

天地图API

第一章 初识天地图API

1. Web API 介绍

天地图Web API 是一套由 JavaScript 语言编写的应用程序接口，它能够帮助您在网站中制作各种类型、行业的地图应用，还可以使地图功能能够以模块化集成在不同类型的系统应用中。

地图 API 是由 JavaScript 语言编写的，您在使用之前需要通过<script> 标签将 API 引用到页面中：

```
<script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>
```

此API 为地图展示版本，主要完成地图的展示和操作功能，如地图定位、放大和缩小等！

兼容性：

浏览器：IE 6.0+、Firefox 3.6+、Opera 9.0+、Safari 3.0+、Chrome

操作系统：Windows、Mac、Linux

2. 简单示例

从一个简单的示例开始学习天地图的地图API。以下代码创建了一个地图并以南京作为地图的中心。

代码如下：

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

  <title>天地图入门</title>

  <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

  <script>

    var map,zoom=12;
```

```

        function onLoad()
        {
            map=new TMap("mapdiv");

            //初始化地图

            map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);

            //设置显示地图的中心点和级别

        };
    </script>

</head>

<body onLoad="onLoad()" style="margin: 0px">

    <div id="mapdiv" style="position:absolute;width:100%;height:100%"></div>

</body>

</html>

```

下面说明一下具体的步骤

1) 准备页面

根据 **HTML** 标准，每一份 **HTML** 文档都应该声明正确的文档类型，我们建议您使用最新的符合 **HTML5** 规范的文档声明：

```
<!DOCTYPE html>
```

2) 引入天地图的地图 JavaScript API 文件

```
<script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>
```

3) 创建地图容器元素

地图需要一个 **HTML** 元素作为容器，这样才能展现到页面上。这里我们创建了一个 **div** 元素。将 **div** 元素的宽和高分别设置为 **100%**，使其充满整个屏幕，或者您也可以计算浏览器窗口的大小并进行设置。

4) 创建地图实例

```
Var map=new TMap("mapdiv");//初始化地图对象
```

TMap 类表示地图，可以通过 **new** 操作符可以创建一个地图实例。其参数可以是元素 **id** 也可以是元素对象。注意在调用此构造函数时应确保容器元素已经添加到地图上。

5) 确定经纬度坐标

```
Var lnglat=new TLatLng(118.79125,32.061);
```

这里我们使用T命名空间下的Point类来创建一个坐标点。TLatLng类描述了一个地理坐标点，其中118.79125表示经度，32.061表示纬度。

6) 地图初始化

```
map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);
```

在创建地图实例后，我们需要对其进行初始化，map.centerAndZoom方法要求设置中心点坐标和地图级别。地图必须经过初始化才可以执行其他操作。

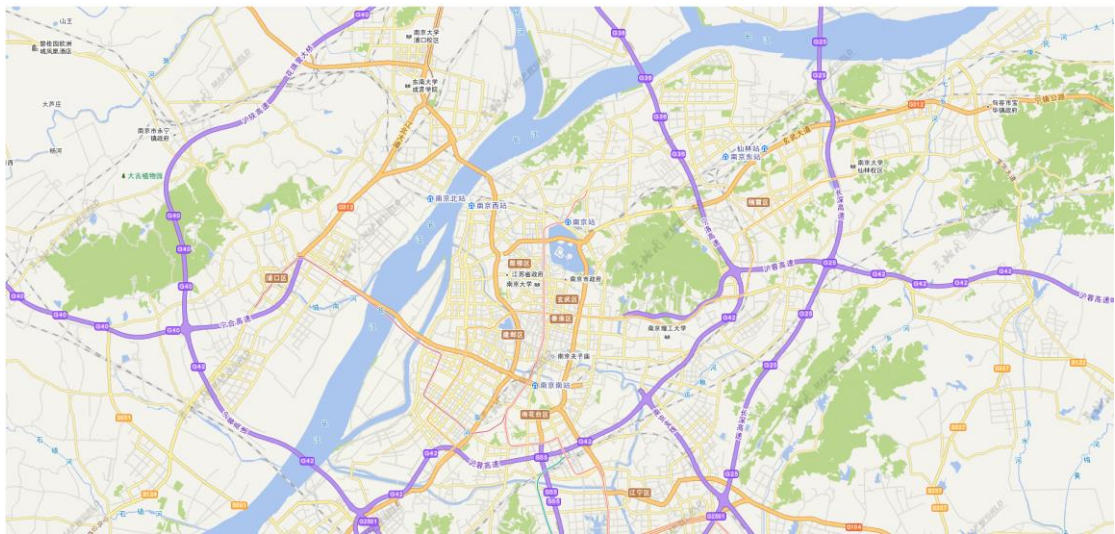


图1-1 初始化地图

第二章 地图控件

1. 知识要点

地图控件是地图API 提供的用户操作地图的界面元素，常用地图控件有：缩放/平移控件、鹰眼控件、比例尺控件、版权控件、地图类型控件。

在天地图API 中，TControl 类是所有控件类的基类，可以通过此类来自定义控件，所有控件均包含Control 类的属性、方法和事件。在下一章我们会具体介绍自定义控件。

常用的地图控件在天地图API 中都有内置的类：

- 1) TNavigationCotrol 类：缩放/平移控件（骨头棒控件），可以缩放和移动地图。PC 端默认在地图的左上方。
- 2) TOverviewMapControl 类：鹰眼地图控件，用来显示一个鹰眼地图，继承自 TControl，拥有基类的所有方法。默认位于地图右下方，是一个可折叠的缩略地图。
- 3) TScaleControl 类：地图比例尺控件，用来实时的显示地图的当前比例尺。默认位于地图的左下方。
- 4) TCopyrightControl 类：此类表示版权控件，可以在地图上添加自己的版权信息。每一个版权信息需要包含如下内容：版权的唯一标识、版权内容和其适用的区域范围。默认位于左下方。
- 5) TMapTypeControl 类：此类是负责切换地图类型的控件，此类继承 TControl。默认在地图的右上角。

2. 功能示例

在地图的使用中，常用控件是不可或缺的。通过下面是一个示例，讲解如何把这些控件加载到地图里面。

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title>常用控件 </title>

<script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

<script>

    var map,zoom=12,control;

    function onLoad() {

        map = new TMap("mapdiv");//初始化地图

        map.centerAndZoom(new TLngLat(118.79125, 32.061), zoom); //设置显示地图的中心,点和
级别

        map.enableHandleMouseScroll() //允许鼠标滚轮缩放地图

        control = new TNavigationControl({

            type: "TMAP_NAVIGATION_CONTROL_LARGE", //缩放平移的显示类型

            anchor: "TMAP_ANCHOR_TOP_RIGHT",          //缩放平移控件显示的位置

            offset: [0, 0],          //缩放平移控件的偏移值

            showZoomInfo: true          //是否显示级别提示信息, true 表示显示, false 表示隐
藏。

        })//创建缩放平移控件

        map.addControl(control); //添加缩放平移控件

        var Overview = new TOverviewMapControl({size:new TSize(180,120),isOpen: true}); //创建鹰
眼控件

        map.addControl(Overview); //添加鹰眼控件

        var scale=new TScaleControl(); //创建比例尺控件

        map.addControl(scale); //添加比例尺控件

        var maptype =new TMapTypeControl(); //创建地图类型控件

        map.addControl(maptype); //添加地图类型控件

        maptype.setLeft(10);

        maptype.setTop(20);

    };

```

```

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

<div id="mapdiv" style="position:absolute;width:100%;height:100%"></div>

</body>

</html>

```

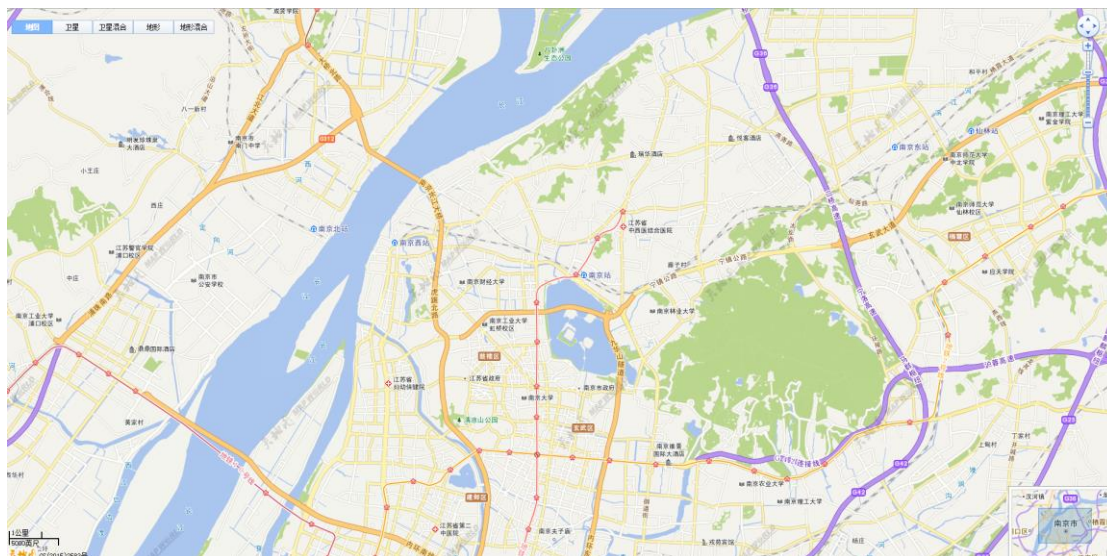


图2-1 常用控件示例图

3. 知识细节

通过图2-1 我们可以看到常用的控件都已经加载在了地图上,但是和他们默认的情况不太一样, 为什么会出现这样的情况, 如何改变控件的默认状态?

在天地图API 的控件类中不仅提供了简单的构造函数,还提供了相应的方法和属性,通过调用这些方法和属性我们可以设置自己想要的控件效果。

1) 通用方法

TControl 类是所有控件类的基类, 所有控件均包含 *Control* 类的属性、方法和事件。

下面是一些*TControl* 类的方法:

- a *setLeft*(length:Numbe) 设置控件相对于地图左边的像素距离。
- b *setRight*(length:Numbe) 设置控件相对于地图右边的像素距离。
- c *setTop*(length:Numbe) 设置控件相对于地图上方的像素距离。

- d `setBottom(length:Number)` 设置控件相对于地图下方的像素距离。
- e `setOpacity(opacity:Number)` 设置控件的透明度。
- f `getOpacity()` 返回控件的透明度。
- g `show()` 显示控件。
- h `hidden()` 隐藏控件。
- i `isHidden()` 控件是否被隐藏。

例如代码示例中:

```
maptype.setLeft(10);  
maptype.setTop(20); // 设置了地图类型控件在地图中的位置。
```

2) 控件位置

控件的位置除了使用 `Set` 方法外, 还可以使用天地图API 内置的 `anchor` 和 `offset` 属性共同控制控件在地图上的位置。 `anchor` 表示控件的停靠位置, 即控件停靠在地图的哪个角。地图尺寸发生变化时, 控件会根据停靠位置的不同来调整自己的位置。 `offset` 指示控件距离地图边界有多少像素。如果两个控件的停靠位置相同, 那么控件可能会重叠在一起, 这时就可以通过设置偏移量使二个控件分开显示。

`anchor` 允许的值为:

`TMAP_ANCHOR_TOP_LEFT` 表示控件定位于地图的左上角。

`TMAP_ANCHOR_TOP_RIGHT` 表示控件定位于地图的右上角。

`TMAP_ANCHOR_BOTTOM_LEFT` 表示控件定位于地图的左下角。

`TMAP_ANCHOR_BOTTOM_RIGHT` 表示控件定位于地图的右下角。

Anchor 使用方法

```
OverviewMap=new TOverviewMapControl( {  
    anchor: "TMAP_ANCHOR_TOP_RIGHT"  
}); // 设置控件到右上角
```

上面的代码把缩略图的位置设置到了右上角, 如下图所示:

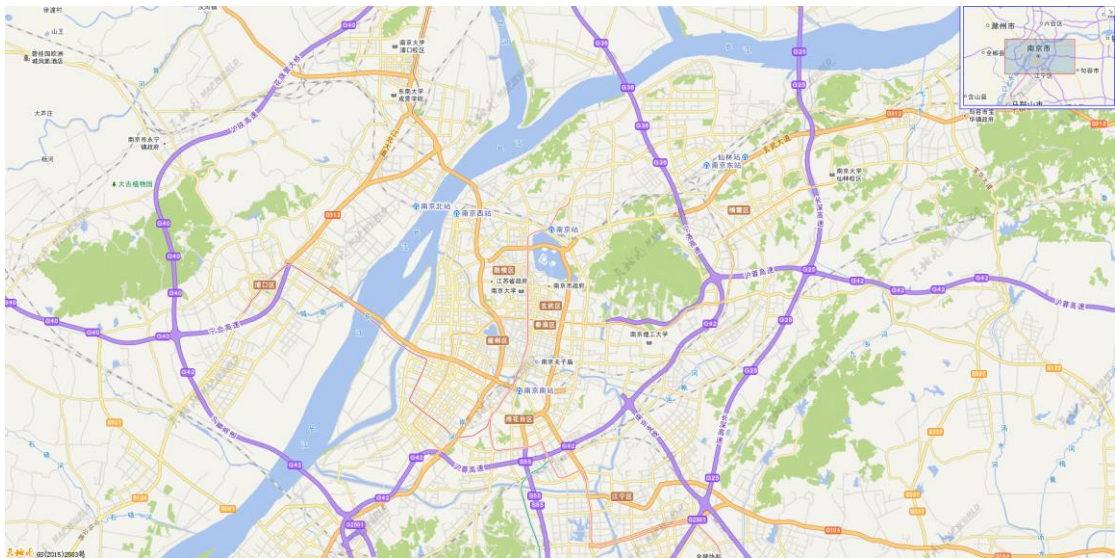


图2-2 鹰眼位置图

Offset 使用方法:*offset*:*[TPixel]*

*TPixel*类是以像素坐标表示的地图上的一个点。*TPixel (x:Number, y:Number)* 。像素坐标中x坐标, x坐标向右增大。 像素坐标中y坐标, y坐标向下增大。根据给定x和y坐标创建地图上的一个像素坐标点。

示例代码:

```
control=new TNavigationControl({
    offset:[30,30]
})//位置偏移向下和向右各 30 像素
map.addControl(control);
```

下图是设置偏移量后的对比图



图2-3 控件偏移图

3) 控件类型

在天地图API 中还提供了丰富的类型参数, 通过查阅天地图API 参考类, 选取自己

喜欢的控件类型。以缩放平移控件为例：在天地图 API 中提供了五种不同的控件类型，分别是：

TMAP_NAVIGATION_CONTROL_LARGE：标准的平移缩放控件（包括平移、缩放按钮和滑块）。

TMAP_NAVIGATION_CONTROL_SMALL：仅包含平移和缩放按钮。

TMAP_NAVIGATION_CONTROL_ZOOMANDSLIDER：包含缩放按钮和滑块。

TMAP_NAVIGATION_CONTROL_ZOOM：仅包含缩放按钮。

TMAP_NAVIGATION_CONTROL_ZOOMHORIZONTAL：仅包含缩放按钮，并且缩放按钮是横排。

下图是各种类型所对应的控件图



图2-4 控件类型图

第三章 自定义控件

1. 创建流程

天地图API 中提供的地图控件毕竟是有限的，往往我们在单独开发的过程中希望使用一些自己定义的控件，来达到我们所需要的效果。在上一章中我们提到了所以控件的基类TControl 类，它是所有控件类的基类，可以通过此类来自定义控件，所有控件均包含Control 类的属性、方法和事件。要创建一个自定义控件，需要完成以下工作：

