

# Интерфейс взаимодействия оператора с порталом oll.tv (ispAPI v.2.1.5)

## Термины и понятия

Сервер — ресурс на котором размещен портал oll.tv.

Оператор (провайдер) — телеком-оператор, с которым есть договорные отношения.

Подписка типа бандл (Bundle) – подписка, включаемая оператором пользователю.

Подписка типа VAS – подписка, которую пользователь покупает самостоятельно через свой личный кабинет на портале oll.tv.

Подписка для доступа к основной услуге — подписка типа бандл, предоставляющая доступ к просмотру контента на портале oll.tv.

Подписка для доступа с доп. экрана — подписка типа бандл, не дающая доступ к контенту, а предоставляющая пользователю возможность привязать дополнительное устройство; активация возможна только после подписки на основную услугу и одновременно может быть активировано несколько подписок данного типа.

## Общая информация

Методы, возвращающие блоки данных, вызываются при помощи HTTP GET запросов на сервер. Сервер, в свою очередь, возвращает данные в виде JSON.

Методы, передающие данные от клиента к серверу, вызываются при помощи HTTP POST, данные в POST запросе передаются в прямом виде, ответ же кодируется в JSON.

## Логика взаимодействия с сервером

Корневая папка для запросов на сервер находится по адресу: <http://oll.tv/ispAPI/>.

Для тестирования используется сервер dev.oll.tv. Доступ на него возможен только для предоставленных ip-адресов.

Первый запрос должен прийти на метод авторизации. Используя логин и пароль для ispAPI, клиентское приложение получает уникальный хеш, который должно передавать при последующих обращениях к серверу.

Пример.

*Запрос:*

```
POST /ispAPI/auth2/
```

```
login=isp#1&password=password#1
```

*Ответ:*

```
{"status": "0", "hash": "o32xf3oddksk33"}
```

После успешной авторизации доступны остальные методы API, с обязательной передачей хеша, используя ключ "hash", например:

```
GET /ispAPI/emailExists?hash=o32xf3oddksk33
```

## Формат ответа

Ответ представляет собой объект в JSON, в котором обязательно присутствует элемент **status**, значение которого в случае успешного выполнения запроса равно нулю, а в случае ошибки — коду ошибки. Второй элемент в общем случае — **data** (где это не так — указано отдельно). Ответ, также,

может содержать поле **warnings** со списком предупреждений (например, об отсутствии в запросе каких-либо полей или неправильном их значении).

Если же какой-либо метод не может быть выполнен по той или иной причине сервер возвращает объект в JSON с полями **status** и **message**, где message – описание ошибки (*перечень возможных ошибок* описан в одноимённом разделе данной документации).

В случае, если хеш ещё не выдан или не может быть верифицирован, элемент status дублируется элементом id для совместимости с ispAPI версии 1.x.x.

## Список методов

### Метод авторизации

`http://oll.tv/ispAPI/auth2/`

Принимает два параметра, `login` и `password`.

Возвращает хеш (элемент объекта `hash`).

Возможные коды ошибок: 111, 112, 113.

### 1. Работа с пользователями

#### 1.1. Метод проверки существования пользователя с данным email в БД oll.tv

`http://oll.tv/ispAPI/emailExists/`

Принимает один параметр – **email**.

Возвращает 1 если email существует в базе, и 0 – если нет.

Возможные коды ошибок: 109, 110, 200, 201.

#### 1.2. Метод проверки существования провайдерского аккаунта в БД oll.tv

`http://oll.tv/ispAPI/accountExists/`

Принимает один параметр – **account** – аккаунт абонента на стороне провайдера.

Если такой абонент уже есть у нас в базе – будет возвращён объект с информацией по этому абоненту, в противном случае будет возвращен 0.

Структура возвращаемого JSON:

```
{
  data: {
    ID:
    email:
    birth_date:
    reg_date:
    firstname:
    gender:
    account:
    lastname:
    phone:
    region:
    receive_news:
    index:
    active:
    bought_subs: [
      {
        ID:
        sub_id:
        name:
        service_type:
        parent_id:
        start_date:
        expiration_date:
        device_binding_code:
        activation_type:
      }
      ...
    ]
  }
}
```

```
    status:0  
}
```

Возможные коды ошибок: 109, 110, 200.

