EPP specification
it.com domains

v.1.7 (31.10.2023)

Contents

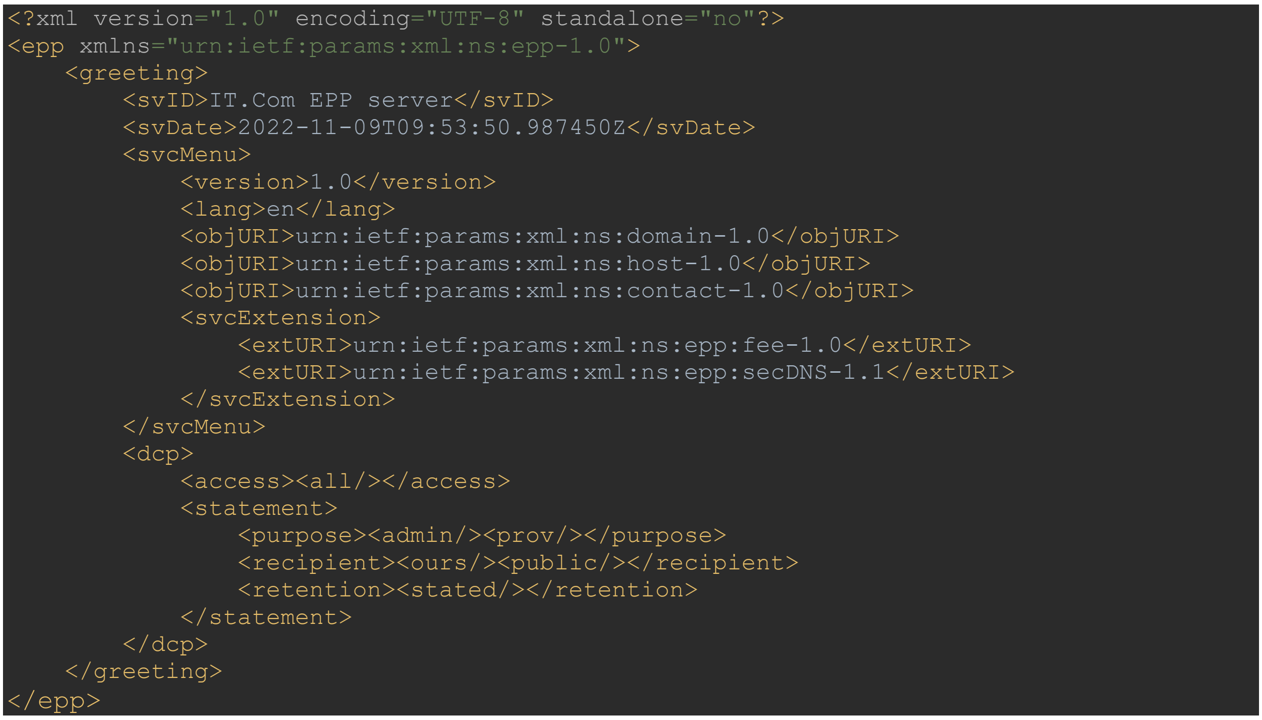
The interaction with the system is carried out through the exchange of parameters (command-response) in XML format.

The connection to the server is carried out via the TCP / TLS protocol using the certificate created during the registration of the client (registrar).

When creating a certificate, the login of the client (registrar) must be specified in the CN block.

1. Introduction

After establishing a TCP/TLS connection, the server sends a greeting element according to RFC 5730:



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <greeting>
        <svID>IT.Com EPP server</svID>
        <svDate>2022-11-09T09:53:50.987450Z</svDate>
        <svcMenu>
            <version>1.0</version>
            <lang>en</lang>
            <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
            <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>
            <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
            <svcExtension>
                <extURI>urn:ietf:params:xml:ns:epp:fee-1.0</extURI>
                <extURI>urn:ietf:params:xml:ns:epp:secDNS-1.1</extURI>
            </svcExtension>
        </svcMenu>
        <dcp>
            <access><all/></access>
            <statement>
                <purpose><admin/><prov/></purpose>
                <recipient><ours/><public/></recipient>
                <retention><stated/></retention>
            </statement>
        </dcp>
    </greeting>
</epp>
```

Production and OT&E Environments

Registrars should create unique logins and SSL certificates in each of these environments separately. In each environment, you can add your IP addresses for protected access and you can generate signed SSL Certificates by clicking on the EPP Settings tab.

Production Dashboard:	https://registrar.it.com/login
Production EPP Hostname:	epp.it.com
Production WHOIS Hostname:	whois.it.com
Production RDAP Hostname:	rdap.it.com
OT&E Dashboard:	https://ote-registrar.it.com/login
OT&E EPP Hostname:	ote-epp.it.com
OT&E WHOIS Hostname:	ote-whois.it.com
OT&E RDAP Hostname:	ote-rdap.it.com

API key creation

There are two ways to create an API key to work with our EPP service. You may automatically create a signed key and certificate, or you may attach your own Certificate Signing Request (CSR), and we will send you a signed SSL Certificate. This can be done inside either the OT&E or Production dashboards under EPP Settings.

2. login - start a session

Authenticates the user and starts the session.

The <clID> element must contain the login specified during registration.

The <pw> element must contain a password. The password is case sensitive.

It also checks the validity of the user certificate and the presence of the IP address from which the connection was made in the user's White List, if it is filled in the registrar's account.

The <clTRID> element must contain the session identifier. The identifier must be present in other commands.

The remaining elements of the command must match those specified in the example.

Example:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <login>
            <clID>epp@it.com</clID>
            <pw>XXXXXXXX</pw>
            <options>
                <version>1.0</version>
                <lang>en</lang>
            </options>
            <svcs>
                <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
                <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
                <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>
                <svcExtension>
                    <extURI>urn:ietf:params:xml:ns:epp:fee-1.0</extURI>
                    <extURI>urn:ietf:params:xml:ns:secDNS-1.1</extURI>
                </svcExtension>
            </svcs>
        </login>
        <clTRID>IT.COM-login-1656487504012</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results:

- 1000 - Command completed successfully
- 2200 - User not found
- 2200 - User not registrar
- 2200 - User is blocked
- 2200 - Bad password
- 2200 - Bad certificate
- 2200 - IP is not in white list for user

Example

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <trID>
            <clTRID>IT.COM-login-1656487504012</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

3. logout - ending the session

Ends the session with the ID specified in the <clTRID> element.

The remaining elements of the command must match those specified in the example.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <logout/>
        <clTRID>IT.COM-logout-1656496559772</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results:

- 1500 - Command completed successfully; ending session

Example

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1500">
            <msg>Command completed successfully; ending session</msg>
        </result>
        <trID>
            <clTRID>IT.COM-logout-1656496559772</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

4. domain:check - checks if the domain exists in the registry

Checks if the registry contains a domain with the name specified in the <domain:name> element. Multiple <domain:name> elements are allowed if multiple domain names need to be validated.

The <clTRID> element must contain the session identifier.

If you need to get information about the cost of registering a domain, you can use the <extension> element with the <fee:check> element. The <fee:check> element must contain one or more <fee:command> elements with a name parameter that can take the following values:

- create - domain lease
- renew – domain lease extension

The <fee:command> element must contain one or more <fee:period unit="y"> elements (unit can only take the value "y"), with a period value in years for which the price is requested. Value must be integers from 1 to 10

The remaining elements of the command must match those specified in the example.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <check>
            <domain:check>
                <domain:name>test-epp3.it.com</domain:name>
            </domain:check>
        </check>
        <extension>
            <fee:check xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">
                <fee:command name="create">
                    <fee:period unit="y">1</fee:period>
                </fee:command>
            </fee:check>
        </extension>
        <clTRID>TEST-domain-check-1656507674719</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results :

- 1000 - Command completed successfully
- 2004 - Parameter value range error
- 2004 - Command failed
- 2201 - Authorization error

In the case of a result code of 1000, the domain name and the avail parameter are returned in the <domain:name> element , which can be equal to:

- "1" - the domain is available for registration
- "0" - the domain is not available for registration

If the value is "0", a <domain:reason> element is also returned, containing a description of the reason for the domain name being unavailable.

If the command had a <fee:check> element, the response includes an <extension> element with a <fee:chkData> element. It contains information about the domain class (standard, premium) and the price of registration and (or) renewal for terms in accordance with the parameters passed in the command.

Example of a response if the domain is **available**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <domain:chkData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:cd>
                    <domain:name avail="1">test-epp3.it.com</domain:name>
                </domain:cd>
            </domain:chkData>
        </resData>
        <extension>
            <fee:chkData xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">
                <fee:currency>USD</fee:currency>
                <fee:cd avail="1">
                    <fee:objID>test-epp3.it.com</fee:objID>
                    <fee:class>standard</fee:class>
                    <fee:command name="create">
                        <fee:period unit="y">1</fee:period>
                        <fee:fee>25.00</fee:fee>
                    </fee:command>
                </fee:cd>
            </fee:chkData>
        </extension>
        <trID>
            <clTRID>TEST-domain-check-1656507674719</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

Example of a response if the domain is **not available**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <domain:chkData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:cd>
                    <domain:name avail="0">test-epp3.it.com</domain:name>
                    <domain:reason>In use</domain:reason>
                </domain:cd>
            </domain:chkData>
        </resData>
        <extension>
            <fee:chkData xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">
                <fee:currency>USD</fee:currency>
                <fee:cd avail="0">
                    <fee:objID>test-epp3.it.com</fee:objID>
                    <fee:reason>In use</fee:reason>
                </fee:cd>