1) 创建自定义控件

在天地图API中创建一个自定义控件，要调用THtmlElementControl类，此控件是向地图上添加自定义的控件。 此类继承TControl，拥有TControl的所有方法。THtmlElementControl的构造函数是：THtmlElementControl(div)，div:可以是一个DIV的ID号，也可以是DIV的对象。

例如：

```
var mapTypeStyle = document.getElementById("mapTypeStyle");  
var mapTyleControl = new THtmlElementControl(mapTypeStyle);//创建了一个自定义的控件。
```

2) 初始化自定义控件

创建完成的自定义控件，只是完成了创建的工作，我们对于它的位置，颜色，透明度等，都没有设置，通过TControl 类的方法，可以设置这些参数。

例如：

```
mapTyleControl.setRight(10);  
mapTyleControl.setTop(10);//把自定义控件放置到离右边和顶部各 10 像素的位置。
```

3) 添加自定义控件

添加自定义控件和添加常用控件一样，调用map.addControl()方法即可。

2. 示例展示

下面通过两个实例，来详细讲解自定义控件的完整过程。

1) 自定义样式地图类型控件

```
<!DOCTYPE html>
```

```

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

    <title> 自定义样式地图类型控件</title>

    <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

    <script>

        var map, zoom=12;

        function onLoad() {

            map = new TMap("mapdiv"); //初始化地图

            map.centerAndZoom(new TLngLat(118.79125, 32.061), zoom);

            //设置显示地图的中心,点和级别

            map.enableHandleMouseScroll(); //允许鼠标滚轮缩放地图

            var mapTypeStyle = document.getElementById("mapTypeStyle");

            var mapTyleControl = new THtmlElementControl(mapTypeStyle);

            //创建自定义地图类型控件

            mapTyleControl.setRight(10);

            mapTyleControl.setTop(10);

            map.addControl(mapTyleControl); 添加地图类型控件

        };

        function switchingMapType(obj) {

            switch(obj.value) {

                case "TMAP_NORMAL_MAP": setNormal();

                    break;

                case "TMAP_SATELLITE_MAP": setSatellite();

                    break;

                case "TMAP_HYBRID_MAP": setHybrid();

                    break;

                case "TMAP_TERRAIN_MAP": setTerrain();

            }

        }

    }

</script>


```

```

        break;

        case "TMAP_TERRAIN_HYBRID_MAP":setTerrainHybrid();

        break;  }

    }

    function setNormal() {

        map.setMapType(TMAP_NORMAL_MAP);

        }//地图

    function setSatellite() {

        map.setMapType(TMAP_SATELLITE_MAP);

        }//卫星

    function setHybrid() {

        map.setMapType(TMAP_HYBRID_MAP);

        }//卫星混合

    function setTerrain() {

        map.setMapType(TMAP_TERRAIN_MAP);

        }//地形

    function setTerrainHybrid() {

        map.setMapType(TMAP_TERRAIN_HYBRID_MAP);

        }//地形混合

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

    <div id="mapdiv" style="position:absolute;width:100%;height:100%"></div>

    <div id="mapTypeStyle">

        <select id="mapTypeSelect" onChange="switchingMapType(this);">

            <option value="TMAP_NORMAL_MAP">地图</option>

            <option value="TMAP_SATELLITE_MAP">卫星</option>

            <option value="TMAP_HYBRID_MAP">卫星混合</option>

```

```

        <option value="TMAP_TERRAIN_MAP">地形</option>

        <option value="TMAP_TERRAIN_HYBRID_MAP">地形混合</option>

    </select>

</div>

</body>

</html>

```



图3-1 自定义样式地图类型控件

2) 自定义样式缩放控件

```

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

    <title>自定义缩放控件</title>

    <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

    <script>

        var map,zoom=12;

        function onLoad(){

            map=new TMap("mapdiv");//初始化地图

```

```

        map.centerAndZoom(new TLatLng(118.79125,32.061),zoom); //设置显示地图的中心点
和级别

        map.enableHandleMouseScroll(); //允许鼠标滚轮缩放地图

        var zoomInObj = document.getElementById("zoomIn");

        var zoomInControl = new THtmlElementControl(zoomInObj);

        zoomInControl.setRight(50);

        zoomInControl.setTop(10);

        map.addControl(zoomInControl); //创建并添加放大控件

        var zoomOutObj = document.getElementById("zoomOut");

        var zoomOutControl = new THtmlElementControl(zoomOutObj);

        zoomOutControl.setRight(5);

        zoomOutControl.setTop(10);

        map.addControl(zoomOutControl); //创建并添加缩小控件

    };

</script>

</head>

<body onload="onLoad()" style="margin:0px">

    <div id="mapdiv" style="position:absolute;width:100%;height:100%"></div>

    <div
        id="zoomIn"
        style="font-size:12px;border:solid
        2px
        blue;background:#fff;padding:2px;line-height:15px;cursor:pointer;"
        onclick="map.zoomIn();"> 放 大
    </div>

    <div
        id="zoomOut"
        style="font-size:12px;border:solid
        2px
        blue;background:#fff;padding:2px;line-height:15px;cursor:pointer;"
        onclick="map.zoomOut();"> 缩 小
    </div>

</body>

</html>

```




图3-2 自定义样式缩放控件

第四章 覆盖物

1. 覆盖物基本类库

叠加物或覆盖到地图上的内容，我们统称为地图覆盖物。如标注、矢量图形元素(折线、多边形、矩形、椭圆、圆)、信息窗口等。覆盖物拥有自己的地理坐标，当您拖动或缩放地图时，它们会相应的移动。在天地图API 中内置了许多覆盖物类。

1) TOverlay 类

叠加层类都直接或间接继承于此基类。如果希望在地图上显示自定义的叠加层对象类型，可以继承这一基类。

2) TLabel 类

表示地图上的文本标注，根据给定的参数设置文本标注的内容、文本标注位置的偏移量、标注的地理位置。

3) TMarker 类

表示地图上一个图像标注，改变参数可以更改标注的地理位置和标注所用的图标对象。

4) TPolyline 类

表示折线的地图叠加层，通过坐标数组创建一条折线，通过相应的参数设置折线的颜色、宽度（以像素为单位）、透明度（范围0-1 之间）、样式。

5) TPolygon 类

表示一个多边形覆盖物，同样是由坐标数组构建，可以设置边线颜色、填充颜色（当参数为空时，折线覆盖物将没有填充效果）、边线的宽度（以像素为单位）、边线透明度（范围0-1之间）、填充的透明度（范围0-1之间）、边线的样式（solid或dashed）。

6) TRect 类

表示地图上的一个矩形图形对象，通过 TBounds 类指定坐标范围建立一个矩形，TBounds(TLngLat:lngmin,TLngLat:latmin,TLngLat:lngmax,TLngLat:latmax)，lngmin：经度最小值，latmin：纬度最小值，lngmax：经度最大值，latmax：纬度最大值。通过参数可以设置矩形边框颜色、填充颜色、矩形边框的宽度（以像素为单位）、矩形边框和填充的透明度（范围0-1 之间）、矩形边框的样式（solid 或dashed）。

7) TEllipse 类

表示地图上的椭圆覆盖物，构造形式和参数属性类似于矩形覆盖物。

8) TCircle 类

表示地图上的圆覆盖物，通过设置中心点经纬度坐标和圆的半径来确定一个圆形覆盖物，设置参数更改圆形的边线颜色、填充颜色、边线的宽度、边线透明度、填充的透明度、边线的样式。

9) TInfoWindow 类

表示地图上包含信息的窗口，信息窗口的位置为地理坐标。默认情况下在地图上打开的信息窗底端的尖角将指向其地理坐标，在标注上打开的信息窗底端尖角的位置取决于标注所用图标的infoWindowOffset 属性值，可以为信息窗添加偏移量来改变默认位置。

2. 基本类库的调用

下面完整的代码详细讲解上述覆盖物类的使用。

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

    <title>添加覆盖物</title>

    <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

    <script>

        var map,zoom=12;

        function onLoad()

        {

            map=new TMap("mapdiv");//初始化地图

            map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);//设置显示地图的中心,点

            和级别

            map.enableHandleMouseScroll();//允许鼠标滚轮缩放地图
```

```

var marker = new TMarker(new TLatLng(118.89125,32.161));//创建标注对象
map.addOverlay(marker);//向地图上添加标注
var label=new TLabel({
    text: "<a href='http://www.tianditu.com'
target='_blank'>http://www.tianditu.com</a>",
    offset: new TPixel(0, 0),
    position: new TLatLng(118.49125, 32.161)
})//创建文本注记
map.addOverlay(label);//添加文本注记
var points = [];
points.push(new TLatLng(118.79125,32.162));
points.push(new TLatLng(118.79125,32.061));
points.push(new TLatLng(118.59123,32.164));
points.push(new TLatLng(118.69123,32.066));
var line = new TPolyline(points,{strokeColor:"red", strokeWeight:6, strokeOpacity:1});//创

```

建线

```

map.addOverlay(line);
var points1 = [];
points1.push(new TLatLng(118.77125,32.162));
points1.push(new TLatLng(118.79125,32.161));
points1.push(new TLatLng(118.69123,32.064));
points1.push(new TLatLng(118.60123,32.066));
polygon = new TPolygon(points1,
    {strokeColor:"blue", strokeWeight:6, strokeOpacity:0.5, fillOpacity:0.5});//创建

```

面对象

```

map.addOverlay(polygon);
var config = {
    strokeColor: "black",//折线颜色

```

果

```
fillColor: "#FFFFFF", //填充颜色。当参数为空时, 折线覆盖物将没有填充效  
  
strokeWeight: "5px", //折线的宽度, 以像素为单位  
opacity: 0.5, //折线的透明度, 取值范围0 - 1  
strokeStyle: "solid" //折线的样式, solid 或 dashed  
};  
  
var bounds = new TBounds(118.79125, 32.061, 118.89125, 32.161);  
  
var rect = new TRect(bounds, config);  
  
map.addOverLay(rect);  
  
var config1 = {strokeColor: "green", //圆边线颜色  
  
fillColor: "#FFFFFF", //填充颜色  
  
strokeWeight: "5px", //圆边线线的宽度, 以像素为单位  
  
strokeOpacity: 0.5, //圆边线线的透明度, 取值范围0 - 1  
  
fillOpacity: 0.7, //填充的透明度, 取值范围0 - 1  
  
strokeStyle: "solid" //圆边线线的样式, solid 或 dashed  
};  
  
var bounds1 = new TBounds(118.89125, 32.031, 118.95125, 32.161); //创建椭圆对象  
  
var ellipse = new TEllipse(bounds1, config1); //向地图上添加椭圆  
  
map.addOverLay(ellipse);  
  
var config2 = {  
  
strokeColor: "blue", //圆边线颜色  
  
fillColor: "#FFFFFF", //填充颜色。  
  
strokeWeight: "5px", //圆边线线的宽度, 以像素为单位  
  
strokeOpacity: 0.5, //圆边线线的透明度, 取值范围0 - 1  
  
fillOpacity: 0.5, //填充的透明度, 取值范围0 - 1  
  
strokeStyle: "solid" //圆边线线的样式, solid 或 dashed  
};  
  
var circle = new TCircle(new TLatLng(118.88125, 32.001), 5000, config2); // 定义该圆的显
```

示区域

```
map.addOverLay(circle);//向地图上添加圆  
  
var lnglat=new TLngLat(118.79125,31.961);//信息窗口地理位置  
  
var infoWin=new TInfoWindow(lnglat,new TPixel([0,-34]));//创建信息窗口对象 TPixel  
为信息窗位置偏移值  
  
infoWin.setLabel("添加的信息窗口");//设置信息窗口要显示的内容  
  
map.addOverLay(infoWin);//向地图上添加信息窗口  
  
};
```

```
</script>
```

```
</head>
```

```
<body onload="onLoad()" style="margin:0px;">
```

```
<div id="mapdiv" style="position:absolute;width:100%;height:100%"></div>
```

```
</body>
```

```
</html>
```

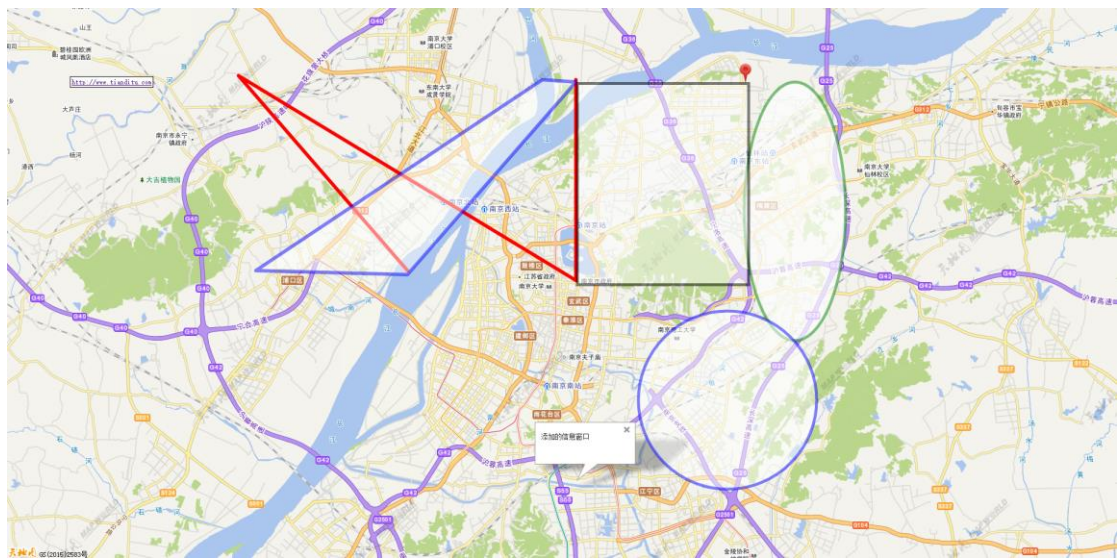


图4-1 覆盖物示意图

3. 自定义覆盖物

地图覆盖物和地图控件一样，在天地图 API 中，不仅提供了简单的构造函数，还提供了丰富的属性、方法和事件。可以对覆盖物添加更多的功能，使覆盖物变的可编辑，

能够更加灵活的使用。具体内容参考天地图API 类库。下面通过几个简单的例子来讲解方法和事件的调用。

1) 编辑方法

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

    <title>编辑线</title>

    <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

    <script>

        var map,zoom=12;

        var line=null;

        function onLoad()

        {

            map=new TMap("mapdiv");

            //初始化地图

            map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);

            //设置显示地图的中心点和级别

            map.enableHandleMouseScroll();

            //允许鼠标滚轮缩放地图

            var points = [];

            points.push(new TLngLat(118.79825,32.062));

            points.push(new TLngLat(118.79425,32.064));

            points.push(new TLngLat(118.79623,32.066));

            points.push(new TLngLat(118.79123,32.068));

            points.push(new TLngLat(118.79322,32.070));

            line = new TPolyline(points,{strokeColor:"red", strokeWeight:6, strokeOpacity:1});//

            创建线
```

```

        map.addOverLay(line);
    }
</script>

</head>

<body onload="onLoad()" style="margin:0px;">

<div style="position:absolute;right: 50px;">

    <input type="button" value="启动编辑" onclick="line.enableEdit();">

    <input type="button" value="禁止编辑" onclick="line.disableEdit();">

</div>

<div id="mapdiv" style="position:absolute;width:100%;height:100%;top: 30px;"></div>

</body>

</html>