### 1.3. Метод добавления (регистрации) нового пользователя в БД oll.tv

<http://oll.tv/ispAPI/addUser/>

Принимает следующие параметры, из которых первые два – обязательны:

1. **email**
2. **account**
3. **birth\_date** (допускаются форматы YYYY-MM-DD или DD.MM.YYYY)
4. **gender** (значения М или F, по-умолчанию: М)
5. **firstname** (по-умолчанию: «Гость»/«Гостья»)
6. **password** (если не задан — сгенерируется автоматически, минимум 6 символов)
7. **lastname**
8. **phone** (пример: 0501234567)
9. **region**
10. **receive\_news** (значения 1 или 0, по-умолчанию: 1)
11. **send\_registration\_email** (слать ли пользователю сообщение о регистрации, значения 1 или 0, по-умолчанию: 1)
12. **index**(почтовый индекс или другой идентификатор региональной привязки абонента, для отображения дополнительных региональных каналов)

Возвращает ID созданного пользователя.

Возможные коды ошибок: 109, 110, 115, 116, 120, 200, 301.

### 1.4. Метод получения списка пользователей

<http://oll.tv/ispAPI/getUserList/>

Принимает следующие необязательные параметры:

1. **offset**
2. **limit** (не больше 1000, по умолчанию равно 1000)
3. **with\_subs** — возвращать ли в ответе информацию об активных подписках (массив `bought_subs` как в методе `getUserInfo`)

Метод возвращает список пользователей oll.tv (с подробной информацией), которые “привязаны” к провайдеру.

Структура возвращаемого JSON:

```
{  
  data: [  
    {  
      ID:  
      email:  
      birth_date:  
      reg_date:  
      firstname:  
      gender:  
      account:  
      lastname:  
      phone:  
      region:  
      receive_news:  
      index:  
      active:  
      bought_subs: [  
        {
```

```

        ID:
        sub_id:
        name:
        service_type:
        parent_id:
        start_date:
        expiration_date:
        device_binding_code:
        activation_type:
    }
    ...
]
}
...
]
status:0
}

```

Возможные коды ошибок: 109, 110, 200.

### 1.5. Метод установки/изменения провайдерского аккаунта и привязывания пользователя к провайдеру

<http://oll.tv/ispAPI/changeAccount>

После выполнения данного метода, если пользователь не был ранее привязан к провайдеру, пользователю на почту будет выслано письмо с уведомлением о том, что он будет привязан к провайдеру, и ссылкой, перейдя по которой, он завершит процедуру привязки к провайдеру.

Принимает следующие обязательные параметры:

1. **email** — кому изменить
2. **account** — новое значение

Не возвращает ничего кроме статуса.

Возможные коды ошибок: 109, 110, 200, 201, 404, 505.

### 1.6. Метод отвязки пользователя от оператора

<http://oll.tv/ispAPI/deleteAccount>

Принимает один обязательный параметр:

1. **account** или **email** или **id** или **ds\_account**

Не возвращает ничего кроме статуса.

Возможные коды ошибок: 109, 110, 200, 404, 505.

### 1.7. Метод изменения email

<http://oll.tv/ispAPI/changeEmail>

Провайдер может менять email только привязанным к нему пользователям. Метод принимает два параметра:

1. **email** — кому изменить
2. **new\_email** — новый адрес

Не возвращает ничего кроме статуса.

Возможные коды ошибок: 109, 110, 200, 201, 205, 404, 505.

