**影像**

CesiumJS 支持多种高精度影像图层服务, 包括 [Cesium ion](https://cesium.com/ion). 使用Cesium ion可以从栅格数据转换成CesiumJS支持加载的影像图层，并分级分块的加载。图层可以排序和颜色混合，每个图层的亮度、对比度、饱和度、色调能被动态改变。本篇教程将介绍图像层概念和相关的CesiumJS api。

要加载自己的图层，可以参照前面介绍的，先将数据添加到Cesium ion上，然后根据Asserts id进行加载。

**快速开始**

打开Sandcastle 中的 [Hello World](https://sandcastle.cesium.com/?src=Hello%20World.html) 示例。这个示例创建了一个viewer，通过Cesium ion加载了Bing Maps Aerial imagery影像图层 （注意defaultAccessToken设置成自己的token信息）

Cesium.Ion.defaultAccessToken **=** 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJkYmEwNWZmMC1lZTlhLTQ0YzQtYTk5OS04MjY0MzgyMmE3MGQiLCJpZCI6MTQ0NzcsInNjb3BlcyI6WyJhc3IiLCJnYyJdLCJpYXQiOjE1NjU2NTkwOTZ9.gimVCxsdX7dVIJBhUPO4rHUZh6iHvH-orUikRWlTT9w';

**var** viewer **=** **new** Cesium.Viewer('cesiumContainer', {

imageryProvider : Cesium.createWorldImagery({

style : Cesium.IonWorldImageryStyle.AERIAL\_WITH\_LABELS

}),

baseLayerPicker : **false**

});

**Cesium ion 账户**

Cesium ion是提供空间数据管理服务的，必须先注册，通过token和资产id实现图层加载。



当您放大和缩小时，层会根据需要加载数据。

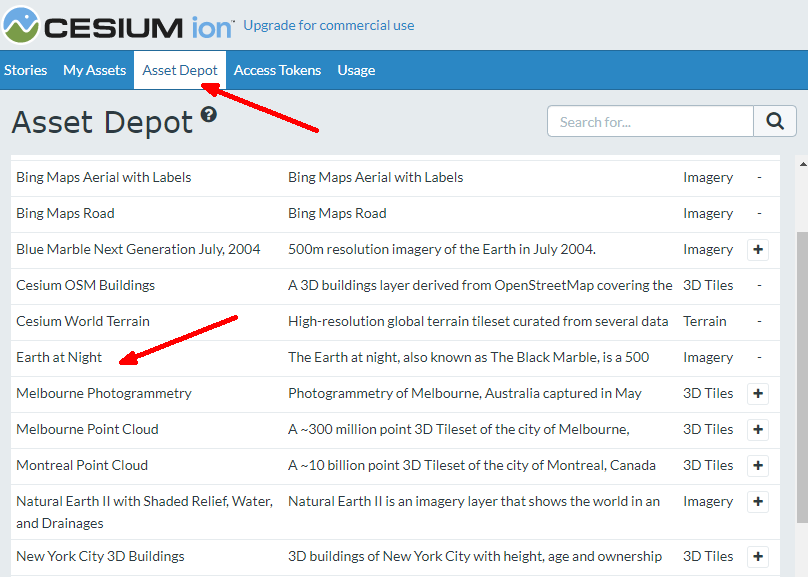
以下代码是添加另一层图像层（夜晚的地球）：

**var** layers **=** viewer.scene.imageryLayers;

**var** blackMarble **=** layers.addImageryProvider(**new** Cesium.IonImageryProvider({ assetId: 3812 }));



先要在Cesium ion资产管理中添加[Earth at Night](https://cesium.com/ion/assetdepot/3812)，并记住Asserts Id



