

# PPVision 客户端

## 开发指南

版本 6.0.1

北京威远图易数字科技有限公司

2017 年 6 月 9 日

## 目录

PPVision 客户端.....	1
开发指南.....	1
目录.....	2
1 简介.....	4
2 安装与部署.....	4
2.1 客户端.....	4
2.2 服务端目录结构.....	4
2.3 运行环境.....	4
3 主界面.....	5
4 操作方式.....	6
4.1 视图浏览.....	6
4.2 定位操作.....	6
4.3 定位模式.....	7
5 开发接口.....	7
5.1 方法.....	7
5.2 事件.....	14

# 1 简介

PPVision 客户端是基于 HTML5 技术的实景、全景组件，该控件可访问 PPServer 发布的全景或可测量影像资源，同时也提供了在图像上进行空间采集与量测的功能，开发人员可以轻易地将该控件嵌入到基于 Web 应用的网页系统。

主要功能

- ◆ 浏览全景或可测量影像
- ◆ 空间三维量测
- ◆ 空间三维采集

## 2 安装与部署

### 2.1 客户端

客户端无需安装部署，但需要支持 HTML5、WebGL 的浏览器，例如 Chrome、Firefox。

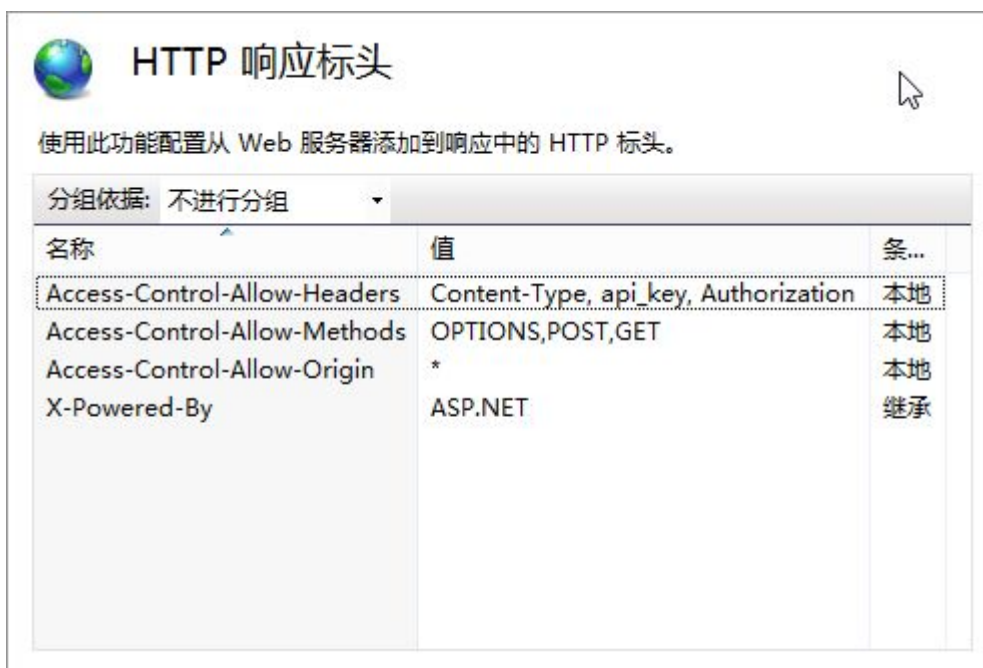
### 2.2 服务端目录结构

/ppv	ppvision 组件
/icon	图标文件夹
/js	ppvision 组件脚本

### 2.3 运行环境

Web 服务需要允许跨域访问。

IIS 配置如下：



### 3 主界面



## 4 操作方式

### 4.1 视图浏览

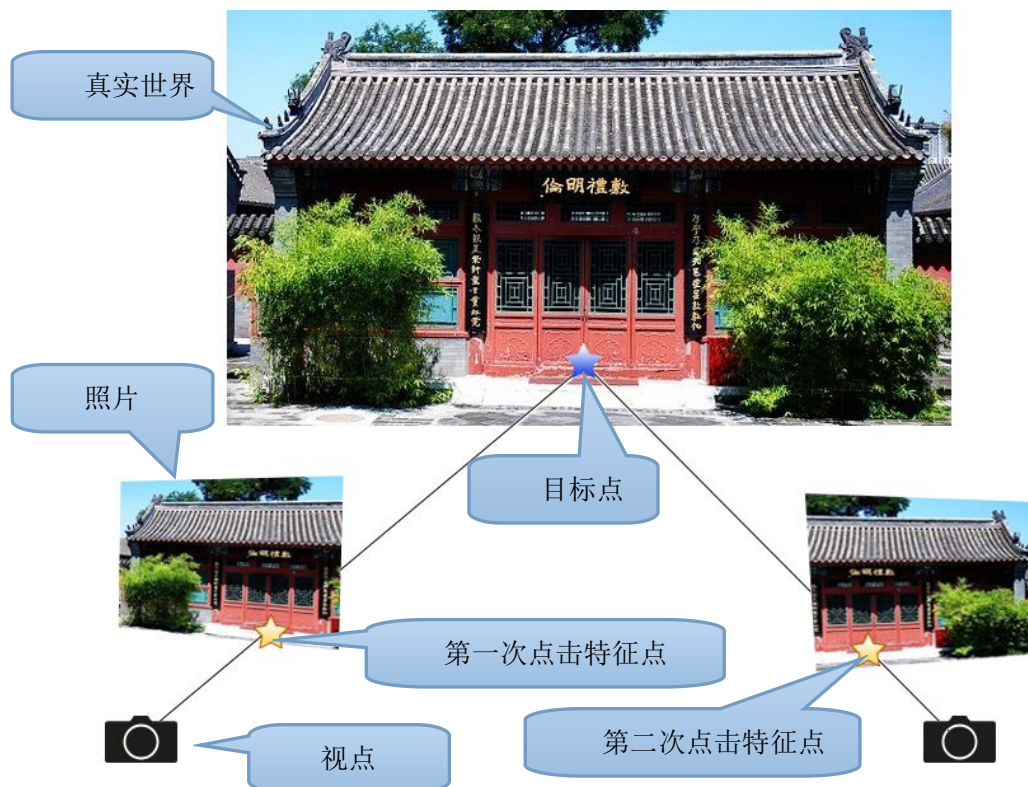
滚轮	前进、后退
方向箭头	前进、后退
翻页键	快进、快退
空格键	播放、停止播放
F9	全屏
路线指示箭头	前进、转换路线
历史数据	查看、转换历史帧

### 4.2 定位操作

采集线、面，测量长度、面积

左键	添加节点
中键	撤销一步
右键	完成

## 4.3 采样模式



### 摄影测量法

在第一张照片上点取一个特征，自动切换到下一帧，并显示一条绿色辅助线，再次点击特征点，完成定位。

#### 注意：摄影测量法操作原则

- 1 只能测量静态物体，不能测量运动物体
- 2 尽量在物体最近的距离测量，以免误差放大
- 3 尽量放大图像，点击位置精确到像素
- 4 找到明确的特征点，比如顶点、角点、交点、文字或图案。不要用在杆体、标线等无法重复定位的特征
- 5 不要测量重复的物体，比如栅栏的中段，因为到第二张照片上很难确定第一张照片点击的位置

### 地面采样法

如果帧数据包含地面高度信息，可以使用地面采样模式，在一帧照片上点击一次，完成定位。