### 1.8. Метод получения данных о пользователе

<http://oll.tv/ispAPI/getUserInfo>

Принимает один обязательный параметр:

1.  **account** или **email** или **id** или **ds\_account**

Возвращает объект с данными:

- ▲ email
- ▲ account
- ▲ registration\_date
- ▲ bought\_subs (массив с купленными подписками типа бандл, где каждая подписка представлена объектом с необходимой информацией).
- ▲ alt\_mail (в случае если аккаунт от укртелекома)
- ▲ master\_code – мастер-код привязки устройств абонента, может использоваться вместо кодов привязки от основных услуг или дополнительных экранов(пока только на дев среде).

Структура возвращаемого JSON:

```
{
  data: {
    email:
    account:
    ds_account:
    registration_date
    index:
    altemail:
    bought_subs: [
      {
        ID:
        sub_id:
        name:
        service_type:
        parent_id:
        start_date:
        expiration_date:
        device_binding_code:
        activation_type:
      }
      ...
    ]
  }
  status:0
}
```

Возможные коды ошибок: 109, 110, 200, 404, 505.

## 1.9. Метод изменения пользовательских данных

<http://oll.tv/ispAPI/changeUserInfo>

Провайдер может менять данные только привязанным к нему пользователям.

Метод принимает следующие параметры, из которых первый – обязательный:

1. **account** или **email** или **id** или **ds\_account**
2. **birth\_date** (допускаются форматы YYYY-MM-DD или DD.MM.YYYY)
3. **gender** (значения M или F, по-умолчанию: M)
4. **password**
5. **firstname**
6. **lastname**
7. **phone**
8. **region**
9. **index** (почтовый индекс или другой идентификатор региональной привязки абонента)
10. **multicast** (0 — использовать только unicast)

Не возвращает ничего кроме статуса.

Возможные коды ошибок: 109, 110, 200, 404, 505.

### 1.10. Метод сброса пароля родительского контроля

<http://oll.tv/ispAPI/resetParentControl>

Принимает один обязательный параметр:

1. **account** или **email** или **id** или **ds\_account**

Не возвращает ничего кроме статуса.

Возможные коды ошибок: 109, 110, 200, 404, 505.

### 1.11. Метод регенерации мастер-кода для привязки устройств(пока только на дев среде)

<http://oll.tv/ispAPI/resetMasterCode>

Принимает один обязательный параметр:

1. **account** или **email** или **id** или **ds\_account**

Не возвращает ничего кроме статуса.

Формат ответа: {"status": "0", "data": "1"}

Возможные коды ошибок: 109, 110, 200, 404, 505.

## 2. Работа с подписками

### 2.1. Метод включения пользователю бандл-подписки

<http://oll.tv/ispAPI/enableBundle>

Принимает три обязательных параметра:

1. **account** или **email** или **id** или **ds\_account**
2. **sub\_id**
3. **type**

Параметр **sub\_id** принимает идентификатор подписки, которую необходимо включить пользователю; необходим в случае если провайдеру доступно более одного варианта бандл-подписки. Подписка на доступ с доп. экрана не может быть включена раньше подписки на основную услугу.

Параметр **type** предназначен для передачи уточняющей информации об активации и может принимать одно из следующих значений:

- *subs\_free\_device* - Новый контракт - 24 мес и оборудование за 1 грн
- *subs\_buy\_device* - Новый контракт - покупка оборудования
- *subs\_rent\_device* - Новый контракт - аренда оборудования
- *subs\_no\_device* - Новый контракт - без оборудования
- *subs\_renew* - Восстановление текущего контракта

В случае удачи возвращает строку, содержащую код привязки дополнительных устройств(если такая возможность входит в бандл-подписку) или 1, в противном — 0.

Возможные коды ошибок: 109, 110, 200, 404, 407, 408, 505, 506.

### 2.2. Метод выключения у пользователя бандл-подписки

<http://oll.tv/ispAPI/disableBundle>

Принимает три обязательных параметра:

1. **account** или **email** или **id** или **ds\_account**
2. **sub\_id**
3. **type**

Параметр **sub\_id** принимает идентификатор подписки, которую необходимо выключить пользователю; необходим в случае если провайдеру доступно более одного варианта бандл-

подписки. Подписка на основную услугу не может быть отключена при наличии активных подписок на доп. Экраны или дополнительных пакетов (ФутболHD, МегаХит, Аренда оборудования).