```

上述代码中调用了 *enableEidt* 和 *disableEidt* 方法使折线具有了可编辑性。下图是折线的两种状态，左边是未编辑的折线，右边是启用编辑后的折线。

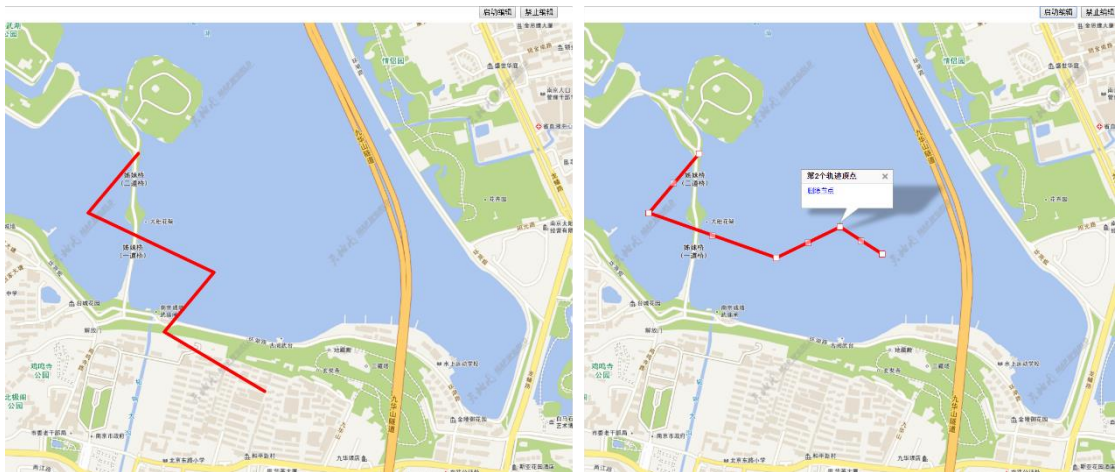


图4-2 编辑线示意图

2) 用户自定义窗口信息

```

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

```

```

<title>用户自定义窗口信息</title>

<script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

<script>

    var map,zoom=12;

    function onLoad()

    {

        map=new TMap("mapdiv");

        //初始化地图

        map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);

        //设置显示地图的中心,点和级别

        map.enableHandleMouseScroll();

        //允许鼠标滚轮缩放地图

        marker = new TMarker(new TLatLng(118.79125,32.061));//创建标注对象

        map.addOverLay(marker);//向上地图上添加标注

        TEvent.addListener(marker,"click",onClick);//注册标注的点击事件

    }

    function onClose(){

        map.removeOverLay(customerWinInfo);

    }

    function onClick(){

        var html=[];

        html.push('<div style="background:#CCCC99;height:20px;color:#000;width:135px;">');

        html.push('<span style="width:100px;float:left;margin-left:2px;background:"> 自定义信

息窗口</span>' +

            '<span style="width:30px;float:right;margin-right:2px;" onclick="onClose();">

关闭</span>');

        html.push('</div>');

        html.push('<div id="deliver-legend-ctrl" style="background:#fff;border:1px solid

```



```
#C0C0C0;">');
```

```
html.push('<table cellspacing="0" cellpadding="0" style="width:130px;border:1px solid
```

```
#ff0000;">');
```

```
html.push('<tr align="center" style="height:10px;">');
```

```
html.push('<td></td>');
```

```
html.push('<td><a href="javascript:void(0);"></a></td>');
```

```
html.push('</tr>');
```

```
html.push('    <tr align="center">');
```

```
html.push('        <td> 经度:</td>');
```

```
html.push('        <td>'+marker.getLngLat().getLng()+ '</td>');
```

```
html.push('    </tr>');
```

```
html.push('    <tr align="center">');
```

```
html.push('        <td> 纬度:</td>');
```

```
html.push('        <td>'+marker.getLngLat().getLat()+ '</td>');
```

```
html.push('    </tr>');
```

```
html.push('    <tr style="height:10px;">');
```

```
html.push('        <td></td>');
```

```
html.push('        <td><a href="javascript:void(0);"></a></td>');
```

```
html.push('    </tr>');
```

```
html.push('</table>');
```

```
html.push('</div>');
```

```
var config = {
```

```
    offset:new TPixel(0,0),
```

```
    position:marker.getLngLat()
```

```
};
```

```
customerWinInfo=new TLabel(config);
```

```
customerWinInfo.setTitle("");
```

```
customerWinInfo.setLabel(html.join("));
```

```

customerWinInfo.getObject().style.zIndex = 10000;

map.addOverLay(customerWinInfo);

var obj = customerWinInfo.getObject();

var width = parseInt(obj.offsetWidth);

var height = parseInt(obj.offsetHeight);

var icon = this.getIcon();

var anchor_icon = icon.getAnchor();

var pixel = new TPixel(width/2,height/2-anchor_icon[1]);

customerWinInfo.setOffset(pixel);

}

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

    <div id="mapdiv" style="position:absolute;width:100%;height:100%"></div>

</body>

</html>

```

上述代码实现了点击标注后，触发标注点击事件，弹出一个自定义的信息窗口，里面是该标注点的经纬度坐标。



图4-3 自定义信息窗口图

3) 聚合marker

用来解决加载大量标注点到地图上产生覆盖现象的问题，并提高性能。代码如下：

```

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

    <title> 聚合 marker</title>

    <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

    <script type="text/javascript" src="http://api.tianditu.com/js/maptools.js"></script>

    <script>

        var map, zoom=5, markerClusterer;

        function onLoad()

        {

            map=new TMap("mapdiv");//初始化地图

            map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);

            //设置显示地图的中心点和级别

            map.enableHandleMouseScroll();//允许鼠标滚轮缩放地图

            map.enableContinuousZoom();

            //启用连续缩放功能

            var MAX = 200;

            var markers = [];

            for (var i = 0; i < MAX; i++) {

                var lnglat = new TLatLng(Math.random() * 40 + 85, Math.random() * 30 + 21);

                markers.push(new TMarker(lnglat));

            }

            var config = {markers:markers};

            markerClusterer = new TMarkerClusterer(map,config);

            //最简单的用法, 生成一个marker 数组, 然后调用markerClusterer 类即可。

        }

    </script>

```

</head>

<body onload="onLoad()" style="margin:0px;">

<div id="mapdiv" style="position:absolute;width:100%;height:100%"></div>

</body>

</html>

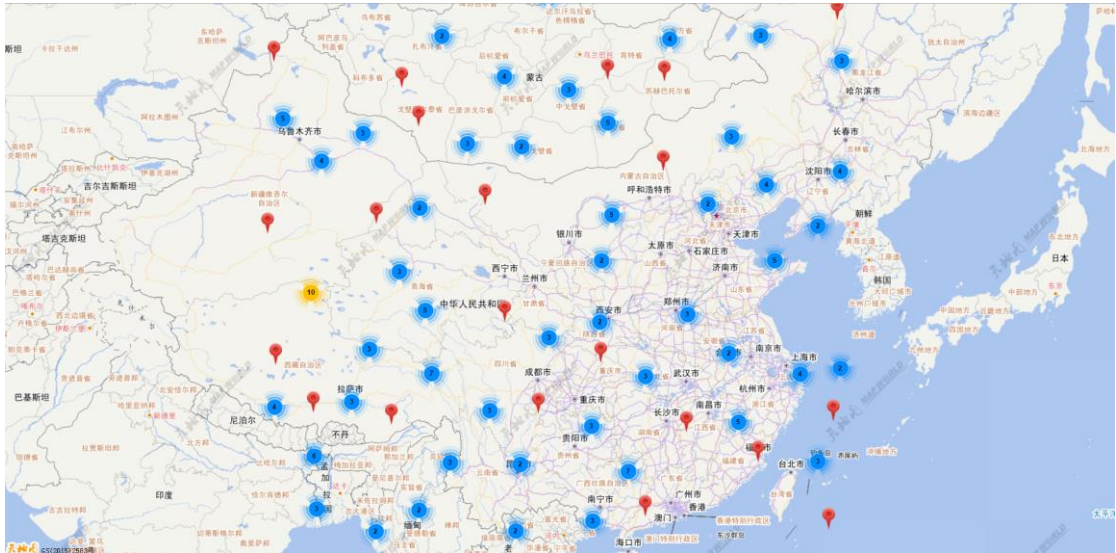


图4-4 初始聚合图

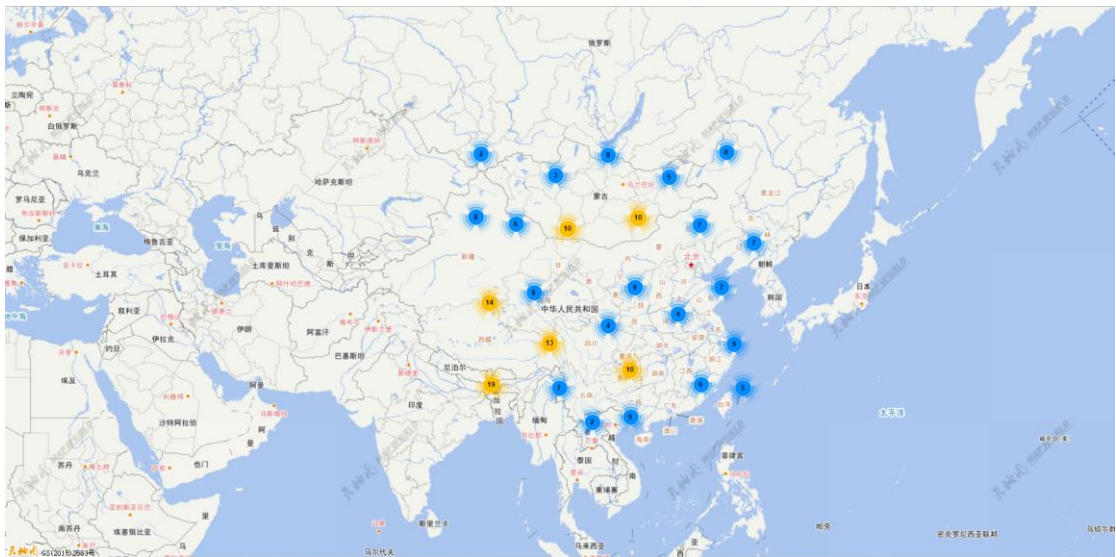


图4-5 聚合后标注图

第五章 事件

JavaScript 使我们有能力创建动态页面。事件是可以被 JavaScript 侦测到的行为。网页中的每个元素都可以产生某些可以触发 JavaScript 函数的事件。比方说，我们可以在用户点击某按钮时产生一个 `onClick` 事件来触发某个函数。事件在 HTML 页面中定义。

1. 关于事件

在天地图API中也定义了许多基本的事件。如：点击事件、拖拽事件、滑动事件等。在天地图API中调用这些事件，还需要通过 `TEvent` 类。此类用于注册事件处理程序和触发自定义事件。`TEvent` 类提供了很多静态方法。

1) `TEvent.addListener(source:Object, event:String, handler:Function)`

为源对象 (`source`) 的自定义事件 (`event`) 注册事件处理程序 (`handler`)，返回一个可用于最终注销处理程序的句柄。事件处理程序的 `this` 指向源对象(`source`)。

2) `TEvent.removeListener(handle:TEventListener)`

删除使用 `addListener()` 注册的事件处理程序。

3) `TEvent.clearListeners(source:Object || Node, event:String)`

删除使用 `addListener()` 在指定对象 (`source`) 上为指定事件 (`event`) 注册的所有事件处理程序。

4) `TEvent.trigger(source:Object, event:String, args:Array)`

在源对象 (`source`) 上触发自定义事件 (`event`)，可选参数作为自定义事件调用函数的参数传递。

5) `TEvent.bind(source:Object, event:String, object:Object, method:Function)`

将指定对象 (`object`) 的方法 (`method`) 调用注册为源对象 (`source`) 自定义事件 (`event`) 的事件处理程序，返回一个可用于最终注销处理程序的句柄。

6) `TEvent.getCallback(object:Object, method:Function)`

返回调用指定对象 (`object`) 上的方法 (`method`) 的闭包。

7) TEvent.deposeNode(object:Object)

删除指定节点及其子节点上注册的所有事件处理程序，用来清除节点，防止内存溢出。

8) TEvent.cancelBubble(event:String)

中止事件处理程序的执行以取消事件冒泡，并返回事件处理结果false。

9) TEvent.returnTrue(event:String)

中止事件处理程序的执行以取消事件冒泡，并返回事件处理结果true。

2. 简单示例

下面通过三个例子简单的介绍一下点击事件、拖拽事件和滑动事件。

1) 点击事件

代码示例:

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title>注册点击事件</title>

<script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

<script>

    var map,zoom=12,mapclick;

    function onLoad()

    {

        map=new TMap("mapdiv");

        //初始化地图

        map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);

        //设置显示地图的中心,点和级别

        map.enableHandleMouseScroll();
```

```

        //允许鼠标滚轮缩放地图
    }

    function addMapClick()
    {
        removeMapClick();

        //移除地图的点击事件

        mapclick = TEvent.addListener(map,"click",function(p){

            var lnglat = map.fromContainerPixelToLngLat(p);//将像素坐标转换成经纬度坐标

            alert(lnglat.getLng()+" "+lnglat.getLat());

        }); //注册地图的点击事件
    }

    function removeMapClick()
    {
        TEvent.removeListener(mapclick);

        //移除地图的点击事件
    }

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

<div id="">

    地图点击事件:

    <input type="button" value="注册" onClick="addMapClick();"/>

    <input type="button" value="移除" onClick="removeMapClick();"/>

</div>

<div id="mapdiv" style="position:absolute;width:100%;height:100%;top:30px"></div>

</body>

</html>

```

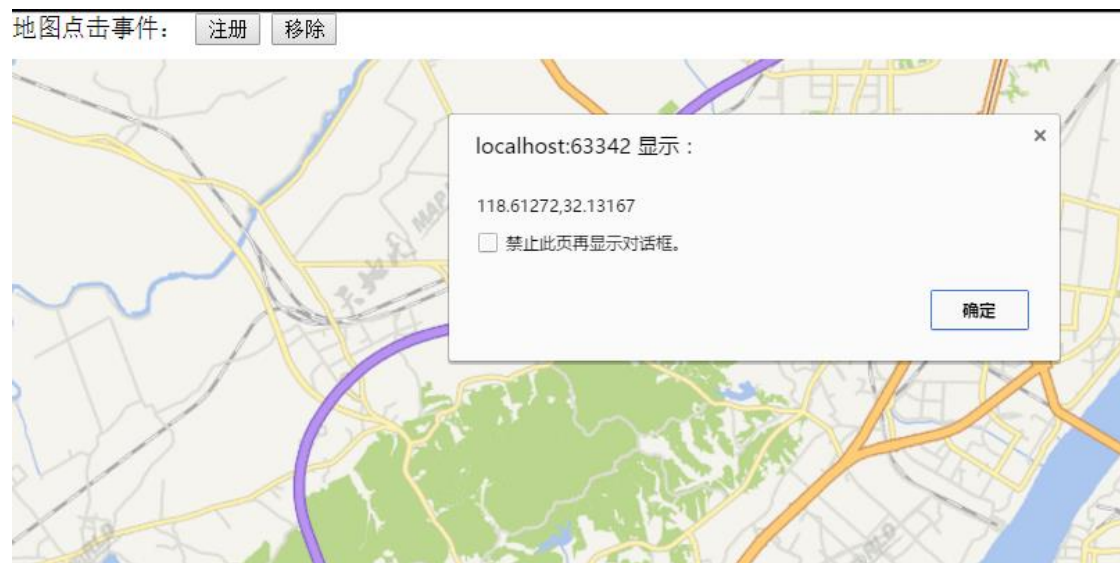


图5-1 点击事件

2) 拖拽事件

代码示例:

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

