NIC.GE
EPP Service Guide

v 0.8

Table of Contents

What is EPP

EPP (Extensible Provisioning Protocol) defines an unified way how registrars can communicate with registries of domain names by exchanging XML messages.

Detailed information and documentation for EPP can be found in the RFC document on Internet Engineering Task Force website:

- RFC-5730
  Extensible Provisioning Protocol (EPP)
  https://tools.ietf.org/html/rfc5730

- RFC-5731
  Extensible Provisioning Protocol (EPP) Domain Name Mapping
  https://tools.ietf.org/html/rfc5731

- RFC-5733
  Extensible Provisioning Protocol (EPP) Contact Mapping
  https://tools.ietf.org/html/rfc5733

- RFC-5734
  Extensible Provisioning Protocol (EPP) Transport over TCP
  https://tools.ietf.org/html/rfc5734

How EPP Works

1. A typical EPP conversation starts when client connects to the server using an SSL Certificate provided by the NIC.
2. Client logs in using it's account username and password provided by NIC.
3. Client reads details and sends creates/update/delete commands to the server.
4. Client waits for new things to be done, periodically polling for notification messages. Session can be kept alive by periodically calling any epp command, but the most appropriate commands are wither hello or poll.
5. Connection is ended either by client when nothing else needs to be done or by server, when the connection times out due to no command being send for specific period of time.

Note:

- Nic.ge provides only synchronous EPP communication mode.
- Nic.ge uses only generic parameters defined by the RFCs. No extensions are used.
- Host object mapping (RFC-5732) is not implemented as network records are attributes of each domain name in our data model (as defined in section 1.1 of RFC-5731).

Connection to NIC.GE EPP

Access to testing and production environment will be provided by the NIC. After registration as a registrar and a corresponding review process the following will be provided:

1. Client certificate in P12/PFX format and it's password
2. Username and password
3. Registrar prefix
   (unique prefix with which that registrar must create object IDs)
4. VPN connection details

Note:

- Certificates might optionally be provided in two separate files, Public Key and a password encrypted Private Key
- Whitelisted public IP access might optionally be provided instead of a VPN tunnel

Important!

All object identifiers must be prefixed with the "Registrar Prefix" supplied by the registrar (ex. "R9"). If, for example, an object identifier in registrar's side is "e9c49b4bab73c", the value sent to EPP server should be "R9-e9c49b4bab73c".

Max length for the ID is 16 characters.

Test server is available at **epp-test.nic.ge** on standard EPP port 700 (TCP), located in an isolated network accessible only through VPN connection or a whitelisted public IP . TLS Client certificate authentication is used for connections.

Production server is available at **epp.nic.ge** on standard EPP port 700 (TCP), located in an isolated network accessible only through VPN connection or a whitelisted public IP . TLS Client certificate authentication is used for connections.

Session Management

Session is started when a new TCP connection to server is opened over TLS v1.2.

The login command begins an authorized session and gives access to chosen EPP services the server offers. Session is closed in response to logout request issued by the client.

Server will have limitations on number of concurrent connections opened from the same client. Session would timeout after being idle for more than 1 minute.

Hello command can be issued at regular intervals to keep the session alive.

As soon as the client connects to the server, after successful TLS negotiation, the server will send a greeting to the client. It is the same message that is replied after Hello command.

Hello command

When a new TCP connection to server is initiated, implicit hello command is processed by the server and a greeting message is returned.

Example of hello command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <hello />
</epp>
```

Example of hello command response

```
<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">
  <greeting>
    <svID>GEDAS</svID>
    <svDate>2018-12-03T15:52:21.2498329+04:00</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:contact-1.0</objURI>
    </svcMenu>
    <dcp>
      <access>
        <all />
      </access>
      <statement>
        <purpose>
```