Параметр **type** предназначен для передачи уточняющей информации о деактивации и может принимать одно из следующих значений:

- *subs\_break\_contract* - Разрыв договора
- *subs\_negative\_balance* - Отрицательный баланс
- *subs\_malfunction* - Технические неполадки
- *subs\_vacation* – Каникулы
- *subs\_cancel* – отказ абонента(для отложенного старта услуги)

Все пользовательские устройства, которые были привязаны с помощью кода привязки, который был выдан по данной подписке, будут автоматически отвязаны от пользователя.

В случае успеха возвращает 1, в противном — 0.

Возможные коды ошибок: 109, 110, 200, 404, 407, 408, 504, 505.

### 2.3. Метод проверки состояния бандл-подписки у пользователя

<http://oll.tv/ispAPI/checkBundle>

Принимает два обязательных параметра:

1. **account** или **email** или **id** или **ds\_account**
2. **sub\_id**

Параметр **sub\_id** принимает идентификатор подписки, по которой необходимо выполнить проверку; необходим в случае если провайдеру доступно более одного варианта бандл-подписки.

В случае если подписка активна возвращает 1, в противном — 0.

Для получения более подробной информации используйте метод `getUserInfo` (1.8).

Возможные коды ошибок: 109, 110, 200, 404, 407, 505.

### 2.4. Метод смены у пользователя бандл-подписки

<http://oll.tv/ispAPI/changeBundle>

Принимает три обязательных параметра:

1. **account** или **email** или **id** или **ds\_account**
2. **old\_sub\_id**
3. **new\_sub\_id**

Параметр **old\_sub\_id** принимает идентификатор подписки, которая активна у пользователя в данный момент, а **new\_sub\_id** — идентификатор подписки, на которую осуществляет смена. Смену можно производить только с подписками на основной тарифный план, при этом активные доп. экраны будут автоматически перенесены на новый основной тарифный план с сохранением кодов привязки.

В случае, если у пользователя неактивна подписка, идентификатор которой передается в параметре **old\_sub\_id**, возвращает 0.

В случае, если подписка, идентификатор которой передается в параметре **new\_sub\_id**, уже активна возвращает 1.

Не возвращает ничего кроме статуса.

Возможные коды ошибок: 109, 110, 200, 404, 407, 504, 505.

### 2.4. Метод получения списка активных подписок провайдера за период

<http://oll.tv/ispAPI/getAllPurchases>

Принимает два параметра:

1. **start\_date**
2. **page**



Параметры `start_date` – начало отчетного периода, обязателен. Параметр `page` – номер страницы, каждая страница возвращает до 10000 `purchases`, не обязателен

Структура возвращаемого JSON:

```
{
  data: {
    purchases: [
      0: {
        TransactionID:
        SubscriberDSID:
        SubscriberProviderID:
        TariffName:
        TariffID:
        PurchaseDate:
        StartDate:
        ExpirationDate:
        StartOfReportingPeriod:
        EndOfReportingPeriod:
        RetailPrice:
        ProviderPrice:
        DaysWorkBundle
      }
      ...
    ]
    Count:
    DaysInMonth:
    ProviderID:
    ProviderName:
    Page:
  }
  status: 0
}
```

Возможные коды ошибок: 109, 110, 200.

### 3. Работа с устройствами

#### 3.1. Метод добавления и привязки нового устройства пользователю

<http://oll.tv/ispAPI/addDevice/>

Принимает следующие обязательные параметры:

1. **account** или **email** или **id** или **ds\_account**
2. **serial\_number** — серийный номер устройства
3. **mac** — mac-адрес устройства
4. **device\_type** — тип устройства (например: `stb`, `dune`, `smarttv`, `lge`, `samsung`, `ipad` и т.д.)
5. **device\_model** — модель устройства
6. **binding\_code** — код привязки устройства, выданный во время активации подписки для доступа с доп. экрана
7. **type**

Из параметров **mac** и **serial\_number** обязателен один любой, но *желательно* использование обоих вместе.