```
<title>拖拽地图后获取中心点经纬度</title>
```

```
<script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>
```

```
<script>
```

```
var map,zoom=12,mapmoveend;
```

```
function onLoad()
```

```
{
```

```
map=new TMap("mapdiv");
```

```
//初始化地图
```

```
map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);
```

```
//设置显示地图的中心,点和级别
```

```
map.enableHandleMouseScroll();
```

```
//允许鼠标滚轮缩放地图
```



```

    }

    function addMapMoveend()

    {

        removeMapMoveend();//移除地图的移动停止事件,防止多次注册事件

        mapmoveend = TEvent.addListener(map,"moveend" function(lnglat){

            alert(lnglat.getLng()+" "+lnglat.getLat());

        });

    }

    function removeMapMoveend()

    {

        TEvent.removeListener(mapmoveend);

    }//移除地图的移动停止事件

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

<div id="">

    地图停止事件:

    <input type="button" value="注册" onClick="addMapMoveend();"/>

    <input type="button" value="移除" onClick="removeMapMoveend();"/>

</div>

<div id="mapdiv" style="position:absolute;width:100%;height:100%;top:30px"></div>

</body>

</html>

```

地图停止事件:

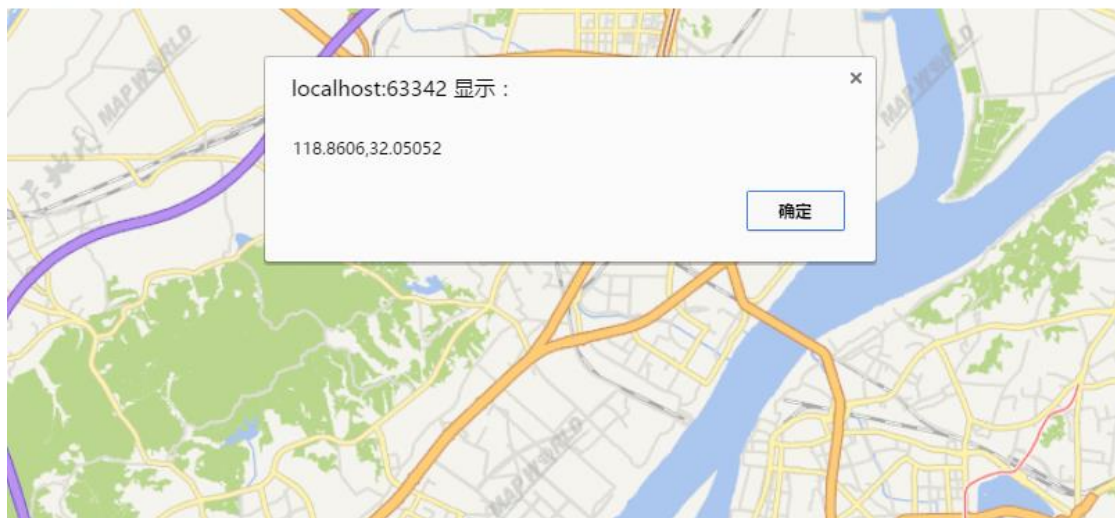


图5-2 拖拽事件

3) 滑动事件

代码示例:

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

```
<title> 鼠标在地图上滚动时触发事件</title>
```

```
<script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>
```

```
<script>
```

```
var map,zoom=12,mapmousemove;
```

```
function onLoad()
```

```
{
```

```
map=new TMap("mapdiv");
```

```
//初始化地图
```

```
map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);
```

```
//设置显示地图的中心,点和级别
```

```
map.enableHandleMouseScroll();
```

```
//允许鼠标滚轮缩放地图
```

```

    }

    function addMapMousemove(){

        removeMapMousemove();//移除掉以前的注册事件

        mapmousemove = TEvent.addListener(map,"mousemove",function(p){

            var lnglat = map.fromContainerPixelToLngLat(p);//将像素坐标转换成经纬度坐标

            document.getElementById("info").value = lnglat.getLng()+","+lnglat.getLat();

        });

    }//注册鼠标在地图上的滑动事件

    function removeMapMousemove(){

        TEvent.removeListener(mapmousemove);

    }//移除鼠标在地图上的滑动事件

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

<div id="">

    鼠标在地图上的滑动事件:

    <input type="button" value="注册" onClick="addMapMousemove();"/>

    <input type="button" value="移除" onClick="removeMapMousemove();"/>

    鼠标所在位置坐标:

    <input type="text" id="info" value=""/>

</div>

<div id="mapdiv" style="position:absolute;width:100%;height:100%;top:30px"></div>

</body>

</html>

```



图5-3 滑动事件

第六章 地图图层

1. 自定义图层

在天地图API 中,通过TTileLayer 类实现在地图上叠加自定义的地图图层。TTileLayer 类的构造函数是 TTileLayer(opts:TTileLayerOptions)。

1) 相关参数

a isPng

判断图块的格式是否为 PNG。默认为false。

b opacity

设置图块的透明度 (0.0-1.0)。默认值为 1.0 不透明。

c tileUrlTemplate

指定图块网址模板, 该模板可以针对每个图块请求而展开, 以根据现有的图块坐标系引用唯一的图块。将模板置于 TTileLayer 构造函数中可允许您使用此坐标系动态检索图块。模板的格式应该为: `http://host/tile?x={X}&y={Y}&z={Z}.png`, 其中 X 和 Y 分别指纬度和经度图块坐标, Z 指缩放级别, 比如: `http://host/tile?x=3&y=27&z=5.png`。

d errorImg

没图时候叠加层应该显示的图, 默认为空。

e zIndex

叠加层的zIndex 顺序。

f minResolution

此图块层的最低缩放级别。

g maxResolution

此图块层的最高缩放级别。

2) TTileLayer 类方法

a getObject() 获取包含图片层的Div 对象

b dispose() 销毁创建的layer 对象

c refresh() 重新加载所有添加到此 TTileLayer 的可见图块

- d `hide()` 隐藏此叠加层使之不可见, 但保留它在叠加层堆栈中的位置
- e `isHidden()` 如果图块层叠加层不可见, 则返回 `true`。否则, 返回 `false`
- f `show()` 显示先前不可见的 `TTileLayer`
- g `setOpacity()` 设置透明度
- h `setGetTileUrl(function:Function)` 设置取图函数 (`function(x,y,z){return some_url+x+y+z;}`, `x,y,z` 为地图的块号, 返回值为图片的路径)
- i `setZindex()` 设置 `z-index`
- j `getImg(x:Number,y:Number,z:Number)` 返回块号对应的 `html` 对象, 可能是 `div` 或者 `image`, 注意操作完这个对象以后解除所有引用防止内存泄漏 `x` 参数配置横向块号、`y` 参数配置纵向块号、`z` 参数配置缩放等级
- k `getAllImg()` 返回包含所有 `image` 的一个对象, 通过对象的 `id` 可以取到块号, 通过 `getObject()` 可以得到 `image` 或者 `div`

3) 代码示例

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

  <title>添加自定义图层</title>

  <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

  <script>

    var map, zoom=12;

    function onLoad()

    {

      map=new TMap("mapdiv");

      //初始化地图

      map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);

      //设置显示地图的中心点和级别

      map.enableHandleMouseScroll();
```

```

//允许鼠标滚轮缩放地图

var config = {};

config.getTileUrl = function(x,y,z){

    return

    "http://t0.tianditu.cn/img_w/wmts?"+"SERVICE=WMTS&REQUEST=GefTile&VERSION=1.0.0

    &LAYER=img&STYLE=default&TILEMATRIXSET=w&FORMAT=tiles&" + "TILECOL=" + x

    + "&TILEROW=" + y + "&TILEMATRIX=" + z;

}

var lay = new TTileLayer(config);//创建自定义图层对象

lay.setGetTileUrl(config.getTileUrl);//设置取图函数

map.addLayer(lay);//将图层增加到地图上

}

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

    <div id="mapdiv" style="position:absolute;width:100%;height:100%"></div>

</body>

</html>

```

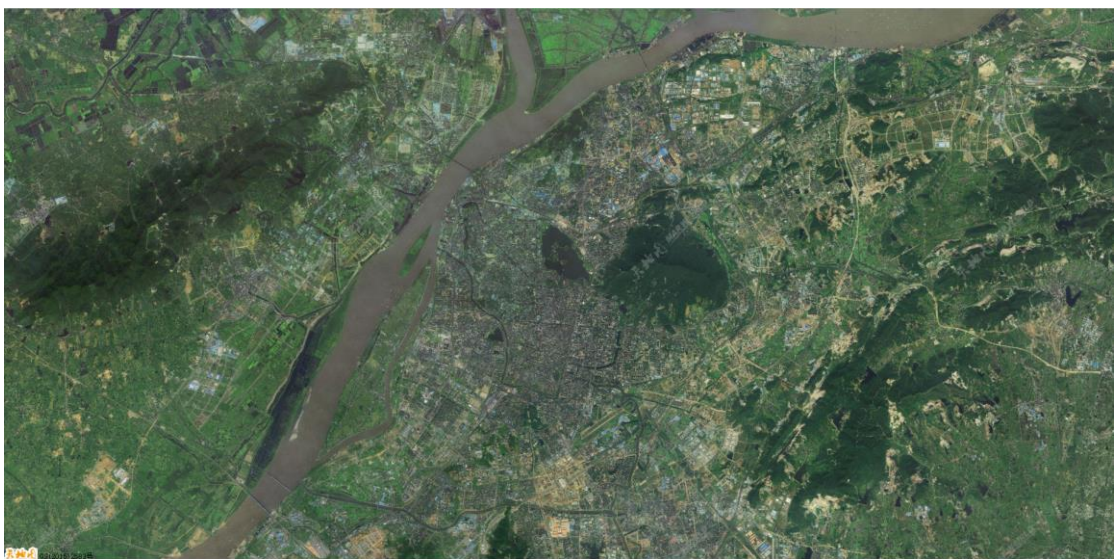


图6-1 自定义图层

2.WMS 图层

在天地图API 中，叠加自定义的WMS 地图图层，需要通过TTileLayerWMS 类来实现，TTileLayerWMS 类的构造函数是TTileLayerWMS (name,url, opts)，Name 是WMS 图层的名称，url 是WMS 的服务地址，opts 是TTileLayerWMS 接口的可选参数（见表6-1）。它还提供了getName ()和loadURL ()方法，前者可以获取WMS 图层的名称，后者可以获取图层的属性信息。

表6-1 WMS 函数接口属性表

属性	类型	说明
REQUEST	string	操作名称。必选
SERVICE	string	服务类型标识符，参数值为wms。必选
VERSION	string	请求服务的版本，参数值为1.1.1。必选
LAYERS	string	用","分隔的多个图层列表。
STYLES	string	每个请求图层的用","分隔的描述样式。必选
SRS	string	地图投影类型。必选
BBOX	string	显示范围（左下角，右上角，坐标值用","分隔）。必选
WIDTH	number	输出地图图片的像素宽。必选
HEIGHT	number	输出地图图片的像素高。必选
FORMAT	string	输出图像的类型，参数值为'image/png'。必选
TRANSPARENT	boolean	输出图像背景是否透明。
BGCOLOR	string	十六进制的背景颜色。
EXCEPTIONS	string	异常处理文档，参数值为application/vnd.ogc.se_xml。
TILED	boolean	是否为栅格瓦片，true 表示为栅格瓦片。注：接口目前只支持瓦片的形式。

下面是一个简单的示例，实现叠加超图WMS 图层。

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

  <title>叠加超图WMS 图层</title>

  <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

  <script>

    var map,zoom=12,wmsLayer;
```



```

var listener = null;

function onLoad()
{
    map = new TMap("mapdiv",{projection: "EPSG:4326"});

    //初始化地图

    map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);

    //设置显示地图的中心点和级别

    map.enableHandleMouseScroll();

    //允许鼠标滚轮缩放地图

    map.setMapType(TMAP_SATELLITE_MAP);

    //设置地图显示为卫星
}

function getWMS(layers,url,config)
{
    if(wmsLayer)
    {
        map.removeLayer(wmsLayer);
    }

    wmsLayer = new TTileLayerWMS("layers",url,config);//创建WMS 图层对象

    map.addLayer(wmsLayer);//将WMS 图层添加到地图上
}

function addSuperMapLayer(layers,url)
{
    var config = {

        REQUEST:"GetMap", //操作名称

        VERSION:"1.1.1", //请求服务的版本

        SERVICE:"WMS", //服务类型标识符

        LAYERS:layers,

```

```

        TRANSPARENT:true, //输出图像背景是否透明

        STYLES:"", //每个请求图层的用","分隔的描述样式

        FORMAT:"image/png", //输出图像的类型

        SRS:map.getCode(), //地图投影类型

        WIDTH:256, //输出地图图片的像素宽

        HEIGHT:256 //输出地图图片的像素高

    };

    getWMS(layers,url,config);

}

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

<div id="toll" style="margin:0px;padding:0px; border: 0px;background: #f4f4ed;position:
absolute;width:100%;height: 100%">

    <input type="button" value="叠加超图WMS 服务图层"

onClick="addSuperMapLayer('China','http://support.supermap.com:8090/iserver/services/map-china400
/wms111/China')"/>

    <input type="button" value="删除超图WMS 服务图层"

onClick="map.removeLayer(wmsLayer);"/>

</div>

<div id="mapdiv" style="position:absolute;width:100%;height:100%;top:30px"></div>

