目录

简介 1.1

基本概念 1.2

开发者账号申请 1.3

GraphGL API 1.4

- 上传图像 1.4.1

Javascript SDK 1.5

- 演示应用 1.5.1
- 常见问题 1.5.2
- 详细文档 1.5.3

Altizure 开发平台

欢迎使用 Altizure！

在 Altizure 您除了可以使用全球领先的二维图像到三维图像的自动建模服务，还可以通过 Altizure 在线浏览和应用技术，轻松分享和使用您的三维模型，为您带来无限乐趣和价值。

加入 Altizure 开发平台，可以把 Altizure 上提供的建模和浏览服务和您的业务和应用做更深入的集成，释放三维数据带来的无限可能。

让我们开始实景三维世界的探险吧。

- 基本概念
- 开发者账号申请
- GraphGL API
  - 上传图像
- Javascript SDK
  - 演示应用
  - 常见问题
  - 详细文档

更多了解 Altizure 欢迎访问：

- 探索 Altizure 的三维世界：altizure.cn/explore
- 关注微信公众号：altizure_info
- 关注官方博客：blog.altizure.cn
- 离线文档：pdf, epub

联系我们 support@altizure.com

该文档最后修改于 Thu Dec 21 2017 20:04:53 GMT+0800 (HKT)

基本概念

在开始使用 Altizure 开发平台之前，您需要大概了解以下概念。您如果使用过一些其他开发平台，例如微信开放平台、支付宝开放平台，这些概念都是稀松平常的，您可以轻易上手。

1. 开发者账号

如果您还没有一个 altizure 账号，请马上在注册页面免费申请一个。

然后您需要按照开发者账号申请指示，并把它升级为开发者账号。所有开发的权限都会和这个账号关联。请确保这个账号的安全和保密。

2. 应用 (App)

应用是一个您在 Altizure 开发平台上创建的最基本应用管理单元。每个应用有相应的权限级别，安全设置，唯一的 ID，令牌和鉴权密码。

一个开发者账号，可以申请多个应用，用来区分不同的业务和流程。

应用令牌 **(App key)**

每个应用唯一的令牌，用于识别开发平台的服务调用。用户在授权相关应用访问其 Altizure 平台上的信息后，可以通过该令牌识别相关应用，终止相关授权。

应用密钥 **(App secret)**

用于服务器端调用时鉴别这个 API 调用是来自相应的开发者账号。切勿在任何前端代码里泄露这个密钥。只能在服务器端程序内使用。

3. 用户令牌 (User token)

这个令牌是应用在需要获得用户授权后得到的鉴权令牌，用于访问用户授权的相关 Altizure 上的信息。开发者可以存储这个令牌，而不应该以任何形式诱导用户提供密码或者存储用户的密码。

4. GraphQL API

GraphQL API 是一组以 GraphQL 为协议提供的在线 API。这套 API，提供了获取和修改 Altizure 上基础数据的能力。详情可以浏览：GraphGL API

5. Javascript SDK

这是一个以 Javacript 为开发语言的 SDK。提供了丰富的三维浏览和编辑的功能。详情可以浏览：Javascript SDK

该文档最后修改于 Mon Oct 30 2017 17:19:01 GMT+0800 (HKT)

开发者账号申请

非常抱歉现在我们还不接受公开申请。只有受邀请的合作伙伴才可以申请开发者账号。

关于最新的开发者信息，请关注我们的社交媒体和官方博客：

- 微信公众号：altizure_info
- 中文官方博客：blog.altizure.cn

该文档最后修改于 Mon Oct 30 2017 17:19:01 GMT+0800 (HKT)

Altizure Graphql API

GraphQL API 是一组以 GraphQL 为协议提供的在线 API。这套 API 提供了获取和修改 Altizure 上基础数据的能力。

您如果在找如何快速加载和渲染 Altizure 上的三维数据，请查看我们的 Javascript SDK 页面。

1. 使用前准备

- Altizure 开发者账号 (必须)
- 应用令牌 (App key) (必须)
- 用户令牌 (User token) (部分调用需要)

