Dear reader,

this folder contains *chaTree* - a fully functioning video chat embedment in an oTree5 project including the opportunity to store audio level data. If you use *chaTree*, please cite the accompanying paper: Promann, T., J.-P. Mayer, G. Muehlheusser, A. Roider, E. Tereschenko, and N. Wallmeier (2025): "chaTree: An oTree addon allowing face-to-face communication in online group experiments," mimeo, University of Hamburg.

To successfully use the video chat embedment, basic oTree skills as well as accounts from 'Vonage' and 'MongoDB' are necessary requirements. Vonage provides the video chat, while MongoDB provides a database for temporal storage of video chat user IDs and audio level data. This way, each member of a given experimental group will be allocated to the same video chat, while members of the next group will be allocated to a separate video chat room and so on. The audio level data can be informative with respect to the intensity and structure of discussions.

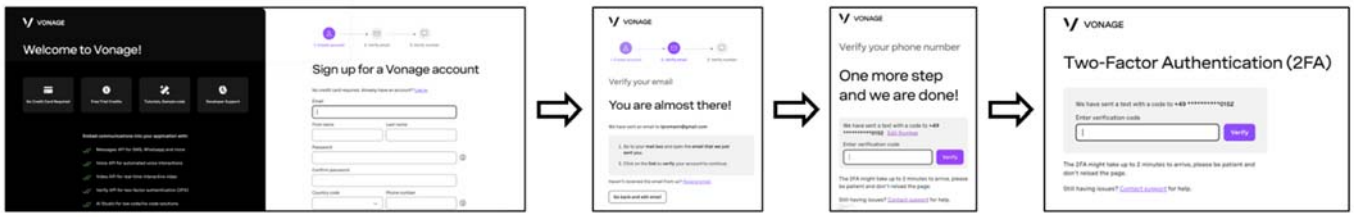
Below, each step for creating these accounts and adding the crucial credentials to the provided code sample is described in detail.

The provided oTree5 project containing the video chat is self-sufficient. We recommend to test the functionality of this project using your own Vonage and MongoDB credentials before you integrate the video chat in your own oTree5 project.

This document entails additional comments on how to set up the necessary python environment and an example of how to integrate the video chat in your own project.

Creating a Vonage account:

1. Use the link https://ui.idp.vonage.com/ui/auth/registration to sign up for a Vonage account.

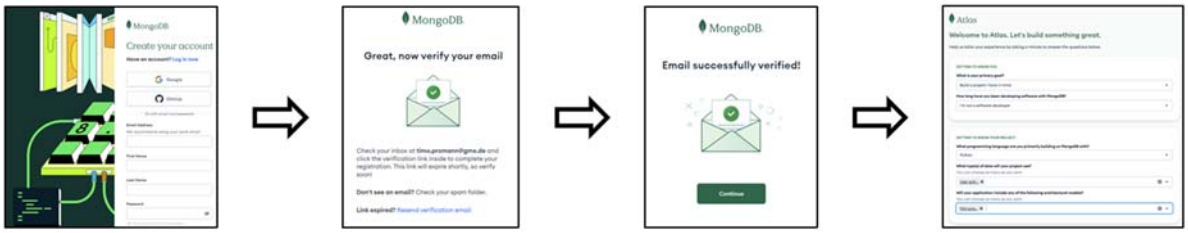


2. Create a new application in your Vonage account.
    a. In the menu on the left-hand side of your dashboard: scroll down to "Applications", click on it.
    b. Click the purple button saying "create a new application".
    c. Within your application, under "Capabilities" activate "Video".
    d. Within your application, under "Authentication" click "Generate public and private key". This downloads a file named "**private.key**". Save this file for later.
    e. Within your application, give your application a name.
    f. Click the purple button "Generate new application".
    g. Copy and save the displayed "**APPLICATION ID**" for later.