</body>

</html>

```

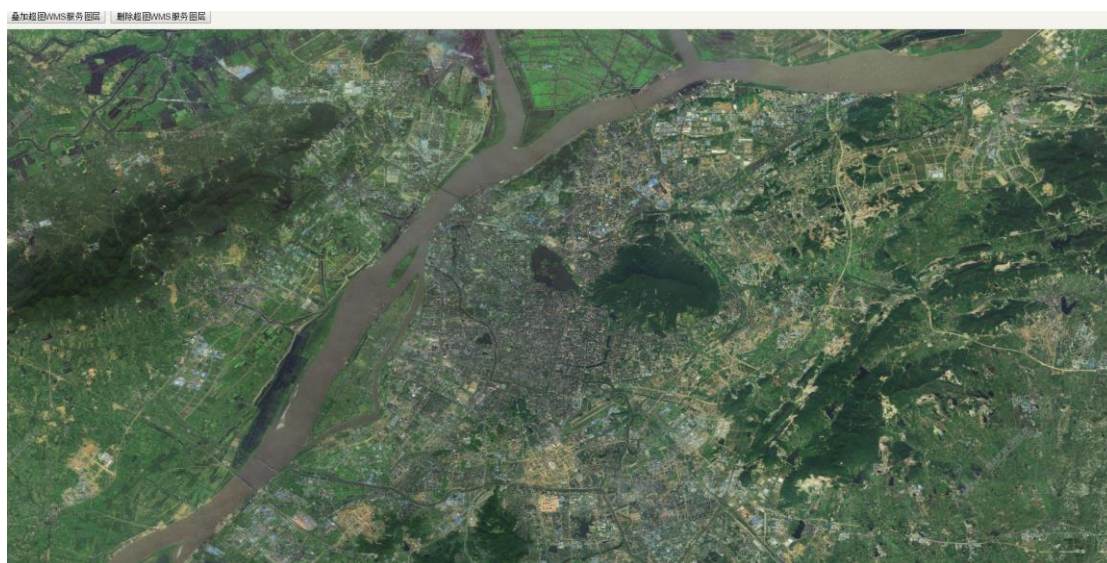


图6-2 WMS 图层叠加前

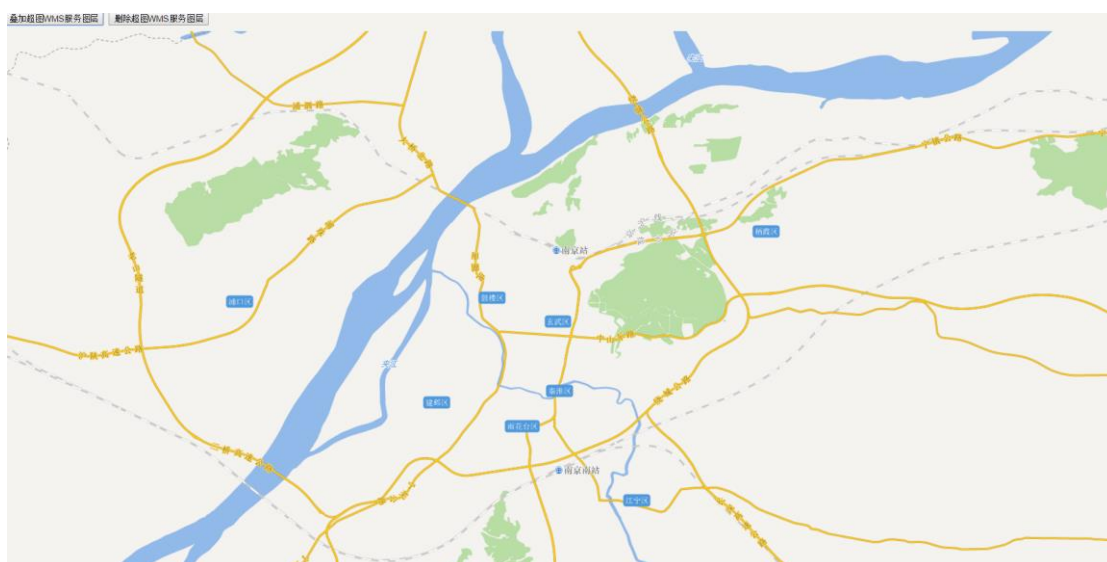


图6-3 WMS 图层叠加后

第七章 地图工具

1. 标注工具

标注工具，用来让用户在地图上标注一个点，可以通过该工具获得用户标点的经纬度位置。`TMarkTool(map:TMap[,opts:TMarkToolOptions])`构造标注工具的函数，设置相应的方法，控制是否开启标注工具，调用`mouseup` 事件可以显示标注点的地理坐标。下面是一个简单的示例：

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

  <title> 标注工具</title>

  <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

  <script>

    var map,zoom=12,marker,markerTool;

    var listener =null;

    function onLoad()

    {

      map =new TMap("mapdiv");           //初始化地图

      map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);           //设置显示地图的中

心点和级别

      map.enableHandleMouseScroll();

      //允许鼠标滚轮缩放地图

      markerTool = new TMarkTool(map);

      //创建标注工具对象

      TEvent.addListener(markerTool,"mouseup",mouseup); //注册标注的mouseup 事件

    }

  </script>

</head>

</html>
```

```

function mouseup(point){

    marker = new TMarker(point);

    map.addOverlay(marker);

    markerTool.close();

}

//鼠标在地图上添加一个标记

function editMarker(){

    if(marker==null)

    {

        alert('请先画点再进行编辑! ');

        return;

    }

    else

    {   marker.enableEdit();

        listener = TEvent.bind(marker,"dragend", marker, function{lnglat){

            TEvent.removeListener(listener);

            alert("当前坐标: "+lnglat.getLng()+" "+lnglat.getLat());

        });

    }

} //启动编辑点标记

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

    <div id="toll" style="margin:0px;padding:0px; border: 0px;background: #f4f4ed;position: absolute;width:100%;height: 100%">

        标注工具

        <input type="button" value="开 启" onClick="markerTool.open();"/>

        <input type="button" value="关 闭" onClick="markerTool.close();"/>

```

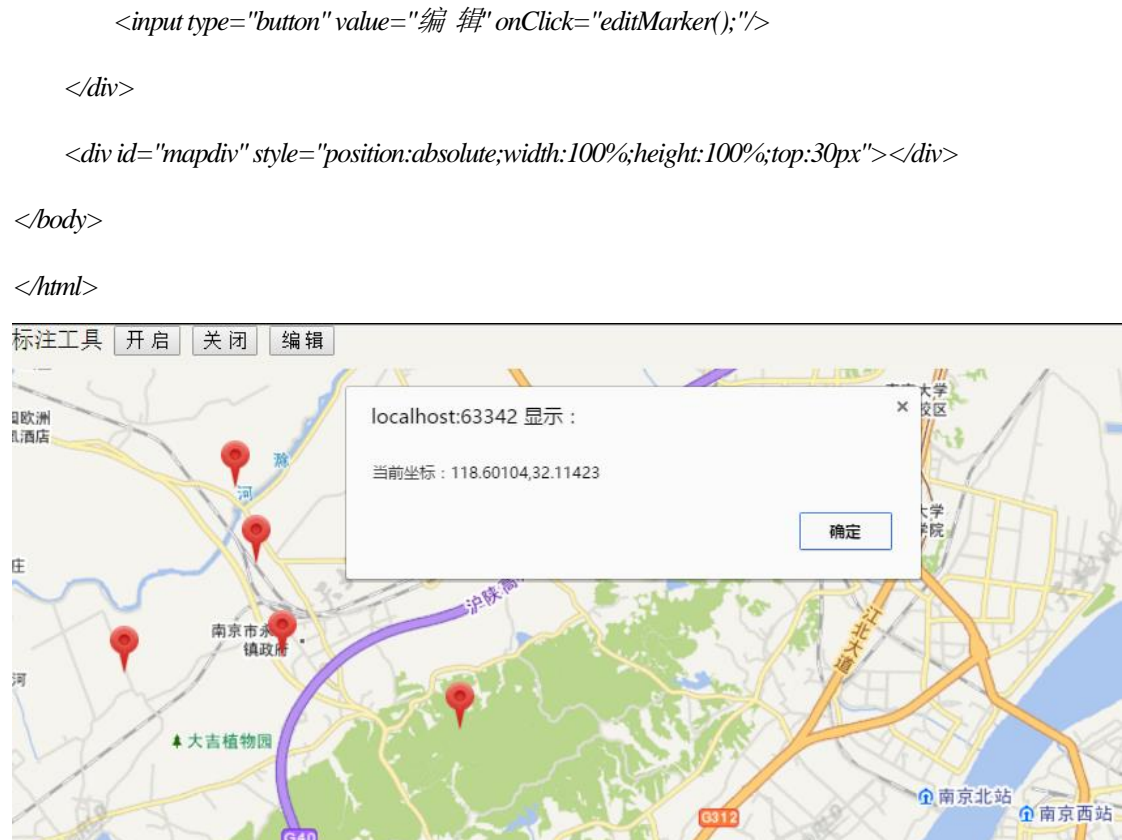


图7-1 标注工具

2. 矩形工具

矩形工具，用来实现在地图上选择一个矩形区域或绘制矩形的功能。矩形工具的构造函数是 `TRectTool(map:Tmap[,opts:TRectToolOptions])`。在天地图API 中，提供了清除、开启和关闭三种方法和 `draw()` 事件，在用户完成拉框后触发该事件。下面是简单的示例：

```

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title>矩形工具</title>

<script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

<script>

var map,zoom=12,rect,rectTool;

```

```

function onLoad()

{

    map=new TMap("mapdiv");

    //初始化地图

    map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);

    //设置显示地图的中心点和级别

    map.enableHandleMouseScroll();

    //允许鼠标滚轮缩放地图

    var config = {

        strokeColor: "blue",//折线颜色

        fillColor: "#FFFFFF", //填充颜色。当参数为空时，折线覆盖物将没有填充效果

        strokeWeight: "5px",//折线的宽度，以像素为单位

        opacity: 0.5,    //折线的透明度，取值范围0 - 1

        strokeStyle: "solid" //折线的样式，solid 或dashed

    };

    rectTool = new TRectTool(map,config);

    //创建矩形工具

    TEvent.addListener(rectTool,"draw",onDrawRect);

    //注册矩形工具绘制完成后的事件

}

function onDrawRect(bounds){

    rect = new TRect(bounds);

    map.addOverLay(rect);

    //添加矩形

    rectTool.close();

    //关闭矩形工具

}

```

```

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

    <div id="toll" style="margin:0px;padding:0px; border: 0px;background: #f4f4ed;position:
absolute;width:100%;height: 100%">

        矩形工具

        <input type="button" value="开启" onClick="rectTool.open();"/>

        <input type="button" value="关闭" onClick="rectTool.close();"/>

    </div>

    <div id="mapdiv" style="position:absolute;width:100%;height:100%;top:30px"></div>

</body>

</html>

```

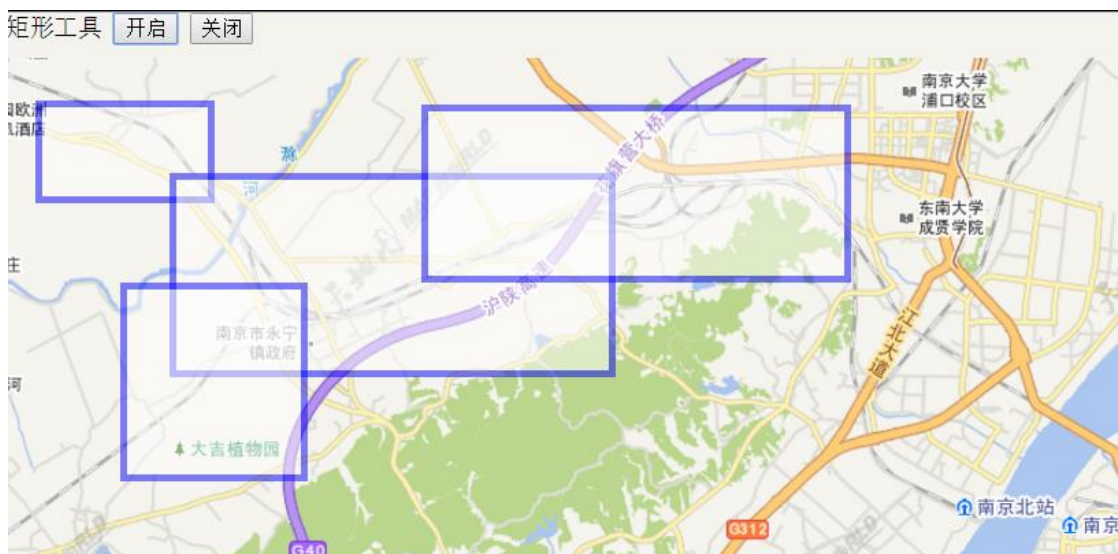


图7-2 矩形工具

3. 折线工具

折线工具，可以通过事件来获取用户绘制的折线，包含测距功能。折线工具的构造函数是 `TPolylineTool(map:TMap[,opts:TPolylineToolOptions])`。通过设置线的参数，改变线的样式。设置相应的方法，控制是否开启折线工具。通过 `draw()` 事件，用户双击完成一次折线后调用该事件，`addpoint()` 事件，用户测距过程中，每次点击底图添加节点时触

发事件。代码示例如下:

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

  <title>折线工具</title>

  <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

  <script>

    var map, zoom=12, lineTool;

    function onLoad()

    {

      map=new TMap("mapdiv");//初始化地图

      map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);

      //设置显示地图的中心点和级别

      map.enableHandleMouseScroll();//允许鼠标滚轮缩放地图

      var config = {

        strokeColor:"blue", //折线颜色

        strokeWeight:"3px", //折线的宽度，以像素为单位

        strokeOpacity:0.5, //折线的透明度，取值范围0-1

        strokeStyle:"solid" //折线的样式，solid 或 dashed

      };

      lineTool = new TPolylineTool(map,config); //创建测距工具对象

      TEvent.addListener(lineTool,"draw",onDrawLine); //注册测距工具绘制完成后的事件

    }

    function onDrawLine(bounds,line,obj) {

      lineTool.close();

    } //关闭测距工具

  </script>
```

```

</head>

<body onload="onLoad()" style="margin:0px;">

    <div id="toll" style="margin:0px;padding:0px; border: 0px;background: #f4f4ed;position:
absolute;width:100%;height: 100%">

        折线工具

        <input type="button" value="开启" onClick="lineTool.open();" />

        <input type="button" value="关闭" onClick="lineTool.close();" />

    </div>

    <div id="mapdiv" style="position:absolute;width:100%;height:100%;top:30px"></div>

</body>

</html>

```



图7-3 折线工具

4. 多边形工具

多边形工具，可以通过事件来获取用户绘制的多边形，包含测面积功能。它的构造函数是 `TPolygonTool(map:TMap[,opts: TPolygonToolOptions])`。上述工具一样都具有类似的方法和事件。完成测面功能的方法是 `getArea(points:TLngLat [])`，可以计算由地理点组成的面积。示例代码如下：

```

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

    <title> 多边形工具</title>

    <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

    <script>

        var map,zoom=12,polygonTool;

        function onLoad()

        {

            map=new TMap("mapdiv");

            //初始化地图

            map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);

            //设置显示地图的中心点和级别

            map.enableHandleMouseScroll();

            //允许鼠标滚轮缩放地图

            var config = {

                strokeColor:"blue",//折线颜色

                fillColor:"#FFFFFF",    //填充颜色。当参数为空时，折线覆盖物将没有填充

                //效果

                strokeWeight:"3px",//折线的宽度，以像素为单位

                strokeOpacity:0.5, //折线的透明度，取值范围0-1

                fillOpacity:0.5          //填充的透明度，取值范围0-1

            };

            polygonTool = new TPolygonTool(map,config);

            //创建测面工具对象

            TEvent.addListener(polygonTool,"draw",onDrawPolygon);

            //注册测面工具绘制完成后的事件

```

```

    }

    function onDrawPolygon(bounds,line)

    {

        polygonTool.close();

        }//关闭测面工具

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

    <div id="toll"style="margin:0px;padding:0px;  border: 0px;background: #f4f4ed;position:

absolute;width:100%;height: 100%">

        多边形工具

        <input type="button" value="开启" onClick="polygonTool.open();" />

        <input type="button" value="关闭" onClick="polygonTool.close();" />

    </div>

    <div id="mapdiv" style="position:absolute;width:100%;height:100%;top:30px"></div>

