

Протокол Клиент-Сервер UpDAU

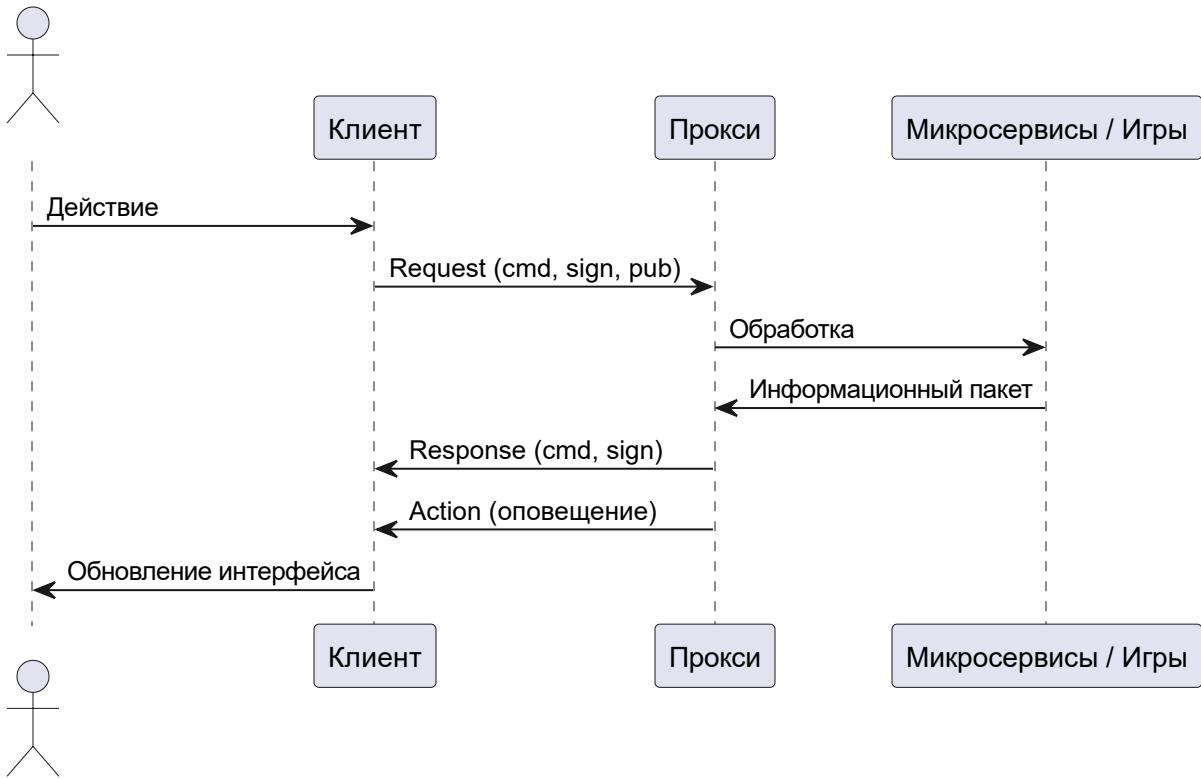
В данном документе описывается протокол взаимодействия между клиентским программным обеспечением и сервером (Прокси) системы UpDAU.

Основные понятия

- **Клиент** — программный комплекс, реализующий логику на стороне пользователя.
- **Канал** — виртуальная сущность, идентифицируемая уникальным ключом (`pub`), предназначенная для управления состояниями и идентификации данных. См. [Состояния каналов](#).
- **Прокси** — серверный компонент, обеспечивающий работу с пользователями, транспортом и каналами, а также маршрутизацию запросов.
- **Стол** — "игровое окружение", модель, реализующая логику конкретного игрового стола. Стол связан с игрой, которая обрабатывает сообщения и рассыпает оповещения участникам.
- **Участники** — каналы, присоединенные к столу. Все участники получают оповещения о событиях.
- **Боксы** — посадочные места за столом (нумерация с 0).
- **Игровые критерии** — набор параметров (ключ-значение), описывающих правила и условия игры (ставка, количество игроков и т.д.).

Подробное описание жизненных циклов этих объектов доступно в документе [Жизненные циклы](#).