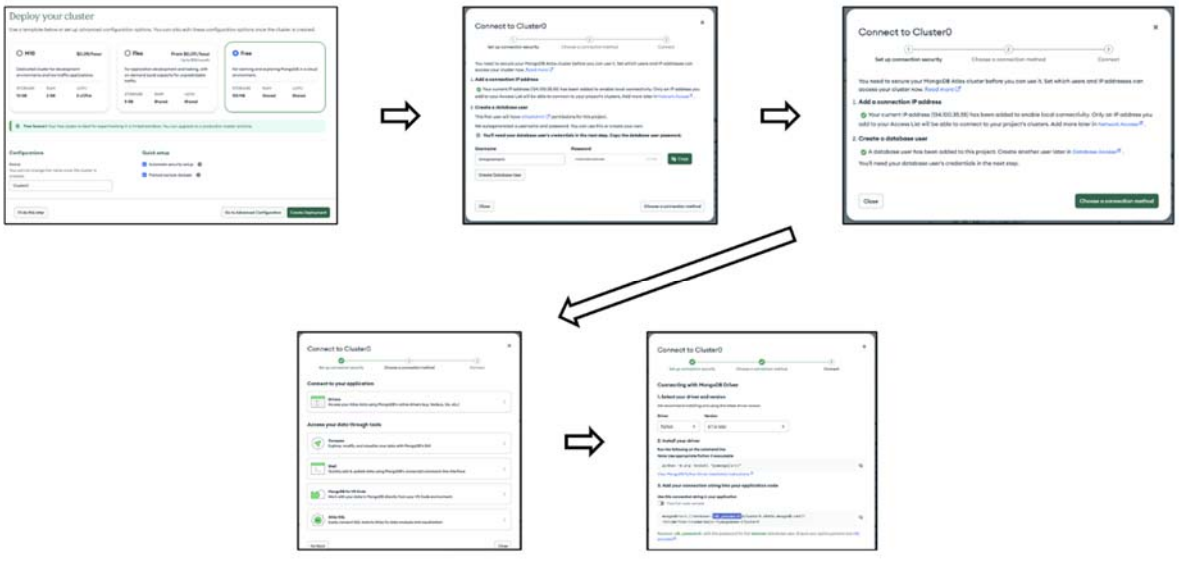
3. In a subsequent step, you will need your **APPLICATION ID** and the path to the storage location of your **private.key** file.

Creating a MongoDB account:

1. Use the link https://account.mongodb.com/account/register to sign up for a MongoDB account.
    a. On the first welcome screen, select the following choices:
        i. What is your primary goal?
            ➢ "Build a project I have in mind"
        ii. How long have you been developing software with MongoDB?
            ➢ "I'm not a software developer"
        iii. What programming language are you primarily building on MongoDB with?
            ➢ "Python"
        iv. What type(s) of data will your project use?
            ➢ "User activity/messaging data"
        v. Will your application include any of the following architectural models?
            ➢ "Not sure/None"



2. Immediately after signing up configure your account by proceeding as follows:
    a. On the page "Deploy your Cluster": of the three given options, select the free version, then click on the green button "Create Deployment".
    b. On the page "Connect to Cluster0": set a **username and a password for your database**. As you will need this password later, please write it down. Click on "Create Database User".
    c. On the next window, click on "Choose a connection method".
    d. Still on "Connect to Cluster0", click on "Drivers".
    e. Under "3. Add your connection string into your application code", copy the connection string and replace "<db_password>" with your database password (set beforehand in step 2b). The string needs a few seconds to be generated, so be patient with this step. You can also click on "Show password" to directly include your database password in the connection string. You will only need this connection string in oTree!
    f. Finally, click on "Done".