            </fee:chkData>
        </extension>
        <trID>
            <clTRID>TEST-domain-check-1656507674719</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

Note: the *fee extension is only implemented for the <domain:check> command*

5. domain:info – Retrieval of information about the domain

Returns information about the domain whose name is specified in the <domain:name> element.

The <clTRID> element must contain the session ID.

The remaining elements of the command must match those specified in the example.

Example command:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <info>
            <domain:info>
                <domain:name>test-epp2.it.com</domain:name>
            </domain:info>
        </info>
        <clTRID>IT.COM-domain-info-1656505411798</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results :

- 1000 - Command completed successfully
- 2303 - Object does not exist

In the case of a result code of 1000, the <domain:infData> element contains information about the domain, as shown in the examples. If the domain was registered by the registrar on whose behalf the request was made, the full domain information is returned in the response. In other cases - abbreviated information.

If the session in which the command is received uses the secDNS-1.1 extension (the login command has an <extURI> element with the value "urn:ietf:params:xml:ns:secDNS-1.1") and SEC DNS information exists for the domain, the response with complete information about the domain, an <extension> element will be included with a nested <secDNS:infData> element that contains the elements:

- <secDNS:maxSigLife> - DS record lifetime in seconds One or more
- <secDNS:dsData> elements, each containing the following elements:
    - <secDNS:keyTag> - key tag value
    - <secDNS:alg> - algorithm value
    - <secDNS:digestType> - digest type
    - <secDNS:dsData> - digest value

An example of a response if the command is executed by the registrar who registered the domain.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:name>test-epp2.it.com</domain:name>
                <domain:roid>1110</domain:roid>
                <domain:clID>epp@it.com</domain:clID>
                <domain:status s="ok"/>
                <domain:registrant>IT.COM-REGISTRANT132</domain:registrant>
                <domain:contact type="admin">IT.COM-ADMIN132</domain:contact>
                <domain:contact type="tech">IT.COM-TECH132</domain:contact>
                <domain:contact type="billing">IT.COM-BILLING132</domain:contact>
                <domain:ns>
                    <domain:hostObj>ns1.dn.it.com</domain:hostObj>
                    <domain:hostObj>ns2.dn.it.com</domain:hostObj>
                </domain:ns>
                <domain:crDate>2022-06-29T12:22:53.528469Z</domain:crDate>
                <domain:exDate>2023-06-30T00:00:00Z</domain:exDate>
                <domain:authInfo>
                    <domain:pw>XXXXXXXXX</domain:pw>
                </domain:authInfo>
            </domain:infData>
        </resData>
        <extension>
            <secDNS:infData xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
                <secDNS:dsData>
                    <secDNS:keyTag>12345</secDNS:keyTag>
                    <secDNS:alg>3</secDNS:alg>
                    <secDNS:digestType>1</secDNS:digestType>
                    <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
                </secDNS:dsData>
            </secDNS:infData>
       </extension>
        <trID>
            <clTRID>IT.COM-domain-info-1656505411798</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

An example of the answer in other cases.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:name>test-epp2.it.com</domain:name>
                <domain:roid>1110</domain:roid>
                <domain:clID>epp@it.com</domain:clID>
            </domain:infData>
        </resData>
        <trID>
            <clTRID>IT.COM-domain-info-1656505411798</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

6. contact:check - check if the contact exists in the registry

Checks if the registry contains a contact with the ID specified in the <contact:id> element.

The <clTRID> element must contain the session ID.

The remaining elements of the command must match those specified in the example.

Example command :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <check>
            <contact:check>
                <contact:id>IT.COM-REGISTRANT132</contact:id>
            </contact:check>
        </check>
        <clTRID>IT.COM-contact-check-165650537333</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results :

- 1000 - Command completed successfully

The <contact:id> element returns the contact ID and the avail parameter , which can be equal to:

- "1" - the contact is available for registration
- "0" - the contact is not available for registration

If the value is "0", a <contact:reason> element is also returned, containing a description of the reason for the contact being unavailable.

Sample response if contact is available

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <contact:chkData xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
                <contact:cd>
                    <contact:id avail="1">IT.COM-REGISTRANT132</contact:id>
                </contact:cd>
            </contact:chkData>
        </resData>
        <trID>
            <clTRID>IT.COM-contact-check-165650537333</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

Example of a response if the contact is not available

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <contact:chkData xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
                <contact:cd>
                    <contact:id avail="0">IT.COM-REGISTRANT132</contact:id>
                    <contact:reason>In use</contact:reason>
                </contact:cd>
            </contact:chkData>
        </resData>
        <trID>
            <clTRID>IT.COM-contact-check-165650537333</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

7. host:check - checks if the host exists the registry

Checks if the registry contains a host with the name specified in the <host:name> element.

The <clTRID> element must contain the session identifier.

The remaining elements of the command must match those specified in the example.

Example command :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <check>
            <host:check>
                <host:name>ns1.dn.it.com</host:name>
            </host:check>
        </check>
        <clTRID>IT.COM-host-check-1656505373469</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results :

- 1000 - Command completed successfully

<host:id> element returns the host name and the avail parameter , which can be equal to:

- "1" - the host is available for registration
- "0" - the host is not available for registration

If the value is '0', a <host:reason> element is also returned, containing a description of the reason for the host being unavailable.

Example response if the host is **available for creation**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <host:chkData xmlns:host="urn:ietf:params:xml:ns:host-1.0">
                <host:cd>
                    <host:id avail="1">ns1.dn.it.com</host:id>
                </host:cd>
            </host:chkData>
        </resData>
        <trID>
            <clTRID>IT.COM-host-check-1656505373469</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

Sample response if the host is **not available/exists**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <host:chkData xmlns:host="urn:ietf:params:xml:ns:host-1.0">
                <host:cd>
                    <host:id avail="0">ns1.dn.it.com</host:id>
                    <host:reason>In use</host:reason>
                </host:cd>
            </host:chkData>
        </resData>
        <trID>
            <clTRID>IT.COM-host-check-1656505373469</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

8. contact:info – Retrieval of the information about a contact

Checks for the presence in the registry system of a contact with the name specified in the <host:name> element.

Example command :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <info>
            <contact:info>
                <contact:id>IT.COM-BILLING132</contact:id>
            </contact:info>
        </info>
        <clTRID>IT.COM-contact-info-1656512737831</clTRID>
    </command>
</epp>
```