2. API 网址和文档

API 调用和文档的入口：

- 国际站 api.altizure.com/graphql 。
- 中国站 api.altizure.cn/graphql 。

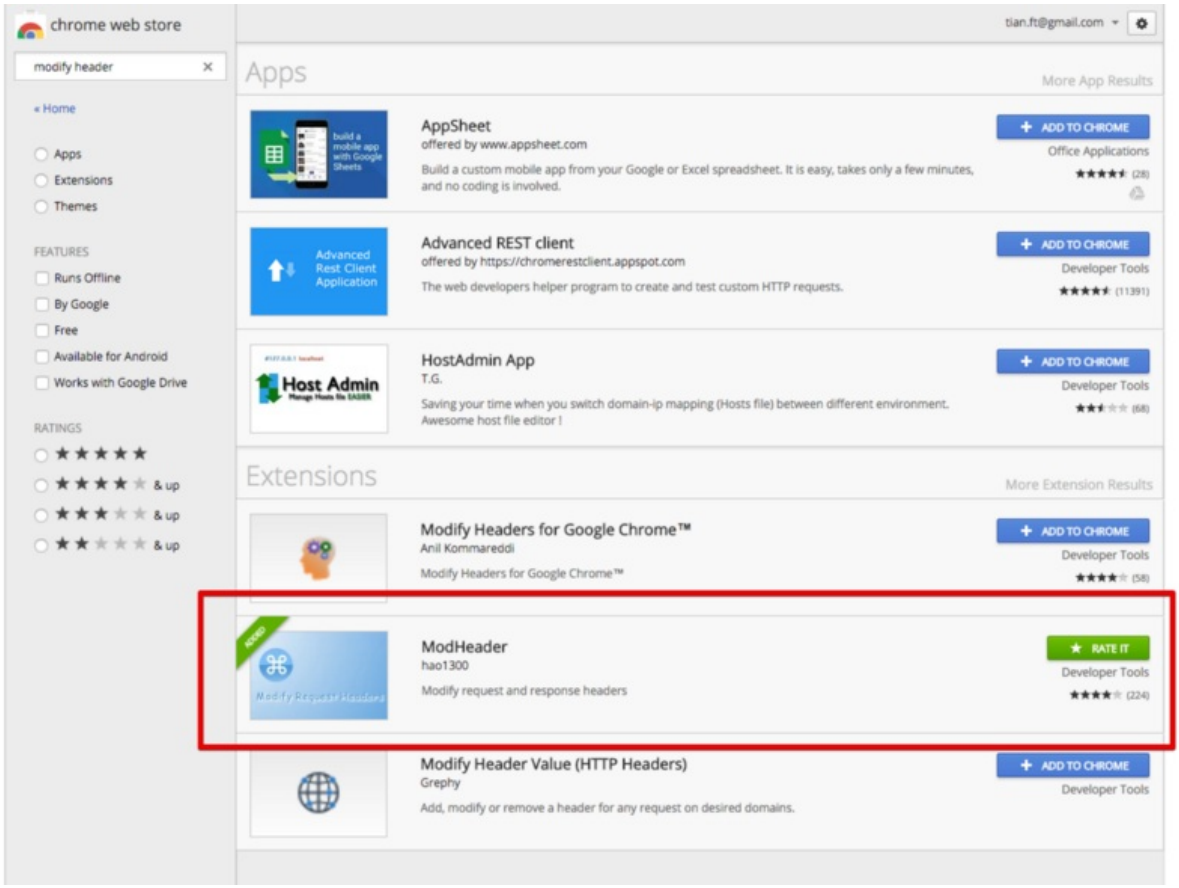
3. 在浏览器中测试 API

在浏览器中测试 API，可以在浏览文档的同时测试，提高开发效率。确定数据正确后，才把相关的调用复制到代码中集成，过程轻松简单。

这里的教程以谷歌 Chrome 浏览器为例子，其他支持插件的浏览器也可以用类似的方法做测试。

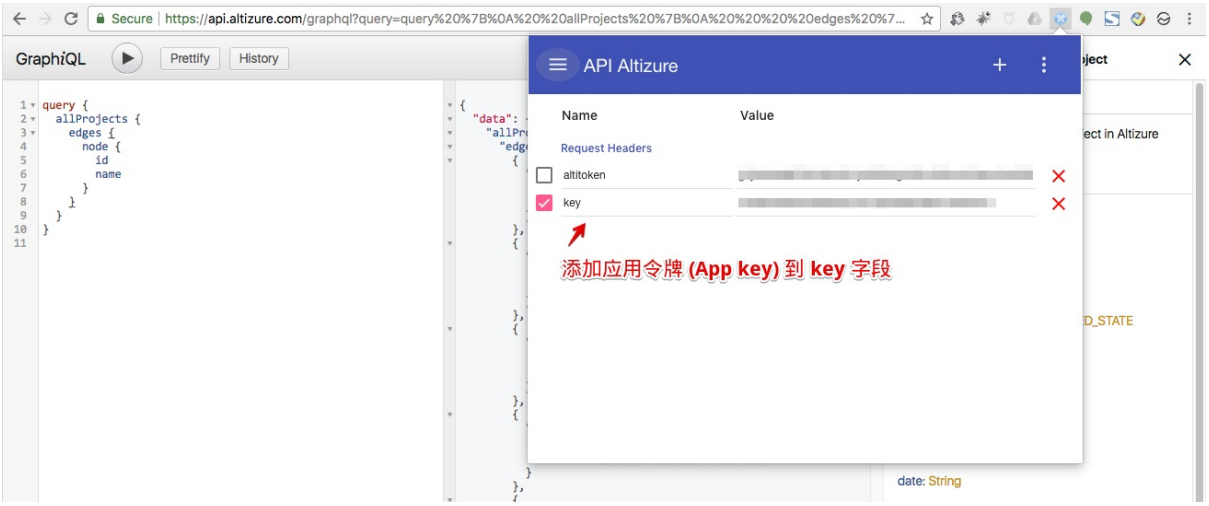
安装插件

安装一个可以修改 http 访问请求 header 的插件。这里我们以 ModHeader 为例。在谷歌 Chrome 扩展商店里面搜索 `ModHeader` ，并点击安装即可。



设置插件修改 **http header**

在 http request header，添加 `key` 字段，填入应用令牌 (App key)。



设置完成后访问 api.altizure.cn/graphql，可以看到以下界面，集成了“查询区”，“查询结果”和“文档”三个模块。

试着在查询区输入一下字段查询一个组图像的 GP 数。

```
query {
  support {
    sizeToGigaPixel(width: 4000, height: 3000, numImg: 100)
  }
}
```

并按查询按钮，便可获得查询结果。这个样例 API 调用会计算 100 张 4000x3000 的图像的总 GP 数。

4. 在代码中集成 API 调用

您可以使用任何可以发起 http post request 的程序库来调用 GraphQL API。

例如：

JQurey in Javascript

```
$.ajax({
    type: 'POST',
    url: 'https://api.altizure.cn/graphql',
    headers: {
      altitoken: '用户令牌',
      key: '应用令牌'
    },
    data: 'query=' + 'GraphGL的查询字符串'
  })
```

5. 获取用户令牌

您需要通过 OAuth 2.0 的标准流程来获取用户令牌。

其中调取 OAuth 窗口的链接为:

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

其中 **appKey** 是您的应用令牌，**redirect_uri** 是您注册该令牌时绑定在上面的域名。

您的应用/网页需要在客户端调出浏览器或者 WebView 来访问这个链接。接下来一个授权窗口会呈献给用户用于输入用户名和密码以便授权您的应用访问该用户的信息。您的应用的说明信息会被现实在这个窗口，便于用户识别开发者的身份。

用户授权之后，登录窗口会重新想到 **redirect_uri** 。重新向的链接会包含一个用户令牌 **access_token**。您既可以在客户端页面处理这个令牌，也可以实现一个服务器端的路由在服务器端接收这个令牌。

对于移动应用，**redirect_uri** 应该填入您的应用的识别符 application bundle identifier name (iOS) 或者包名 package name (android)。这些信息都必须在申请应用令牌时提供给 Altizure 以便记录在册。

您可以访问这个网址 here 查看一个极简的登录实现。

6. 常见问题

6.1 国际站和中国站如何选择？

请选择与您链接最快的一个。可以在程序中做些自动判断，如果一个站点掉线了可以切换成另外一个。

6.2 GraphQL API 详细文档在哪儿？

请直接用浏览器参照以上教程访问 api.altizure.cn/graphql，即可直接看到所有查询接口的文档。

7. 了解更多

- 深入学习 GraphQL
- 通过 Altizure Javascript SDK 实现丰富的三维浏览编辑功能
- 更多 GraphQL 相关工具 Awesome GraphGL

该文档最后修改于 Fri Dec 08 2017 17:30:17 GMT+0800 (HKT)

上传图像

概况

为了获得最佳的上传速度，用于三维重建的图像将直接从客户电脑传输到我们的亚马逊或者阿里云的存储空间里。 传输和存储中，你的图像将被严格保密，其他任何人都不可以读取和修改你的图像。 所以上传开始前需要开发者通过我们的 API 获取一个临时的加密 url (亚马逊 S3) 或者 STS 鉴权令牌 (阿里云 OSS)。因为在 altizure 上每个项目中不允许有同名文件，所以我们的 API 会根据你所上传文件的校验码分配一个唯一的文件名，而不是直接使用原文件的名字作为上传的名字。上传完成后便可以调用启动项目的 API，开始三维重建。