Схема взаимодействия



Формат сообщений

Обмен данными может происходить в формате **XML** или **JSON**. Поддерживаются три типа сообщений: запрос, ответ и оповещение.

Request (Запрос)

Отправляется клиентом серверу.

```
<request cmd="..." [pub="..."] [sign="..."]>
    [content]
</request>
```

- **cmd** — имя команды.
- **pub** — идентификатор канала (необязателен для общих команд).
- **sign** — произвольная метка для отслеживания асинхронного ответа.
- **content** — тело команды (обычно XML-структура).

Response (Ответ)

Отправляется сервером в ответ на конкретный запрос.

```
<response cmd="..." [pub="..."] [sign="..."]>
    [content]
</response>
```

Атрибуты `pub` и `sign` соответствуют значениям из запроса.

Action (Оповещение)

Асинхронное сообщение от сервера, не привязанное к конкретному запросу.

```
<action cmd="..." pub="...">>
    [content]
</action>
```

Pack (Пакет сообщений)

Сервер или клиент могут группировать сообщения для последовательной обработки.

```
<pack>
    <response ... />
    <action ... />
    ....
</pack>
```

Общие команды (без привязки к каналу)

PING (Проверка соединения)

Позволяет проверить активность соединения и замерить задержку.

Запрос:

```
<request cmd="ping" [sign="xxx"]>arbitrary data</request>
```

Ответ:

```
<response cmd="ping" [sign="xxx"]>arbitrary data</response>
```

VER (Версия сервера)

Возвращает информацию о версии сборки прокси-сервера.

Запрос:

```
<request cmd="ver" [sign="xxx"]/>
```

Ответ:

```
<response cmd="ver" [sign="xxx"]>
  <version>2.0</version>
  <build>1.0.6451.33774</build>
</response>
```

GAMES (Доступные игры)

Возвращает список игр и их настроек (конвенций), доступных для матчинга.

Запрос:

```
<request cmd="games" [sign="xxx"]/>
```

Ответ:

```

<response cmd="games" [sign="xxx"]>
  <game name="durak" alias="durak" min_players="2" max_players="4">
    <type name="durak_podkidnoi" view="Дурак подкидной">
      <bet>
        <currency name="FUN">
          <option default="true" view="100 фишек">100</option>
          <option view="300 фишек">300</option>
        </currency>
      </bet>
      <criterias>
        <var name="num_seats" view="Количество игроков">
          <option default="true" view="2">2</option>
          <option view="3">3</option>
          <option view="4">4</option>
        </var>
      </criterias>
    </type>
  </game>
</response>

```

AUTH (Авторизация)

Авторизация пользователя в системе. Используется токен, полученный на White Label (WL).

Запрос:

```

<request cmd="auth" [sign="xxx"]>
  <platform value="web" />
  <bundle value="com.globo.durak" />
  <token value="AUTH_TOKEN_HERE" />
</request>

```

Ответ (успех):

```
<response cmd="auth" [sign="xxx"]>
  <user>
    <userinfo uid="123" nick="Player1" lvl="5" exp="1500" />
  </user>
  <channels>
    <channel pub="1" tag="main">
      <state value="free" />
    </channel>
  </channels>
</response>
```

- `channels` — содержит список уже открытых каналов пользователя (если есть).

Ответ (ошибка):

```
<response cmd="auth" [sign="xxx"]>
  <error code="401">Invalid token</error>
</response>
```

USERS (Игроки онлайн)

Возвращает список игроков в сети с их статусами (играют, наблюдают, ищут игру).

Запрос:

```
<request cmd="users" [sign="xxx"]/>
```

Ответ:

```
<response cmd="users" [sign="xxx"]>
  <users>
    <user id="123" wl="site1" platform="web" on="1">
      <matching><criteria game="durak" bet="100" /></matching>
      <play><game table="77" type="durak_podkidnoi" currency="FUN" bet="100"/></play>
    </user>
  </users>
</response>
```

ONLINE (Статистика онлайн)

Количество пользователей по типам игр.

Запрос:

```
<request cmd="online" [sign="xxx"]/>
```

Ответ:

```
<response cmd="online" [sign="xxx"]>
  <users connections="1500" online="1200" />
    <type name="durak_podkidnoi" players="450" watchers="50" />
</response>
```

Команды работы с каналами (требуют pub)

Большинство игровых команд требуют указания идентификатора канала pub .