</body>

</html>

```



图7-4 多边形工具

5. 画圆工具

画圆工具，用来实现在地图上画圆的功能。它的构造函数是 `TCircleTool` (`map:Tmap[,opts:TCircleToolOptions]`)。与覆盖物的圆一样可以通过参数设置圆的样式。也有开启和关闭方法。通过 `draw` () 事件可以得到所绘制圆的半径。示例代码如下：

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

  <title>画圆工具</title>

  <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

  <script>

    var map,zoom=12,circleTool,circle;

    function onLoad()

    {

      map=new TMap("mapdiv");

      //初始化地图

      map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);

      //设置显示地图的中心,点和级别

      map.enableHandleMouseScroll();

      //允许鼠标滚轮缩放地图

      var config = {

        strokeColor:"#000000", //圆边线颜色

        fillColor:"#FFFFFF", //填充颜色。

        strokeWeight:"3px",//圆边线线的宽度，以像素为单位

        strokeOpacity:0.5, //圆边线线的透明度，取值范围0 - 1

        fillOpacity:0.5, //填充的透明度，取值范围0 - 1

        strokeStyle:"solid" //圆边线线的样式，solid 或dashed
```

```

};

circleTool = new TCircleTool(map,config);

//创建画圆工具对象

TEvent.addListener(circleTool,"draw",onDrawCircle);

//注册画圆工具在绘制过程中的事件

TEvent.addListener(circleTool,'drawend',onDrawCircleEnd);

//注册画圆工具绘制完成后的事件

}

function onDrawCircleEnd(circle)

{

    var center = circle.getCenter();

    var radius = circle.getRadius();

    circle = new TCircle(center,radius);

    map.addOverLay(circle);

    circleTool.close();

}

function onDrawCircle(center,radius)

{

    document.getElementById('info').value="圆的半径是 "+parseInt(radius)+" 米";

}

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

    <div id="toll"style="margin:0px;padding:0px; border: 0px;background: #f4f4ed;position:

absolute;width:100%;height: 100%">

        画圆工具

        <input type="button" value="开启" onClick="circleTool.open();"/>

        <input type="button" value="关闭" onClick="circleTool.close();"/>

```

```

        <input type="text" id="info" value=""/>

    </div>

    <div id="mapdiv" style="position:absolute;width:100%;height:100%;top:30px"></div>

</body>

</html>

```



图7-5 画圆工具

第八章 右键菜单示例

1. 创建右键菜单

通过 `TContextMenu` 类，可以在地图上添加自定义内容的右键菜单。使用方法如下：

```
var menu = new TContextMenu();
```

创建出来的菜单里面没有内容，要通过 `TContextMenu` 类的方法来初始化右键菜单。

- 1) `addItem()` 添加菜单项
- 2) `getItem()` 返回指定索引位置的菜单项，第一个菜单项的索引为0
- 3) `removeItem()` 移除菜单项
- 4) `addSeparator()` 添加分隔符
- 5) `removeSeparator()` 移除指定索引位置的分隔符，第一个分隔符的索引为0
- 6) `getItems()` 返回所有的 `TMenuItem`，是一个数组

7) `getAllSeparato()`返回所有分割线，是一个数组

除了`addItem()`添加菜单项，在天地图API 中还提供了 `TMenuItem` 类， 此类可以设置菜单项显示的文字、点击菜单时的回调函数、菜单项的宽度。调用相应的方法，可以启用或关闭菜单项。

2. 右键菜单示例

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

  <title>添加带分割线的右键菜单</title>

  <script src="http://api.tianditu.com/js/maps.js" type="text/javascript"></script>

  <script>

    var map,zoom=12;

    function onLoad()

    {

      map=new TMap("mapdiv");//初始化地图

      map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);

      //设置显示地图的中心点和级别

      map.enableHandleMouseScroll();//允许鼠标滚轮缩放地图

      var menu  = new TContextMenu();//创建右键菜单对象

      var txtMenuItem = [

        {

          text:'放大',

          callback:function(){map.zoomIn()}

        },

        {
```



```

        text:'缩小',

        callback:function(){map.zoomOut()}

    },

    {

        text:'放置到最大级',

        callback:function(){map.setZoom(18)}

    },

    {

        text:'查看全国',

        callback:function(){map.setZoom(4)}

    },

    {

        text:'获得右键点击处坐标',

        isDisable:false,

        callback:function(lnglat){alert(lnglat.getLng()+"," + lnglat.getLat());}

    }

    ];

    for(var i=0;i<txtMenuItem.length; i++)

    {

        var options = new TMenuItemOptions();//创建右键菜单参数接口对象

        options.width = 100;//设置右键菜单的宽度

        menu.addItem(new

TMenuItem(txtMenuItem[i].text,txtMenuItem[i].callback,options)); //添加菜单项

        if(i==1|| i==3){

            menu.addSeparator();//创建分割线

        }

    }

    map.addContextMenu(menu);//添加右键菜单

}

```

```

</script>

</head>

<body onload="onLoad()" style="margin:0px;">

<div id="mapdiv" style="position:absolute;width:100%;height:100%"></div>

</body>

</html>

```



图8-1 右键菜单示例图

第九章 服务

1. 搜索

在天地图API 中, 提供了TLocalSearch 类, 此类用于位置检索、周边检索和范围检索。它的构造函数是TLocalSearch(map,opts)。Map 为地图对象, opts 是可选参数, 可以通过pageCapacity 设置每页的容量和onSearchComplete 设置检索结束后的回调函数。

1) TLocalsearch 类方法

a 检索方式

在天地图中, 提供了根据检索词、根据范围和检索词和根据中心点、半径与检索词的三种检索方式, 具体方法见下表。

方法	返回值	说明
<code>search(keyword:String,type:Number)</code>	<code>none</code>	根据检索词发起检索。 参数说明: <code>keyword</code> : 关键字。 <code>type</code> : 搜索类型1 表示普通搜索;2 表示视野内搜索;4 表示普通建议词搜索;5 表示公交规划建议词搜索;7 表示 纯POI 搜索(不搜公交线);10 表示拉框搜索
<code>searchInBounds(keyword:String, bounds:TBounds)</code>	<code>none</code>	根据范围和检索词发起范围检索。 <code>keyword</code> : 关键字。 <code>bounds</code> : 搜索范围
<code>searchNearby(keyword:String,center:TlngLat, radius:Number)</code>	<code>none</code>	根据中心点、半径与检索词发起周边0 检索。 参数说明: <code>keyword</code> : 关键字。 <code>center</code> : 中心点。。 <code>Radius</code> : 搜索半径

b 检索类型

方法	返回值	说明
<code>setSpecifyAdminCode(code:Number)</code>	<code>none</code>	设置搜索类型。 参数说明: <code>code</code> : 行政区的国标码。
<code>setQueryType(type:Number)</code>	<code>none</code>	设置搜索类型。 参数说明: <code>type</code> : 搜索类型1 表示普通搜索;2 表示视野内搜索;4

		表示普通建议词搜索;5 表示公交规划建议词搜索;7 表示 纯POI 搜索(不搜公交线)。
<code>getQueryType()</code>	<i>number</i>	返回搜索类型。

c 检索结果

方法	返回值	说明
<code>getResults()</code>	<i>TLocalSearchResult</i>	返回最近一次检索的结果。
<code>clearResults()</code>	<i>none</i>	清除最近一次检索的结果。
<code>gotoPage(page:Number)</code>	<i>none</i>	检索特定页面的结果。 参数说明: <i>page</i> : 检索特定页面数值。
<code>setSearchCompleteCallback(fun:Function)</code>	<i>none</i>	设置检索结束后的回调函数。 参数说明: <i>fun</i> : 回调函数。回调函数参数为 <i>TLocalSearchResult</i> , 具体内容详见 <i>TLocalSearchResult</i> 类。

d 页面设置

方法	返回值	说明
<code>setPageCapacity(count:Number)</code>	<i>none</i>	设置每页容量。 参数说明: <i>count</i> : 每页容量。
<code>getPageCapacity()</code>	<i>number</i>	返回每页容量。
<code>firstPage()</code>	<i>none</i>	检索第一页。
<code>nextPage()</code>	<i>none</i>	检索下一页。
<code>previousPage()</code>	<i>none</i>	检索上一页。
<code>lastPage()</code>	<i>none</i>	检索最后一页。
<code>getCountNumber()</code>	<i>number</i>	返回总记录数。
<code>getCountPage()</code>	<i>none</i>	返回共分总页数。
<code>getPageIndex()</code>	<i>string</i>	返回当前页。

2) TLocalSearchResult 类

此类表示 *TLocalSearch* 的检索结果, 没有构造函数, 通过 *TLocalSearch.getResults()* 方法或 *TLocalSearch* 的 *onSearchComplete* 回调函数的参数得到。*TLocalSearch.getResults()* 方法有以下几种:

a getResultType()

返回搜索类型, 1 表示普通搜索 2 表示视野内搜索;4 表示普通建议词搜索;5 表示公交规划建议词搜索;7 表示 纯POI 搜索(不搜公交线)。

b getCount()

返回检索总条数

c *getKeyword()*

返回检索关键词

d *getPois()*

返回点信息，当*getResultType* 值为1 时返回。

数据结构如下:

```
[
  {
    "phone": "", //电话
    "lonlat": "116.313488 39.979416", //坐标
    "address": "北京市海淀区", //地址
    "name": "中关村", //Poi 点名称
    "poiType": "102", //poi 类型 (102 表示公交站, 普通poi 该参数可以不返回)
  },
  ...
]
```

e *getStatistics()*

返回统计信息，当*getResultType* 值为2 时返回。

数据结构如下:

```
{
  "priorityCitys": [ //推荐显示城市
    {
      "count": "38341", //统计数量
      "name": "上海市", //城市名称
      "adminCode": "156310000" //城市国标码
    },
    {
      "count": "37346",
      "name": "北京市",
      "adminCode": "156110000"
    },
    ...
  ],
  "keyword": "饭店", //搜索的关键字
  "countryCount": 1, //搜索的国家数量
  "citysCount": 383, //搜索的城市数量
  "allAdmins": [ //各省包含信息集合
    {
```

```

"count": "82739", //统计数量
"name": "广东省", //省名称
"adminCode": 156440000, //省国标码
"childAdmins": [ //包括各市级集合
    {
        "count": "19038", ///统计数量
        "name": "广州市", //城市名称
        "adminCode": "156440100" //城市国标码
    },
    {
        "count": "16578",
        "name": "深圳市",
        "adminCode": "156440300"
    },
    ...
]
},
{
    "count": "62891",
    "name": "四川省",
    "adminCode": 156510000,
    "childAdmins": [
        {
            "count": "30673",
            "name": "成都市",
            "adminCode": "156510100"
        },
        {
            "count": "4907",
            "name": "绵阳市",
            "adminCode": "156510700"
        },
        ...
    ]
},
...
],
"provinceCount": 34 //搜索的省份数量
}

```

f *getArea()*

返回行政区省信息，当*getResultType* 值为3 时返回。

数据结构如下:

```
{
  "level": "11", //显示级别
  "lonlat": "116.38094,39.923615", //定位中心点坐标
  "name": "北京市", //名称
  "points": [ //行政区边界坐标
    {
      "region": "116.625875 41.058853,116.68501 41.040147,116.692716 41.019134,116.677056 40.998562,..."
    }
  ],
  "type": "1" //1 表示正常区域, 2 表示特殊区域
}
```

g **getSuggests()**

返回建议词信息, 当**getResultType** 值为4 时返回。

数据结构如下:

```
[
  {
    "address": "", //地址
    "name": "北京市", //名称
    "gbCode": "000" //国标码
  },
  {
    "address": "北京市海淀区",
    "name": "北京北京",
    "gbCode": "156110108"
  },
  ...
]
```

h **getPrompt()**

返回提示信息。

数据结构如下:

```
[
  {
    "keyword": "123456", //关键字
    "admins": [
      {
        "name": "中国", //搜索的行政区范围
        "adminCode": 1560000000 //行政区码
      }
    ]
  }
]
```

```

    }
  ],
  "type": 2
}
/*

```

当 Type=1 时，会给出一个 admin 一个 keyword 还有 admincode，此时一般的提示为'是否在 XXX 搜索名称含 XXX 的结果'

当 Type = 2 时，会给出一个 admin 一个 keyword 还有 admincode，此时一般提示为'在 XXX 没有搜索到相关的结果'

当 Type = 3 时，会给出多个 admin 及对应的 admincode，没有 keyword，此时一般的提示为

广东省汕尾市城区

山西省晋城市城区

山西省大同市城区

山西省长治市城区

山西省阳泉市城区

只列出名称即可，点击后，直接用这些关键字搜索即可完成行政区跳转

```

*/
},
{
  "admins": [
    {
      "name": "中国",
      "adminCode": 156000000
    }
  ],
  "type": 4
}
]

```

i getLineData()

返回线路信息，当 getResultType 值为 5 时返回。

数据结构如下：

```

[
  {
    "poiType": "103", // poi 点的 poi 类型 (poiType = 102 表示公交站，其它值表示普通 poi)
    "stationNum": "24", // 站数
    "name": "1 路 四惠枢纽站 靛厂新村", // 线路名称
    "uuid": "23089" // 线路的 id
  },
  {
    "poiType": "103",
    "stationNum": "24",

```



```
"name": "1 路 靛厂新村 四惠枢纽站",  
"uid": "23088"  
},  
...  
]
```

3) 简单示例

a 关键字本地搜索

代码如下:

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>  
  
  <meta name="keywords" content="天地图"/>  
  
  <title>根据关键字本地搜索</title>  
  
  <style type="text/css">  
  
    .search{font-size:13px;margin: auto;background:white}  
  
    .ls{line-height:27px; padding-left:7px;background:white}  
  
    .prompt{display: none; font-size:13px; border:1px solid #999999;background:white}  
  
    .statistics{display: none; font-size:13px; border:1px solid #999999; overflow-y: scroll;  
height:150px;background:white}  
  
    .suggests{display: none; font-size:13px; border:1px solid #999999;background:white}  
  
    .lineData{display: none; font-size:13px; border:1px solid #999999;background:white}  
  
    .result{display:none; font-size:12px; border:1px solid #999999; line-height:27px;  
padding-left:7px;background:white}  
  
  </style>  
  
  <script type="text/javascript" src="http://api.tianditu.com/js/maps.js"></script>  
  
  <script type="text/javascript" src="http://api.tianditu.com/js/service.js"></script>  
  
  <script>  
  
    var map,zoom = 12,localsearch;
```

<!-- 地图初始化和创建搜索对象-->

```
function onLoad()
```

```
{
```

```
    map=new TMap("mapDiv");//初始化地图对象
```

```
    map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);//设置显示地图的中心点
```

和级别

```
    map.enableHandleMouseScroll();//允许鼠标滚轮缩放地图
```

```
    map.enableDoubleClickZoom();//允许双击地图放大
```

```
    var config = {
```

```
        pageCapacity:10,    //每页显示的数量
```

```
        onSearchComplete:localSearchResult //接收数据的回调函数
```

```
    };
```

```
    localsearch = new TLocalSearch(map,config);//创建搜索对象
```

```
}
```

```
function localSearchResult(result)
```

```
{
```

```
    clearAll();//清空地图及搜索列表
```

```
    prompt(result);//添加提示词
```

```
    switch(parseInt(result.getResultType()))
```

```
    {
```

```
        case 1:
```

```
            //解析点数据结果
```

```
            pois(result.getPois());
```

```
            break;
```

```
        case 2:
```

```
            //解析推荐城市
```

```
            statistics(result.getStatistics());
```

```
            break;
```

```

        case 3:

            //解析行政区划边界

            area(result.getArea());

            break;

        case 4:

            //解析建议词信息

            suggests(result.getSuggests());

            break;

        case 5:

            //解析公交信息

            lineData(result.getLineData());

            break;

    } //根据返回类型解析搜索结果
}

function prompt(obj)
{
    var prompts = obj.getPrompt();

    if(prompts)
    {
        var promptHtml = "";

        for(var i=0;i<prompts.length;i++)
        {
            var prompt = prompts[i];

            var promptType = prompt.type;

            var promptAdmins = prompt.admins;

            var meanprompt = prompt.DidYouMean;

            if(promptType == 1)
            {

