Table of Contents

Altizure Development Platform

Welcome to Altizure Development Platform！

On Altizure, you can not only experience the most advanced 3D reconstruction technology to turn 2D photos to realistic 3D models, but also use the online 3D publishing service to share and enjoy the 3D models.

Join Altizure open platform to integrate the powerful 3D reconstruction and publishing services on Altizure to your business workflow. Unleash the power of realistic 3D models!

Let's start the journey.

Learn more about Altizure at:

- Explore the 3D world on Altizure: altizure.com/explore
- Facebook page
- Official blog: blog.altizure.com
- Offline documentations: pdf, epub

Contact us support@altizure.com

Last modified at Sun Nov 19 2017 14:30:14 GMT+0800 (HKT)

Concepts

Before starting to develop with Altizure, here we introduce a few concepts. If you have used some development SDK for other online platform, e.g. facebook app or github sdk, you should be familiar with them.

1. Developer account

If you do not have an altizure account yet, please get one on sign up page.

Then you can apply for developer account. All development tools and privileges will be linked with this account. Please keep this account secure and safe.

2. App

To be added

App key

To be added

App secret

To be added

3. User token

To be added

4. GraphQL API

Altizure GraphQL API is a set of API in the syntax of GraphQL. The API allows developers to fetch and modify the data on Altizure. Learn more at GraphGL API

5. Javascript SDK

Altizure Javascript SDK allows you to integrate rich 3D experience with our realistic 3D models to your business workflow. Learn more at Javascript SDK

Last modified at Mon Oct 30 2017 19:30:15 GMT+0800 (HKT)

Developer Account

We are sorry that the developer account is not yet for public application. It is only for invited partners at the moment.

For latest information about our open platform, please keep following us at:

- Facebook page
- Official blog: blog.altizure.com

Last modified at Mon Oct 30 2017 19:30:15 GMT+0800 (HKT)

Altizure Graphql API

Altizure GraphQL API is a set of API in the syntax of GraphQL. The API allows developers to fetch and modify the data on Altizure.

1. Prerequisite

- Altizure developer account
- App key
- User token (optional)

2. API Endpoint and Documentations

API endpoints and documentations: api.altizure.com/graphql.