以下我们详细介绍上传的每个步骤。

1. 选择最近的 **bucket**

开发者可以通过 `GeoIPInfo.nearestBuckets` 这个查询来获取当前用户最近的 Bucket 用于上传，以达到最快的上传速度。

2. 计算图像校验码

为了保证图像上传的稳定性和数据的一致性。在上传每张图像前开发者必须

- 计算的图像的 **sha1sum** 校验码
- 调用 mutation `hasImage(pid, sha1sum)` 检查图像是否已经项目 `pid` 里存在。
- 如果图像存在则跳过该图像无需上传；如果图像不存在则上传图像。

3. 上传

3.1 上传到阿里云的 **OSS**

在选择了最近的 OSS bucket 后，我们需要按照以下流程进行上传：

- 通过 `uploadImageOSS(pid, bucket, filename, type, sha1sum)` 获取 **STS** 令牌及其他上传所需的图像 meta info，包括上传 图像的 **id** 和 哈希化的文件名。STS 令牌的有效期只有 1 小时，对相关 bucket 的 `/pid` 只有写权限。如果 STS 令牌过期了，开发者必须利用同样的 mutation `uploadImageOSS(pid, bucket, filename, type, sha1sum)` 进行更新。如果令牌还未过期，开发者在上传新的图像时无需请求新的令牌，只需用同样的 STS 请求 `uploadImageOSS` 中的 `image` 项。虽然开发者可以每上传一张图像都请求新的 STS 令牌，但是由于阿里云的签发令牌的相应很慢每次都请求新的令牌会导致上传不太稳定，所以我们推荐在 STS 令牌未过期前，不要重复申请新的令牌。如果您使用阿里云 OSS Javascript SDK 中的 `Wrapper` ，请注意在 STS 过期前不要重复创建该类型对象，只有在 STS 过期后才需要重新创建。创建过多 `Wrapper` 对象会导致 `The sockets is full` 的错误。
- 获取了 STS 令牌后和相应的 meta image info 后，调用 mutation `startImageUpload(id)` 通知服务器一张图像即将开始上传。
- 使用相应的 STS 令牌把图像上传到 OSS 的 bucket 里 `${pid}/${image.filename}` 的路径下。其中 `${image.filename}` 是上传前调用 `uploadImageOSS` 所返回的图像 meta info。
- 上传完成后调用 mutation `doneImageUpload(id)` 通知服务器，一张图像的上传完成了。

3.2 上传到亚马逊的 **S3**

我们未能在中国大陆区提供亚马逊 **S3** 的节点，请中国大陆的开发者使用阿里云的 **OSS** 节点。

如果最近的上传节点是在 S3，开发者可以使用 S3 上传图像。

- 在每张图像上传前，调用 mutation `uploadImageS3(pid, bucket, filename, type, checksum)` 获取一个有效期为 3 小

时的加密 url 和相关的图像 meta info。
- 调用 `startImageUpload(id)` 通知服务器一张图像的上传即将开始。
- 通过标准的 HTTP PUT 命令把图像的文件上传到加密的 url 上，文件名为 `${image.filename}`。上传时需要把传输的内容标记为 `Content-type: JPEG`。相片上传后无需调用 mutation `doneImageUpload(id)`

4. 等待图像预处理

图像上传到 OSS 或者 S3 后，Altizure 的服务器会对图像进行验证和预处理，确保图像都是正确有效的。最终每个图像都会被标记为 `Ready` 或者 `Invalid`。如果 `Ready` 的图像数目和你想上传的图像数目一样，就可以继续调用 mutation `startReconstructionWithError(id, options)` 来开始三维重建。如果有效图像的数目不对，开发者可以通知用户重新上传图像确保图像正确，重新上传中可以利用 mutation `hasImage` 来判断图像是否已经存在服务器上来决定是否重新上传，只上传服务器上不完整或者不存在的图像。

开发者也可以不等待 Altizure 服务器对图像进行验证，在图像上传完后，便调用 mutation `preStartReconstruction(id, options)`。如果这样，服务器在验证完所有图像后，就会用已有的有效图像开始三维重建。

了解更多

- 了解更多关于 STS 的信息
- 阿里云 OSS Javascript SDK: 链接

Last modified at Mon Dec 11 2017 18:30:15 GMT+0800 (HKT)

