1. Аутентификация и регистрация профиля

1.1. Аутентификация с действующим сертификатом

Клиент выбирает действующий сертификат (base64) и подписывает его простой подписью этим же сертификатом.

Далее выполняем запрос: /api/auth/ с параметрами:

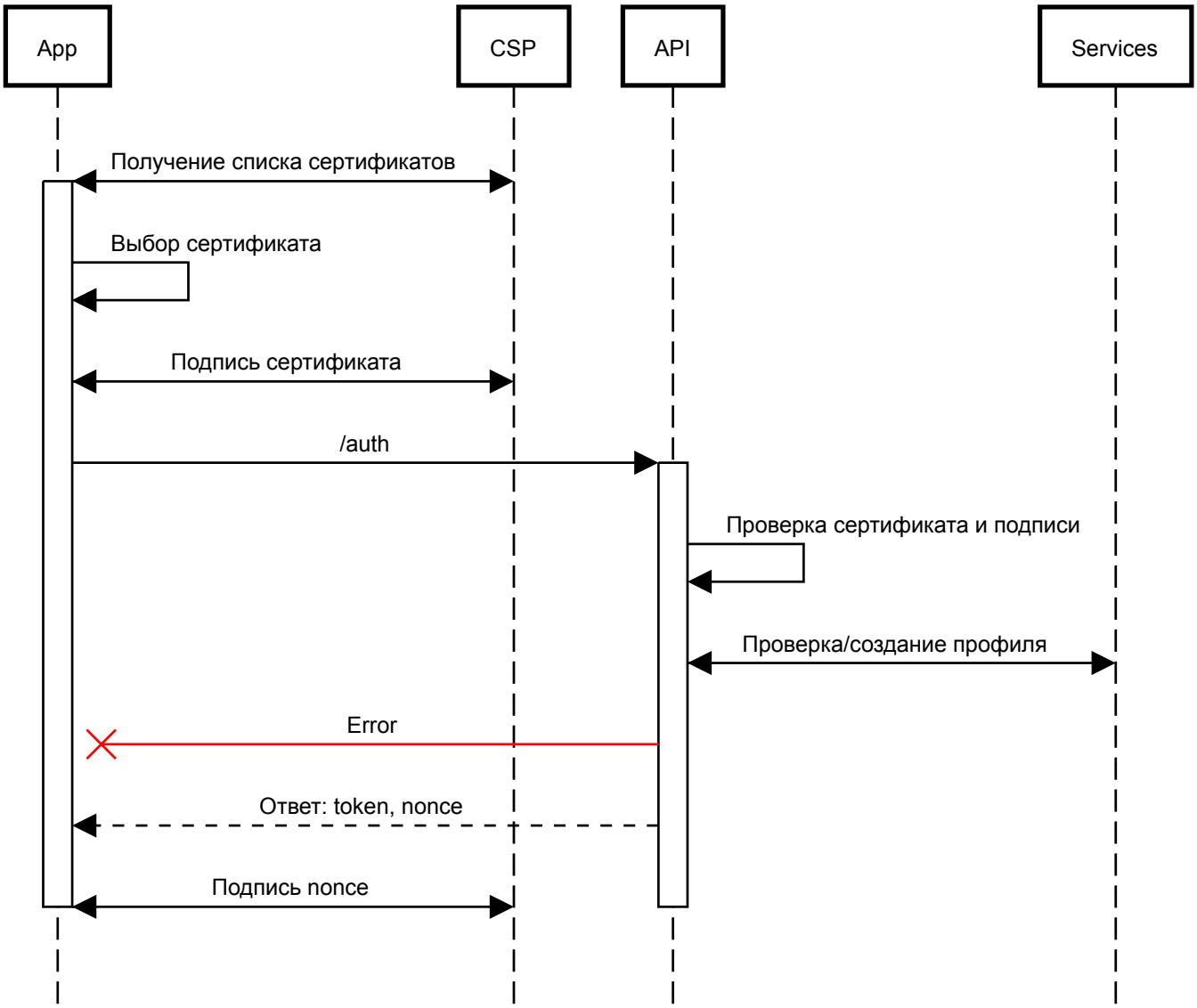
```
POST https://api.komandor.app/api/v2/auth/
Content-Type: application/json


{
  "cert": "(base64) Текущий сертификат",
  "sign": "(base64) Простая подпись сертификата"
}
```

Сервис проверяет сертификат на действительность и корректность подписи, получает или создает профиль, и возвращает объект с данными (или код ошибки) для подтверждения аутентификации:

```
{
  "success": true,
  "data": {
    "nonce": "(base64) Параметр nonce",
    "token": "(jwt) Временный токен аутентификации"
  }
}
```

- `nonce` - случайна строка для подписи или шифрования
- `token` - JWT токен с данными `nonce` и `cid` для подтверждения аутентификации на втором этапе. Срок действия токена - 1 минута.

Подписываем параметры `nonce` и выполняем второй этап.