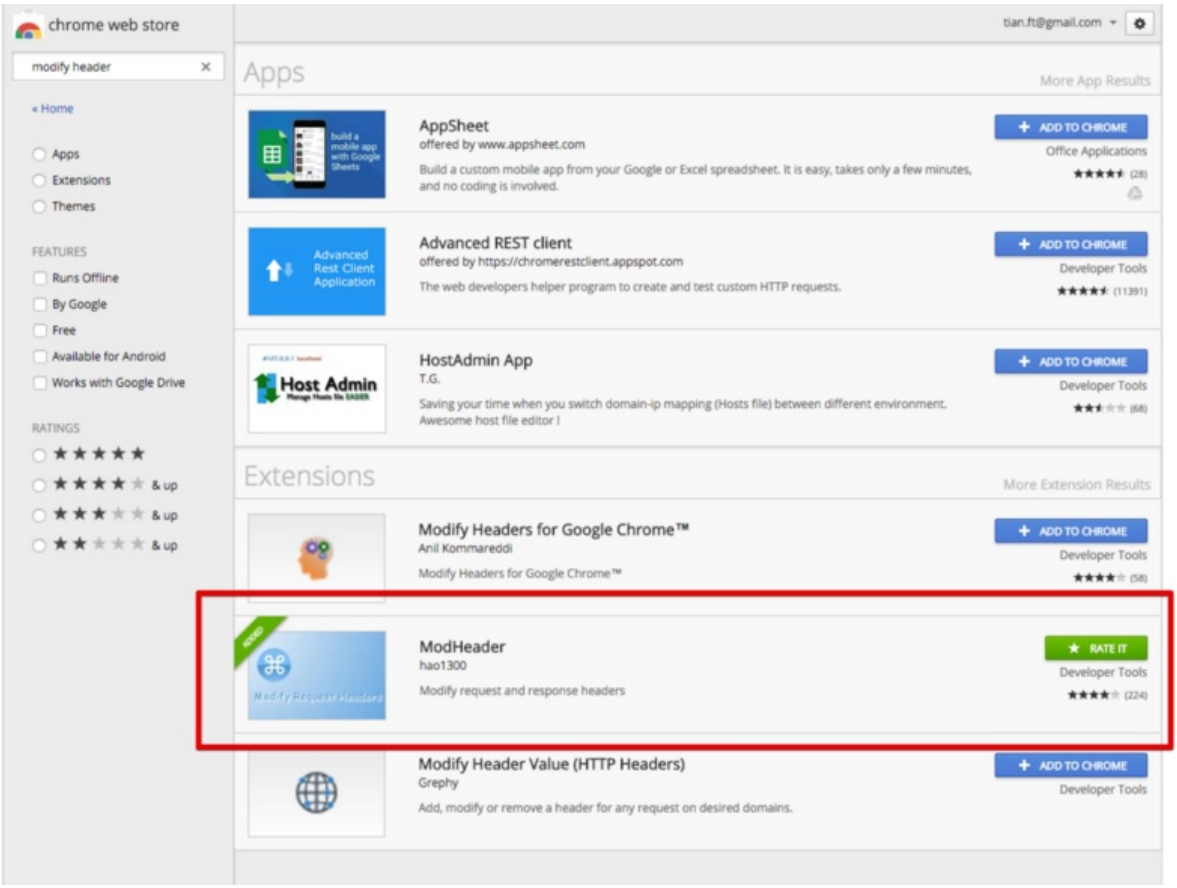
3. Try out API in your browser

It is very convenience to test the API in browsers, because it provides instant feedback on the query results and detailed inline documentations. After the testing, you can easily copy and paste the query string to your code and trigger the API call.

We take Google Chrome as an example. Other browsers supporting extensions, e.g. Firefox, should work too.

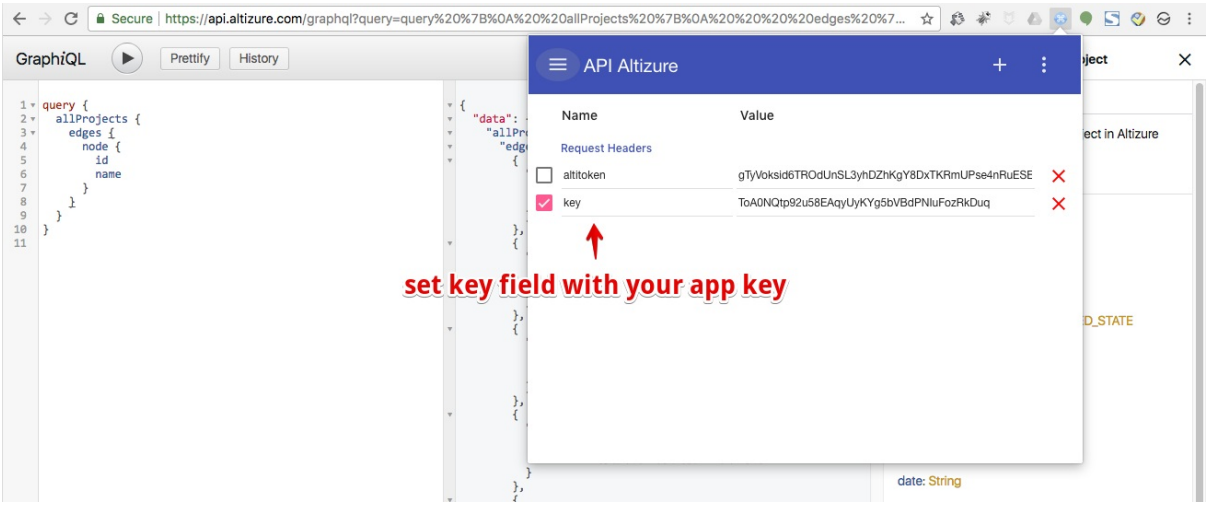
Install extension

First, please install an extension that can modify the http request header. Here ModHelper is used. Please search and install `ModHelper` in the extension store of Google Chrome.



Modify http header

Please use ModHelper to add `key` field in request header with app key as the value.



Visit api.altizure.com/graphql after setting the key. You can find three sections: "query section", "result section", and "documentation", on the page.

Now please fill the following query string to the query section to get the ID and name of public projects.

```
query {
  allProjects {
    edges {
      node {
        id
        name
```

```
        }
      }
    }
  }
```

Please click the action button to get the query results.

4. Integrate the API in your code

The API can be called by any libs and programming languages that can issue a http post request.

For example:

JQurey in Javascript

```
$.ajax({
    type: 'POST',
    url: 'https://api.altizure.com/graphql',
    headers: {
      altitoken: 'user token',
      key: 'app key'
    },
    data: 'query=' + 'GraphGL query string'
  })
```

5. Obtain user token

User token is obtained via the standard OAuth 2 flow.

