



EPP Manual
it.com Domains
v.1.8 (23.08.2024)

Contents

1.	Introduction.....	3
2.	The <login> Command – Starting a Session	4
3.	The <logout> Command – Ending a Session.....	6
4.	Hello Format.....	6
5.	The <domain:check> Command – checking if a domain exists in the registry	7
6.	The <domain:info> Command – Retrieving Information about a Domain	10
7.	The <contact:check> Command – Checking if a Contact Exists in the Registry	12
8.	The <host:check> Command – Checking if a Host Exists in the Registry ...	13
9.	The <contact:info> Command – Retrieving Information about a Contact ..	15
10.	The <contact:create> Command – Creating a Contact in the Registry.....	16
11.	The <domain:create> Command – Registering a Domain.....	18
12.	The <contact:update> Command – Modifying a Contact	22
13.	The <domain:update> Command – Updating a Domain Name	23
14.	The <domain:update> Command with the <rgp:update> Extension – Restoring a Domain	27
15.	The <domain:renew> Command – Renewing a Domain	29
16.	Domain Name Transfer Process	31
17.	The <domain:transfer> Command – Managing a Domain Transfer.....	32
18.	The <contact:delete> Contact – Deleting a Contact	35
19.	The <domain:delete> Command – Deleting a Domain	36
20.	Poll System – Retrieving Messages from the Registry.....	37
21.	Other Domain Name Statuses	40

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. The <login> Command – Starting a Session

The <login> command 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 on the user's whitelist, 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 <svcs> element must contain one or more <objURI> elements that contain namespace URIs representing the objects to be managed during the session. The <svcs> element may contain an optional <svcExtension> element that contains one or more <extURI> elements that identify object extensions to be used during the session.

Command example:

```
<?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 – Wrong password
- 2200 – Not valid certificate
- 2200 – IP address is not whitelisted for a user

Response example:

```
<?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. The <logout> Command – Ending a Session

The <logout> command ends the session with the ID specified in the <clTRID> element.

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

Command 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

Response 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. Hello Format

An EPP client may request a <greeting> from an EPP server at any time between a successful <login> command and a <logout> command by sending a <hello> to a server. An EPP <hello> must be an empty element with no child elements.

Command example:

```
<?xml version="1.0" ?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <hello/>
</epp>
```

In response to the command a greeting element will be returned in accordance with 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>
```

5. The <domain:check> Command – checking if a domain exists in the registry

The <domain:check> command 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 maximum number of <domain:name> elements per one <domain:check> command is limited to 100.

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 – initial domain registration
- renew – domain renewal
- transfer – domain transfer
- restore – domain restore

The <fee:command> element, except for restores, 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.

Command 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 case of 1000 result code, the domain name and the availability 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.

Response example if a 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>
```

Response example if a 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>
```

6. The <domain:info> Command – Retrieving Information about a Domain

The <domain:info> command returns information about the domain 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.

Command example:

```
<?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 case of 1000 result code, 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

Response example if the command is executed by the registrar that registered a 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>XXXXXXXXXX</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>
```

Response example 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>
```

7. The <contact:check> Command – Checking if a Contact Exists in the Registry

The <contact:check> command 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.

Command example:

```
<?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 availability 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.

Response example if a 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>
```

Response example 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>
```

8. The <host:check> Command – Checking if a Host Exists in 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.

Command example:

```
<?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 available 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.

Response example if a 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>
```

Response example if a 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>
```

9. The <contact:info> Command – Retrieving Information about a Contact

The <contact:info> command checks contact's presence in the registry system with the name specified in the <host:name> element.

Command example:

```
<?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>
```

Response example:

```
<?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>
      <cITRID>IT.COM-contact-info-1656512737831</cITRID>
      <svTRID/>
    </trID>
  </response>
</epp>
```

10. The <contact:create> Command – Creating a Contact in the Registry

The <contact:create> command 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 <cITRID> element must contain the session ID.

Command example:

```
<?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:postallInfo>
          <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:postallInfo>
        <contact:voice>+44.123453445665</contact:voice>
        <contact:email>user@mail.org</contact:email>
        <contact:authInfo>
          <contact:pw>XXXXXXXXXXXX</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <cITRID>IT.COM-contact-create-1656505373454</cITRID>
  </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

Response example:

```
<?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>
```

11. The <domain:create> Command – Registering a Domain

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

- <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:registrar> – registrar contact ID. Must be indicated
- <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> – an authorization code for the domain

The command may 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> – algorithm 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 command may also contain an <extension> element with a nested <fee:create> element that contains the following elements:

- <fee:currency> – not necessary element; only "USD" is permitted as a value
- <fee:fee> – domain registration fee. If the element presents in the command it will be successfully performed only in case if the fee entered in the <fee:fee> element matches the corresponding one configured in the registry

For premium domain registrations the <fee:create> element must be indicated.

The <clTRID> element must contain the session ID.

Command example:

```
<?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>XXXXXXXXXX</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>
      <fee:create xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">
        <fee:currency>USD</fee:currency>
        <fee:fee>500.00</fee:fee>
      </fee: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 not supported by the registry or the contact indicated for it does not exist)
- 2302 – Object exists

- 2004 – Parameter value range error, returned in the following cases:
 - Registration term indicated for the domain is outside of the permitted range from one to ten years
 - The domain is on the registry's block list and not available for registration
 - The number of name servers indicated for the domain is outside of the permitted range (from two to six)
 - IP addresses are missing for a glue record
 - <extension> element with a <fee:create> element are missing for premium domain registrations
- 2003 – Required parameter missing: <domain:registrant> element is required
- 2104 – Billing failure (registrar's account balance and credit limit are not sufficient for a registration to be completed)
- 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

If a session started with the <login> command included the <extURI> element for object fee extension the response to the <domain:create> command will contain the <extension> element with the <fee:creData> element that includes the following elements:

- <fee:currency> – transaction currency
- <fee:fee> – domain registration fee
- <fee:balance> – remaining registrar's balance after the withdrawal of the domain registration fee

- <fee:creditLimit> – registrar’s credit limit

Response example:

```
<?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>
    <extension>
      <fee:creData xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">
        <fee:currency>USD</fee:currency>
        <fee:fee>500.0</fee:fee>
        <fee:balance>-500.0</fee:balance>
        <fee:creditLimit>1000.0</fee:creditLimit>
      </fee:creData>
    </extension>
    <trID>
      <clTRID>IT.COM-domain-create-1656505373493</clTRID>
      <svTRID/>
    </trID>
  </response>
</epp>
```

12. The <contact:update> Command – Modifying a Contact

The <contact:update> command 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.

Command example:

```
<?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:postallInfo 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:postallInfo>
        </contact:chg>
      </contact:update>
    </update>
  </command>
</epp>
```

```

    </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

Response example:

```

<?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>

```

13. The <domain:update> Command – Updating a Domain Name

The <domain:update> command 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
- OPTIONAL <domain:chg> element that contains object attribute values to be changed

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 provided in the Other Domain Name Statuses section of this document. If a domain has the clientUpdateProhibited status the <domain:update> command can only remove it, and any other requested changes won't be applied

A <domain:chg> element contains the following child elements:

- <domain:registrant> element that contains the identifier of contact object to be associated with the domain object as its registrant. This object identifier must be known to the server before the contact object can be associated with the domain object. The element cannot have an empty value
- <domain:authInfo> element that contains authorization information associated with the domain object

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 DNSSEC elements of the domain, both those added by this command and those that has 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.

Command example:

```
<?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:update>
    </extension>
  </command>
</epp>
```

```

    </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
- 2004 – Parameter value range error returned in the following cases:
 - The number of name servers indicated for the domain is outside of the permitted range (from two to six)
 - IP addresses are missing for a glue record

Response example:

```

<?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>

```

```
</response>  
</epp>
```

14. The <domain:update> Command with the <rgp:update> Extension – Restoring a Domain

Once a domain is deleted from the registry the redemptionPeriod starts during which domain restore is available. In order to restore the domain the <domain:update> command is used with the <rgp:update> extension.

The following elements must be entered for the command:

- <domain:name> – the domain to be restored
- At least one of the following elements must be empty: <domain:add>, <domain:rem>, <domain:chg>

The command must contain the <extension> element, containing the <rgp:update> element, containing the <rgp:restore op="request"/> element.

Also the <extension> element may contain the <fee:update> element, containing the following elements:

- <fee:currency> – not necessary element; only "USD" is permitted as a value
- <fee:fee> – domain restore fee. If the element presents in the command it will be successfully performed only in case if the fee entered in the <fee:fee> element matches the corresponding one configured in the registry

The <clTRID> element must contain the session ID.