Отладочные команды (не доступны в релизе)

SLEEP

Задерживает исполнение запросов на прокси (по умолчанию 50мс). Ответ не возвращается.

Используется для отладки гонок состояний.

```
<request cmd="sleep" [sign="xxx"]>50</request>
```

DEBUG

Включает расширенное логирование и форматирование XML в ответах.

```
<request cmd="debug" [sign="xxx"]/>
```

OPEN (Открыть канал)

Создает новый или переиспользует существующий канал. Каждый канал привязан к пользователю.

Запрос:

```
<request cmd="open" [pub="xxx"] [sign="xxx"]>
  <token value="AUTH_TOKEN" [tag="my-tag"]/>
</request>
```

Ответ:

```
<response cmd="open" pub="xxx" [sign="xxx"]>
  <state value="free"/>
  <startup>
    <wallet value="5000" currency="FUN" />
  </startup>
</response>
```

- Если `pub` не указан, сервер назначит его автоматически.
- Канал может возвращать текущее состояние игры, если пользователь уже в игре (см. [Состояния каналов](#)).

DETACH (Отсоединить канал)

Отвязывает канал от текущего транспортного соединения. Состояние канала в системе при этом не меняется.

Запрос:

```
<request cmd="detach" pub="xxx" [sign="xxx"]/>
```

FIND (Поиск игры / Матчинг)

Поиск соперников по заданным критериям. После сбора нужного количества участников игра начнется автоматически.

Запрос:

```
<request cmd="find" pub="xxx" [sign="xxx"]>
    <criteria game="durak_podkidnoi" currency="FUN" players="2,3,4" bet="100,300" />
</request>
```

- **players, bet** — можно указывать несколько значений через запятую.
- Сервер заблокирует максимальную сумму из указанных ставок.

CREATE (Создать игру)

Прямое создание игры (обычно для одиночных игр или слотов).

Запрос:

```
<request cmd="create" pub="xxx" [sign="xxx"]>
    <criteria game="slot-AL" />
</request>
```

JOIN (Присоединиться по ID)

Присоединение к конкретному столу (например, по приглашению или из списка столов).

Запрос:

```
<request cmd="join" pub="xxx" [sign="xxx"]>
    <table id="table_123" [box="0"] [stack="1000"] />
</request>
```

- **id** — ID стола.
- **box** — желаемое место (0-N).
- **stack** — сумма для входа (если требуется).

SUBSCRIBE (Подписка на лобби)

Клиент подписывается на обновления в "комнате" (обычно по типу игры). Сервер будет присыпать оповещения EVENT о создании/удалении столов.

Запрос:

```
<request cmd="subscribe" pub="xxx" [sign="xxx"]>
  <room id="durak" [filter="FUN"]/>
</request>
```

Ответ:

```
<response cmd="subscribe" pub="xxx">
  <room id="durak">
    <table id="table1" created="10" players="2" watchers="0" bet="100" currency="FUN" gameType="Durak">
      <player id="1" name="User1" status="active" bet="50" currency="FUN" role="player" type="user"/>
      <player id="2" name="User2" status="active" bet="50" currency="FUN" role="player" type="user"/>
    </table>
  </room>
</response>
```

UNSUBSCRIBE (Отписка)

Отменяет все активные подписки на комнаты для данного канала.

Запрос:

```
<request cmd="unsubscribe" pub="xxx" [sign="xxx"]/>
```

TABLEINFO (Информация о столе)

Получение списка игроков и наблюдателей конкретного стола.

Запрос:

```
<request cmd="tableinfo" pub="xxx">
  <table id="table1"/>
</request>
```

FINDCANCEL (Отмена поиска)

Отменяет ранее запущенный через FIND процесс матчинга.

Запрос:

```
<request cmd="findcancel" pub="xxx" [sign="xxx"]/>
```

BEGIN (Готовность к игре)

Подтверждает готовность начать игру после того, как сервер приспал оповещение `found`.

Запрос:

```
<request cmd="begin" pub="xxx" [sign="xxx"]/>
```

GAME (Игровой протокол)

Команда для передачи игровых действий (ход, пас и т.д.) непосредственно в игровую логику. См. примеры в [Протокол игры Дурак](#).