```

```

        promptHtml += "<p>您是否要在"+promptAdmins[0].name+"</strong>
搜索更多包含<strong>"+obj.getKeyword()+"</strong>的相关内容? <p>";

    }

    else if(promptType == 2)

    {

        promptHtml += "<p>在<strong>"+promptAdmins[0].name+"</strong>
没有搜索到与<strong>"+obj.getKeyword()+"</strong>相关的结果。 <p>";

        if(meanprompt)

        {

            promptHtml += "<p> 您 是 否 要 找 : <font weight='bold'
color='#035f9e'><strong>"+meanprompt+"</strong></font><p>";

        }

    }

    else if(promptType == 3)

    {

        promptHtml += "<p style='margin-bottom:3px;'>有以下相关结果, 您是
否要找: <p>"

        for(i=0;i<promptAdmins.length;i++)

        {

            promptHtml += "<p>"+promptAdmins[i].name+"<p>";

        }

    }

}

if(promptHtml != "")

{

    document.getElementById("promptDiv").style.display = "block";

    document.getElementById("promptDiv").innerHTML = promptHtml;

}

```

```

    }

} //解析提示词

//解析点数据结果

function pois(obj)

{

    if(obj)

    {

        var divMarker = document.createElement("div");//显示搜索列表

        var zoomArr = [];//坐标数组， 设置最佳比例尺时会用到

        for(var i=0;i<obj.length;i++)

        {

            //闭包

            (function(i){

                var name = obj[i].name;//名称

                var address = obj[i].address;//地址

                var lnglatArr = obj[i].lonlat.split(" "); //坐标

                var lnglat = new TLngLat(lnglatArr[0],lnglatArr[1]);

                var winHtml = "地址:" + address;

                var marker = new TMarker(lnglat);//创建标注对象

                map.addOverLay(marker);//地图上添加标注点

                TEvent.bind(marker,"click",marker,function(){

                    var info = this.openInfoWinHtml(winHtml);

                    info.setTitle(name);

                }); //注册标注点的点击事件

                zoomArr.push(lnglat);

                //在页面上显示搜索的列表

                var a = document.createElement("a");

                a.href = "javascript:/" ;
            })(i);
        }
    }
}

```

```

        a.innerHTML = name;

        a.onclick = function(){

            showPosition(marker,name,winHtml);

        }

        divMarker.appendChild(document.createTextNode((i+1)+"."));

        divMarker.appendChild(a);

        divMarker.appendChild(document.createElement("br"));

    })(i);

}

//显示地图的最佳级别

map.setViewport(zoomArr);

//显示搜索结果

divMarker.appendChild(document.createTextNode('                共

'+localsearch.getCountNumber()+ ' 条 记 录 ， 分 '+localsearch.getCountPage()+ ' 页 ， 当 前 第

'+localsearch.getPageIndex()+ ' 页'));

        document.getElementById("searchDiv").appendChild(divMarker);

        document.getElementById("resultDiv").style.display = "block";

    }

}

//显示信息框

function showPosition(marker,name,winHtml)

{

    var info = marker.openInfoWinHtml(winHtml);

    info.setTitle(name);

}

//解析推荐城市

function statistics(obj)

{

```

```

    if(obj)
    {
        //坐标数组， 设置最佳比例尺时会用到

        var pointsArr = [];

        var priorityCitysHtml = "";

        var allAdminsHtml = "";

        var priorityCitys = obj.priorityCitys;

        if(priorityCitys)
        {
            //推荐城市显示

            priorityCitysHtml += "在中国以下城市有结果<ul>";

            for(var i=0;i<priorityCitys.length;i++)
            {
                priorityCitysHtml += "<li>" + priorityCitys[i].name +
                "("+priorityCitys[i].count+")</li>";
            }

            priorityCitysHtml += "</ul>";
        }

        var allAdmins = obj.allAdmins;

        if(allAdmins)
        {
            allAdminsHtml += "更多城市<ul>";

            for(var i=0;i<allAdmins.length;i++)
            {
                allAdminsHtml += "<li>" + allAdmins[i].name +
                "("+allAdmins[i].count+")";

                var childAdmins = allAdmins[i].childAdmins;

                if(childAdmins)

```

```

        {

            for(var m=0;m<childAdmins.length;m++)

            {

                allAdminsHtml += "<blockquote>" + childAdmins[m].name +

                "("+childAdmins[m].count+")</blockquote>";

            }

        }

        allAdminsHtml += "</li>"

    }

    allAdminsHtml += "</ul>";

}

document.getElementById("statisticsDiv").style.display = "block";

document.getElementById("statisticsDiv").innerHTML = priorityCitysHtml +

allAdminsHtml;

}

}

//解析行政区划边界

function area(obj)

{

    if(obj)

    {

        //坐标数组， 设置最佳比例尺时会用到

        var pointsArr = [];

        var points = obj.points;

        for(var i=0;i<points.length;i++)

        {

            var regionLngLats = [];

            var regionArr = points[i].region.split(",");

```



```

        for(var m=0;m<regionArr.length;m++)
        {
            var lnglatArr = regionArr[m].split(" ");
            var lnglat = new TLatLng(lnglatArr[0],lnglatArr[1]);
            regionLngLats.push(lnglat);
            pointsArr.push(lnglat);
        }
        //创建线对象
        var line = new TPolyline(regionLngLats,{strokeColor:"blue", strokeWeight:3,
strokeOpacity:1, strokeStyle:"dashed"});
        //向地图上添加线
        map.addOverlay(line);
    }
    //显示最佳比例尺
    map.setViewport(pointsArr);
}
}
//解析建议词信息
function suggests(obj)
{
    if(obj)
    {
        //建议词提示，如果搜索类型为公交规划建议词或公交站搜索时，返回结果为
        公交信息的建议词。
        var suggestsHtml = "建议词提示<ul>";
        for(var i=0;i<obj.length;i++)
        {
            suggestsHtml += "<li>" + obj[i].name + "&nbsp;&nbsp;&nbsp;<font

```

```

style='color:#666666'>"+obj[i].address+"</font></li>";

    }

    suggestsHtml += "</ul>";

    document.getElementById("suggestsDiv").style.display = "block";

    document.getElementById("suggestsDiv").innerHTML = suggestsHtml;

    }

}

//解析公交信息

function lineData(obj)

{

    if(obj)

    {

        //公交提示

        var lineDataHtml = "公交提示<ul>";

        for(var i=0;i<obj.length;i++)

        {

            lineDataHtml += "<li>" + obj[i].name + "&nbsp;&nbsp;&nbsp;<font

style='color:#666666'>共"+obj[i].stationNum+"站</font></li>";

        }

        lineDataHtml += "</ul>";

        document.getElementById("lineDataDiv").style.display = "block";

        document.getElementById("lineDataDiv").innerHTML = lineDataHtml;

    }

}

//清空地图及搜索列表

function clearAll()

{

    map.clearOverLays();

```

```

        document.getElementById("searchDiv").innerHTML = "";

        document.getElementById("resultDiv").style.display = "none";

        document.getElementById("statisticsDiv").innerHTML = "";

        document.getElementById("statisticsDiv").style.display = "none";

        document.getElementById("promptDiv").innerHTML = "";

        document.getElementById("promptDiv").style.display = "none";

        document.getElementById("suggestsDiv").innerHTML = "";

        document.getElementById("suggestsDiv").style.display = "none";

        document.getElementById("lineDataDiv").innerHTML = "";

        document.getElementById("lineDataDiv").style.display = "none";

    }

</script>

</head>

<body onLoad="onLoad()" style="position: relative; margin: 0px">

<div id="mapDiv" style="position: absolute; width: 100%; height: 900px; z-index: -1"></div>

<div style="position: fixed; top: 50px; right: 50px; z-index: 1">

    <!-- 搜索面板 -->

    <div class="search">

        <input type="text" id="keyWord" value="南京工业大学" style="border: 0px solid
#999999; font-size: large"/>

        <input type="button"
onClick="localsearch.search(document.getElementById('keyWord').value)" value="搜索"/>

    </div>

    <br/>

    <!-- 提示词面板 -->

    <div id="promptDiv" class="prompt"></div>

    <!-- 统计面板 -->

    <div id="statisticsDiv" class="statistics"></div>

```

```

<!-- 建议词面板 -->

<div id="suggestsDiv" class="suggests"></div>

<!-- 公交提示面板 -->

<div id="lineDataDiv" class="lineData"></div>

<!-- 搜索结果面板 -->

<div id="resultDiv" class="result">

    <div id="searchDiv"></div>

    <div id="pageDiv">

        <input type="button" value="第一页" onClick="localsearch.firstPage()"/>

        <input type="button" value="上一页" onClick="localsearch.previousPage()"/>

        <input type="button" value="下一页" onClick="localsearch.nextPage()"/>

        <input type="button" value="最后一页" onClick="localsearch.lastPage()"/>

        <br/>

        转到第<input type="text" value="1" id="pageId" size="3"/>页

        <input                                     type="button"
onClick="localsearch.gotoPage(parseInt(document.getElementById('pageId').value));" value="转到"/>

    </div>

</div>

</div>

</body>

</html>

```

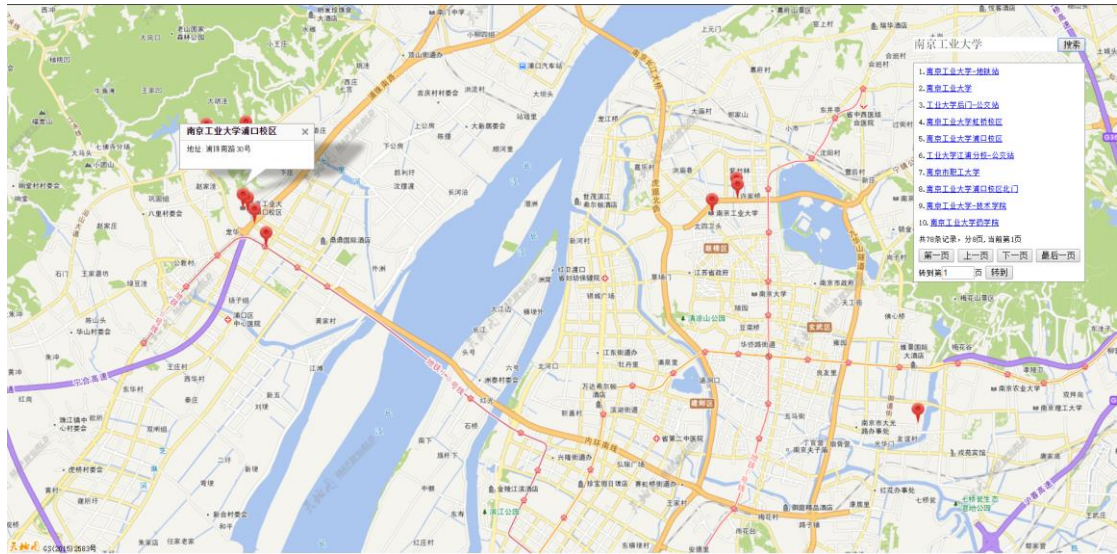


图9-1 关键字搜索

b Bounds 本地搜索

代码如下:

```
<!DOCTYPE html>

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=utf-8"/>

<meta name="keywords" content="天地图"/>

<title>根据 Bounds 本地搜索</title>

<style type="text/css">

    .search{font-size:13px; padding-left:7px;background:white;}

    .statistics{display: none; font-size:13px; border:1px solid #999999; overflow-y: scroll;
height:150px;background:white;}

    .result{display:none; font-size:12px; border:1px solid #999999; line-height:27px;
padding-left:7px;background:white;}

</style>

<script type="text/javascript" src="http://api.tianditu.com/js/maps.js"></script>

<script type="text/javascript" src="http://api.tianditu.com/js/service.js"></script>
```

```

<script>

    var map;

    var zoom = 12;

    var localsearch;

    var bounds = new TBounds(118.75125, 32.031, 118.79125, 32.061);

    function onLoad()

    {

        //初始化地图对象

        map=new TMap("mapDiv");

        //设置显示地图的中心点和级别

        map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);

        //允许鼠标滚轮缩放地图

        map.enableHandleMouseScroll();

        //允许双击地图放大

        map.enableDoubleClickZoom();

        var config = {

            pageCapacity:10,    //每页显示的数量

            onSearchComplete:localSearchResult //接收数据的回调函数

        };

        //创建搜索对象

        localsearch = new TLocalSearch(map,config);

        //创建矩形

        createRect();

    }

    function localSearchResult(result)

    {

        //清空地图及搜索列表

        clearAll();
    }

```

```

//创建矩形

createRect();

//根据返回类型解析搜索结果

switch(parseInt(result.getResultType()))

{

    case 1:

        //解析点数据结果

        pois(result.getPois());

        break;

    case 2:

        //解析统计城市

        statistics(result.getStatistics());

        break;

}

}

//解析点数据结果

function pois(obj)

{

    if(obj)

    {

        //显示搜索列表

        var divMarker = document.createElement("div");

        //坐标数组， 设置最佳比例尺时会用到

        var zoomArr = [];

        for(var i=0;i<obj.length;i++)

        {

            //闭包

            (function(i){

```

```

//名称

var name = obj[i].name;

//地址

var address = obj[i].address;

//坐标

var lnglatArr = obj[i].lonlat.split(" ");

var lnglat = new TLngLat(lnglatArr[0],lnglatArr[1]);

var winHtml = "地址:" + address;

//创建标注对象

var marker = new TMarker(lnglat);

//地图上添加标注点

map.addOverLay(marker);

//注册标注点的点击事件

TEvent.bind(marker,"click",marker,function(){

    var info = this.openInfoWinHtml(winHtml);

    info.setTitle(name);

});

zoomArr.push(lnglat);

//在页面上显示搜索的列表

var a = document.createElement("a");

a.href = "javascript:/" ;

a.innerHTML = name;

a.onclick = function(){

    showPosition(marker,name,winHtml);

}

divMarker.appendChild(document.createTextNode((i+1)+". "));

divMarker.appendChild(a);

divMarker.appendChild(document.createElement("br"));

```



```

        })(i);

    }

    //显示搜索结果

    divMarker.appendChild(document.createTextNode('
        '+localsearch.getCountNumber()+ ' 条 记 录 ， 分 '+localsearch.getCountPage()+ ' 页 ， 当 前 第
        '+localsearch.getPageIndex()+ ' 页');

    document.getElementById("searchDiv").appendChild(divMarker);

    document.getElementById("resultDiv").style.display = "block";

    }

    else

    {

        alert("无结果");

    }

}

//显示信息框

function showPosition(marker,name,winHtml)

{

    var info = marker.openInfoWinHtml(winHtml);

    info.setTitile(name);

}

//解析统计城市

function statistics(obj)

{

    if(obj)

    {

        //坐标数组， 设置最佳比例尺时会用到

        var pointsArr = [];

        var priorityCitysHtml = "";
    
```

```

var allAdminsHtml = "";

var priorityCitys = obj.priorityCitys;

if(priorityCitys)
{
    //推荐城市显示

    priorityCitysHtml += "在中国以下城市有结果<ul>";

    for(var i=0;i<priorityCitys.length;i++)
    {
        priorityCitysHtml += "<li>" + priorityCitys[i].name +
        "("+priorityCitys[i].count+")</li>";
    }

    priorityCitysHtml += "</ul>";
}

var allAdmins = obj.allAdmins;

if(allAdmins)
{
    allAdminsHtml += "更多城市<ul>";

    for(var i=0;i<allAdmins.length;i++)
    {
        allAdminsHtml += "<li>" + allAdmins[i].name +
        "("+allAdmins[i].count+")";

        var childAdmins = allAdmins[i].childAdmins;

        if(childAdmins)
        {
            for(var m=0;m<childAdmins.length;m++)
            {
                allAdminsHtml += "<blockquote>" + childAdmins[m].name +
                "("+childAdmins[m].count+")</blockquote>";
            }
        }
    }
}

