

# How to Run the 'Cloud User Manager' Demo Scenes

## Prerequisites

1. This package uses Azure Cognitive Services for face detection and user recognition. These services are free of charge, if you don't exceed a certain limit (30000 requests per month, 20 per minute, as of now).
2. In order to use the Cognitive Services for face detection and user recognition, you need to subscribe for Face API. To get a subscription key, go to this page: <https://azure.microsoft.com/en-us/try/cognitive-services/> and press the big blue 'Get API Key'-button next to 'Face API'.
3. You will be asked to sign-in with your Microsoft and Azure accounts. This is the right moment to sign-up for Microsoft and Azure accounts, if you don't have those already.
4. Go to <https://portal.azure.com> On the Azure dashboard, press '+New' button, search and add the 'Face API'-cognitive service.
5. On the 'Face API' control panel, press the 'Show access keys' to see your face-api subscription keys.
6. Go back to the Azure dashboard. Press '+New' button, search and add the 'Emotion API'-cognitive service.
7. On the 'Emotion API' control panel, press the 'Show access keys' to see your emotion-api subscription keys.
8. Create an empty Unity project and import this package into it.

## Face Detection Demo

1. Open Assets/CloudUserManager/DemoScenes/FaceDetectionDemo-scene.
2. Select the CloudFaceController-game object in Hierarchy. Copy the Face-API subscription key from Prerequisites-p.5 above to the 'Face Subscription Key'-setting of the CloudFaceManager-component.
3. Copy the Emotion-API subscription key from Prerequisites-p.7 above to the 'Emotion Subscription Key'-setting of the CloudFaceManager-component.
4. Check, if the locations of Face-service and Emotion-service match the Azure service locations.
5. Run the scene. The upper left part of the screen shows the output of your machine's web-camera, if there is any. Click the web camera image, to take a photo for face detection and analysis.
6. Alternatively, you can click the lower left window, and select a jpeg-image for face detection and analysis.
7. The selected picture or camera photo will be displayed on the lower left window, along with the detected faces on it. All detected faces will be surrounded by rectangles with different colors.
8. Further information about the detected faces will be displayed in the same color on the right part of the screen. The information includes gender, age and smile-status for each detected face. If you have supplied emotion subscription key, the emotion status of each detected face will be shown as well.

## User Recognition Demo

1. Open 'File / Build Settings' and add UserRecognitionScene0 & UserRecognitionScene1 to 'Scenes in Build'.
2. Open Assets/CloudUserManager/DemoScenes/UserRecognitionScene0-scene.
3. Select the CloudFaceController-game object in Hierarchy. Copy the Face-API subscription key from Prerequisites-p.5 above to the 'Face Subscription Key'-setting of the CloudFaceManager-component.
4. Emotion detection is not needed here, so you can leave the 'Emotion Subscription Key'-setting of the CloudFaceManager-component blank.
5. Check, if the location of the Face-service matches the Azure Face-API service location.

6. Look at CloudUserManager-component. The 'User Group Id'-setting is set to the group name that will be used for user recognition. This group will keep the users, their names, face info and data. By default, it is 'demo-users', but you can change it, as to your preference. This way you could have several different groups of users, for different games. The group will be created, if not found.
7. Run the scene. The upper left part of the screen shows the output of your machine's web-camera. Click the camera image, to take a photo for user recognition.
8. Alternatively, you can click on the lower left window, to select a jpeg-image for user recognition.
9. The selected picture or camera photo will be displayed on the lower left window, along with the detected faces on it. The detected faces will be surrounded by rectangles with different colors.
10. The panel on the right part of the screen will display the recognized users, as well as not-recognized faces. The recognized users will be displayed along with their user names. To log-in (i.e. go to the 2<sup>nd</sup> scene) select one of the recognized users.
11. You can also add any of the not-recognized faces as new user. To do it, write its user-name and press the respective 'Create User'-button.

## Group Management App

1. Open Assets/CloudUserManager/GroupManager/CloudGroupManager-scene. This scene is not a demo, but rather back-office application for group & user management. Do mind, you can't see the user faces, because the face images cannot be downloaded from the cloud. You will only see the user names, IDs and face IDs.
2. Before you start, just like in the demo scenes, select the CloudFaceController-game object in Hierarchy. Copy the Face-API subscription key from Prerequisites-p.5 above to the 'Face Subscription Key'-setting of the CloudFaceManager-component.
3. Check, if the location of the Face-service matches the Azure Face-API service location.
4. Make sure that 'User Group Id'-setting of CloudUserManager-component is set to the same group name you have used for user recognition, i.e. in the User-recognition demo above.
5. Run the scene. You will see the current list of users in the specified user group. You can refresh the list at any time by pressing the Refresh-button, in the lower right corner of the window.
6. Select any user in the list to see more information about it. You will see its user name, person ID and face ID.
7. You can change the user name, add or modify custom data for the user, or delete the user and all its data.

## How to Include Face Detection in Your Unity Project

1. Copy the Assets/CloudUserManager/CloudFaceScripts-folder from this package to your project.
2. Create an empty game object in your scene. Name it 'CloudFaceController'.
3. Add CloudFaceManager-script as component to the newly created CloudFaceController-object.
4. Copy the Face-API subscription key from Prerequisites-p.5 above to the 'Face Subscription Key'-setting of the CloudFaceManager-component. Do the same for the Emotion-API subscription key, if applicable.
5. Check, if the locations of Face-service and Emotion-service match the Azure service locations.
6. You can copy the CloudFaceDetector.cs-script from DemoScenes/Scripts-folder to a folder in your project and then use it as component in the scene. Don't forget to set its image source.
7. You can also use the public API-functions of the CloudFaceManager-component in your scripts, just like the CloudFaceDetector.cs-script in the demo does. Use its source as an example.

## How to Include User Recognition in Your Unity Project

1. Copy the Assets/CloudUserManager/CloudFaceScripts-folder from this package to your project.
2. Create an empty game object in your scene. Name it 'CloudFaceController'.
3. Add CloudFaceManager-script as component to the newly created CloudFaceController-object.
4. Copy the Face-API subscription key from Prerequisites-p.5 above to the 'Face Subscription Key'-setting of the CloudFaceManager-component.
5. Check, if the location of the Face-service matches the Azure Face-API service location.
6. Add CloudUserManager as component to the CloudFaceController-game object. Set its 'User Group Id'-setting to the name of the user group you'd like to utilize.
7. You can copy CloudUserRecognizer.cs-script from DemoScenes/Scripts-folder to a folder in your project and then use it as component in the scene.
8. Alternatively, you can use the public API-functions of the CloudUserManager-component in your scripts, just like the CloudUserRecognizer.cs-script in the demo does. Use its source as an example.

## More Information, Support and Feedback

Web: <https://rfilkov.com/2016/05/02/cloud-user-manager/>

Facebook: <https://www.facebook.com/issbgcom>

Twitter: <https://twitter.com/roumenf>