Example response :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <contact:infData xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
                <contact:id>IT.COM-BILLING132</contact:id>
                <contact:status s="linked"/>
                <contact:postalInfo type="int">
                    <contact:name>Pavel Kuzovkin</contact:name>
                    <contact:org>Company</contact:org>
                    <contact:addr>
                        <contact:street>Street Street</contact:street>
                        <contact:city>City</contact:city>
                        <contact:pc>kuzovkin</contact:pc>
                        <contact:cc>GB</contact:cc>
                    </contact:addr>
                </contact:postalInfo>
                <contact:voice>+44.123453445665</contact:voice>
                <contact:email>pavel_kuzovkin@mail.ru</contact:email>
            </contact:infData>
        </resData>
        <trID>
            <clTRID>IT.COM-contact-info-1656512737831</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

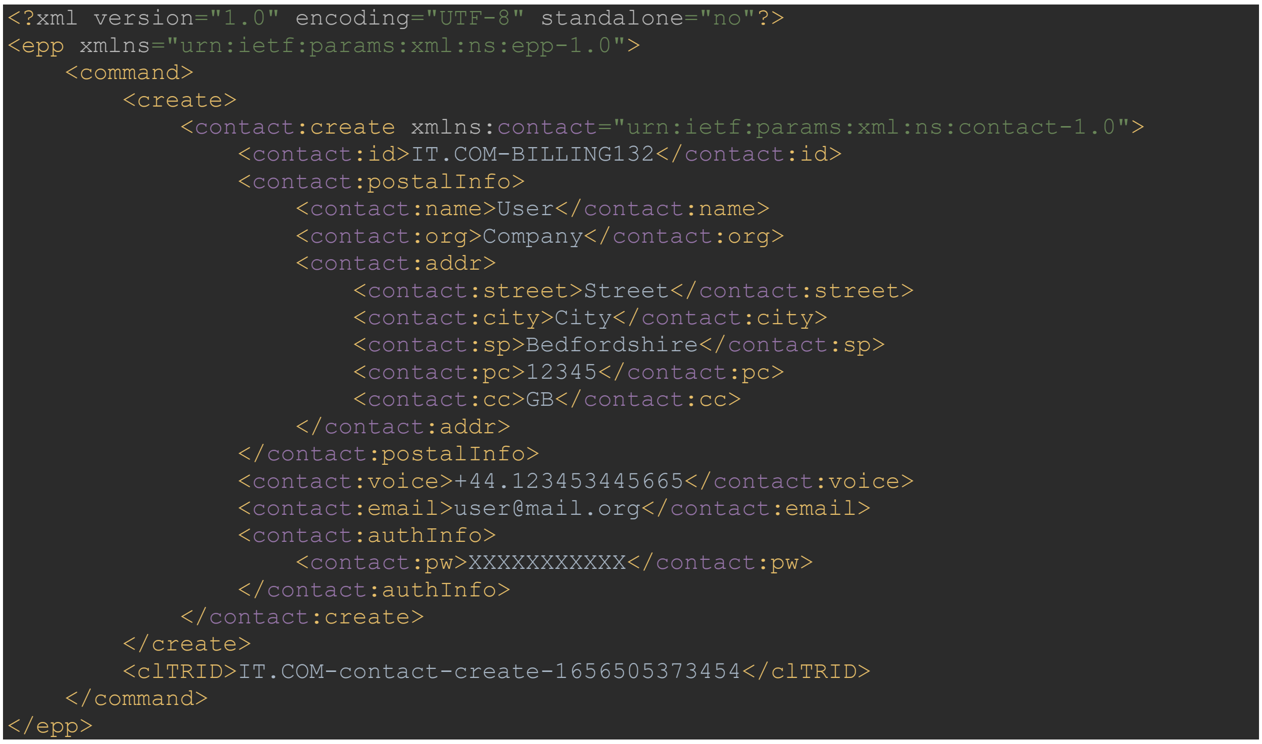
9. contact:create - create a contact in the registry

Creates a contact in the registry. The command must contain a <contact:create> element with the following elements:

- <contact:id> - unique identifier for the created contact
- <contact:postalInfo> - postal information of the contact. Only the element type is int (parameter type="int"). If the parameter is not specified, the int type is assumed by default. If type loc is specified, the command will fail with an error. There must be one such element. Contains elements:
    - <contact:name> - contact name
    - <contact:org> - the name of the organization the contact is associated with
    - <contact:addr> - the address information associated with the contact. It contains the following elements:
        - <contact:street> - street. There may be several such elements.
        - <contact:city> - city
        - <contact:sp> - state, region, region
        - <contact:pc> - zip code/po box
        - <contact:cc> - country
    - <contact:voice> - phone number
    - <contact:email> - email address
    - <contact:fax> - fax phone number
    - <contact:authInfo> This must contain an element which will contain <contact:pw> - the authorization code for the contact

The <clTRID> element must contain the session ID.

Example command :



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <create>
            <contact:create xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
                <contact:id>IT.COM-BILLING132</contact:id>
                <contact:postalInfo>
                    <contact:name>User</contact:name>
                    <contact:org>Company</contact:org>
                    <contact:addr>
                        <contact:street>Street</contact:street>
                        <contact:city>City</contact:city>
                        <contact:sp>Bedfordshire</contact:sp>
                        <contact:pc>12345</contact:pc>
                        <contact:cc>GB</contact:cc>
                    </contact:addr>
                </contact:postalInfo>
                <contact:voice>+44.123453445665</contact:voice>
                <contact:email>user@mail.org</contact:email>
                <contact:authInfo>
                    <contact:pw>XXXXXXXXXXX</contact:pw>
                </contact:authInfo>
            </contact:create>
        </create>
        <clTRID>IT.COM-contact-create-1656505373454</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results :

- 1000 - Command completed successfully
- 2201 - Authorization error

In the case of a result code of 1000, the following elements are returned in the response:

- <contact:id> - contact ID
- <contact:crDate> - date/time the contact was created

Example response :

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <contact:creData xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
                <contact:id>IT.COM-BILLING132</contact:id>
                <contact:crDate>2022-06-29T12:22:53.465712Z</contact:crDate>
            </contact:creData>
        </resData>
        <trID>
            <clTRID>IT.COM-contact-create-1656505373454</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

10.domain:create - Creating a domain name

Creates a domain in the registry system. The command must contain a <domain:create> element with the following elements:

- <contact:id> - unique identifier for the created contact
- <domain:name> - domain name
- <domain:period> with unit parameter - domain registration period. The parameter can take the value " y " - the value is specified in years. Element value must be between 1 and 10
- <domain:ns> - can contain up to 6 <domain:hostObj> or <domain:hostAttr> elements, which contains the NS name of the domain server. Hosts for the specified names are created automatically.
- The <domain:hostAttr> elements are used when specifying the NS server with the Glue record. It contains nested elements:
  - <domain:hostName> - full name of the NS server. The name must belong to the domain being created
  - Zero or more <domain:hostAddr> elements - the IP address of the NS server. Each element can contain an IP attribute, which is the type of the IP address. Valid attribute values are "v4", "v6". If the attribute is absent, the type "v4" is assumed by default. The meaning of the element is the IP-address of the corresponding type.
- <domain:registrant> - registrar contact ID
- <domain:contact> - identifier contact types admin, tech, billing. The contact type is indicated by the type parameter
- <contact:authInfo> This must contain the element <contact:pw> - the authorization code for the domain