Altizure Javascript SDK

这是一个以 Javacript 为开发语言的 SDK。提供了丰富的三维浏览和编辑的功能。SDK 的主要目的是：

- 简化加载和渲染海量三维数据的开发工作
- 简化各种数据源的数据在实景三维底图上的整合和高效显示
- 简化实景三维数据和行业应用的开发工作

使用 Altizure Javascript 3D SDK 并结合 Electron 和 React Native 等混合开发的工具，开发者可以轻松开发出高质量的实景三维桌面和移动应用，助力你的商业应用。

1. 快速开始指南

引用 **SDK**

在网页的 head 部分，引用我们的 SDK。

```
<!-- 设置编码, 确保 utf8 字符的正确显示 -->
<meta charset="utf-8">
<!-- 设置 viewport, 确保移动端的正确渲染 -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
<!-- 引用sdk -->
<script type="text/javascript" src="https://beta.altizure.com/sdk"></script>
```

其中我们提供三个版本的 sdk 引用链接：

- 最新版：`<script type="text/javascript" src="https://beta.altizure.com/sdk"></script>`
- 稳定版：`<script type="text/javascript" src="https://www.altizure.com/sdk"></script>`
- 中国版：`<script type="text/javascript" src="https://www.altizure.cn/sdk"></script>`

创建三维显示容器

我们的 sdk 会完全接管三维数据的下载和渲染，用户需要创建一个 `div` 指定相关用于渲染的容器的位置和大小。

```
<body>
  <div id="page-content"></div>
</body>
```

创建三维引擎对象

我们的三维引擎是以最新的 Altizure 地球的引擎作为基础，新建对象时需要把它附着在一个作为显示容器的 `div` 里。

```
// 创建一个参数配置对象
let options = {
  altizureApi:{
    // 填入您的 app key
    key: 'your-app-key'
  }
}

// 创建地球渲染引擎对象, 附着在 page-content 这个 div 上
let sandbox = new altizure.Sandbox('page-content', options)
```

其中 `'page-content'` 是上面创建三维显示容器的 `div` 的 id。 `options` 用于配置新建的引擎对象，更多参数可以参考下面的范例和详细文档。

小结

所有代码组合起来就是：

```
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
  <script type="text/javascript" src="https://beta.altizure.com/sdk"></script>
</head>
<body>
  <div id="page-content"></div>
  <script>
    let options = {
      altizureApi:{
        // 填入您的 app key
        key: 'your-app-key'
      }
    }

    let sandbox = new altizure.Sandbox('page-content', options)
  </script>
</body>
</html>
```

把这段代码保存成一个 html 文件放在一个文件夹如 `<path>/altizure-sdk-test/earth.html` 中，然后在控制台中键入：

```
cd <path>/altizure-sdk-test/
python -m SimpleHTTPServer
```

再通过浏览器访问 `http://127.0.0.1:8000/earth.html` 就可以加载这个 Altizure 地球了。

你也可以访问演示页面观看这段代码的效果。

只需要简单几行代码，我们便可以创建出一个可以加载全球实景三维模型的视图。惊不惊喜？激不激动？

接下来我们直接通过范例代码来学习 sdk 的丰富功能。

2. 范例

您可以直接访问 altizure.github.io/sdk.examples/examples.sdk.html 来直接尝试各种范例的效果。

2.1 概念释义

以下我们简单解释一下出现在范例里的元素的概念

- Sandbox (沙盒)： altizure.Sandbox (沙盒) 是整个三维应用的核心，它负责管理整个三维场景的数据和绘制。通过对沙盒进行定制和添加数据，可以定制出非常强大的三维应用。这是编写三维应用的主要入口。
- Earth (地球)：altizure.Earth (地球) 是负责渲染管理 Altizure 地球的核心，通常它是作为沙盒的底图。我们提供丰富的选项来方便您定义这个地球的外观。一般来说，如果创建了 Sandbox 就无需再创建一个 Earth 对象。

您可以参考 github.com/altizure/sdk.examples 的教程把这范例代码下载下来，在本地建立服务器进行尝试。您只需要对其中的部分函数做些简单修改便可以和您现有的系统进行整合。