**注意：**地面采样法只能采集地面上的对象，不能采集高于、低于地面的对象。而且定位精度较差。

## 5 开发接口

### 5.1 变量

#### 1. needsUpdate

在 onPosition 中设置 `ppv.needsUpdate = true`; 可以强制 ppv 更新所有图层要素

### 5.2 方法

#### 1. getVersion

查询当前 ppv 版本

返回

一个字符串

#### 2. setPref

设定 ppv 系统参数

参数

def 参数定义，一个对象。可以只设置需要的

```
{
  bgcolor:0x000000,    //'rgb(255,255,255)', 背景色
  fullView:FullMode.trans,//照片填充窗口的模式
  enableArrow:true,    //方向箭头
  enableHistory:true,  //历史表盘
  scope:100,           //可视范围，米，只对点要素起作用
  thumb:'Small',       //缩略图尺寸，可选['Middle', 'Small']，缺省'Middle'
  //方向箭头位置
  arrows:{
    forward:5,         //向前，米
    below:1.8,         //降低，米
    lean:15,           //前倾，度
  },
  //放大镜
  magnifier:{
    size:256,          //放大镜尺寸，像素
    zoom:5.0,          //放大倍数
    fix:false,         //是否启动放大镜后，固定位置不变。
  }
}
```

```

//另一种方式是，放大镜随鼠标移动
},
// 快捷键
key:{
    play:32,          //播放    space
    fforward:33,      //快进    page up
    fbackward:34,     //快退    page down
    forward:38,        //前进    up
    backward:40,       //后退    down
    fullscreen:120,    //全屏    f9
},
//标注样式
label:{
    fontface:'微软雅黑', //字体
    fontsize:15,          //字高，像素
    textcolor:'#000000',  //文字颜色
    borderThickness:1,    //边框线宽，如果取 0 值，将禁用边框
    borderFillet:0,       //边框圆角
    borderColor:'rgba(0,0,0,0.8)', //边框颜色
    backgroundColor:'rgba(255,255,255,0.8)', //背景颜色
},
}

