# **OpenVidu** Video Chat Client (Budgie)

## Setting Up The Working Environment

OpenVidu Video Chat Client project’s product name is **Budgie**.

The package contains 2 services.

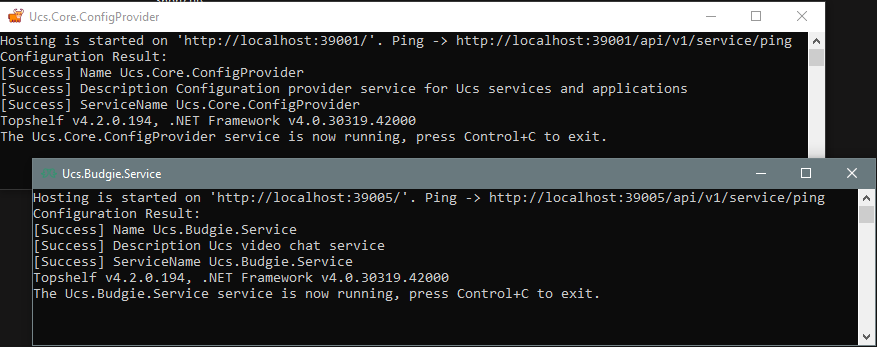
* **Ucs.Core.ConfigProvider**: The **self-hosted** (can respond to http requests) service where our applications get their configuration. Uses local port **39001**.
* **Ucs.Budgie.Service**: The self-hosted server side service that provides session&token for Budgie and communicates with turn, stun, ice servers of OpenVidu. Uses local port **39005**.

To run those services first you need to define their local **port allocation**.

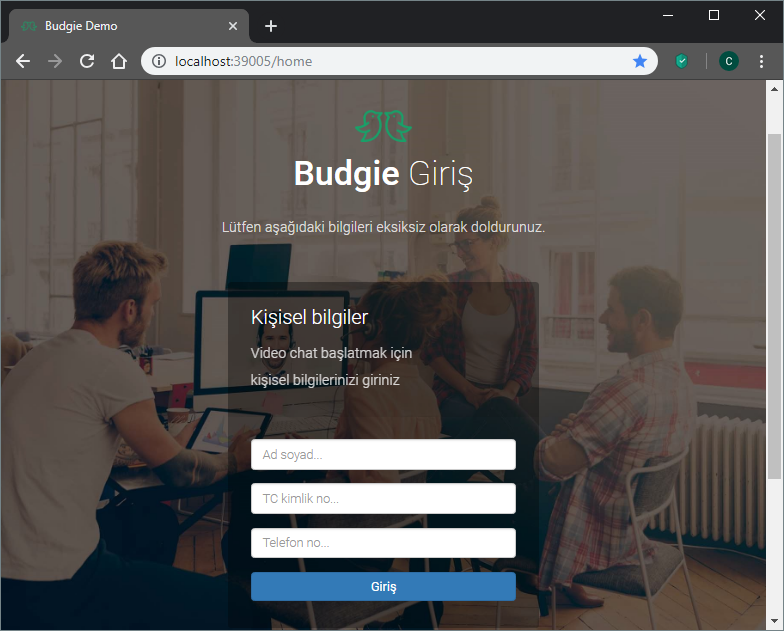
* Open **command prompt** with **administrator** privilege (run as administrator).
* Execute the commands below:
  + Netsh http add urlacl url=http://+:39001 user=everyone
  + Netsh http add urlacl url=http://+:39005 user=everyone

Execute **Ucs.Core.ConfigProvider** shortcut, after a few seconds execute **Ucs.Budgie.Service** shortcut**.**

Two console applications will start. Minimize and keep them open.

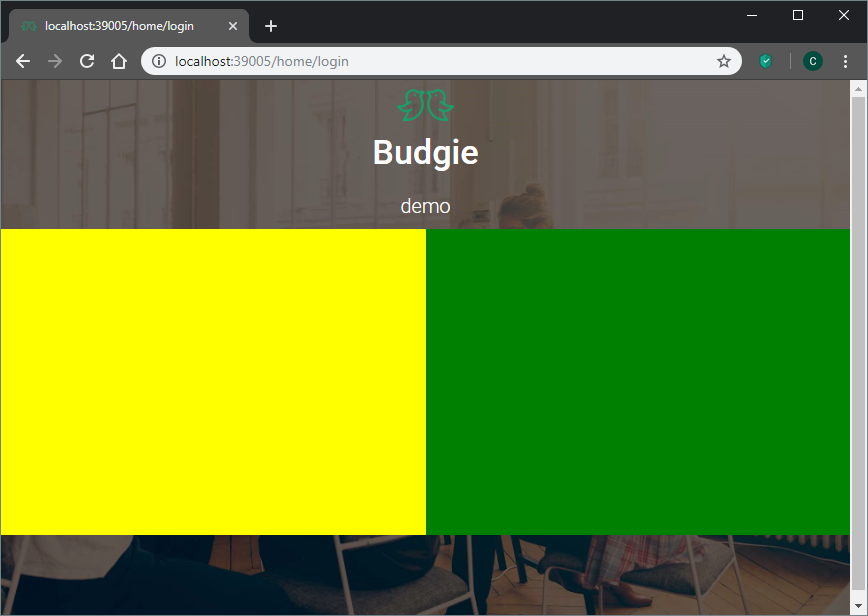


At this point, open a **Chrome** browser and navigate to [**http://localhost:39005/home**](http://localhost:39005/home). A login page will show up.