Параметр **binding\_code** обязателен для провайдеров, которые работают с подписками для доступа с доп. устройств.

Параметр **type** предназначен для передачи уточняющей информации об привязке и может принимать одно из следующих значений:

- **device\_free** - Новый контракт - 24 мес и оборудование за 1 грн

- *device\_buy* - Новый контракт - покупка оборудования
- *device\_rent* - Новый контракт - аренда оборудования
- *device\_change* - Сервисная замена текущего оборудования

Не возвращает ничего кроме статуса.

Возможные коды ошибок: 109, 110, 119, 200, 203, 301, 302, 303, 304, 305, 404, 405, 505.

### 3.2. Метод отвязки существующего устройства от пользователя

<http://oll.tv/ispAPI/delDevice/>

Принимает следующие обязательные параметры:

1. **account** или **email** или **id** или **ds\_account**
2. **mac** — mac-адрес устройства
3. **serial\_number** — серийный номер устройства
4. **type**

Из параметров **mac** и **serial\_number** обязателен один любой, но *желательно* использование обоих вместе.

Если передается параметр **account** – перед удалением устройства будет произведена проверка на то, что это устройство привязано именно к этому пользователю.

Параметр **type** предназначен для передачи уточняющей информации об отвязке и может принимать одно из следующих значений:

- *device\_break\_contract* - Окончание контракта
- *device\_change* - Сервисная проблема оборудования

Не возвращает ничего кроме статуса.

Возможные коды ошибок: 109, 110, 117, 200, 203, 406, 501.

### 3.3. Метод проверки существования устройства в БД oll.tv

<http://oll.tv/ispAPI/deviceExists/>

Принимает следующие параметры:

1. **mac** — mac-адрес устройства
2. **serial\_number** — серийный номер устройства

Обязателен один любой параметр, но *желательно* использование обоих вместе.

Если устройство привязано к пользователю, который является абонентом другого оператора, в ответ будет возвращена ошибка 501.

Возвращает объект с информацией по устройству, если оно было найдено, или же 0 в противном случае.

Структура возвращаемого JSON:

```
{
  data: {
    ID:
    serial_number:
    mac:
    date_added:
    USER: {
      ID:
      email:
      birth_date:
      reg_date:
      firstname:
      gender:
      account:
      lastname:
      phone:
    }
  }
}
```

```

    region:
    receive_news:
    index:
    active:
    bought_subs: [
        {
            ID:
            sub_id:
            name:
            service_type:
            parent_id:
            start_date:
            expiration_date:
            device_binding_code:
            activation_type:
        }
        ...
    ]
    }
    activation_type:
    binding_code:
}
status:0
}

```

Возможные коды ошибок: 109, 110, 200, 501.

### 3.4. Метод получения списка устройств

<http://oll.tv/ispAPI/getDeviceList/>

Принимает следующие необязательные параметры:

1. **account** или **email**
2. **offset**
3. **limit** (не больше 1000, по умолчанию равно 1000)

Возвращает список устройств провайдера с указанием дополнительных данных устройства и пользователя.

Структура возвращаемого JSON:

```

{
  data: [
    {
      ID:
      serial_number:
      mac:
      date_added:
      USER: {
        ID:
        email:
        birth_date:
        reg_date:
        firstname:
        gender:
        account:
        lastname:
        phone:
        region:
        receive_news:
        index:
        active:
      }
    }
  ]
}

```

```

    bought_subs: [
      {
        ID:
        sub_id:
        name:
        service_type:
        parent_id:
        start_date:
        expiration_date:
        device_binding_code:
        activation_type:
      }
      ...
    ]
  }
  binding_code:
}
...
]
status:0
}

```

Возможные коды ошибок: 109, 110, 200.

### 3.5. Метод получения списка устройств абонента с описанием устройств

<http://oll.tv/ispAPI/getUserDevices/>

Принимает следующие необязательные параметры:

1. **account** или **email** или **id** или **ds\_account**

Возвращает список устройств абонента с указанием дополнительных данных устройства, таких как тип и модель устройства.