```

其中fullView 可选值:

```

var FullMode= {
    fill:1,          //填充，黑边，保持全部数据可见
    trans:2,         //镂空，露出背景，保持全部数据可见
    clip:3,          //裁剪，充满div，数据不全
    stretch:4,      //拉伸，充满div，会变形
}

```

**注意：**ppv 在网页中布局与css 样式，请参考样例页面中的设计

### 3. getPref

获得当前 ppv 系统设置

返回

参数对象，见 setPref 定义

### 4. setServer

设置服务器地址，并验证服务。

参数



*url*      *WebService 服务地址*

例如: *"http://192.168.11.233:8022/PPVisionServer.asmx"*

## 5. locate

打开全景或可测量影像, 会触发 onLocate 事件

### 参数

*type*      *图像类型*

0      *ppv uniform*

3      *imajbox*

4      *streetview*

*lon, lat*      *地理位置*

*key*      *用户密钥*

### 事件

触发 onLocate 事件

## 6. locateById

打开全景或可测量影像, 会触发 onLocate 事件

### 参数

*type*      *图像类型, 同 locate*

*id*      *照片 id*

*key*      *用户密钥*

### 事件

触发 onLocate 事件

## 7. step

前进或后退

### 参数

*step*      *帧数, 负数表示后退*

## 8. play

开始自动沿轨迹播放图像

## 9. stop

停止播放图像

## 10. isPlaying

是否正在播放图像

返回

*true/false*

## 11. setTool

设置当前工具

参数

*cid* 工具定义

```
var Tool=
{
    none:0,
    //
    measurePoint:1,
    measureLength:2,
    measureArea:3,
    measureZ:4,
    measureFacade:5,
    measureAngle:6,
    measureSlope:7,
    createPoint:11,
    createPolyline:12,
    createPolygon:13,
    caliplane:14,
    //
    pick:21,
    remove:22,
    fullscreen:23,
    lrs:24,
    //
    play:31,
    stop:32,
    next:33,
    prev:34,
    branch:35,
```

```
        back:36,  
        history:37,  
        //  
    };
```

#### 事件

触发 *onTool* 事件

## 12. getTool

查询当前工具

#### 返回

工具 *id*，见 *setTool* 参数 *cid* 定义

## 13. setSampleMode

设置采样模式。摄影测量法，定位一个点需要在两个影像点击两次，标定面法只需要点击一次，但需要预先标定面。

#### 参数

*mode*     采样模式

```
var SampleMode=  
{  
    none:0,  
    cloud:1,  
    photo:2,  
    ground:3,  
    plane:4,  
    object:5,  
    depth:6,  
};
```

#### 事件

触发 *onSampleMode* 事件

## 14. getSampleMode

查询当前采样模式

#### 返回

采样模式，见 *setSampleMode* 参数 *mode* 定义

## 15. getFrame

查询当前照片 id

返回

照片 id

## 16. lookAt

定位视线方向，通过当前视点位置与目标点坐标，确定一条射线

**注意：**仅全景有效

参数

lon, lat, alt          目标点，经纬度坐标

## 17. addLayer

添加一个图层

参数

def      图层定义

    样例 1 :

```
{
    name:"路灯",      //图层名称
    type:"Point",     //Point, Line, Polygon 三种类型，并非强约束
    color:0xffff00,   //要素颜色
    opacity:0.5,      //透明度
    size:40,          //图标大小
    icon:"icon/创建点.png", //图标
    fontSize:15,      //文字尺寸
};
```

    样例 2 :

```
{
    name:"Road",      //图层名称
    type:"Line",      //Point, Line, Polygon 三种类型，并非强约束
    color:"rgb(255,0,255)", //RGBA or RGB，颜色
    opacity:1,        //透明度
    lineWidth:2,      //线宽，单位是像素，windows 平台仅支持1
    fontSize:15,      //文字尺寸
}
```