```
          <prov />
        </purpose>
        <recipient>
          <ours />
        </recipient>
        <retention>
          <stated />
        </retention>
      </statement>
    </dcp>
  </greeting>
</epp>
```

Login command

Login command authenticates client for the rest of the TCP session.
The following parameters should be provide:

- **clID**
  Identificator (client ID) assigned to the registrar for EPP account. Client ID and password are chosen by the client before activating EPP account.

- **pw**
  Password for the EPP account.

Session will be closed if more than 5 failed login attempts happen in a row.

Example of login command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <login>
      <clID>ClientX</clID>
      <pw>epp123</pw>
      <options>
        <version>1.0</version>
        <lang>en</lang>
      </options>
      <svcs>
        <objURI>urn:ietf:params:xml:ns:obj1</objURI>
        <objURI>urn:ietf:params:xml:ns:obj2</objURI>
        <objURI>urn:ietf:params:xml:ns:obj3</objURI>
        <svcExtension>
          <extURI>http://custom/obj1ext-1.0</extURI>
        </svcExtension>
      </svcs>
    </login>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of login command response

```
<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">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-750e4efc</svTRID>
    </trID>
  </response>
</epp>
```

Logout command

Logout command terminates session and closes TCP connection (please, do not terminate connection without logging out).

Example of logout command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <logout />
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of logout command response

```
<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">
  <response>
    <result code="1500">
      <msg>successful &lt;logout&gt; command</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-664dfe05</svTRID>
    </trID>
  </response>
</epp>
```

Service Messages

Server manages a message queue for registrars to deliver information about certain events. One example is a domain name registrar change request (transfer) is initiated, approved, rejected or canceled by the ther party.

Polling for new messages

a EPP <poll> command with op="req" attribute present requests for the next unread service message waiting for registrar in the queue.

Example of poll request command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <poll op="req" />
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of empty poll command response

```
<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">
  <response>
    <result code="1300">
      <msg>&lt;poll&gt; request command and the server message queue is
empty.</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-1ecb64cd</svTRID>
    </trID>
  </response>
</epp>
```

Example of poll command response with a transfer approval message

```
<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">
  <response>
    <result code="1301">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="6" id="d936fe4e-0587-403c-a97c-a2264d057ad7">
```

```
      <qDate>2019-01-22T13:28:37.020002</qDate>
      <msg>Transfer requested.</msg>
    </msgQ>
    <resData>
      <trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <name>example.ge</name>
        <trStatus>Server Approved</trStatus>
        <reID>loosing-reg</reID>
        <reDate>2019-01-22T13:28:36.774567</reDate>
        <acID>gaining-reg</acID>
        <acDate>2019-01-22T13:28:36.774567</acDate>
      </trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-4e32f692</svTRID>
    </trID>
  </response>
</epp>
```

Acknowledge poll messages

Example of poll ack request command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <poll op="ack" msgID="d936fe4e-0587-403c-a97c-a2264d057ad7" />
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of poll ack command response when no messages left to read

```
<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">
  <response>
    <result code="1300">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-bb3d7e25</svTRID>
    </trID>
  </response>
</epp>
```

Example of poll ack command response with more messages left

```
<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">
  <response>
    <result code="1301">
      <msg>Command completed successfully</msg>
    </result>
    <msgQ count="4" id="bf792a0a-f052-4329-9c5e-14f4e0a4523a">
      <qDate>2019-01-23T13:30:51.628166</qDate>
      <msg>Transfer requested.</msg>
    </msgQ>
    <resData>
      <trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <name>letstestit.ge</name>
        <trStatus>Server Approved</trStatus>
        <reID>loosing-reg</reID>
        <reDate>2019-01-23T11:45:35.008226</reDate>
        <acID>gaining-reg</acID>
        <acDate>2019-01-23T11:45:35.008226</acDate>
      </trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-92b85304</svTRID>
    </trID>
  </response>
</epp>
```

Domain Names

Implementation of domain names mapping closely follows RFC-5731.

Note: IDN domain names are not allowed

Domain Check

Command allows to check availability of a domain name before calling a domain:create request to register it. Command follows standard EPP Domain name mapping RFC-5731.

Example of domain check command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <check>
      <domain:check xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
        <domain:name>example.com</domain:name>
        <domain:name>example.ge</domain:name>
```

```
        <domain:name>eyyyyample.ge</domain:name>
      </domain:check>
    </check>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of domain check command response