The command can also contain an <extension> element with a nested <secDNS:create> element that contains the following elements:

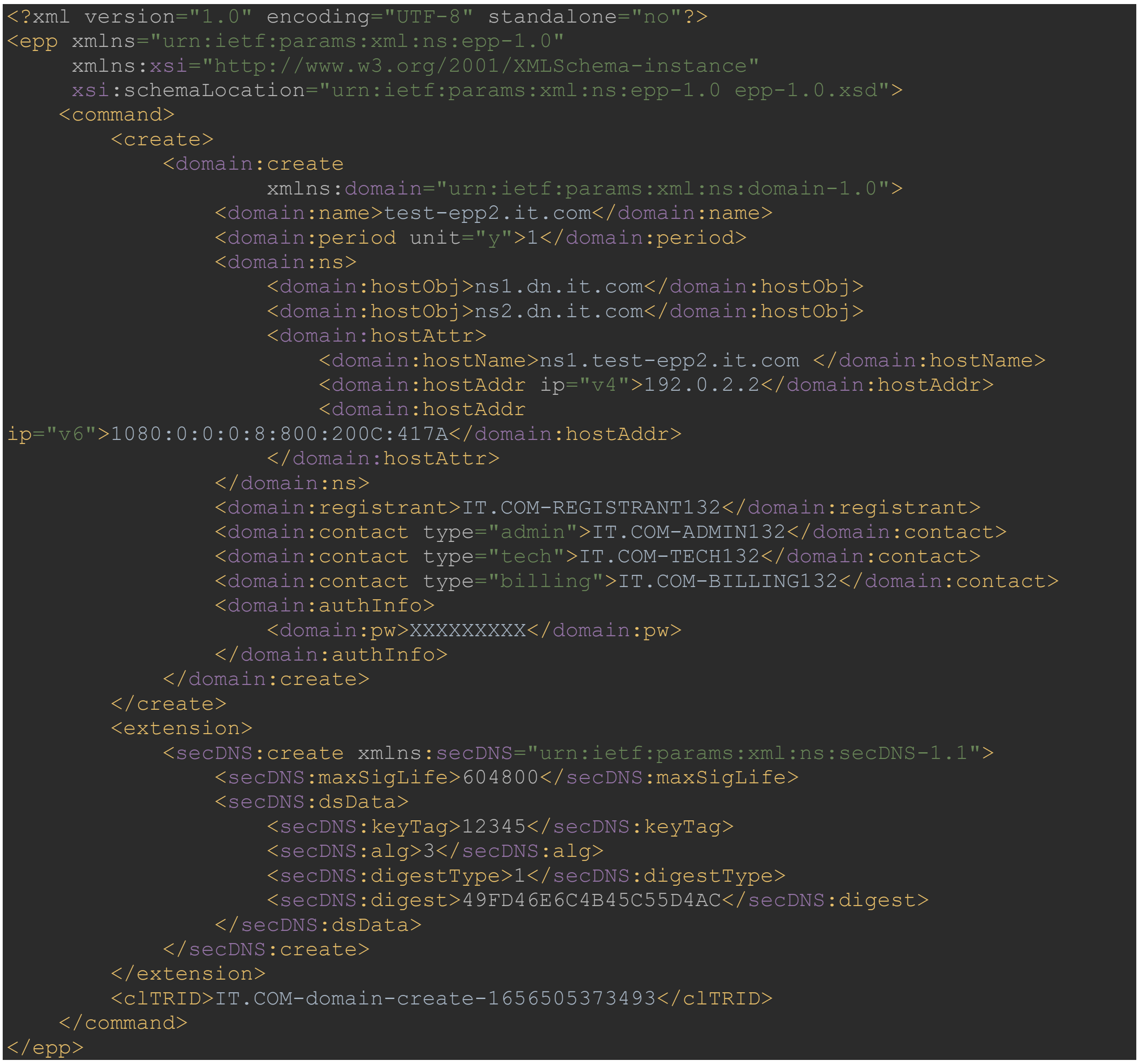
- <secDNS:maxSigLife> - DS record lifetime in seconds
- One or more <secDNS:dsData> elements, each containing the following elements:
  - <secDNS:keyTag> - key tag value
  - <secDNS:alg> - algorith value
  - <secDNS:digestType> - digest type
  - <secDNS:dsData> - digest value

The <secDNS:create> element will be processed only if the session in which the command is received uses the secDNS-1.1 extension (the login command contains an <extURI> element with the value "urn:ietf:params:xml:ns:secDNS-1.1") .

Note: Adding SEC DNS information for a domain is only possible if external NS servers are used for the domain (ns1.it.com, ns2.it.com, ns3.it.com servers should not be used). Otherwise, error 2305, "Object association prohibits operation" will be returned.

The <clTRID> element must contain the session ID.

