

## HELLO, GREETING

Після встановлення з'єднання з EPP-сервером або після отримання запиту <hello> сервер видає у відповідь XML-документ <greeting> ("вітання").

Документ містить наступні елементи:

- <svID> - ім'я сервера;
- <svDate> - поточні дата і час на сервері;
- <svcMenu> - опис сервісів, які підтримуються сервером:
  - <version> - версії протоколу, підтримувані сервером;
  - <lang> - список мов, підтримуваних сервером;
  - <objURI> - посилання на опис структури використовуваних сервером об'єктів (namespace URIs). Клієнт повинен вказувати в командах ці позначення при описі об'єктів в полях xmlns.
  - <svcExtension> - розширення протоколу, підтримувані сервером.
- <dcp> - (data collection policy) - опис політики конфіденційності сервера при передачі даних. Детальний опис елемента є в RFC5730.

EPP клієнт може запитати в сервера документ <greeting> у будь-який час між командами <login> та <logout>, відправивши запит <hello>.

### Приклад запиту <hello>:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <hello/>
</epp>
```

### Приклад вітання сервера <greeting>:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>LVIV.UA EPP/WEB Server - epp.uar.net</svID>
    <svDate>2018-11-30T18:00:34.963Z</svDate>
    <svcMenu>
      <version>1.0</version>
      <lang>en</lang>
      <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
      <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
      <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>
      <svcExtension>
        <extURI>urn:ietf:params:xml:ns:rgp-1.0</extURI>
        <extURI>urn:ietf:params:xml:ns:auxcontact-0.1</extURI>
        <extURI>urn:ietf:params:xml:ns:secDNS-1.1</extURI>
        <extURI>urn:ietf:params:xml:ns:fee-1.0</extURI>
      </svcExtension>
    </svcMenu>
    <dcp>
      <access>
        <all/>
      </access>
      <statement>
        <purpose>
          <admin/>
          <prov/>
        </purpose>
        <recipient>
          <ours/>
          <public/>
        </recipient>
        <retention>
          <stated/>
        </retention>
      </statement>
    </dcp>
  </greeting>
</epp>
```

## LOGIN

Команда EPP <login> використовується для відкриття сесії з EPP сервером у відповідь на видане сервером вітання <greeting>

Команда EPP <login> має бути відправлена перед будь-якою іншою командою. Число невдалих спроб, після яких сервер закриває з'єднання обмежено і визначається параметрами сервера.

- <clID> - елемент містить ідентифікатор клієнта (login). Це рядок завдовжки від 3 до 16 символів, що складається з букв латинського алфавіту, цифр і знаку "-".
- <pw> - елемент містить пароль клієнта - рядок завдовжки від 6 до 30 символів. Інформація в цьому полі регістрозалежна.
- <options> - елемент містить такі піделементи:
  - <version> - версія протоколу.
  - <lang> - мова для повідомлень сервера.Значення параметрів <version> и <lang> повинні збігатися із значеннями, запропонованими сервером у вітанні <greeting>
- <svcs> - містить елементи <objURI>, вказуючі на URI схем просторів імен (namespace), такі, що описують об'єкти, з якими здійснюватиметься робота в ході сесії. Значення параметрів <objURI> повинні збігатися із значеннями, запропонованими сервером у вітанні <greeting>.

### Приклад команди <login>:

```
<?xml version="1.0" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <login>
      <clID>{$clid}</clID>
      <pw>{$passwd}</pw>
      <options>
        <version>1.0</version>
        <lang>en</lang>
      </options>
      <svcs>
        <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
        <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
        <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>
      </svcs>
    </login>
  <clTRID>{$cltrid}</clTRID>
</command>
</epp>
```

### Приклад відповіді сервера на команду <login>:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>cltrid-1234567890-0</clTRID>
      <svTRID>9876543210123</svTRID>
    </trID>
  </response>
</epp>
```

## LOGOUT

Команда EPP <logout> використовується для закриття сесії з EPP сервером

### Приклад команди <logout>:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
      <logout/>
      <clTRID>{${cltrid}}clTRID</clTRID>
    </command>
  </epp>
```