```
<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">
  <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">example.com</domain:name>
        </domain:cd>
        <domain:cd>
          <domain:name avail="0">example.ge</domain:name>
          <domain:reason>Reserved</domain:reason>
        </domain:cd>
        <domain:cd>
          <domain:name avail="0">eyyyyample.ge</domain:name>
          <domain:reason>Reserved</domain:reason>
        </domain:cd>
      </domain:chkData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-69f045b0</svTRID>
    </trID>
  </response>
</epp>
```

Domain Info

Requests detailed information about a registered domain name. All domain info is returned with requesting a domain owned by registrator.

Only limited number of info is returned when requesting information for domain not owned by the registrator.

If a domain name request has a valid authInfo password provided for a domain, whole domain information will be provided even for a non-domain owner registrator. This method should be used to verify the transfer code for domain.

Example of domain info command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <domain:info xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name hosts="all">ey51ample.ge</domain:name>
        <domain:authInfo>
          <domain:pw>2fooBAR</domain:pw>
        </domain:authInfo>
      </domain:info>
    </info>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of domain info command response

```
<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">
  <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>ey51ample.ge</domain:name>
        <domain:roid>796150a6-4e44-48af-b515-e6831f9397ca</domain:roid>
        <domain:registrant>jd1234991</domain:registrant>
        <domain:contact type="Tech">sh8013</domain:contact>
        <domain:contact type="Billing">sh8013</domain:contact>
        <domain:hostAttr>
          <domain:hostName>ns2.yoursrs.com</domain:hostName>
        </domain:hostAttr>
        <domain:hostAttr>
          <domain:hostName>ns1.yoursrs.com</domain:hostName>
          <hostAddr ip="v4" xmlns="">217.170.2.195</hostAddr>
        </domain:hostAttr>
        <domain:clID></domain:clID>
        <domain:crID>c787c2ee-998d-4735-83c0-96c6278db47d</domain:crID>
        <domain:crDate>2018-12-03T15:32:34.980145</domain:crDate>
        <domain:exDate>2020-12-03T15:32:34.980145</domain:exDate>
        <domain:authInfo>
          <domain:pw>2fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-1abbae67</svTRID>
    </trID>
  </response>
</epp>
```

Domain Create

At least the following input fields are required to create a domain:

- Domain name
- Authinfo
- Registrang
- Minimum 2 NS records

In case the NS record hosts are subdomains of the domain name itself (for example when domain is “example.com” and it’s name server is “ns1.example.com), it is mandatory to specify the host IPv4 address as well. Otherwise the resolve will not take place.

Example of domain create command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>ey51ample.ge</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostAttr>
            <domain:hostName>ns1.yoursrs.com</domain:hostName>
            <domain:hostAddr>217.170.2.195</domain:hostAddr>
          </domain:hostAttr>
          <domain:hostAttr>
            <domain:hostName>ns2.yoursrs.com</domain:hostName>
          </domain:hostAttr>
        </domain:ns>
        <domain:registrant>jd1234991</domain:registrant>
        <domain:contact type="admin">sh8013</domain:contact>
        <domain:contact type="tech">sh8013</domain:contact>
        <domain:authInfo>
          <domain:pw>2fooBAR</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of domain create response

```
<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">
  <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>ey51ample.ge</domain:name>
        <domain:crDate>2018-12-03</domain:crDate>
        <domain:exDate>2020-12-03</domain:exDate>
      </domain:creData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-9515c99f</svTRID>
    </trID>
  </response>
</epp>
```

Domain Renew

Example of domain renew command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <renew>
      <domain:renew xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
        <domain:name>ey51ample.ge</domain:name>
        <domain:curExpDate>2020-12-03</domain:curExpDate>
        <domain:period unit="y">1</domain:period>
      </domain:renew>
    </renew>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of domain renew response

```
<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">
  <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>ey51ample.ge</domain:name>
        <domain:exDate>2019-12-03</domain:exDate>
      </domain:renData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-7aade9d0</svTRID>
    </trID>
  </response>
</epp>
```

Domain Update

The command is used to update domain information, including the domain password, that is used for domain transfer.

Before changing the domain transfer password, registrator should perform a 2FA verification with the client to confirm that the change is really requested by the client. 2FA should be performed outside the registrator's portal. A good example is a link sent to an the email of domain owner requiring to click in order to complete the process.

Once the client clicks the link (verifies), the registrator can generate a new transfer code and set it using Domain Update command.

Example of domain update command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <update>
      <domain:update xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
        <domain:name>domain.ge</domain:name>
          <domain:add>
            <domain:ns>
              <domain:hostAttr>
                <domain:hostName>ns3.yoursrs.com</domain:hostName>
                <domain:hostAddr>217.170.2.191</domain:hostAddr>
               </domain:hostAttr>
              </domain:ns>
          </domain:add>
          <domain:rem>
            <domain:ns>
              <domain:hostAttr>
                <domain:hostName>ns1.yoursrs.com</domain:hostName>
              </domain:hostAttr>
            </domain:ns>
          </domain:rem>
      </domain:update>
    </update>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of domain update response