由于它是最后添加的并且覆盖了整个地球，黑色图层层覆盖了Bing层。我们可以用layers.lower(blackMarble);将图层移动到底部，但是让我们将其与Bing层混合，这样我们就能更好地理解这两个层之间的关系：

blackMarble.alpha **=** 0.5; *// 0.0 透明. 1.0 不透明.*



下面，增加亮度：

blackMarble.brightness **=** 2.0; *// > 1.0 增加. < 1.0 减少.*



最后，增加第三个图层，覆盖特定的区域

***layers***.addImageryProvider(**new** Cesium.SingleTileImageryProvider({  
 **url** : **'./data/CesiumLogo.png'**,  
 **rectangle** : Cesium.Rectangle.fromDegrees(-75.0, 28.0, -67.0, 29.75)  
}));



完整代码：

*// Your access token can be found at: https://cesium.com/ion/tokens.*Cesium.Ion.**defaultAccessToken** = **'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJkYmEwNWZmMC1lZTlhLTQ0YzQtYTk5OS04MjY0MzgyMmE3MGQiLCJpZCI6MTQ0NzcsInNjb3BlcyI6WyJhc3IiLCJnYyJdLCJpYXQiOjE1NjU2NTkwOTZ9.gimVCxsdX7dVIJBhUPO4rHUZh6iHvH-orUikRWlTT9w'**;  
*// Initialize the Cesium Viewer in the HTML element with the `cesiumContainer` ID.***const *viewer*** = **new** Cesium.Viewer(**'cesiumContainer'**);   
**var *layers*** = ***viewer***.scene.imageryLayers;  
**var *blackMarble*** = ***layers***.addImageryProvider(**new** Cesium.IonImageryProvider({ **assetId**: 3812 }));   
***blackMarble***.**alpha** = 0.8; *// 0.0 透明. 1.0 不透明****blackMarble***.**brightness** = 2.0; *// > 1.0 增加. < 1.0 减少.****layers***.addImageryProvider(**new** Cesium.SingleTileImageryProvider({  
 **url** : **'./data/CesiumLogo.png'**,  
 **rectangle** : Cesium.Rectangle.fromDegrees(-75.0, 28.0, -67.0, 29.75)  
}));

**准备影像瓦片服务**

Sandcastle中的资产标签包含了更多由cesium托管的图像瓦片集，这些图片可以通过几行代码添加到CesiumJS应用程序中。许多这样的瓦片集也可用于本地使用。

**更多影像接口**

像上面使用的前两层那样的高分辨率图像太大，无法放入内存中，因此图像被分成更小的图像，称为瓦片，可以根据视图浏览范围，只请求加载范围内的图片数据传输到客户端。Cesium支持使用图像接口请求瓦片的几种标准。大多数图像接口使用基于HTTP的REST接口来请求瓦片。图像提供者根据请求的格式和瓦片的组织方式而有所不同。分为以下几类：

* [WebMapServiceImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/WebMapServiceImageryProvider.html): [Web Map Service](http://www.opengeospatial.org/standards/wms) (WMS), 一个OGC标准，用于从分布式地理空间数据库请求地理区域的地图分幅。.
* [TileMapServiceImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/TileMapServiceImageryProvider.html): [Tile Map Service](https://cesium.com/docs/cesiumjs-ref-doc/TileMapServiceImageryProvider.html) (TMS), 用于访问地图分幅的REST接口。可以用 [Cesium ion](https://cesium.com/ion) 或 [GDAL2Tiles](http://www.klokan.cz/projects/gdal2tiles/)来创建。
* [WebMapTileServiceImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/WebMapTileServiceImageryProvider.html): [OpenGIS Web Map Tile Service](http://www.opengeospatial.org/standards/wmts) (WMTS), 一个OGC标准，用于在互联网上提供预先准备好的地理参考地图块。
* [OpenStreetMapImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/OpenStreetMapImageryProvider.html): 访问OpenStreetMap发布的数据。
* [BingMapsImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/BingMapsImageryProvider.html): 使用Bing地图REST服务访问互动程序。Bing地图密钥可以创建在https://www.bingmapsportal.com/。.
* [ArcGisMapServerImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/ArcGisMapServerImageryProvider.html): 使用Esri ArcGIS服务器REST API访问由ArcGIS MapServer托管的数据。[GoogleEarthEnterpriseMapsProvider](https://cesium.com/docs/cesiumjs-ref-doc/GoogleEarthEnterpriseMapsProvider.html): 访问谷歌数据。.
* [MapboxImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/MapboxImageryProvider.html): 使用Mapbox API访问瓦片[SingleTileImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/SingleTileImageryProvider.html): 创建一个单独的影像范围。.
* [UrlTemplateImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/UrlTemplateImageryProvider.html): 创建自定义服务加载。使用URL模板连接到大量图像源。例如，TMS的URL模板是//path to tiles/{z}/{x}/{reverseY}.jpg。
* [TileCoordiantesImageryProvider](https://cesium.com/docs/cesiumjs-ref-doc/TileCoordinatesImageryProvider.html): 演示如何在将地球划分为多个分幅，方法是在每个分幅周围绘制边界并用级别、X和Y坐标标记。