1.2. Аутентификация с просроченным сертификатом и активным профилем

Клиент выбирает просроченный сертификат (base64) и не подписывая его выполняем запрос: /api/auth/ с параметрами:

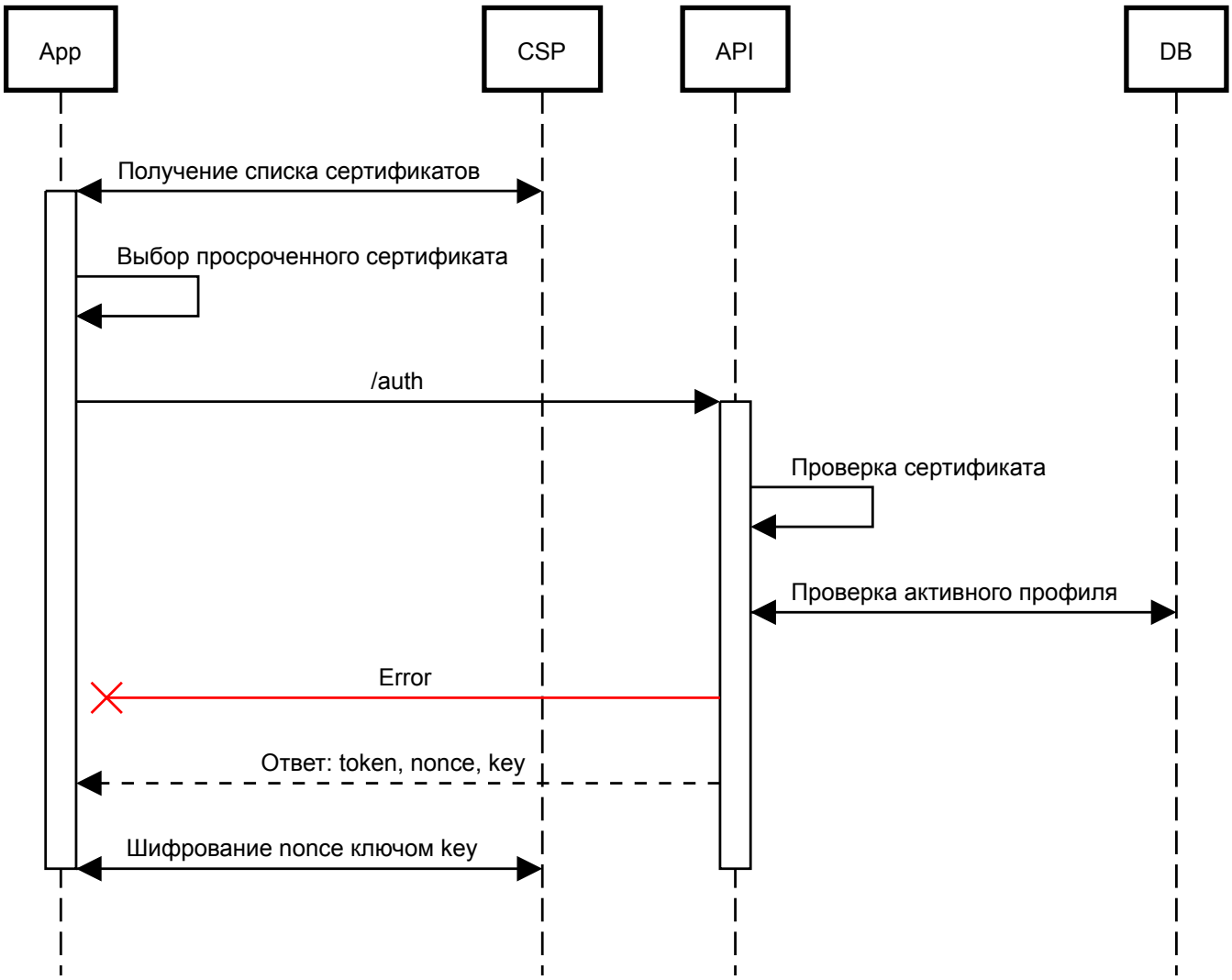
```
POST https://api.komandor.app/api/v2/auth/
Content-Type: application/json

{
  "cert": "(base64) Текущий сертификат"
}
```

Сервис проверяет зарегистрированный сертификат и профиль, возвращает объект с данными (или код ошибки) для подтверждения аутентификации:

```
{
  "success": true,
  "data": {
    "nonce": "(base64) Параметр nonce",
    "token": "(jwt) Временный токен аутентификации",
    "key": "(base64) Ключ шифрования"
  }
}
```

При просроченном сертификате мы не можем отправлять сообщения т.к. нет возможности поставить подпись, но можем сделать синхронизацию всех чатов на новый сертификат.

Шифруем параметр `nonce` ключом `key` и выполняем второй этап.