Command example:

```
<?xml version="1.0" encoding="UTF-8"?>  
<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>test-domain-restore.it.com</domain:name>  
        <domain:add/>  
        <domain:rem/>  
        <domain:chg/>  
      </domain:update>  
    </update>  
    <extension>  
      <rgp:update xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0">  
        <rgp:restore op="request"/>  
      </rgp:update>  
      <fee:update xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">  
        <fee:currency>USD</fee:currency>  
        <fee:fee>10.00</fee:fee>  
      </fee:update>
```

```
</extension>
<cITRID>TEST-1724334151</cITRID>
</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 restore a domain not supported by the registry)
- 2004 – Parameter value range error: operation price does not match the requested price
- 2104 – Billing failure
- 2304 – Object status prohibits operation

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

- The <extension> element containing the <rgp:upData> element containing the <rgp:rgpStatus s="pendingRestore"/> element

If a session started with the <login> command included the <extURI> element for object fee extension the response to the <domain:update> command with the <rgp:update> extension will contain the <extension> element with the <fee:upData> element that includes the following elements:

- <fee:currency> – transaction currency
- <fee:fee> – domain restore fee
- <fee:balance> – remaining registrar's balance after the withdrawal of the domain restore fee
- <fee:creditLimit> – registrar's credit limitResponse example:

Response example:

```
<?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>
    <extension>
      <rgp:upData xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0">
        <rgp:rgpStatus s="pendingRestore"/>
      </rgp:upData>
    </extension>
  </response>
</epp>
```

```

<fee:updData xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">
  <fee:currency>USD</fee:currency>
  <fee:fee>10.0</fee:fee>
  <fee:balance>7156.0</fee:balance>
  <fee:creditLimit>245.0</fee:creditLimit>
</fee:updData>
</extension>
<trID>
  <cITRID>TEST-1724334151</cITRID>
  <svTRID>UPDATE_DOMAIN-160-38b99e3e-9fbf-4583-9d96-d1fa5e74df7a</svTRID>
</trID>
</response>
</epp>

```

15. The <domain:renew> Command – Renewing a Domain

The <domain:renew> command extends the domain registration period for a 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 command may also contain an <extension> element with a nested <fee:renew> element that contains the following elements:

- <fee:currency> – not necessary element; only "USD" is permitted as a value
- <fee:fee> – domain renewal fee. If the element presents in the command it will be successfully performed only in case if the fee entered in the <fee:fee> element matches the corresponding one configured in the registry

The <cITRID> element must contain the session ID.

Command example:

```

<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <renew>
      <domain:renew>
        <domain:name>test-domain-renew.it.com</domain:name>
        <domain:curExpDate>2025-07-02</domain:curExpDate>
        <domain:period unit="y">1</domain:period>
      </domain:renew>
    </renew>
  </command>
</epp>

```

```
</renew>
<extension>
  <fee:renew xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">
    <fee:currency>USD</fee:currency>
    <fee:fee>5.00</fee:fee>
  </fee:renew>
</extension>
<cITRID>TEST-1724335780</cITRID>
</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
- 2304 – Object status prohibits operation
- 2005 – Parameter value syntax error
- 2004 – Parameter value range error
- 2104 – Billing failure
- 2004 – Parameter value range error: operation price does not match the requested price

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

- <domain:name> – the renewed domain
- <domain:exDate> – new expiration date for the domain

If a session started with the <login> command included the <extURI> element for object fee extension the response to the <domain:renew> command will contain the <extension> element with the <fee:renData> element that includes the following elements:

- <fee:currency> – transaction currency
- <fee:fee> – domain renewal fee
- <fee:balance> – remaining registrar's balance after the withdrawal of the domain renewal fee

- <fee:creditLimit> – registrar's credit limit

Response example:

```
<?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>test-domain-restore.it.com</domain:name>
        <domain:exDate>2026-07-02T00:00:00Z</domain:exDate>
      </domain:renData>
    </resData>
    <extension>
      <fee:renData xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">
        <fee:currency>USD</fee:currency>
        <fee:fee>5.0</fee:fee>
        <fee:balance>7151.0</fee:balance>
        <fee:creditLimit>245.0</fee:creditLimit>
      </fee:renData>
    </extension>
    <trID>
      <clTRID>TEST-1724335780</clTRID>
      <svTRID>RENEW_DOMAIN-160-3b4f71b0-1d8f-402f-b242-205c3f41bb13</svTRID>
    </trID>
  </response>
</epp>
```

16. Domain Name Transfer Process

The following process for transferring domains between registrars applies:

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.

17. The <domain:transfer> Command – Managing a Domain Transfer

The <domain:transfer> 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
- query – getting information about the transfer

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.