### Приклад відповіді сервера на команду <logout>:

```
<?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">
  <response>
    <result code="1500">
      <msg>Command completed successfully; ending session</msg>
    </result>
    <trID>
      <clTRID>cltrid-1234567890-0</clTRID>
      <svTRID>9876543210123</svTRID>
    </trID>
  </response>
</epp>
```

## POLL

Команда EPP <poll> призначена для читання службових повідомлень EPP-сервера, адресованих клієнтові.

Якщо черга повідомлень не порожня, то успішна відповідь на команду <poll op="req"> поверне перше повідомлення з черги повідомлень. У відповіді серверу міститься унікальний ідентифікатор повідомлення (необхідний для підтвердження отримання повідомлення) та лічильник, що показує загальну кількість повідомлень в черзі. Наприклад:

```
<msgQ count="45" id="123"/>
```

Також повідомлення про наявність непрочитаних повідомлень може бути додане до відповідей на інші команди, наприклад <login>

Після отримання повідомлення клієнт повинен відповісти на нього явним підтвердженням прийому. Після отримання підтвердження сервер видалить повідомлення з черги, зменшить лічильник черги і зробить доступним для читання наступне повідомлення в черзі, якщо воно існує.

Системні повідомлення створюються для клієнтів, що попали під дію над об'єктом. Наприклад, інформація про надходження команди <transfer> ставиться в чергу реєстраторові, який уповноважений прийняти або відхилити запит.

Повідомлення в черзі зберігаються обмежений час, який визначається параметрами Реєстру. Якщо клієнти дуже довго не видаляють прочитані повідомлення (не підтверджують їх прийом), то найстаріші повідомлення можуть бути автоматично видалені.

Команда <poll> має атрибут "op", що визначає дію, що буде зроблена командою:  
"req" - читання першого непрочитаного повідомлення з черги повідомлень;  
"ack" - підтвердження прочитання повідомлення.

### Приклад команди <poll op="req">:

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

### Приклад відповіді на команду <poll op="req">:

```
<?xml version="1.0" encoding="UTF-8"?>
<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; ack to dequeue</msg>
    </result>
    <msgQ count="45" id="123">
      <qDate>2018-11-30T21:00:05.941Z</qDate>
      <msg lang="en">
        <![CDATA[<offlineUpdate>
          <domain>
            <name>test.lviv.ua</name>
            <change>CONTACTS_CHANGED</change>
            <details>Contacts updated by housekeeping</details>
          </domain>
        </offlineUpdate>]]>
      </msg>
    </msgQ>
    <trID>
      <clTRID>cltrid-1234567890-0</clTRID>
      <svTRID>9876543210123</svTRID>
    </trID>
  </response>
</epp>
```

В атрибуті "id" елементу msgQ вказується номер поточного повідомлення, в атрибуті "count" - кількість повідомлень, що залишилися в черзі.

Клієнт повинен підтвердити прийом кожної відповіді, щоб виключити повідомлення з черги і зробити подальші повідомлення доступними для читання.

Команда <poll> с атрибутом "ack" повинна мати додатковий атрибут "msgID", в якому потрібно зазначити номер повідомлення в черзі.

**Приклад підтвердження за допомогою команди <poll op="ack">:**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
    <command>
      <poll op="ack" msgID="{Sid}" />
      <clTRID>{cltrid}</clTRID>
    </command>
  </epp>
```

**Приклад відповіді на команду <poll op="ack"> з підтвердженням читання повідомлення:**

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
    <msgQ count="44" id="124"/>
    <trID>
      <clTRID>cltrid-1234567890-0</clTRID>
      <svTRID>9876543210123</svTRID>
    </trID>
  </response>
</epp>
```