Example command :



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
    <command>
        <create>
            <domain:create
                    xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:name>test-epp2.it.com</domain:name>
                <domain:period unit="y">1</domain:period>
                <domain:ns>
                    <domain:hostObj>ns1.dn.it.com</domain:hostObj>
                    <domain:hostObj>ns2.dn.it.com</domain:hostObj>
                    <domain:hostAttr>
                        <domain:hostName>ns1.test-epp2.it.com </domain:hostName>
                        <domain:hostAddr ip="v4">192.0.2.2</domain:hostAddr>
                        <domain:hostAddr
ip="v6">1080:0:0:0:8:800:200C:417A</domain:hostAddr>
                    </domain:hostAttr>
                </domain:ns>
                <domain:registrant>IT.COM-REGISTRANT132</domain:registrant>
                <domain:contact type="admin">IT.COM-ADMIN132</domain:contact>
                <domain:contact type="tech">IT.COM-TECH132</domain:contact>
                <domain:contact type="billing">IT.COM-BILLING132</domain:contact>
                <domain:authInfo>
                    <domain:pw>XXXXXXXXX</domain:pw>
                </domain:authInfo>
            </domain:create>
        </create>
        <extension>
            <secDNS:create xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
                <secDNS:maxSigLife>604800</secDNS:maxSigLife>
                <secDNS:dsData>
                    <secDNS:keyTag>12345</secDNS:keyTag>
                    <secDNS:alg>3</secDNS:alg>
                    <secDNS:digestType>1</secDNS:digestType>
                    <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
                </secDNS:dsData>
            </secDNS:create>
        </extension>
        <clTRID>IT.COM-domain-create-1656505373493</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results :

- 1000 - Command completed successfully
- 2303 - object does not exist (when trying to create a domain in a zone not supported by the registry)
- 2302 - Object exists
- 2004 - Parameter value range error (when specifying a lease term outside the range from 1 to 10 years)
- 2100 - Unimplemented protocol version
- 2001 - Command syntax error
- 2102 - Unimplemented option
- 2305 - Object association prohibits operation

In the case of a result code of 1000, the following elements are returned in the response:

- <domain:name> - the name of the created domain name
- <domain:crDate> - date/time of domain name creation
- <domain:ex Date> - date/time

Example response :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:name>test-epp2.it.com</domain:name>
                <domain:crDate>2022-06-29T12:22:53.528469Z</domain:crDate>
                <domain:exDate>2023-06-30T00:00:00Z</domain:exDate>
            </domain:creData>
        </resData>
        <trID>
            <clTRID>IT.COM-domain-create-1656505373493</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

11.contact:update – Contact Modification

Modifies the contact with the specified ID in the registry. The command must contain a <contact:update> element with elements similar to those described for the contact:create command, except for <contact:authInfo>.

The <clTRID> element must contain the session identifier.

Example command :

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <update>
            <contact:update xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
                <contact:id>IT.COM-REGISTRANT132</contact:id>
                <contact:chg>
                    <contact:postalInfo type="int">
                        <contact:name>User</contact:name>
                        <contact:org>New Company</contact:org>
                        <contact:addr>
                            <contact:street>Street Street</contact:street>
                            <contact:city>City</contact:city>
                            <contact:sp></contact:sp>
                            <contact:pc> </contact:pc>
                            <contact:cc>GB</contact:cc>
                        </contact:addr>
                    </contact:postalInfo>
                    <contact:voice>+44.123453445665</contact:voice>
                    <contact:fax></contact:fax>
                    <contact:email>user@mail.org</contact:email>
                </contact:chg>
            </contact:update>
        </update>
        <clTRID>IT.COM-contact-chg-1656513842717</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results :

- 1000 - Command completed successfully
- 2201 - Authorization error
- 2303 - Object does not exist

Example response :

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <trID>
            <clTRID>IT.COM-contact-chg-1656513842717</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

12.domain:update - domain name update

Updates the DNS nameservers for the domain in the registry. The command must contain the <domain:update> element with the following elements:

- <domain:name> is the name of the domain for which the changes are being made
- **OPTIONAL** <domain:add> element containing attribute values to be added to the object
- **OPTIONAL** <domain:rem> element containing attribute values to be removed from the object

The <domain:add> and <domain:rem> elements contain the following child elements

- **OPTIONAL** <domain:ns> element, which can contain up to 6 <domain:hostObj> or <domain:hostAttr> elements, which contains the name servers. Hosts for new names are created automatically. The content of the <domain:hostAttr> element corresponds to that described for the domain:create command. If the <domain:rem> element contains a <domain:hostAttr> element, then its nested <domain:hostAddr> elements are ignored and any existing Glue records corresponding to the NS server specified in the <domain:hostName> element are removed.

- Zero or more <domain:status> elements with an “s” parameter containing the status code to be applied to or removed from the domain. The list of statuses that registrars can assign to a domain is given in the Custom domain statuses section.

The command can also contain an <extension> element with a nested <secDNS:update> element that contains at least one of the <secDNS:add>, <secDNS:rem>, <secDNS:chg> elements. The "urgent" attribute on the <secDNS:update> element is not supported. Error 2102 "Unimplemented option" will be returned if a command with "urgent" attribute set to "true" is received.

Possible nested <secDNS:update> elements:

- <secDNS:rem> - remove SEC DNS elements. May contain an element <secDNS:all>true</secDNS:all> to remove all DNS SEC information, or one or more <secDNS:dsData> elements with nested elements:
  - <secDNS:keyTag> - keytag value
  - <secDNS:alg> - algorithm value
  - <secDNS:digestType> - digest type
  - <secDNS:dsData> - digest value

- <secDNS:add>> - add SEC DNS elements. Contains one or more <secDNS:dsData> elements with nested elements as described above
- <secDNS:chg> - change the lifetime of SEC DNS elements. Contains a nested <secDNS:maxSigLife> element that contains the number of seconds to live for DNS SEC elements. The specified value will be set for all DNS SEC elements of the domain, both those added by this command and those that already existed.

The <secDNS:create> element will be processed only if the session in which the command is received uses the secDNS-1.1 extension (the login command contains an <extURI> element with the value "urn:ietf:params:xml:ns:secDNS-1.1") .

Note: Adding SEC DNS information for a domain is only possible if external NS servers are used for the domain (ns1.it.com, ns2.it.com, ns3.it.com servers should not be used). Otherwise, error 2305, "Object association prohibits operation" will be returned. Also, this error will be returned if you try to change the NS servers to ns1.it.com, ns2.it.com, ns3.it.com for a domain that has SEC DNS entries

The <clTRID> element must contain the session ID.