```
<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">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
```

```
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-0fe21a5e</svTRID>
    </trID>
  </response>
</epp>
```

Domain Delete

The command is used to delete domain name. After issuing this command domain name will become available for others to create.

Example of domain delete command

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
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>domain.ge</domain:name>
     </domain:delete>
   </delete>
   <clTRID>ABC-12345</clTRID>
 </command>
</epp>
```

Example of domain update response

```
<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">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-0fe21a5e</svTRID>
    </trID>
  </response>
</epp>
```

Domain Transfer Request

Before transfering a domain name, the source (current) registrator of the domain should perform a 2FA verification with the client to confirm that the transfer is really requested by the client. 2FA should be performed outside the registrator's portal. A good example is a link sent to an the email of domain owner requiring to click in order to complete the process.

Once (if) the client's transfer request is verified with the second factor, the source registrator can create a random password (transfer code), update the domain password with Domain Update command and give the transfer code to the client. Transfer code should be delivered to the email of the domain owner (registrar).

Transfer code should comply with the following parameters:

- Can contain: numbers, uppercase letters and lowercase letters
- Should be at least 8 characters long
- Should be no more than 16 characters long

Once the client obtains the transfer code, transfer process can be continued at the target (new) registrator by submitting the domain name and transfer code (password) to a new registrator.

Registrator can use the provided domain name and password to request domain information and verify that the password is correct. Full domain details, including the password will be returned.

If the password is confirmed, registrator should send a Transfer Request command passing a transfer code as a domain password. If the code is correct, the domain name will be automatically transferred to new registrator. No additional approvals will be required from any side.

Once the transfer is completed, domain password will be reset so that the existing code cannot be used.

The old registrator can use one of two ways to check if the domain transfer has been completed:

1. Call Domain Info and check it's current registrator
2. Server will deliver a successful transfer notification next time when the registrator preforms Polling for new messages.

No other transfer operations are implemented.

Note: Transfer code is valid for N days (N might be changed by the administrator). If the code is not used for N days, it will be automatically reset. Once the code is reset, new Domain Update command will be required along with the 2FA confirmation all over again.

Example of request query command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
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>example.ge</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:authInfo>
          <domain:pw>2fooBAR</domain:pw>
        </domain:authInfo>
```

```
      </domain:transfer>
    </transfer>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of transfer query response

```
<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">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.ge</domain:name>
        <domain:trStatus>Pending</domain:trStatus>
        <domain:reID>11</domain:reID>
        <domain:reDate>2018-12-03T16:53:53.352882</domain:reDate>
        <domain:acID>1</domain:acID>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-57f42a63</svTRID>
    </trID>
  </response>
</epp>
```

Contacts

Contact Check

Example of contact check command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <check>
      <contact:check xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0 contact-1.0.xsd">
        <contact:id>jd1234991</contact:id>
        <contact:id>sah8013</contact:id>
        <contact:id>8013sah</contact:id>
      </contact:check>
    </check>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of contact check response

```
<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">
  <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">jd1234991</contact:id>
          <contact:reason>In Use</contact:reason>
        </contact:cd>
        <contact:cd>
          <contact:id avail="1">sah8013</contact:id>
        </contact:cd>
        <contact:cd>
          <contact:id avail="1">8013sah</contact:id>
        </contact:cd>
      </contact:chkData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
```

```
      <svTRID>Gedas-667bf4fd</svTRID>
    </trID>
  </response>
</epp>
```

Contact Info

The command is used to retrieve the contact information. Contact information can only be retrieved by the registrar who owns the domain.