Запрос:

```
<request cmd="game" pub="xxx">
  <game cmd="turn" card="AD" />
</request>
```

GETWALLET (Состояние кошелька)

Запрашивает баланс пользователя (по токену или для открытого канала).

Запрос:

```
<request cmd="getwallet" pub="xxx">
  <currency id="FUN"/>
</request>
```

GETPROFILE (Профиль пользователя)

Запрашивает публичную информацию о пользователях по их ID.

Запрос:

```
<request cmd="getprofile" pub="xxx">
  <user id="123"/>
  <user id="456"/>
</request>
```

Чат

Команды управления внутриигровым или общим чатом.

CHATJOIN (Вход в чат)

```
<request cmd="chatjoin" pub="xxx">
  <room id="general"/>
</request>
```

CHATMSG (Отправить сообщение)

```
<request cmd="chatmsg" pub="xxx">Привет всем!</request>
```

CHATLEAVE (Выход из чата)

```
<request cmd="chatleave" pub="xxx"/>
```

События чата (action cmd="chat")

- event="setup" — параметры комнаты.
- event="join" / event="leave" — вход/выход участников.
- event="msg" — новое сообщение.

GAMEOVER (Сдаться)

Принудительное завершение игры по инициативе игрока.

REVANCHE (Реванш)

Предложение сыграть снова с тем же составом и правилами. Игра начнется, когда все участники пришлют эту команду.

REVANCHECANCEL (Отказ от реванша)

Отмена предложения или отказ от участия в реванше.

WATCH / UNWATCH (Наблюдение)

Позволяет подключиться к столу в качестве зрителя или покинуть его.

```
<request cmd="watch" pub="xxx">
    <table id="table1"/>
</request>
```

LEAVE (Временный уход)

Игрок покидает стол, но игра продолжается (например, управление передается боту или просто ожидается возвращение). Канал перестает получать игровые события.

RETURN (Возврат)

Возврат за стол после команды LEAVE .

GETSTAT (Статистика)

Запрос статистики игрока (общей или по конкретной игре).

```
<request cmd="getstat" pub="xxx">
    <game type="durak_podkidnoi"/>
</request>
```

Оповещения (Actions)

Оповещения приходят клиенту асинхронно в открытый канал.

Общий формат: `<action cmd="ИМЯ" pub="ID_КАНАЛА">[content]</action>`

auth

Оповещение о параллельной авторизации под тем же аккаунтом на другом устройстве.

begin

Оповещение о фактическом старте игры. Содержит начальное состояние:

```
<action cmd="begin" pub="xxx">
  <gameinfo id="58" gameId="10" bet="100" currency="FUN" gameType="durak" />
  <self box="0" />
  <players>
    <player uid="123" box="0" timeBank="30" />
  </players>
</action>
```

timerstart / timerstop / timer

Управление и уведомление о работе игровых таймеров.

wait / waitreset / waitstop / timeout

Управление состоянием ожидания хода.

- `wait` — сервер ждет действия от игрока `box`.
- `timeout` — время на раздумья вышло.

partyend / gameover

Уведомления о завершении партии или всей игры с результатами.

```
<action cmd="gameover" pub="xxx">
  <box id="0" place="1" points="10" win="200" />
  <box id="1" place="-1" points="5" win="0" />
</action>
```

wallet

Изменение баланса кошелька.

- `type="BET"` — списание/блокировка ставки.
- `type="WIN"` — начисление выигрыша.
- `type="LOSE"` — фиксация проигрыша.

state

Изменение состояния канала. См. [Состояния каналов](#).

event (События лобби)

Результат подписки SUBSCRIBE .

Имя события	Описание	Содержимое DATA
new	Создан новый стол	<table id="..." players="..." game="..." bet="..." />

found (Соперники найдены)

Приходит после команды find . Означает, что сервер подобрал игроков. Клиент должен ответить командой begin .

```
<action cmd="found" pub="xxx">
  <criteria game="durak" num_seats="2" />
  <players>
    <player uid="123" level="5" />
    <player uid="456" level="3" />
  </players>
</action>
```

Диаграмма матчинга