Example command :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <update>
            <domain:update xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:name>api-test.it.com</domain:name>
                <domain:add>
                    <domain:ns>
                    <domain:hostAttr>
                        <domain:hostName>ns1.api-test.it.com</domain:hostName>
                        <domain:hostAddr>192.0.2.2</domain:hostAddr>
                    </domain:hostAttr>
                    </domain:ns>
                    <domain:status s="clientHold"></domain:status>
                </domain:add>
                <domain:rem>
                    <domain:ns>
                        <domain:hostObj>ns1.it.com</domain:hostObj>
                        <domain:hostObj>ns2.it.com</domain:hostObj>
                        <domain:hostObj>ns3.it.com</domain:hostObj>
                    </domain:ns>
                    <domain:status s="clientUpdateProhibited"></domain:status>
                </domain:rem>
            </domain:update>
        </update>
        <extension>
            <secDNS:update xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
                <secDNS:rem>
                    <secDNS:dsData>
                        <secDNS:keyTag>12345</secDNS:keyTag>
                        <secDNS:alg>3</secDNS:alg>
                        <secDNS:digestType>1</secDNS:digestType>
                        <secDNS:digest>38EC35D5B3A34B33C99B</secDNS:digest>
                    </secDNS:dsData>
                </secDNS:rem>
                <secDNS:add>
                    <secDNS:dsData>
                        <secDNS:keyTag>12346</secDNS:keyTag>
                        <secDNS:alg>3</secDNS:alg>
                        <secDNS:digestType>1</secDNS:digestType>
                        <secDNS:digest>38EC35D5B3A34B44C39B</secDNS:digest>
                    </secDNS:dsData>
                </secDNS:add>
                <secDNS:chg>
                    <secDNS:maxSigLife >605900<secDNS:maxSigLife >
                </secDNS:chg>
            </secDNS:update>
       </extension>
        <clTRID>IT.COM-domain-update-1667979341872</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible result:

- 1000 - Command completed successfully
- 2201 - Authorization error
- 2303 - Object does not exist
- 2001 - Command syntax error
- 2100 - Unimplemented protocol version
- 2102 - Unimplemented option
- 2305 - Object association prohibits operation

Example response :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <trID>
            <clTRID>IT.COM-domain-update-1667979341872</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

13.domain:renew – Domain Name Renewal

Extends the validity of a domain by the specified number of years. The command must contain a <domain:renew> element, which must contain the elements:

- <domain:name> - domain name
- <domain:curExpDate> - current domain expiration date. To exclude an erroneous re-renewal of a domain when a renewal command is received again
- <domain:period> with unit parameter - domain renewal period. The parameter can take the value "y" - the value is specified in years. Element value must be between 1 and 10

The <clTRID> element must contain the session ID.

Example command:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <renew>
            <domain:renew>
                <domain:name>api-test.it.com</domain:name>
                <domain:curExpDate>2024-11-02</domain:curExpDate>
                <domain:period unit="y">1</domain:period>
            </domain:renew>
        </renew>
        <clTRID>IT.COM-domain-renew-1667988957382</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible result:

- 1000 - Command completed successfully
- 2201 - Authorization error
- 2303 - Object does not exist
- 2304 - Object status prohibits operation
- 2005 - Parameter value syntax error
- 2004 - Parameter value range error
- 2104 - Billing failure

Example response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <resData>
            <domain:renData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:name> api-test.it.com </domain:name>
                <domain:exDate>2025-11-02T00:00:00Z</domain:exDate>
            </domain:renData>
        </resData>
        <trID>
            <clTRID>IT.COM-domain-renew-1667988957382</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

14.Domain Name Transfer Process

The procedure for transferring domains between registrars is as follows:

1. The domain owner (registrar's client) receives AuthInfo (EPP code) for the domain from the registrar through which the domain is currently registered (hereinafter referred to as the registrar - sender). It also requests the removal of the clientTransferProhibited and serverTransferProhibited statuses, if such statuses are set.
2. The domain owner, through the registrar - the recipient of the domain, forms a request to transfer the domain and provides AuthInfo.
3. Registrar - the recipient sends the EPP domain:transfer command to the registry to create a transfer request. You can specify the domain validity period in the command. This period starts from the moment the transfer request is made and is 1 year by default.
4. The registry notifies the originating registrar of the domain transfer request via the poll messaging system. Registrars must receive messages from the registry using the poll command.
5. The sender registrar has 120 hours to approve or reject the domain transfer. After the expiration of this period, the transfer is automatically confirmed.
6. After confirmation of the transfer, the domain is transferred to the recipient registrar within 60 seconds. All domain-related objects (NS and DNS records, WHOIS contacts) are transferred to the receiving registrar.

15.domain:transfer - creating, confirming, rejecting, canceling a domain transfer

The command is used to manage domain transfers. The command must contain the <transfer> element with the “op” parameter, the value of which determines the operation performed by the command:

- request – creating a transfer request
- cancel – cancellation of the transfer by the receiving registrar that created the transfer request
- approve – confirmation of the transfer by the sending registrar
- reject – transfer rejection by the sending registrar

The <transfer> element must contain a <domain:transfer> element containing the following elements:

- A <domain:name> element containing the fully qualified name of the domain for which the transfer request is to be created, approved, denied, or cancelled.
- An **OPTIONAL** <domain:period> element containing the number of years to be added to the domain's registration period after the transfer process is complete. The element may contain a "unit" parameter with a value of "y". The value of the element must be between 1 and 10. This element will only be used when creating a transfer request, and will be ignored in all other cases.

- An **OPTIONAL** <domain:authInfo> element containing domain authorization information (EPP code). Will only be used when creating a transfer request, and will be ignored in all other cases.