Формат ответа:

```
[{"device_type":"stb","device_model":"MAG255","serial":"072014n008241","mac":"001a791de759","date_added":"2015-02-16 18:43:42","binding_code":""}]
```

Возможные коды ошибок: 109, 110, 200, 404.

## 4. Работа с видео

### 4.1. Метод получения токена для проигрывания видео

<http://oll.tv/ispAPI/getViewToken/>

Принимает следующие обязательные параметры:

1. **account** или **email** или **id** или **ds\_account**
2. **itemId** — ID единицы контента
3. **sessionId** — ID сессии пользователя (из ответа на <http://oll.tv/api/partnersPlayer/init>)

Возвращает одноразовый токен с которым пользователь может получить ссылку на поток видео (<http://oll.tv/api/partnersPlayer/media>).

Структура возвращаемого JSON:

```
{
  token:
  status: 0
}
```

Возможные коды ошибок: 200, 400, 403, 404, 505.

## Перечень возможных ошибок

Status	Message	Описание	Методы
109	Hash expired	Время действия хеша истекло или хеш не верен	Все, кроме auth
110	Authorization missed	Хеш не указан	Все, кроме auth
111	Auth failed	Неверный логин или пароль	auth
112	Login empty	Не указан логин	auth
113	Password empty	Не указан пароль	auth
115	Email already exists	Указанный имейл уже есть в БД	addUser
116	Email validation failed	Формат указанного имейла неверен	addUser
117	Result user account does not match provided	Не указан аккаунт или он не совпадает с аккаунтом на который подвязано устройство	delDevice
119	Device with provided mac or/and serial number already exist	Устройство с указанным мак-адресом и/или серийным номером уже присутствует в БД и за кем-то закреплено	addDevice
120	Wrong date format	Неверный формат даты	addUser
200	Required fields missed	Отсутствуют необходимые параметры	Все, кроме auth
201	Field email is required	Отсутствует необходимый параметр email	changeAccount changeEmail emailExists
203	Neither mac nor serial_number was found in your request	Отсутствуют параметры mac и serial_number	addDevice delDevice
205	Field new_email is required	Отсутствует необходимый параметр new_email	changeEmail
301	Registration failed. Contact technical support	Ошибка добавления устройства пользователю или регистрации нового пользователя	addDevice addUser
302	Wrong MAC address	Неверный формат мак-адреса	addDevice
303	Wrong Serial number	Неверный формат серийного номера	addDevice
304	Invalid binding code	Неверный код привязки устройства	addDevice
305	No devices can be binded by this code	Достигнут лимит кол-ва устройств, которые можно привязать по указанному коду привязки	addDevice
400	Bad Request	Проблема с запросом или параметрами запроса	getViewToken
403	Forbidden	Отказано в доступе	getViewToken
404	Account not found	Пользователь не найден в БД	addDevice changeAccount

			changeBundle changeEmail changeUserInfo checkBundle deleteAccount disableBundle enableBundle getUserInfo resetParentControl
405	Not eligible device_type	Недопустимое значение в параметре device_type	addDevice
406	Device not found in our DB	Устройство не найдено в БД или оно отвязано от пользователя	delDevice
407	Subscription not found	Подписка, указанная в параметрах sub_id, new_sub_id или old_sub_id, не найдена	changeBundle checkBundle disableBundle enableBundle
408	Subscription order violation	Нарушение очередности отключения или включения услуги согласно подписке	disableBundle enableBundle
501	Access denied	Устройство привязано к пользователю другого провайдера	delDevice
504	User already deactivated	Услуга была уже выключена ранее	changeBundle disableBundle
505	User is attached to another operator	Пользователь привязан к другому провайдеру	addDevice changeAccount changeBundle changeEmail changeUserInfo checkBundle deleteAccount disableBundle enableBundle getUserInfo resetParentControl
506	Account is not active	Аккаунт пользователя не активен	enableBundle