse from CPAN!

```

#!/usr/bin/perl

use LWP::UserAgent;
use Data::Dumper;
use Class::Date qw(:errors date localtime gmdate now -DateParse);

```

```
my $dmapiURL = "https://dmapi.joker.com/request";

my $ua = LWP::UserAgent-> new;
my $req = HTTP::Request-> new(GET => $dmapiURL. '/login?username='.shift().'&password='.shift());
my $res = $ua-> request($req);

unless ($res-> is_success) {
    print "Failed: ", $res-> status_line, "n";
    exit -1;
}

my $auth=$res-> as_string;
my $output;

if ($auth =~ /Auth-Sid:s*(w+)/ms) {
    $req=HTTP::Request-> new(GET => $dmapiURL. '/query-domain-list?auth-sid=.'.$1);
    $res = $ua-> request($req);

    if ($res-> is_success) {
        $output=$res-> content;
    }
    else {
        print "Failed: ", $res-> status_line, "n";
        exit -2;
    }
}
else {
    print "Failed: Got no auth-id from DMAPi:n";
    print $res-> content;
    exit -3;
}

my @lines=split ("n",$output);
my $month=now + ("2M");
my %list;

my $p=0;
```

```

for my $line(@lines) {
#Skip first line
next unless $p or $line =~ /^s*$/;
unless ($p) { $p=1; next};
my ($fqdn,$exp)=split (" t]+$line);
next unless localdate($exp) < $month;
$list{$fqdn}=$exp;
}

for ( map {$_-> [0]}
sort {
$a-> [1] < => $b-> [1]
||
$a-> [2] cmp $b-> [2]
}
map {[$_ , localdate $list{$_},$_]}
keys %list ) {
print "$_:".list{$_}."n";
}

```

Replace admin-c in multiple domains

```

#
# replace admin-c in multiple domains
#
#!/usr/bin/perl
use LWP::UserAgent;
use Data::Dumper;
my $ua = LWP::UserAgent->new;
my $req = HTTP::Request->new(GET =>
'https://dmapi.joker.com/request/login?username='.<shift().&password='.<shift());
my $res = $ua->request($req);
#--Admin-C :
my %ADMIN = ( de =>'CODE-12345' , org => 'CORG-12345', com => 'CCOM-12345', 'eu' => 'c12345' );
unless ($res->is_success) {
    print "Failed: ", $res->status_line, "\n";
    exit -1;
}

```

```

my $auth=$res->as_string;
my $output;
if ($auth =~ /Auth-Sid:\s*([a-z0-9]+)/m ) {
$auth = $1;
} else {
    print "Failed: Got no auth-id from DMAPI:\n";
    print $res->content;
    exit -3;
}
while ( <DATA> ) {
    chomp;
    my $fqdn = $_;
    ## for 3rd level domains this must be changed!
    my ( $sld, $tld ) = split ( /\. / , $fqdn );
    my $admin = $ADMIN{ $tld };
$req=HTTP::Request->new(GET => 'https://dmapi.joker.com/request/domain-modify?domain='.$fqdn.'&admin-
c='.$admin.'&auth-sid='.$auth);
$res = $ua->request($req);
    if ($res->is_success) {
$output=$res->content;
        print $output;
    } else {
        print "Failed: ", $res->status_line, "\n";
        exit -2;
    }
}
__DATA__
domain1.de
domain2.eu
domain3.org
domain4.com
__END__

```

PHP

Introduction

Please note that there is a full DMAPI based interface written in PHP available. It is free to download and modify, and could be [downloaded from Github](#).

Please read more at section [Joker PHP client](#).

The published examples in PHP are parts of this project and are stripped in terms of functionality and error handling. It is strongly recommended to use the complete code base from Github.