Example command:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
    <command>
        <transfer op="request">
            <domain:transfer
                    xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:name>test-transfer2.it.com</domain:name>
                <domain:period unit="y">1</domain:period>
                <domain:authInfo>
                    <domain:pw>jGp8c?_7w@r7m$J%</domain:pw>
                </domain:authInfo>
            </domain:transfer>
        </transfer>
        <clTRID>IT.COM-domain-transfer-1668441440156</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results:

- 1000 - Command completed successfully
- 2200 - Authentication error
- 2202 - Invalid authorization information
- 2300 - Object pending transfer
- 2303 - Object does not exist
- 2304 - Object status prohibits operation
- 2004 - Parameter value range error (when specifying a lease term outside the range from 1 to 10 years)

In the case when code 1000 is returned, the <domain:trnData> element containing the child elements is returned in the response:

- <domain:name> containing the fully qualified name of the domain object
- <domain:trStatus> containing the status of the most recent transfer request
- <domain:reID> containing the identifier of the receiving registrar
- <domain:reDate> containing the date and time of the transfer request
- <domain:acID> containing the ID of the sending client
- <domain:acDate> containing the date and time of the requested or completed response. For a pending request, the value specifies the date and time by which a response is required before the server takes automatic response action. For all other statuses, the value specifies the date and time the request was completed.
- <domain:exDate>, contains the expiration of the domain object if the transfer request was or will be successful

The element <domain:trStatus> can contain the following values:

- pending
- clientApproved
- clientCancelled
- clientRejected
- serverApproved
- serverCancelled

Example response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1001">
            <msg>Command completed successfully; action pending</msg>
        </result>
        <resData>
            <trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:name>test-transfer2.it.com</domain:name>
                <domain:trStatus>pending</domain:trStatus>
                <domain:reID>registrar1@it.com</domain:reID>
                <domain:reDate>2022-11-14T15:57:20.190425Z</domain:reDate>
                <domain:acID>registrar2@it.com</domain:acID>
                <domain:acDate>2022-11-16T15:57:20.190425Z</domain:acDate>
                <domain:exDate>2023-11-14T15:57:20.190425Z</domain:exDate>
            </trnData>
        </resData>
        <trID>
            <clTRID>IT.COM-domain-transfer-1668441440156</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

16.poll – Retrieving messages from the registry system

The poll system is used to detect and retrieve service messages queued by the server for individual registrars. If the message queue for the registrar is not empty, then a successful response to the poll command returns the first message from the message queue. Each response returned from the server includes the server's unique message ID, which MUST be provided to acknowledge receipt of the message, and a counter indicating the number of messages in the queue. Once a message has been received by a registrar, it MUST respond to the message with an explicit acknowledgment to acknowledge that the message has been received. The server dequeues the message and dequeues the queue counter upon receipt of an acknowledgment from the registrar, making the next message in the queue (if any) available for retrieval.

The command contains the <poll> element with the “op” parameter, which takes the following values:

- req - get the first message from the queue
- ack – confirmation of message receipt. In this case, the <poll> element must also contain the “msgID” parameter containing a unique message identifier, the value of which corresponds to the value of the "id" attribute copied from the <msg> element of the response to the <poll> command

Example command <poll op=”req”>:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <poll op="req"></poll>
        <clTRID>637250b05f32b</clTRID>
    </command>
</epp>
```

The response returns the code and description of the result of the command execution and the client session ID.

Possible results:

- 1300 - Command completed successfully; no messages
- 1301 - Command completed successfully; ack to dequeue

Moreover, the following elements are returned in the response:

  - **OPTIONAL** <msqQ> element. Present if there are messages in the queue. Contains parameters:
    - qDate – date and time when the message was added to the queue
    - msg - message text
  - **OPTIONAL** resData element is a block of data associated with the message. The example shows a data block for a message associated with receiving a domain transfer request.
- Example response <poll op=”req”>:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1301">
            <msg>Command completed successfully; ack to dequeue</msg>
        </result>
        <msgQ count="1" id="1">
            <qDate>2022-11-14T08:33:12.119774Z</qDate>
            <msg>Transfer requested.</msg>
        </msgQ>
        <resData>
            <domain:trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
                <domain:name>test-transfer1.it.com</domain:name>
                <domain:trStatus>pending</domain:trStatus>
                <domain:reID>epp-test@it.com</domain:reID>
                <domain:reDate>2022-11-14T08:33:12.114618Z</domain:reDate>
                <domain:exDate>2023-11-14T08:33:12.114619Z</domain:exDate>
            </domain:trnData>
        </resData>
        <trID>
            <clTRID>637250b05f32b</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

Example command <poll op=”ack”>:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
        <poll op="ack" msgID="1"></poll>
        <clTRID>637250b09ef39</clTRID>
    </command>
</epp>
```

Example response <poll op=”ack”>:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1000">
            <msg>Command completed successfully</msg>
        </result>
        <msgQ count="0" id="1"/>
        <trID>
            <clTRID>637250b09ef39</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

Example response <poll op=”req”> when there are no messages in the queue to registrar:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <response>
        <result code="1300">
            <msg>Command completed successfully; no messages </msg>
        </result>
        <trID>
```

```
            <clTRID>6374820b05f32b</clTRID>
            <svTRID/>
        </trID>
    </response>
</epp>
```

17.Other Domain Name Statuses

EPP Registrars may add or remove the following EPP status codes to domains:

- clientHold – NS records will not be entered into the zone for the domain name while this status code is applied. When the status is set, the previously installed NS records are deleted. Upon subsequent removal of the status, they are restored.
- clientTransferProhibited – transfer requests for this domain name will be automatically rejected while this status code is applied.
- clientRenewProhibited – when this status code is applied, all requests by the sponsoring client to renew the domain are rejected.
- clientUpdateProhibited – when this status code is applied, all requests by the sponsoring client to update the domain name are rejected (except if the request includes removing this status from the domain name).
- clientDeleteProhibited – when this status code is applied, all requests by the sponsoring client to delete the domain name are rejected.