对范例和使用方法有任何疑问可以在 issue page 进行提问和交流。

2.2 范例列表

- 2.2.1 Altizure 地球基本加载范例

    - 默认地球加载
    - 设置地球加载开场动画
    - 设置地球加载图层
    - 设置月球为底图
    - 改变背景
- 2.2.2 插入 Marker 范例
    - 插入 Altizure 项目
        - 设置水面
    - 插入自定义标签
    - 插入多边形和体块
    - 插入折线
    - 插入 obj 模型
    - 插入文字标签
    - 插入带标签折线
    - 插入圆柱形折线
    - 插入 canvas 标签
- 2.2.3 交互事件
    - 鼠标事件
    - 解绑事件
- 2.2.4 获取坐标
    - 获取地球表面坐标
    - 获取模型表面坐标
    - 窗口坐标获取地球坐标
    - 地球经纬度转换窗口坐标
    - 读取高程
- 2.2.5 相机操作
    - 相机姿态设置
    - 相机飞行设置
    - 设置相机移动限制
    - 控制相机移动
    - 设置相机矩阵
- 2.2.6 其他
    - 裁剪项目
    - 体积测量

3. 常见问题

可以访问常见问题页面了解更多问题解答。演示应用页面展示了更加复杂的 Altizure Javascript SDK 应用范例。访问详细文档了解更多细节。

4. 了解更多

- ThreeJS
- WebGL
- OpenGL
- Vulkan
- 详解 OpenGL 坐标变换 OpenGL Transformation

—

该文档最后修改于 Thu Dec 21 2017 20:04:53 GMT+0800 (HKT)

Altizure JavaScript SDK 演示应用

这里我们会演示如何用 Altizure Javascript SDK 解决一些实际应用场景的问题，让 Altizure 的实景模型可以更好地结合到你的商业场景重，发挥实景三维模型的无穷潜力。

您对这些演示有任何问题都欢迎到 issue page 留言，我们会尽力解答。

1. 时间轴

时间轴可以非常直观地展示一个场景按照时间的变化。这是工程进度管理等任何与时间相关的应用都必不可少的工具。下面我们就通过几个应用演示几种可视化时间轴的方法。

1.1 格状平铺时间轴

- 演示链接：altizure.github.io/experimental-demo/timeline.html
- 演示重点：多个 Altizure 模型加载，文字标签设置，模型位置调整
- 演示简介：范例里对一个工地的施工过程超过20个不同时间点的三维模型进行了格状的平铺展示。每个模型带有相关的时间文字标签。

1.2 滑动条时间轴

- 演示链接：altizure.github.io/experimental-demo/timeline-slider.html
- 演示重点：模型可见性修改，实景三维内容与 html 界面交互，相机环绕动画，模型动态加载和销毁
- 演示简介：范例里通过一个滑动条让用户可以任意选取一个工地的施工过程一个时间点的三维模型进行浏览和检视。用户还可以按时间先后循序播放这个工地不同时间段的施工状况。范例里也提供了一个自动环绕飞行的功能，方便用户在展示时自动环绕兴趣点循环播放。

2. 传感器

现代的手持设备里有着各种各样的传感器，测量用户的位置、设备姿态、运动速度等等。这个章节的演示会展示如何结合这些传感器信息让你的三维模型更加具有互动性。

2.1 GPS 轨迹记录

- 演示链接：altizure.github.io/experimental-demo/gps-tracker.html
- 演示重点：图片标签加载，标签位置移动，相机跟随动画，地理坐标与屏幕坐标转换，Canvas 二维元素与实景三维元素结合，获取模型表面高程。
- 演示简介：该演示推荐用手机打开。在演示中，你可以打开轨迹记录，这样你的位置就会以演示中的模型的中心点为基准点，实时以一个 Altizure 图标的形式显示在实景三维模型中。同时您的移动轨迹也会用折线显示出来。移动轨迹的折线的渲染模式可以在三维渲染或者 canvas 二维渲染模式之间切换。您还可以打开相机跟踪模式，让相机在您移动的过程中使用跟随你移动。

3. 性能测试

3.1 文字标签渲染

- 演示链接：altizure.github.io/experimental-demo/performance-test.html

- 演示重点：文字标签加载和删除，测试三维渲染性能，获取模型表面高程
- 演示简介：演示默认加载 1000 个文字标签。可以通过修改标签数目，并点击 `Generate` 生成不同数目的标签。可以同时显示的标签数目越多，说明您的电脑的渲染能力越强。

3.2 线渲染

- 演示链接：altizure.github.io/experimental-demo/polycylinder-limittest.html
- 演示重点：管线加载和删除，测试三维渲染性能
- 演示简介：演示默认加载 100 个有 7 个位置随机的节点的管线。可以通过修改管线数目和节点数目，并点击 `Generate` 生成不同节点数的管线。可以同时显示的有更多节点数的管线数目越多，说明您的电脑的渲染能力越强。