List domains of a customer

```
<?php

//sends HTTP request using CURL

function query_host($conn_server, $params = "", $get_header = false)
{

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $conn_server.$params);
    if (preg_match("/^https:\/\//i", $conn_server)) {
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    }

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    if ($get_header) {
        curl_setopt($ch, CURLOPT_HEADER, 1);
    }

    else {
        curl_setopt($ch, CURLOPT_HEADER, 0);
    }

    $result = curl_exec($ch);

    if (curl_errno($ch)) {
        print "curl error";
    }
}
```



```

    }

    else {
        curl_close($ch);
    }

    return $result;
}

//builds query, sends request and gets the answer back

function execute_request($request, $params, &$sessid)
{
    //build the query
    $http_query = "/request/" . $request . "?" . $params . "&auth-sid=".$sessid."&client-
ip=".$_SERVER["REMOTE_ADDR"];
    //send the request
    $raw_res = query_host("https://dmapl.joker.com", $http_query, true);
    $temp_arr = @explode("\r\n\r\n", $raw_res, 2);
    //split the response for further processing
    if (is_array($temp_arr) && 2 == count($temp_arr)) {
        return $temp_arr[1];
    }

    else {
        return false;
    }
}

//basic parsing of the DMAPl header

function parse_response_header($header)
{
    $raw_arr = explode("\n", trim($header));
    $result = array();
    if (is_array($raw_arr)) {
        foreach ($raw_arr as $key => $value)
        {
            $keyval = array();

```

```

        if (preg_match("/^([^\s]+):\s+(.+)\s*$/", $value, $keyval)) {
            $arr[strtolower($keyval[1])] = $keyval[2];
        }

        else {
            print "Header line not parseable - pattern does not match\nRaw header:\n$value";
        }
    }
}

else {
    $arr = "";
    print "Unidentified error\nRaw header:\n$header";
}

return $arr;
}

//parses the reply from DMAP into a header and body
function parse_response($res)
{
    $raw_arr = explode("\n\n", trim($res));
    $arr_elements = count($raw_arr);
    if ($arr_elements > 0) {
        $temp["response_header"] = parse_response_header($raw_arr["0"]);
        $temp["response_body"] = $raw_arr["1"];
    }

    else {
        print "Couldn't split the response into response header and response body\nRaw result:\n$res";
        $temp = "";
    }

return $temp;
}

$response = "";
$authid = "none";
//first obtaining an Auth-ID

```

```

$username = "<your username here>";
$password = "<your password here>";
$fields = "username=".urlencode($username)."&password=".urlencode($password);
$result = execute_request("login", $fields, $authid);
$result = parse_response($result);
$authid = $result["response_header"]["auth-sid"];
if (!$authid) {
    print "no auth-id was obtained - probably due to wrong username or password";
    exit;
}

//now asking for the domain list
$pattern = "*";
$fields = "pattern=".urlencode($pattern);
$result = execute_request("query-domain-list", $fields, $authid);
$result = parse_response($result);

print_r($result["response_header"]);
print "< br /> ";
print_r($result["response_body"]);

```

Python

Introduction

Python is easy to use with DMAPi. Our examples are tested with python 3 on Windows and Linux, but should work on all platforms.

Running the Python example will look like:

```

$ ./dmapi-example.py
Request-URL: https://dmapi.ote.joker.com/request/login
Login: Status-Code: 0

Request-URL: https://dmapi.ote.joker.com/request/query-domain-list
Domain List: Status-Code: 0

```

domain: another-privacy-test.com

expiration_date: 2018-06-30

domain: another-privacy-test.net

expiration_date: 2020-06-30

Request-URL: <https://dmapl.ote.joker.com/request/logout>

Logout: Status-Code: 0

Login and list your domains

```
#!/usr/bin/env python
import requests

dmaplURL = 'https://dmapl.ote.joker.com'
dmaplUser = 'username'
dmaplPassword = 'password'

def main():
    loginResponse = login(dmaplUser,dmaplPassword)
    print("Login: Status-Code:", loginResponse.header['Status-Code'])
    if loginResponse.header['Status-Code'] != '0':
        print(loginResponse.header['Status-Text'])
        return

    sessionId = loginResponse.header['Auth-Sid'];
    print("")
    domainResponse = domainList(sessionId,1,5)
    print("Domain List: Status-Code:", domainResponse.header['Status-Code'])
    print("")
    domains = domainResponse.resultListWithNames()
    for domain in domains:
        for key, value in domain.items():
            print(" %s: %s" % (key, value))
        print("")
    logoutResponse = logout(sessionId)
    print("Logout: Status-Code:", logoutResponse.header['Status-Code'])
```

```

# implement dmapl commands as functions
def login(username,password):
    parameters = { 'username': username, 'password': password }
    message = sendCommand('login', parameters)
    return message;

def logout(sessionId):
    parameters = { 'auth-sid': sessionId }
    message = sendCommand('logout', parameters)
    return message;

def domainList(sessionId, list_from=1, list_to=""):
    parameters = { 'auth-sid': sessionId , 'from': list_from, 'to': list_to }
    message = sendCommand('query-domain-list', parameters)
    return message;

# general dmapl command call
def sendCommand(command,parameter={}):
    try:
        url = dmaplURL+'/request/'+command
        print("Request-URL: ", url)
        response = requests.get(url, params=parameter)
        # print URL with parameters for debugging purposes
        # print("Request-URL: ", response.url)
        if response.status_code != requests.codes.ok:
            raise CommandError("Command Failed! HTTP Status Code: %s" % response.status_code)
        return DmaplResponse(response.text)
    except requests.ConnectionError as e:
        raise CommandError("Connection Error: %s" % str(e))
    except requests.HTTPError as e:
        raise CommandError("Http Error: %s" % str(e))
    except CommandError as e:
        raise
    except Exception as e:
        raise CommandError("Unexpected Error: %s" % str(e))

class DmaplResponse():