**跨域资源共享**

作为安全措施，web浏览器阻止Javascript代码读取来自不同站点的图像。特别是，如果那些图像来自不同的主机名或端口，并且服务器不明确允许以这种方式使用图像，那么像CesiumJS这样的WebGL应用程序就不能将图像用作纹理。

服务器指示图像不包含机密信息，因此，通过在HTTP响应中包含跨源资源共享（CORS）标头，其他站点可以安全地读取它们的像素。

不幸的是，并不是所有的图像服务都支持CORS。对于那些没有的，必须使用与托管应用程序的网站位于同一来源的代理服务器。当使用这样的代理时，web浏览器和CesiumJS客户机上显示的图像好像来自同一个来源。.要将代理与图像提供程序一起使用，请在构造图像提供程序时使用proxy属性。为了开发方便，Cesium包括一个用node.js编写的简单的代理。

layers.addImageryProvider(**new** Cesium.WebMapServiceImageryProvider({

url : **new** Cesium.Resource({

url: '/path/to/imagery',

proxy : **new** Cesium.DefaultProxy('/proxy/')

})

}));

如果您托管的是公共图像，我们鼓励启用这里描述的CORS，而不是使用代理。

**Imagery providers 和 layers**

目前为止，我们可以加载不同的影像，一个影像接口提供者可以请求特定的服务。

**var** layer **=** layers.addImageryProvider(imageryProvider);

这是对以下代码的简写：

**var** layer **=** **new** ImageryLayer(imageryProvider);

layers.add(layer);

我们通常构建一个imagery provider，来创建图层，然后通过属性（show, alpha, brightness,  contrast）控制图层改变样式。

像上述示例中的图层一样，图像图层集合决定了图层的绘制顺序。图层将根据添加顺序从下到上绘制。图像层集合像CesiumJS中的任何其他集合一样，使用add、remove和get等函数进行操作。此外，可以使用 raise, raiseToTop, lower和 lowerToBottom.对图层重新排序。

**学习资源**

查看 Sandcastle关于影像加载的程序示例:

[Imagery Layers](https://sandcastle.cesium.com/?src=Imagery%20Layers.html)

[Imagery Layers Manipulation](https://sandcastle.cesium.com/?src=Imagery%20Layers%20Manipulation.html)

[Imagery Adjustment](https://sandcastle.cesium.com/?src=Imagery%20Adjustment.html)

查看相关API文档:

[All imagery providers](https://cesium.com/docs/cesiumjs-ref-doc/?classFilter=ImageryProvider)

[ImageryLayer](https://cesium.com/docs/cesiumjs-ref-doc/ImageryLayer.html)

[ImageryLayerCollection](https://cesium.com/docs/cesiumjs-ref-doc/ImageryLayerCollection.html)