This page is in Turkish, just ignore what it says, and enter dummy values to textboxes and press the **Giriş** button.

A dummy chat page will show up. **You will develop this page**. **Please feel free to design the page anyway you want**. **If you need to change the design of the login page, you are also free to do it as you wish.**



## About The Code

All the files you will need are under **Ucs.Budgie.Service;** **\contents**, **\scripts**, **\views** folders. You may use and fill up the files under the folders, or you can add or replace them with new files. Only exception is, you cannot change the file names under the **views** folder. Because Ucs.Budgie.Service knows and serves them over http.

Now, open **views\ shared\\_layout.html** in notepad. This is our main page layout that every other page contains. You wil notice the section entries like, **@Section['Head']**. These entries are used by NancyFx view engine (like Razor in asp.net) to combine page resources. You can give any names to sections, and use any count of them if you need.

At this point, open the **views\home\login.html** in notepad and notice **@Master['../Shared/\_Layout']** section at the start. This tells NancyFx view engine to take and insert the sections below into our \_layout.html. Notice the other sections below. They have the same names with the ones in \_layout.html. Therefore NancyFx view engine will know which one to copy and paste where.

Now, open views\home\chat.html in notepad. Check the below javascript block:

<script>

var sessionId;

var sessionToken;

var userName;

$(function () {

$.backstretch("../content/img/backgrounds/9.jpg");

sessionId = "@Model.SessionId";

sessionToken = "@Model.TokenId";

userName = "@Model.UserData.UserName";

joinSession();

});

</script>

Notice **sessionId = “@Model.SessionId”** line, as example. As you see, we have some global variables (sessionId, sessionToken and userName) and we are taking their values from the server side using the NancyFx view engine.

Open **Ucs.Budgie.Service\Scripts\chatapp.js** in notepad. See the sample OpenVidu api implementation.

The product icons are under **Ucs.Budgie.Service\Content\ico** folder, and company logo is under **Ucs.Budgie.Service\Content\img** folder. If you need anything else please let me know.

## Requirements

The requirements I expect from the working application are as follows:

* Video chat between 2 people
* Mute the microphone
* Mute the video (turn it on/off)
* Open a text chat panel by pressing the icon and send-receive text messages. Text messages may contain clickable urls.
* Switch between back and front camera on the mobile phone. If it is possible, also allow the desktop user to switch mobile user’s camera.
* Sharing the desktop.
* Quit chat button.

## OpenVidu

You can access the OpenVidu browser api reference from [here](https://openvidu.io/api/openvidu-browser/). All documentation and samples are [here](https://openvidu.io/docs/home/).

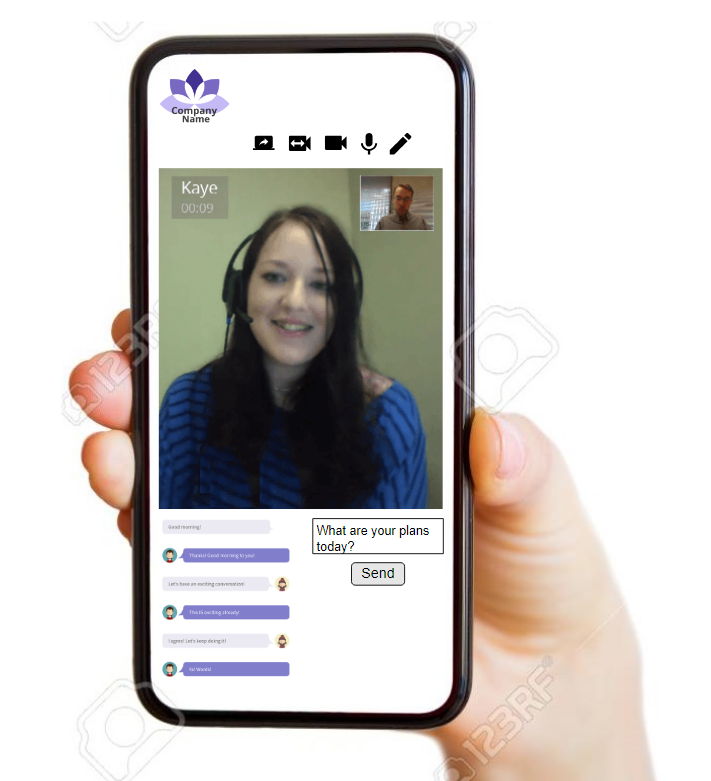
**Please remember, do not use angular, react, ionic, java and node.js in your code. Only use javascript, html, jquery, bootstrap and OpenVidu browser api. If you somehow need to use another js framework, please inform me.**

I also included 2 sample applications that you can use under **OpenVidu Samples** directory. They will configured to connect to our OpenVidu server. You can use them in another browser window to test your application. You can check **openvidu-webcomponent** application under the samples directory to see how the icons will work.

## Sample Screens

We are expecting the screens like below (again, you’re free to make a better design).

### Mobil



### Desktop