4. 如何获取这些演示的代码

这些演示应用都是开源的，您可以通过以下的命令把代码复制到您的电脑查看并运行。

```
git clone https://github.com/altizure/experimental-demo.git
cd experimental-demo
python -m SimpleHTTPServer
```

打开您的浏览器，访问 `http://127.0.0.1:8000/` 便可运行和查看这些演示应用。

5. 了解更多

- Altizure Javascript SDK
- Altizure Javascript SDK 常见问题
- Altizure Javascript SDK 详细文档

—

该文档最后修改于 Thu Jan 11 2018 00:30:17 GMT+0800 (HKT)

常见问题

1 使用相关

1.1 能否提供测试？

您可以直接访问 altizure.github.io/sdk.examples/examples.sdk.html 来直接尝试各种范例的效果。把代码仓库 clone 到本地后即可修改其中的代码进行测试。

1.2 本地测试时无法访问相关模型？

请检查以下配置是否正确：

- 是否已经启动一个 http 服务器？最方便的方法是参考 altizure.github.io/sdk.examples/ 教程中用 python 建立 http 服务器的方法。
- 本地测试请用 `127.0.0.1` 作为地址来访问，而不要使用 `localhost`。

1.3 无法访问 sdk script 链接？

如果无法访问 `https://www.altizure.com/sdk` 或 `https://beta.altizure.com/sdk`，请使用中国站点 `https://www.altizure.cn/sdk`。

1.4 详细文档在哪儿？

请直接参考范例的代码，里面有详细注释。我们会持续更新相关范例，展示最新功能。

1.5 如何获取用户令牌？

请直接参考 GraphQL API 的 `5. 获取用户令牌` 章节。

1.6 Altizure 的三维引擎是用什么语言开发的？

开发语言是 Javascript，最底层三维的接口是 WebGL。但是我们并不经常直接访问 WebGL 底层接口，我们更多的模块是搭建在 ThreeJS 之上。

1.7 能否自定义一些 three.js 的对象然后插入到 Altizure Javascript SDK 的场景里？

暂时不能。因为实景模型的数据量非常巨大，我们的 SDK 对数据的加载和内存管理做了许多优化，暂时未能允许插入任意的 three.js 对象。我们会逐步实现这一点，但现在并无具体上线这个功能的时间表。

1.8 我是否需要学习图形学的知识才能用 Altizure Javascript SDK 编程？

基本不需要，我们尝试把这些复杂的技术实现都进行了封装。您只需要懂得 Javascript 和 html 一套的网页编程就可以进行开发。更重要的是要清晰知道实景三维模型和您的业务是如何结合的。

2 常见用例

2.1 点击模型中某个建筑物，怎么来获取对应的信息，如建筑物的名称？

可以通过鼠标事件获得经纬度，再去相关数据库中查建筑物名称。相关建筑物数据库由开发者自行管理。

2.2 能否直接插入 obj 模型？

可以。如果网络下载带宽不高，过大的 obj 文件可能导致下载时间过长。而且因为受限于浏览器的性能，过大的 obj 也会导致浏览器崩溃。我们一般推荐 obj 模型连同纹理文件不要超过 2 MB。

obj 模型需要满足以下要求：

- obj 模型需要由开发者自行找网络空间存储并获取直接访问的 https 链接。
- 其中 obj 对应的 mtl 文件内的纹理路径需要是相对路径。
- obj 只有三角形面。
- obj 模型里没有非2-流形的点和边，并且没有面积为0的面。

2.3 如果需要更换一个标签的click事件，应该怎么操作？需要把前一个绑定事件解绑吗？

最好解绑，用 `.off(event, func)`

2.4 如何实现分图层显示不同的标签及其他的模型？

可以在打开和关闭图层的时候从图层数据库中读取相关信息，然后动态将相关标签插入到 Sandbox 或者从中删除。如果不想每次开关图层反复插入和删除相关标签，可以用 `marker.visible=true` 或者 `marker.visible=false` 控制每个标签显示与否。

2.5 如何实现测量距离？

通过鼠标事件分别获取两次点击的坐标点，然后求两点距离。

2.6 如何加入一个带格式的文本标签

`TextTagMarker` 是一个带简单格式的文字标签，参考 范例2.6。

如果您需要更复杂的格式定义，您可以用 js 创建一个 canvas, 写入带格式的文本，把 canvas 转为 image, 传给 Tag。(可以把文字和图标都画到这张canvas上。)

2.7 如何获得三维场景拖动事件

响应 Sandbox 对象的 `cameraChange` 事件，`sandbox.on('cameraChange', callback)`。

2.8 地图框选的功能，希望能够在地图上画个方形或圆形，然后可以获取到画的图形的范围值

pick 鼠标 down 和 up 的点,

- pick 到的两个点作为矩形的对角两个端点，矩形的边平行于经纬方向, 使用 polygon marker 绘制矩形
- 以 pick 到的两个点作为直径，中点做圆心，还是用 polygon 就可以，比如画一个 360 边形。已知圆心经纬度，和直径，顶点位置可以很容易算出来。

2.9 如何删除一个标注?