The authorization endpoint is the following url:

```
`https://api.altizure.com/start?client_id=${appKey}&response_type=token&redirect_uri=${redirect_uri}`
```

where **appKey** is your application key, and **redirect_uri** is one of the domains associated with your application.

Your application needs to route/open this url. A form will be shown to your users asking for their authorizations. After your users have authorized the request, the page will be redirected back to your **redirect_uri** with a url hash variable of key: **access_token**.

For mobile application, the **redirect_uri** will be your application's bundle identifier name (iOS) or your package name (android).

For a vanilla JS implementation, please refer to here.

6. FAQ

6.1 How to access the api in Mainland China?

Please use `https://api.altizure.cn/graphql`. It is better to choose a reachable and faster endpoint whenever possible in the logic of your application instead of hardcoding the api endpoint.

6.2 Where to find more detailed documentations on GraphQL API

Please follow the above tutorial and browse api.altizure.cn/graphql with your browser. All documentations are embedded in the web frontend of our api endpoint.

7. Learn more

- Learn more about GraphQL
- Use Altizure Javascript SDK to developer rich 3D application
- More tools on GraphQL: Awesome GraphGL

Last modified at Mon Oct 30 2017 19:30:15 GMT+0800 (HKT)

Using the upload API

Overview

To speed up data transfer, images are uploaded directly from clients to Amazon or Aliyun. Your image are protected securely. Others could not read or modify your images. That is why you need to obtain the authorizations from our api. As filename is the only identifier in the buckets, the image is required to be uploaded to a specific url (S3) or prefix (OSS) so that Altizure knows which image is which.

1. Choose bucket

Choose one of the fastest S3 or OSS buckets. This is largely correlated to your network zone with the edge-points. You may simply query `GeoIPInfo.nearestBuckets` for hints for auto selecting the best edge for your clients.

2. Compute image checksum

Compute the sha1sum hash for each image. Before each upload, check if the current image has already been uploaded by calling the mutation `hasImage(pid, checksum)`. If it is not, proceed to the next step. Otherwise, skip to the next image.

3. Upload to OSS

If an OSS bucket is chosen, obtain STS and the related meta image info (e.g. id and hashed filename) by calling mutation `uploadImageOSS(pid, bucket, filename, type, checksum)`. STS is a temporary (1 hour) security token for the write only permission on the `/pid` prefix. If it has expired, renew with the same mutation. Otherwise, only request the `image` gql fragment from the result, and re-use the same STS for performance reason. As signing from Aliyun is slow, one should not sign a new STS for each image.

Given the STS, images could be uploaded via any compatible protocol or library to OSS. It is required to upload with the specific hashed filename according to the image info returned from the `uploadImageOSS` mutation. Specifically, it is required to be put to the path `${pid}/${image.filename}` inside the bucket.

In order to keep track of the state, just before an image is uploaded, call mutation `startImageUpload(id)` to signal the start of the process. When the upload is done, call mutation `doneImageUpload(id)`.

3. Upload to S3

If a S3 bucket is chosen, uploading is much simpler. For each image, call the mutation `uploadImageS3(pid, bucket, filename, type, checksum)` to obtain a temporary (3 hours) signed url and the related meta image info.

Given the signed url, use the standard HTTP put to put the file to this url with `Content-type: JPEG` or other formats accordingly.

Just before each upload, it is required to call mutation `startImageUpload(id)`. There is no need to signal the end of uploading.

4. Wait for image processing

Uploaded images will be copied, verified and processed. Eventually, the image state will become either `Ready` or `Invalid`. If the total count of `Ready` matches the desired number, you may call mutation `startReconstructionWithError(id, options)` to start the reconstruction. Otherwise, you may consider re-uploading any outstanding image as indicated by the mutation `hasImage`.

If you do not concern about a few number of missing images and want to start the reconstruction immediately once all the images are ready, you could call mutation `preStartReconstruction(id, options)` .

Learn more

- Learn more about STS

Last modified at Sun Nov 19 2017 00:30:14 GMT+0800 (HKT)

Altizure Javascript SDK

Altizure Javascript SDK allows you to integrate rich 3D experience with our realistic 3D models to your business workflow. Combined with tools like Electron and React Native, you can easily develop high quality 3D applications with realistic models for desktop and mobile apps.

1. Basics

Coming soon

2. Sample

Coming soon

3. Reference manual

Coming soon

4. Learn more

- ThreeJS
- WebGL
- OpenGL
- Vulkan
- OpenGL Transformation

Last modified at Wed Nov 15 2017 15:30:16 GMT+0800 (HKT)