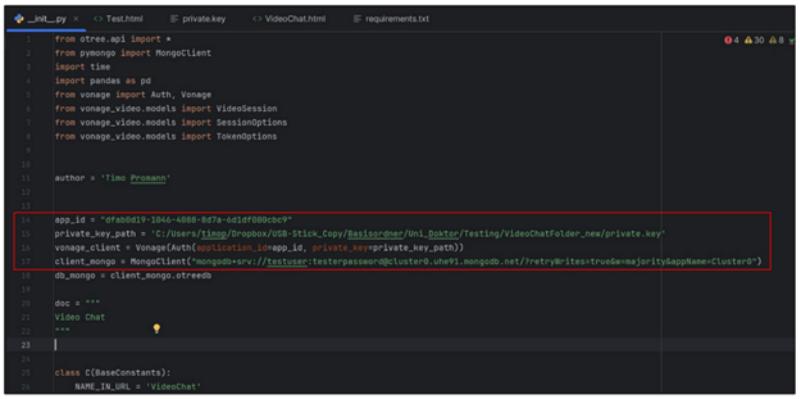


3. Further configuration of your MongoDB account:
    a. On your MongoDB dashboard, on the left-hand sidebar, below “Security” click on “Network Access”.
    b. For the only listed IP Address (your own) click on edit.
    c. Click on the button “Allow access from anywhere”, then click on “Confirm”.

4. If you forgot to note your database user/password: on the left-hand sidebar, below “Security”, click on “Database Access”. There, you can edit your user and set a new password.

Inserting details from Vonage and MongoDB into the video chat embedment code:

1. Download the whole oTree-project contained in the folder “VideoChatFolder” from this GitHub repository.
2. Open the VideoChatFolder you just downloaded with a suitable program, where we recommend PyCharm (and where a free version is available here).
3. Within the oTree-project provided with this README, in the python-file “_init_.py”:
    a. In line 14, replace the app-id with your **APPLICATION ID** from Vonage.
    b. In line 15, replace the private key path with the path leading to your **private.key** file obtained from Vonage.
    c. In line 17, replace the link with the **connection string** you got from MongoDB.



Selecting a folder to store audio level data:

- Within the oTree-project provided with this README, in the python-file “_init_.py”, almost at the bottom of the page within the function “before_next_page”:
    ➢ In line 94, replace the path to the folder in which you would like to store the audio level data.

Final important steps:

1. Setting up the required python environment:
    a. Choose a suitable version of python either from here or install it directly from your laptop's store. Vonage requires a python version of 3.9 or higher, while oTree5 requires a python version of 3.11 or lower. We recommend choosing **python version 3.10**.
    b. Create a PATH to the python folder within the environment variables of your laptop such that the correct python version can be found and applied. You can find instructions for this step here.
    c. In your PyCharm console, use the "cd" command (followed by the path to the folder that contains this oTree-project) to change the current working directory to this oTree-project.
    d. Use the command "pip install -r requirements.txt" to install every tool in the requirements file.
    e. Use the command "otree devserver" to initialize a development server on your laptop.
    f. Enter "http://localhost:8000/" in your browser to open your development server.
    g. Click on "VideoChat" to open a session of this oTree-project.
    h. Click on the "Session-wide demo link" to open one participant window. This participant automatically participates in the first video chat. You will be asked by your browser to enable camera and microphone access.
    i. Click the session-wide demo link again to open a second participant window. This participant automatically participates in the same video chat.
    j. Depending on which group size you set (see line 27 in the "_init_.py" file, behind "PLAYERS_PER_GROUP"), additional participants will either also join this first video chat window or a new video chat window will be created.
2. Integrating the video chat in your oTree5 project:
    a. Copy+paste lines 1-18 from the python-file "_init_.py" to your own "_init_.py" file.
    b. Copy+paste the requirements in your "requirements.txt" file.
    c. Install the requirements using the command from above (step 1d).
    d. Copy+paste lines 46-97 from the python-file "_init_.py" to your own "_init_.py" file as an additional page containing the video chat.
    e. Add the HTML-page "VideoChat" to the HTML-pages of your project.
    f. Add the page VideoChat to your page sequence at the bottom of your "_init_.py" file.