NIC.GE

EPP Service Guide

v 0.2

Table of Contents

What is EPP

EPP (Extensible Provision 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 polls the server for unread notifications.
4. Client reads details and sends creates/update/delete commands to the server.
5. Client waits for new things to be done, periodically polling for notification messages to keep session alive.
6. Connection is ended either by client when nothing else needs to be done or by server, when server if session timeouts.

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 "e9c49b4bab73c1b", the value sent to EPP server should be "R9-e9c49b4bab73c1b"

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 10 minutes or lasting over 1 hour.

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

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>2018-11-16T16:34:01.8883824+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>foo-BAR2</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-be50cb9c</svTRID>
    </trID>
  </response>
</epp>
```

Logout command

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

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>
    <logout />
    <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="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 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-8f7732c0</svTRID>
    </trID>
  </response>
</epp>
```

Dequeuing read messages

When a message has been read, client should acknowledge it and server would make the next message, if any, available for the next <poll> request.
Request contains a <poll op="ack"> element, whose id attribute typically contains the ID of the last read message.

Example of poll achknowledge command

```
Example incomplete. Update pending.
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
</epp>
```

Example of poll command response

```
Example incomplete. Update pending.
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
</epp>
```

Domain Names

Implementation of domain names mapping closely follows RFC-5731.
IDN domain names are not allowed.

Note: IDN domain names are not allowed

The following domain status values are supported:
Example incomplete. Update pending.

- **ok**
  The normal state of an active domain name

- **inactive**
  This state means that domain has no network information attached

- **pendingTransfer**
  Registrar change request for a domain name has been initiated but is not yet reviewed by the other involved side.
- ...

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.net</domain:name>
        <domain:name>example.org</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="1">example.net</domain:name>
        </domain:cd>
        <domain:cd>
          <domain:name avail="1">example.org</domain:name>
        </domain:cd>
      </domain:chkData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-69baeea0</svTRID>
    </trID>
  </response>
</epp>
```

Domain Info

Requests detailed information about a registered domain name. This command follows standard EPP Domain name mapping RFC-5731.

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">example.ge</domain:name>
      </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>example.ge</domain:name>
        <domain:roid>0645c0a0-7e6d-476f-9b27-ec571cc69f62</domain:roid>
        <domain:registrant>8969a0a0-95d7-4bb3-b0d2-e60f35f1d03e</domain:registrant>
        <domain:contact
type="Tech">4b593978-15b3-46ec-a6cd-3f51bfb0afde</domain:contact>
        <domain:contact
type="Admin">c0d172f0-80ef-4795-87f0-0056955e78b9</domain:contact>
        <domain:hostAttr>
          <domain:hostName>ns1.digitalocean.com</domain:hostName>
        </domain:hostAttr>
        <domain:hostAttr>
          <domain:hostName>ns2.digitalocean.com</domain:hostName>
        </domain:hostAttr>
        <domain:hostAttr>
          <domain:hostName>ns3.digitalocean.com</domain:hostName>
        </domain:hostAttr>
        <domain:clID></domain:clID>
        <domain:crID>a0ac0676-be40-47b0-886a-0da52a803faa</domain:crID>
        <domain:crDate>2015-06-19T12:02:39.323</domain:crDate>
        <domain:exDate>2021-06-20T00:00:00</domain:exDate>
        <domain:upID>11</domain:upID>
        <domain:upDate>2018-11-14T12:16:36.592284</domain:upDate>
        <domain:authInfo> <domain:pw>31288CC70F73FD2</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-e99eb0af</svTRID>
    </trID>
  </response>
</epp>
```

Domain Create

<Description>

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>eyxample.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>jd1234</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="2302">
      <msg>Object exists: server received
              a command to create an object that already exists in the
              repository.</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-33818b22</svTRID>
    </trID>
  </response>
</epp>
```

Domain Renew

<Description>

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>example.ge</domain:name>
        <domain:curExpDate>2020-06-20</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="2303">
      <msg>Object does not exist: server received
              a command to query or transform an object that does not
              exist in the repository</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-a73e8d4d</svTRID>
    </trID>
  </response>
</epp>
```

Domain Update

<Description>

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">
        <domain:name>example.ge</domain:name>
        <domain:add>
          <domain:ns>
            <domain:hostObj>ns2.example.com</domain:hostObj>
          </domain:ns>
          <domain:contact type="tech">mak21</domain:contact>
          <domain:status s="clientHold" lang="en">Payment overdue.</domain:status>
        </domain:add>
        <domain:rem>
          <domain:ns>
            <domain:hostObj>ns1.example.com</domain:hostObj>
          </domain:ns>
          <domain:contact type="tech">sh8013</domain:contact>
          <domain:status s="clientUpdateProhibited" />
        </domain:rem>
        <domain:chg>
          <domain:registrant>sh8013</domain:registrant>
          <domain:authInfo>
            <domain:pw>2BARfoo</domain:pw>
          </domain:authInfo>
        </domain:chg>
      </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="2303">
      <msg>Object does not exist: server received
              a command to query or transform an object that does not
              exist in the repository</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-d31969fb</svTRID>
    </trID>
  </response>
</epp>
```

Domain Transfer Request

<Description>

Example of transfer 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 roid="JD1234-REP">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="2300">
      <msg>server received
              a command to confirm, reject, or cancel the transfer of an
              object when no command has been made to transfer the
              object.</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-b3f90075</svTRID>
    </trID>
```

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

Domain Transfer Query

<Description>

Example of transfer 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="query">
      <domain:transfer xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.com</domain:name>
        <domain:authInfo>
          <domain:pw roid="JD1234-REP">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="2303">
      <msg>Object does not exist: server received
              a command to query or transform an object that does not
              exist in the repository</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-87170742</svTRID>
    </trID>
  </response>
</epp>
```

Contacts

Contact Check

<Description>

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>sh8013</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">sh8013</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-f56ad4c0</svTRID>
    </trID>
  </response>
</epp>
```

Contact Info

<Description>

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:authInfo>
          <contact:pw>2fooBAR</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">+1.7035555555</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-a7b87018</svTRID>
    </trID>
  </response>
</epp>
```

Contact Create

<Description>

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>jd1234</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">+1.7035555555</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="2302">
      <msg>Object exists: server received
             a command to create an object that already exists in the
             repository.</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-af970eb3</svTRID>
    </trID>
  </response>
</epp>
```

Contact Update

<Description>

Example of contact update command

```
Example incomplete. Update pending.
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
</epp>
```

Example of contact update response

```
Example incomplete. Update pending.
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
</epp>
```

Contact Delete

<Description>

Example of contact delete 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>
    <delete>
      <contact:delete 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>537-abc</contact:id>
      </contact:delete>
    </delete>
    <clTRID>ABC-12345</clTRID>
```

```
  </command>
</epp>
```

Example of contact delete 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="2303">
      <msg>Object does not exist: server received
              a command to query or transform an object that does not
              exist in the repository</msg>
    </result>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>Gedas-5db68bf5</svTRID>
    </trID>
  </response>
</epp>
```