```

```
def __init__(self,responseBody):
    parts = responseBody.split("\n\n",1)
    if len(parts)>0:
        self.header = self.__parseKeyValueList(parts[0])
    if len(parts)>1:
        self.body = parts[1]
```

```
def __parseKeyValueList(self,text):
    lines = text.split("\n")
    keyValueList = {}
    for line in lines:
        keyValue = line.split(' ',1)
        key = keyValue[0].rstrip(':')
        value = keyValue[1]
        keyValueList[key] = value
    return keyValueList
```

```
def __getSeparator(self):
    if self.header.get('Separator') == 'TAB':
        return "\t"
    else:
        return " "
```

```
def resultList(self):
    lines = self.body.split("\n")
    resultList = []
    separator = self.__getSeparator()
    for line in lines:
        values = line.split(separator)
        resultList.append(line.split(separator))
    return resultList
```

```
def resultListWithNames(self):
    columnNames = self.resultListColumns()
    resultList = []
    if len(columnNames) > 0:
        rawList = self.resultList()
        resultList = []
        for row in rawList:
```

```

        columns = {}
        for idx, column in enumerate(row):
            columns[columnNames[idx]] = column
        resultList.append(columns)
    return resultList

def resultListColumns(self):
    if 'Columns' in self.header:
        columnsText = self.header['Columns']
        columns = columnsText.split(',')
        return columns
    else:
        return []

def resultValues(self):
    return self.__parseKeyValueList(self.body)

class CommandError(Exception):
    def __init__(self, value):
        self.value = value
    def __str__(self):
        return repr(self.value)

# call main function
try:
    main()
except CommandError as e:
    print("Error:", str(e).strip(""))

```

Login and list all A and CNAME records

```

#!/usr/bin/env python
import requests

dmapiURL = 'https://dmapi.joker.com'
dmapiUser = 'username'
dmapiPassword = 'password'

```

```

def main():
    loginResponse = login(dmapiUser,dmapiPassword)
    #print("Login: Status-Code:", loginResponse.header['Status-Code'])
    if loginResponse.header['Status-Code'] != '0':
        print(loginResponse.header['Status-Text'])
        return

    sessionId = loginResponse.header['Auth-Sid'];
    #print("")
    dnsZoneListResponse = dnsZoneList(sessionId,"")
    #print("DNS Zone List: Status-Code:", dnsZoneListResponse.header['Status-Code'])
    dnslist = dnsZoneListResponse.resultList()
    for row in dnslist:
        domain = row[0]
        expiration = row[1]
        #print(" domain: %s" % (domain))
        #print(" expiration: %s" % (expiration))
        #print("")
        dnsZoneResponse = dnsZoneGet(sessionId, domain)
        #print("DNS Zone GET for %s: Status-Code:" % (domain), dnsZoneResponse.header['Status-Code'])
        zoneEntries = dnsZoneResponse.resultList()
        for entry in zoneEntries:
            #print(' '.join(entry))
            if len(entry)<5:
                continue
            eLabel = entry[0]
            eType = entry[1]
            ePriority = entry[2]
            eTarget = entry[3]
            eTTL = entry[4]
            if eType == 'A' or eType == 'CNAME':
                print(("%.%s\t%s" % (eLabel,domain,eTarget)).lstrip('@.'))

    logoutResponse = logout(sessionId)
    #print "Logout: Status-Code:", logoutResponse.header['Status-Code']

# implement dmapi commands as functions
def login(username,password):

