

Patientus OVS PHP client library

Table of content

- [Description](#)
- [Features](#)
- [Requirements](#)
- [Installation](#)
- [Usage](#)
- [Known Issues](#)
- [Release Notes](#)
- [Further Help](#)

Description

This client library is the backend part of the [Patientus Video Frontend Library](#), which is a wrapper for the [Patientus Telemedicine API](#). It provides OVS session creation functionality for the [Patientus Video Frontend Library](#). It is developed and maintained by the [Docplanner](#) Telemedicine team.

How it works

Each client gets their client ID and secret from the [Docplanner](#) Telemedicine team. The client application must implement this client library. By providing the client ID and the secret, the backend part of the client can get an OVS session ID based on a unique identifier (room/appointment/event/conference/...) using this client library. The [Patientus Video Frontend Library](#) will use the OVS session ID to establish an online video consultation.

Features

- Get access token for a client.
- Get an OVS session ID for a doctor.
- Get an OVS session ID for a patient.

Requirements

- PHP 5.6 or later.
- guzzlehttp/guzzle 6.2 or later.

Installation

Composer

To install the via [Composer](#), run `composer require patientus/tele-pat-ovs-client-php-sdk`

Manual Installation

Download the files and include `autoload.php`:

```
require_once('/path/to/Patientus/Client/vendor/autoload.php');
```

Usage

There are a handful of setup steps to go through. Generally, the steps are:

- [Install](#) the client library in the project.
- Configure the library.
- Get an access token and register it.
- Get an OVS session.

Configure the library

```
// Get default configuration
$config = Patientus\OVS\SDK\Configuration::getDefaultConfiguration();

// Set the API host
$config->setHost('API host URL');
```

- API host URL: The URL of the [Patientus Telemedicine API](#).

Get an access token and register it

```
// Create an authorization handler object
$authorization = new \Patientus\OVS\SDK\Handlers\AuthorizationHandler(
    $config
);

// Get an authentication token
$authToken = $authorization->getAuthToken('client_identifier',
    'client_secret');

// Register the authentication token
$config->setAccessToken($authToken);
```

- `client_identifier`: Given client ID.
- `client_secret`: Given client secret.

Get an OVS session

```
// Create an OVS session handler object
$ovsSessionHandler = new \Patientus\OVS\SDK\Handler\OvsSessionHandler(
    $config
);
```

For a doctor (moderator role):

```
$ovsSession = $ovsSessionHandler->getOvsSession(  
    'room_name',  
    \Patientus\OVS\SDK\Consts\ParticipantType::MODERATOR  
);
```

For a patient (publisher role):

```
$ovsSession = $ovsSessionHandler->getOvsSession(  
    'room_name',  
    \Patientus\OVS\SDK\Consts\ParticipantType::PUBLISHER  
);
```

- room_name : An arbitrary name for the room, event, appointment or conference.

Known Issues

There are no known issues at the time of creating this document.

Release Notes

See the changelog for recent changes.

Further Help

To get more help and for any kind of information please email ovs-sdk-support@docplanner.com.