The command may also contain an <extension> element with a nested <fee:transfer> element that contains the following elements:

- <fee:currency> – not necessary element; only “USD” is permitted as a value
- <fee:fee> – domain transfer fee. If the element presents in the command it will be successfully performed only in case if the fee entered

in the <fee:fee> element matches the corresponding one configured in the registry

Command example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <transfer op="request">
      <domain:transfer xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>test-transfer.it.com</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:authInfo>
          <domain:pw>tkZdUakqn%H8/--M</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <extension>
      <fee:transfer xmlns:fee="urn:ietf:params:xml:ns:epp:fee-1.0">
        <fee:currency>USD</fee:currency>
        <fee:fee>5.00</fee:fee>
      </fee:transfer>
    </extension>
    <cITRID>TEST-1724337174</cITRID>
  </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 registration term outside the range from 1 to 10 years)

In 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:relD> 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

If a session started with the <login> command included the <extURI> element for object fee extension the response to the <domain:transfer> command will contain the <extension> element with the <fee:trnData> element that includes the following elements:

- <fee:currency> – transaction currency
- <fee:fee> – domain transfer fee
- <fee:balance> – remaining registrar's balance after the withdrawal of the domain transfer fee
- <fee:creditLimit> – registrar's credit limit

Response example:

```
<?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-transfer.it.com</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:relD>epp@it.com</domain:relD>
        <domain:reDate>2024-08-22T14:32:54.690796Z</domain:reDate>
        <domain:acID>test-registrar1@it.com</domain:acID>
        <domain:acDate>2024-08-27T14:32:54.690796Z</domain:acDate>
        <domain:exDate>2026-08-23T00:00:00Z</domain:exDate>
      </trnData>
    </resData>
    <extension>
      <fee:trnData xmlns:fee="urn:ietf:params:xml:ns:epp-fee-1.0">
        <fee:currency>USD</fee:currency>
        <fee:fee>5.0</fee:fee>
        <fee:balance>7146.0</fee:balance>
        <fee:creditLimit>245.0</fee:creditLimit>
      </fee:trnData>
    </extension>
    <trID>
      <clTRID>TEST-1724337174</clTRID>
      <svTRID>TRANSFER_REQUEST_DOMAIN-160-4cbc4db7-14d1-4973-ac74-
03bfa35d23e8</svTRID>
    </trID>
  </response>
</epp>
```

18. The <contact:delete> Contact – Deleting a Contact

The <contact:delete> command deletes a contact ID indicated in the <contact:id> element.

The <clTRID> element must contain the session ID.

Command example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <delete>
      <contact:delete xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>CONTACT-136872</contact:id>
      </contact:delete>
    </delete>
    <clTRID>TEST-1724338063</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
- 2305 – Object association prohibits operation. Error appears when a contact to be deleted is associated with a domain

Response example:

```
<?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>
      <cITRID>TEST-1724338063</cITRID>
      <svTRID>DELETE_CONTACT-160-86383a45-01d2-4363-bab0-09bb6514912f</svTRID>
    </trID>
  </response>
</epp>
```

19. The <domain:delete> Command – Deleting a Domain

The <domain:delete> command deletes a domain indicated in the <domain:name> element.

The <cITRID> element must contain the session ID.

Command example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <delete>
      <domain:delete xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>test-domain2.it.com</domain:name>
      </domain:delete>
    </delete>
    <cITRID>TEST-1724328898</cITRID>
  </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
- 2305 – Object association prohibits operation

If a session started with the <login> command included the <extURI> element for object fee extension and as a result of domain deletion corresponding refund has been processed to the registrar, the response to the <domain:delete> command will contain the <extension> element with the <fee:delData> element that includes the following elements:

- <fee:currency> – transaction currency
- <fee:fee> – amount refunded for domain deletion
- <fee:balance> – registrar's balance after refund for domain deletion
- <fee:creditLimit> – registrar's credit limit

Response example:

```
<?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>
    <extension>
      <fee:delData xmlns:fee="urn:ietf:params:xml:ns:epp-fee-1.0">
        <fee:currency>USD</fee:currency>
        <fee:credit>-5.0</fee:credit>
        <fee:balance>7216.0</fee:balance>
        <fee:creditLimit>245.0</fee:creditLimit>
      </fee:delData>
    </extension>
    <trID>
      <cITRID>TEST-1724328898</cITRID>
      <svTRID>DELETE_DOMAIN-160-5f12c12d-35ff-4f8f-9581-a4fa73fb1a47</svTRID>
    </trID>
  </response>
</epp>
```

20. Poll System – Retrieving Messages from the Registry

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

<poll op="req"> command example:

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

Additionally, the following elements are returned in the response:

- **OPTIONAL** – the <msgQ> 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** – the <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

<poll op="req"> example response:

```
<?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>
```

<poll op="ack"> example command:

```
<?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>
```

<poll op="ack"> 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>
    <msgQ count="0" id="1"/>
    <trID>
      <clTRID>637250b09ef39</clTRID>
      <svTRID/>
    </trID>
  </response>
</epp>
```

<poll op="req"> example response when there are no messages in the queue for a 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>
      <cITRID>6374820b05f32b</cITRID>
      <svTRID/>
    </trID>
  </response>
</epp>
```

21. 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 from DNS. 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