```



```

parameters = { 'username': username, 'password': password }
message = sendCommand('login', parameters)
return message;

def logout(sessionId):
    parameters = { 'auth-sid': sessionId }
    message = sendCommand('logout', parameters)
    return message;

def domainList(sessionId, pattern="", list_from=1, list_to=""):
    parameters = { 'auth-sid': sessionId , 'from': list_from, 'to': list_to, 'pattern': pattern }
    message = sendCommand('query-domain-list', parameters)
    return message;

def dnsZoneList(sessionId, pattern="", list_from=1, list_to=""):
    parameters = { 'auth-sid': sessionId , 'from': list_from, 'to': list_to, 'pattern': pattern }
    message = sendCommand('dns-zone-list', parameters)
    return message;

def dnsZoneGet(sessionId, domain):
    parameters = { 'auth-sid': sessionId , 'domain': domain }
    message = sendCommand('dns-zone-get', parameters)
    return message;

# general dmapi command call
def sendCommand(command,parameter={}):
    try:
        url = dmapiURL+'/request/'+command
        #print("Request-URL: ", url)
        response = requests.get(url, params=parameter)
        # print URL with parameters for debugging purposes
        # print("Request-URL: ", response.url)
        if response.status_code != requests.codes.ok:
            raise CommandError("Command Failed! HTTP Status Code: %s" % response.status_code)
        return DmapiResponse(response.text)
    except requests.ConnectionError as e:
        raise CommandError("Connection Error: %s" % str(e))
    except requests.HTTPError as e:
        raise CommandError("Http Error: %s" % str(e))

```

```
except CommandError as e:
    raise
except Exception as e:
    raise CommandError("Unexpected Error: %s" % str(e))
```

```
class DmapiResponse():
    def __init__(self,responseBody):
        parts = responseBody.split("\n\n",1)
        if len(parts)>0:
            self.header = self.__parseKeyValueList(parts[0])
        if len(parts)>1:
            self.body = parts[1]
```

```
    def __parseKeyValueList(self,text):
        lines = text.split("\n")
        keyValueList = {}
        for line in lines:
            keyValue = line.split(' ',1)
            key = keyValue[0].rstrip(':')
            value = keyValue[1]
            keyValueList[key] = value
        return keyValueList
```

```
    def __getSeparator(self):
        if self.header.get('Separator') == 'TAB':
            return "\t"
        else:
            return " "
```

```
    def resultList(self):
        lines = self.body.split("\n")
        resultList = []
        separator = self.__getSeparator()
        for line in lines:
            values = line.split(separator)
            resultList.append(line.split(separator))
        return resultList
```

```
    def resultListWithNames(self):
```

```

columnNames = self.resultListColumns()
resultList = []
if len(columnNames) > 0:
    rawList = self.resultList()
    resultList = []
    for row in rawList:
        columns = {}
        for idx, column in enumerate(row):
            columns[columnNames[idx]] = column
        resultList.append(columns)
    return resultList

def resultListColumns(self):
    if 'Columns' in self.header:
        columnsText = self.header['Columns']
        columns = columnsText.split(',')
        return columns
    else:
        return []

def resultValues(self):
    return self.__parseKeyValueList(self.body)

class CommandError(Exception):
    def __init__(self, value):
        self.value = value
    def __str__(self):
        return repr(self.value)

# call main function
try:
    main()
except CommandError as e:
    print("Error:", str(e).strip(""))

```

C#

Applications for domain management with DMAPI are very easy to create using MS Visual Studio. This example code logs into the Joker.com system using DMAPI requests.

It provides a domain listing, whois lookup and "Email verification".

This has been tested using Visual Studio 2008 and 2010, but should work with any version, and also with free Visual Studio Express and .NET versions from 3.5 onwards. Please find the source code as an attachment below.

We also provide this as a windows binary to try without the need of building it before. In case you run Windows 10 with "Smartscreen", you have to click on "more information" to execute this, since this binray of course is not signed by a "manufacturer"...

This is how this will look like:

C#/.NET DMAPI Example