`marker.destruct()` 可以完全销毁一个标注，这个标注所用的资源将被释放。如果您想反复显示和隐藏这个标注，可以设定这个参数 `marker.visible=true`。请参考 2.4 的解答。

2.10 平台的坐标系是什么？

wgs84

2.11 如果我需要通过方法调用实现放大，或缩小，接口是什么？

更改相机高度 `alt` ，参考范例第五章。

2.12 如果我需要通过方法调用旋转、平移，接口是什么？

更改相机经纬度 `lat/lng` 实现平移， `tilt` 更改俯仰视角， `north` 更改旋转视角。请参考范例第五章。

2.13 flyTo方法，是否有飞行速度参数？

有。 `flyTo: function (position, speed)` 。 `speed` 是一个速度的相对倍率，例如设做 2 的话，就是默认速度的 2 倍。

2.14 水面高度设置？

水面高度是在 Altizure 主站上定义的, SDK 目前不允许修改。

2.14 如何加载矢量图层数据？

您需要读取、解析您所使用的矢量数据，转化为 SDK 提供的点线面，加入场景中。

2.15 如何实现局部区域透明显示设置？

您可以使用 Altizure 主站的 裁剪 功能。

2.16 如何实现消息窗口功能？（类似点击弹出信息窗口）

使用 javascript 和 html 写好您想显示的弹窗，再和 marker 的鼠标事件绑定，比如 mouseover 或 click 时显示。

2.17 如何移动标注，或者绑定 GPS 设备的坐标？

只需要修改标注的坐标即可移动标注。需要开发者通过客户端或者服务器和不同设备通信获取设备的 GPS 坐标，便可直接把该坐标设做相关标注的坐标，便确保标注反映了设备的 GPS 位置。可以参考演示应用中的演示 2.1。

2.18 sandbox 里的 pid 是什么？

pid 是 project id 的简写。 浏览 Altizure 的 url (例如，https://www.altizure.com/project/59fe68cebb619d03c446fe85/model) 里面的 `59fe68cebb619d03c446fe85` 就是 pid。

2.19 如何初始化 sandbox 的相机 (camera) 参数？

这里的 camera 参数，是指 sandbox 初始化的时候填的相机位置。它可以通过 sdk 得到，参考 范例 5.1。camera的参数是以经纬度，高度定义的。经纬度可以从谷歌/百度地图拾取，或者更简单的，从范例 4.1拾取，点击想要的地方，就会显示了。console 里有日志, 要离地面近一点也能看到文字标签。 如果您也在使用 Altizure GraphQL API, project 函数中有存地理位置的参数。在 geoInfo 里的 centerLat, centerLng， 可以分别赋值给 lat, lng。高度你可以自己定义，比如 1000(米)。

2.20 平台支持BIM格式的文件吗？

目前不支持，有指定需要再开发。

2.21 创建的标签，有没有方法修改 position 坐标？同一个标签，不销毁重建的情况下，修改其位置/通过鼠标点击地图来设置标签的位置？

```
marker.setPose (position, orientation, scale)
```

这里 `position: {lng, lat, alt}`，`orientation: {x, y, z, w}`，`scale: number`。 如果只改 `position`，另外两个可以设为 `undefined`，如 `setPose(position, undefined, undefined)`。

2.22 将 obj 转成类似 Altizure 原生项目那样的金字塔分层（LOD）数据，有什么要求？

obj 需要符合一些数学规则，才可以做LOD。包括：obj 只有三角形面。obj 模型里没有非2-流形的点和边，并且没有面积为0的面。详情需联系技术人员测试。

2.23 如何给一个已有的 `PolyLineMarker` 添加更多的点？

使用 `altizure.PolyLineMarker::addPointMarker` 如：

```
polylineMarker.addPointMarker(new altizure.LngLatAlt(
  lng, lat, alt
))
```

3. 阅读更多

- 基础范例列表
- 演示应用 展示了更加复杂的应用范例。
- Altizure Javascript SDK 详细文档

—

该文档最后修改于 Thu Dec 21 2017 20:04:53 GMT+0800 (HKT)