1.3. Подтверждение аутентификации

Выполняем запрос: /api/confirmAuth/ для подтверждения аутентификации передав параметры `sign` для действующего сертификата или `data` для просроченного сертификата:

```
POST https://api.komandor.app/api/v2/confirmAuth/
Authorization: Bearer JWT_TOKEN
Content-Type: application/json

{
  "sign?": "(base64) Простая подпись nonce",
  "data?": "(base64) Зашифрованный nonce"
}
```

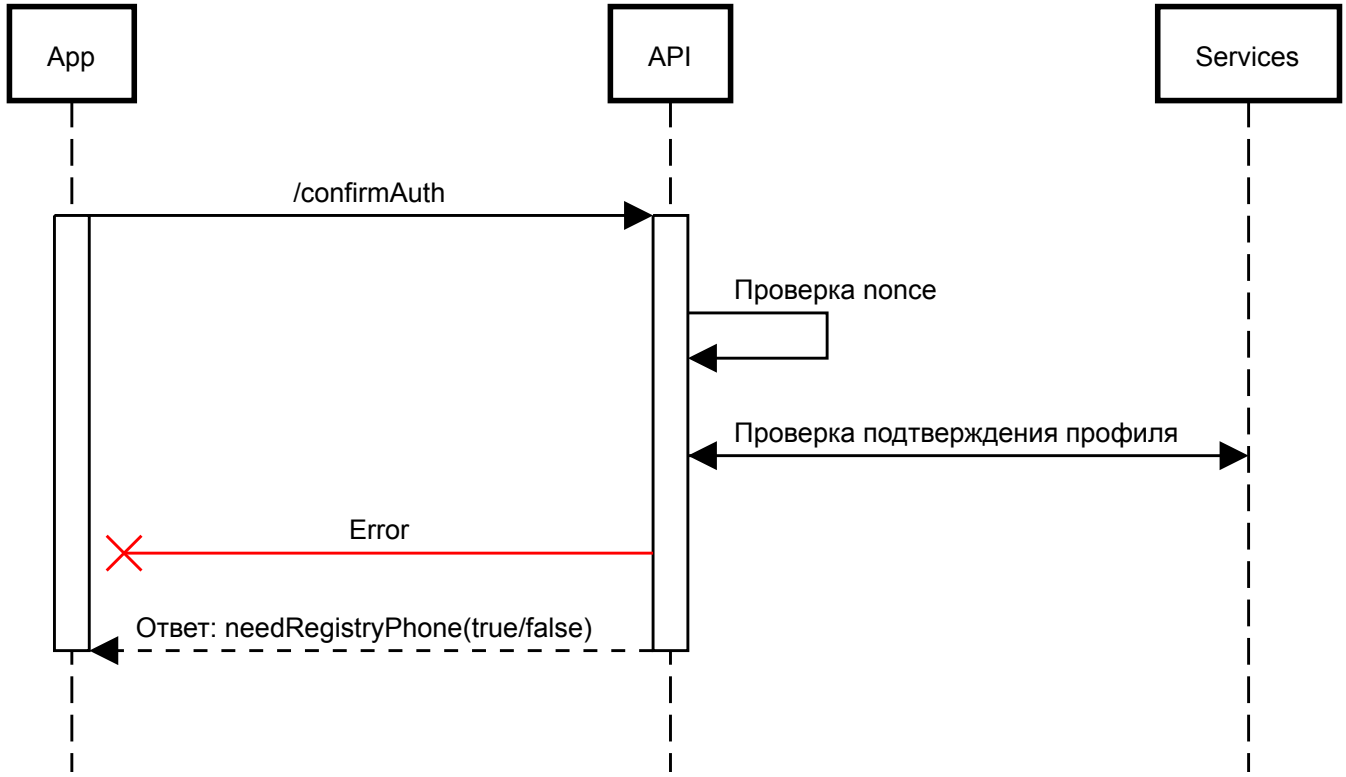
- Далее во всех запросах передается JWT токен сессии. Если токен истек, производим повторную аутентификацию.

Получаем ответ:

```
{
  "success": "(boolean)",
  "data": {
```

```
    "token": "(jwt) Токен сессии",
    "needRegisterPhone": "(boolean)"
  }
}
```

Если параметр `needRegisterPhone = true` то необходимо выполнить регистрацию профиля.



1.4. Регистрация профиля

Для регистрации профиля необходимо, передать Российский мобильный номер телефона для получения SMS с кодом подтверждения.

Выполняем запрос: /api/registryPhone/ с параметрами:

```
POST https://api.komandor.app/api/v2/registryPhone/
Authorization: Bearer JWT_TOKEN
Content-Type: application/json

{
  "phone": "(format) +79XXXXXXXXX"
}
```

Ответ:

```
{
  "success": "(boolean)",
  "data": {
    "timeout": "(number) Таймаут в секундах для повторного запроса кода"
  }
}
```

Для подтверждения номера телефона (привязка к профилю) выполняем запрос: /api/confirmPhone/ с параметрами:

```
POST https://api.komandor.app/api/v2/confirmPhone/
Authorization: Bearer JWT_TOKEN
```

```
Content-Type: application/json

{
  "code": "(string) Код подтверждения"
}
```

Ответ:

```
{
  "success": "(boolean)",
  "error?": "(string) Код ошибки",
  "message?": "(string) Сообщение ошибки",
  "data?": {
    "timeout": "(number) Таймаут в секундах для повторного запроса кода"
  }
}
```

Если код не подтвержден, то будет выдан параметр `timeout` и код ошибки для повторного запроса кода.