返回

图层 handle

## 18. findLayer

根据图层名，查找图层

**参数**

*name*      图层名

**返回**

图层 *handle*

## 19. getLayer

获得图层定义

**参数**

*handle*      图层句柄

**返回**

图层定义，与 *addLayer* 相同

## 20. setLayer

重新定义图层，更新已有图层样式

**参数**

*handle*      图层句柄

*def*          图层定义，与 *addLayer* 相同

## 21. removeLayer

删除图层

**参数**

*handle*      图层 *handle*

## 22. removeAllLayers

删除所有图层

## 23. addFeature

向指定图层添加点、线、面要素

**参数**

*hlayer*      图层 *handle*

*def*          要素定义，GeoJson

支持基本 *GeoJson* 对象

其中 *properties* 不是必须

可识别的属性包括:

*name*: 标注内容

*fid*: 要素 *id*

*toGround*: 高程值是否相对于地面, 如果不设置此属性, 则使用绝对高程

样例

```
{
  "type": "Feature",
  "fid": 1234,
  "name": "灯杆",
  "toGround": 0,
  "geometry": {
    "type": "Point",
    "coordinates": [
      103.7618144853319,
      36.08614306284845,
      1481.648249594895
    ]
  }
}
```

返回

要素 *handle*

## 24. findFeature

根据要素的 *fid* 以及所属图层查找要素

参数

*fid*            要素 *fid* 属性

返回

要素 *handle*

## 25. getFeature

获得要素定义

参数

*handle*        要素句柄

返回

要素定义，与 *addFeature* 相同

## 26. setFeature

重新定义要素，更新已有要素

### 参数

*handle*            要素句柄  
*def*              要素定义，与 *addFeature* 相同

## 27. removeFeature

删除要素

### 参数

*handle*            要素句柄

## 28. removeAllFeatures

删除指定图层的所有要素

### 参数

*hlayer*            图层 *handle*

## 29. selectFeature

选中要素，高亮显示

### 参数

*handle*            要素 *handle*

## 30. visible

隐藏或显示图层或要素

### 参数

*handle*            图层或要素 *handle*  
*visible*           是否可见

## 5.3 事件

### 1. onLocate

响应 locate 和 locateById，可判断是否成功

参数

```
event={  
    state: LocateState.success,  
}
```

其中 state 可选值:

```
var LocateState={  
    success:0,  
    typeError:1,  
    dataError:2,  
    imageError:3,  
    busy:4,  
};
```

### 2. onPosition

响应视点位置改变

参数

```
event={  
    lon:114.012,    经度，单位为度  
    lat:39.231,    纬度，单位为度  
    alt:62.089,    高程，单位为米  
    prev:{  
        id:1234,  
        heading:67,  
    },  
    next:{  
        id:1234,  
        heading:67,  
    },  
    back:{  
        id:1234,  
        heading:67,  
    },  
    branch:[  
        {  
            id:1234,
```



```

        heading:67,

      },
      {id:1234,
        heading:67,
      },
    ],
    history:[
      {
        id:1234,
        heading:67,

      },
      {id:1234,
        heading:67,
      },
    ],
  },
}

```

### 3. onEye

响应视角改变

**参数**

```

event={
  heading:45.23,    当前视线方位角，单位为度
  fovx:45,         水平视角，单位为度
}

```

### 4. onFeatureCreate

响应采集点、线、面工具

**参数**

```

def      要素定义，GeoJson，

```

### 5. onFeatureSelect

响应要素被控件工具选中

**参数**

```

event = {
  layer:hlayer,      图层 handle
  feature:handle,    要素 handle
}

```

```
    layername:"lay_name",    图层名称
    fid:123,                  要素 id
  }
```

## 6. onFeatureRemove

响应要素将被控件工具删除

**参数**

```
event = {
  layer:hlayer,              图层 handle
  feature:handle,            要素 handle
  layername:"lay_name",     图层名称
  fid:123,                   要素 id
}
```

## 7. onMeasure

响应测量工具

**参数**

```
event          不同德测量工具，返回不同的定义
  样例 1: {"length":123.456}
  样例 2: {"length":10.459530,"dz":10.232580,"dxy":2.167046}
```

## 8. onTool

响应工具改变

**参数**

```
event = {tool:11}          不同德测量工具，返回不同的定义
```