There is an exception to the rule in the cases, when contact information needs to be retrieved when transferring a domain name. In this case a special roid attribute should be provided with the auth info in the following manner:

- Roid attribute should contain the domain name (ex. example.com)
- Tag value should contain the password for the domain name

In the example below (command with domain password), contact information will be returned if the password ("PASSFOREXAMPLECOM") corresponts to "example.com" domain name and also the contact is connected to the domain name in any of the following ways:

- Owner
- Tech contact
- Billing contact
- Admin contact

Example of contact info command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <contact:info xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0 contact-1.0.xsd">
        <contact:id>sh8013</contact:id>
      </contact:info>
    </info>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of contact info command with domain password

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <contact:info xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0 contact-1.0.xsd">
        <contact:id>sh8013</contact:id>
        <contact:authInfo>
          <contact:pw roid="example.com">PASSFOREXAMPLECOM</contact:pw>
        </contact:authInfo>
      </contact:info>
```

```
    </info>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of contact info response

```
<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">
  <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>sh8013</contact:id>
        <contact:roid>10706842-1ce1-4e34-8a7c-d73a6656b9e1</contact:roid>
        <contact:postalInfo type="int">
          <contact:name>John Doe</contact:name>
          <contact:org>Example Inc.</contact:org>
          <contact:addr>
            <contact:street>123 Example Dr.</contact:street>
            <contact:street>Suite 100</contact:street>
            <contact:street />
            <contact:city>Dulles</contact:city>
            <contact:sp>VA</contact:sp>
            <contact:pc>20166-6503</contact:pc>
            <contact:cc>US</contact:cc>
          </contact:addr>
          <contact:voice x="1234">+995.599000000</contact:voice>
          <contact:fax>+1.7035555556</contact:fax>
          <contact:email>jdoe@example.com</contact:email>
          <contact:clID></contact:clID>
          <contact:crID>11</contact:crID>
          <contact:crDate>2018-11-05T12:42:27.770434</contact:crDate>
          <contact:authInfo>
            <contact:pw>2fooBAR</contact:pw>
          </contact:authInfo>
        </contact:postalInfo>
      </contact:infData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-85d21946</svTRID>
    </trID>
  </response>
</epp>
```

Contact Create

Example of contact create command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <contact:create xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0 contact-1.0.xsd">
        <contact:id>jd1234991</contact:id>
        <contact:postalInfo type="int">
          <contact:name>John Doe</contact:name>
          <contact:org>Example Inc.</contact:org>
          <contact:addr>
            <contact:street>123 Example Dr.</contact:street>
            <contact:street>Suite 100</contact:street>
            <contact:city>Dulles</contact:city>
            <contact:sp>VA</contact:sp>
            <contact:pc>20166-6503</contact:pc>
            <contact:cc>US</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice x="1234">+995.599000000</contact:voice>
        <contact:fax>+1.7035555556</contact:fax>
        <contact:email>jdoe@example.com</contact:email>
        <contact:authInfo>
          <contact:pw>f2ooBAR</contact:pw>
        </contact:authInfo>
        <contact:disclose flag="0">
          <contact:voice />
          <contact:email />
        </contact:disclose>
      </contact:create>
    </create>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of contact create response

```
<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">
  <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>jd1234991</contact:id>
        <contact:crDate>2018-12-03T15:52:30.4069575+04:00</contact:crDate>
      </contact:creData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-185ed9a0</svTRID>
    </trID>
  </response>
</epp>
```

Contact Update

Example of contact update command

```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
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>jd1234991</contact:id>
        <contact:chg>
          <contact:postalInfo type="int">
            <contact:org />
            <contact:addr>
              <contact:street>124 Example Dr.</contact:street>
              <contact:street>Suite 200</contact:street>
              <contact:city>Dulles</contact:city>
              <contact:sp>VA</contact:sp>
              <contact:pc>20166-6503</contact:pc>
              <contact:cc>US</contact:cc>
            </contact:addr>
          </contact:postalInfo>
          <contact:voice>+995.599000000</contact:voice>
          <contact:fax />
          <contact:authInfo>
            <contact:pw>2fooBAR</contact:pw>
          </contact:authInfo>
          <contact:disclose flag="1">
            <contact:voice />
            <contact:email />
          </contact:disclose>
        </contact:chg>
      </contact:update>
    </update>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

Example of contact update response

```
<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">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-d6e62bfd</svTRID>
    </trID>
  </response>
</epp>
```