```

```

        }

    }

    allAdminsHtml += "</li>"

}

allAdminsHtml += "</ul>";

}

document.getElementById("statisticsDiv").style.display = "block";

document.getElementById("statisticsDiv").innerHTML = priorityCitysHtml +
allAdminsHtml;

}

else

{

    alert("无结果");

}

}

```

//矩形

function createRect()

```

{

    var config = {

        strokeColor:"blue",//折线颜色

        fillColor:"#FFFFFF",    //填充颜色。当参数为空时，折线覆盖物将没有填充

```

效果

```

        strokeWeight:"3px",//折线的宽度，以像素为单位

        opacity:0.5,    //折线的透明度，取值范围0 - 1

        strokeStyle:"solid"//折线的样式，solid 或dashed

    };

    //创建矩形对象

    var rect = new TRect(bounds,config);

```

```

        map.addOverLay(rect);
    }

    //清空地图及搜索列表

    function clearAll()
    {
        map.clearOverLays();

        document.getElementById("searchDiv").innerHTML = "";

        document.getElementById("resultDiv").style.display = "none";

        document.getElementById("statisticsDiv").innerHTML = "";

        document.getElementById("statisticsDiv").style.display = "none";
    }
</script>

</head>

<body onLoad="onLoad()" style="position: relative; margin: 0px">

<div id="mapDiv" style="position: absolute; width: 100%; height: 860px; z-index: -1"></div>

<div style="position: fixed; top: 50px; right: 50px; z-index: 1">

    <!-- 搜索面板 -->

    <div class="search">

        <input type="text" id="keyWord" value="KFC" style="border: 0px solid
#999999; font-size: large"/>

        <input type="button"
onClick="localsearch.searchInBounds(document.getElementById('keyWord').value, bounds)" value="搜
索"/>

    </div>

    <br/>

    <!-- 提示词面板 -->

    <div id="promptDiv" class="prompt"></div>

    <!-- 统计面板 -->

```

```

<div id="statisticsDiv" class="statistics"></div>

<!-- 搜索结果面板 -->

<div id="resultDiv" class="result">

    <div id="searchDiv"></div>

    <div id="pageDiv">

        <input type="button" value="第一页" onClick="localsearch.firstPage()"/>

        <input type="button" value="上一页" onClick="localsearch.previousPage()"/>

        <input type="button" value="下一页" onClick="localsearch.nextPage()"/>

        <input type="button" value="最后一页" onClick="localsearch.lastPage()"/>

        <br/>

        转到第<input type="text" value="1" id="pageId" size="3"/>页

        <input type="button"

onClick="localsearch.gotoPage(parseInt(document.getElementById('pageId').value));" value="转到"/>

    </div>

</div>

</div>

</body>

</html>

```

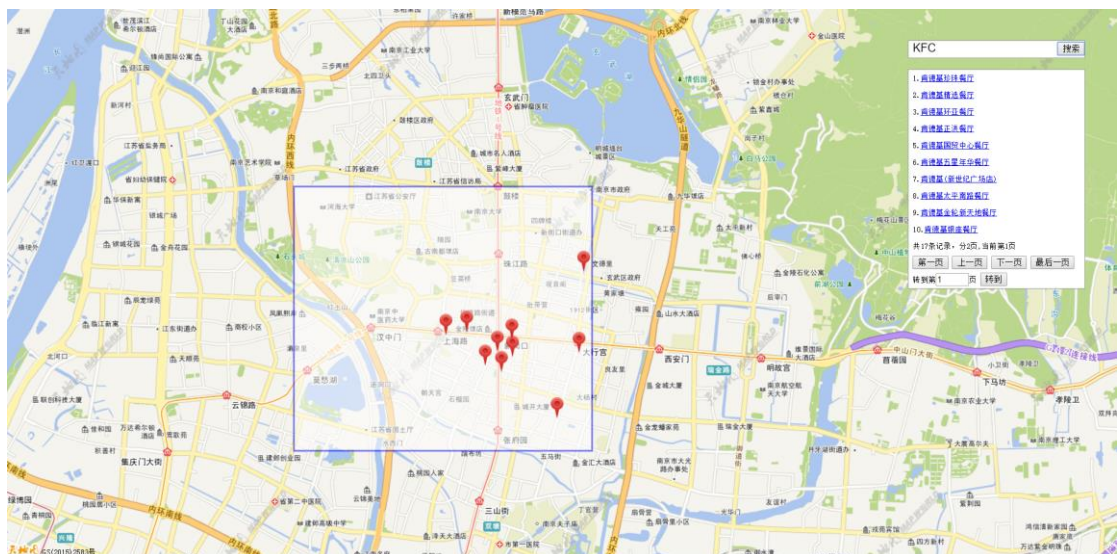


图9-2 范围搜索

c 中心点关键字搜索

代码如下:

```
<!DOCTYPE html>

<html>

<head>

    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>

    <meta name="keywords" content="天地图"/>

    <title>根据中心点关键字周边搜索</title>

    <style type="text/css">

        .search{font-size:13px; line-height:27px; padding-left:7px;background:white;}

        .result{display:none; font-size:12px; border:1px solid #999999; line-height:27px;
padding-left:7px;background:white;}

    </style>

    <script type="text/javascript" src="http://api.tianditu.com/js/maps.js"></script>

    <script type="text/javascript" src="http://api.tianditu.com/js/service.js"></script>

    <script>

        var map,zoom = 12,localsearch;

        var center = new TLatLng(118.79125,32.061);

        var radius = 5000;

        function onLoad()

        {

            map=new TMap("mapDiv");//初始化地图对象

            //设置显示地图的中心点和级别

            map.centerAndZoom(center,zoom);

            //允许鼠标滚轮缩放地图

            map.enableHandleMouseScroll();

            //允许双击地图放大

            map.enableDoubleClickZoom();
```

```

var config = {

    pageCapacity:10,    //每页显示的数量

    onSearchComplete:localSearchResult //接收数据的回调函数

};

//创建搜索对象

localsearch = new TLocalSearch(map,config);

//创建圆

createCircle();

}

function localSearchResult(result)

{

    //清空地图及搜索列表

    clearAll();

    //创建圆

    createCircle();

    //解析点数据结果

    pois(result.getPois());

}

//解析点数据结果

function pois(obj)

{

    if(obj)

    {

        //显示搜索列表

        var divMarker = document.createElement("div");

        //坐标数组， 设置最佳比例尺时会用到

        var zoomArr = [];

        for(var i=0;i<obj.length;i++)

```

```

{

    //闭包

    (function(i){

        //名称

        var name = obj[i].name;

        //地址

        var address = obj[i].address;

        //坐标

        var lnglatArr = obj[i].lonlat.split(" ");

        var lnglat = new TLngLat(lnglatArr[0],lnglatArr[1]);

        var winHtml = "地址:" + address;

        //创建标注对象

        var marker = new TMarker(lnglat);

        //地图上添加标注点

        map.addOverLay(marker);

        //注册标注点的点击事件

        TEvent.bind(marker,"click",marker,function(){

            var info = this.openInfoWinHtml(winHtml);

            info.setTitle(name);

        });

        zoomArr.push(lnglat);

        //在页面上显示搜索的列表

        var a = document.createElement("a");

        a.href = "javascript:/" ;

        a.innerHTML = name;

        a.onclick = function(){

            showPosition(marker,name,winHtml);

        }

    }

}

```



```

        divMarker.appendChild(document.createTextNode((i+1)+""));

        divMarker.appendChild(a);

        divMarker.appendChild(document.createElement("br"));

    }(i);

}

//显示地图的最佳级别

//map.setViewport(zoomArr);

//显示搜索结果

divMarker.appendChild(document.createTextNode('                共
'+localsearch.getCountNumber()+ ' 条 记 录 ， 分 '+localsearch.getCountPage()+ ' 页 ， 当 前 第
'+localsearch.getPageIndex()+ ' 页'));

document.getElementById("searchDiv").appendChild(divMarker);

document.getElementById("resultDiv").style.display = "block";

}

}

//显示信息框

function showPosition(marker,name,winHtml)

{

    var info = marker.openInfoWinHtml(winHtml);

    info.setTitle(name);

}

//圆形

function createCircle()

{

    var config = {

        strokeColor:"blue",    //圆边线颜色

        fillColor:"#FFFFFF",    //填充颜色。

        strokeWeight:"3px",    //圆边线线的宽度，以像素为单位

```

```

        strokeOpacity:0.5, //圆边线线的透明度, 取值范围0 - 1

        fillOpacity:0.5,      //填充的透明度, 取值范围0 - 1

        strokeStyle:"solid"    //圆边线线的样式, solid 或 dashed

    };

    //定义该圆形的显示区域

    var circle = new TCircle(center,radius,config);

    map.addOverLay(circle);

}

//清空地图及搜索列表

function clearAll()

{

    map.clearOverLays();

    document.getElementById("searchDiv").innerHTML = "";

    document.getElementById("resultDiv").style.display = "none";

}

</script>

</head>

<body onLoad="onLoad()" style="position:relative;margin: 0px">

<div id="mapDiv" style="position:absolute;width: 100%; height:860px;z-index: -1"></div>

<div style="position:fixed;top: 50px;right: 50px;z-index: 1">

    <!-- 搜索面板 -->

    <div class="search">

        <input type="text" id="keyWord" value=" 南京工业大学" style="border:0px solid

#999999;font-size:large"/>

        <input type="button"

onClick="localsearch.searchNearby(document.getElementById('keyWord').value,center,radius)" value="

搜索"/>

    </div>

```

```

<br/>

<!-- 搜索结果面板 -->

<div id="resultDiv" class="result">

    <div id="searchDiv"></div>

    <div id="pageDiv">

        <input type="button" value="第一页" onClick="localsearch.firstPage()"/>

        <input type="button" value="上一页" onClick="localsearch.previousPage()"/>

        <input type="button" value="下一页" onClick="localsearch.nextPage()"/>

        <input type="button" value="最后一页" onClick="localsearch.lastPage()"/>

        <br/>

        转到第<input type="text" value="1" id="pageId" size="3"/> 页

        <input type="button"

onClick="localsearch.gotoPage(parseInt(document.getElementById('pageId').value));" value="转到"/>

    </div>

</div>

</div>

</body>

</html>

```

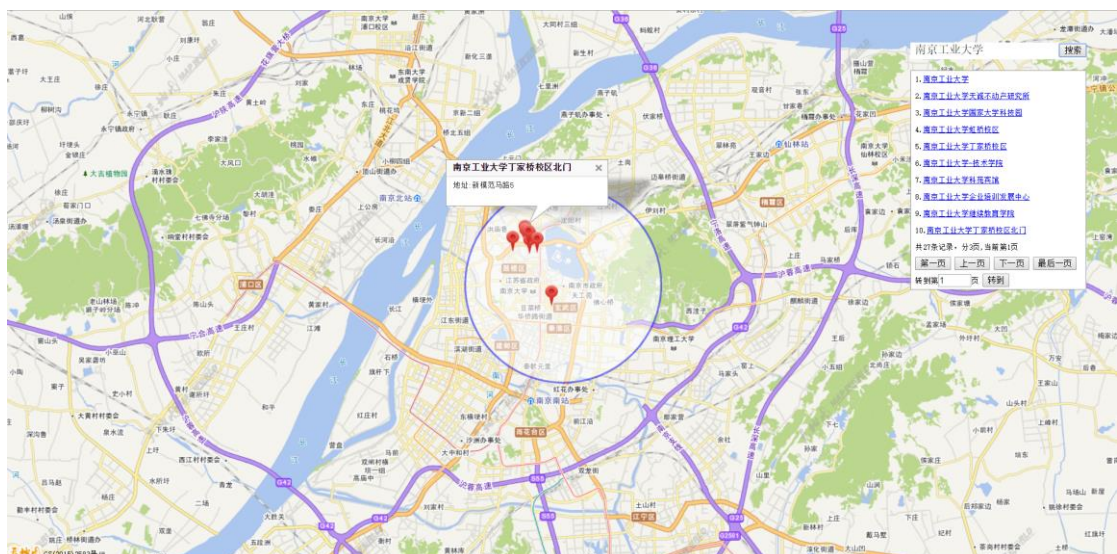


图9-3 圆形搜索

2. 公共交通路线

在天地图 API 中, 提供了两种不同的方式进行公交线路搜索, 一种是公交线路的规划方案, 另一种是公交线路的搜索。

1) 公交线路的规划

在天地图 API 中, `TTransitRoute` 类用于获取公交线路的规划方案, 它的构造函数是 `TTransitRoute(map:TMap,opts: TransitRouteOptions)`, `map` 是地图对象, `opts` 是此函数的可选参数, 包括公交导航的策略参数(`policy`)和检索结束后的回调函数(`onSearchComplete`)。

`TTransitRoute` 类的方法有:

a `search(start:TLngLat,end:TLngLat)`

发起检索, `start` 是起点坐标, `end` 是终点坐标。

b `getResults()`

返回最近一次的检索结果。它的返回值是 `TTransitRouteResult` 类, 此类有两个方法, 一个是返回方案个数 (`getNumPlans()`), 另一个是返回索引指定的方案 (`getPlan(i:Number)`)。索引方案由 `TTransitRoutePlan` 类表示, 可以调用方法返回方案的详细信息段数、描述文本、总距离、总时间、公交导航的策略函数和索引指定的详细信息。

索引指定的详细信息的返回值为 `TTransitRouteLine` 类, 此类表示一条公交线路。它没有构造函数, 提供了 4 种方法。如下表所示。

方法	返回值	说明
<code>getSegmentType()</code>	<code>number</code>	返回线路类型。 1 表示步行; 2 表示公交; 3 表示地铁; 4 表示地铁站内换乘。
<code>getStationStart()</code>	<code>JSON</code>	返回起站点信息。 数据结构如下: { "lonlat": "116.331796,39.991591", //起站点坐标 "name": "五道口站", //起站点名称 "uuid": "133010" //起站点id }

<code>getStationEnd()</code>	JSON	返回起站点信息。 数据结构如下： <pre>{ "lonlat": "116.331796,39.991591", //起站点坐标 "name": "五道口站", //起站点名称 "uuid": "133010" //起站点id }</pre>
<code>getSegmentLine()</code>	number	返回线路内容。 数据结构如下： <pre>[{ "segmentStationCount": 3, //此段线路需要经过的站点数 "segmentTime": 8, //此段线路需要的时间 "segmentDistance": 5933.84980962115, //此段线路的距离 "direction": "地铁 1 3 号线", //此段线路的完整线路名 "linePoint": "116.331796,39.991591;116.331993,39.987717;116.332141,39.98644;...", //此 段线路的坐标 "lineName": "地铁 1 3 号线", //线路名称 "byuuid": "23200" //此段线路id }]</pre>

c `clearResults()`

清除最近一次检索的结果。

d `setPolicy(policy:Number)`

设置路线规划策略，参数为策略常量。

参数说明:

policy: 策略常量。常量如下:

TMAP_TRANSIT_POLICY_LEAST_TIME = 1 表示最少时间

TMAP_TRANSIT_POLICY_LEAST_TRANSFER = 2 表示最少换乘

TMAP_TRANSIT_POLICY_LEAST_WALKING = 4 表示最少步行

TMAP_TRANSIT_POLICY_AVOID_SUBWAYS = 8 表示不乘地铁

e `setSearchCompleteCallback(fun:Function)`

设置检索结束后的回调函数。

f `getStatus()`

返回状态码。

状态码如下:

0 表示正常返回线路;
1 表示找不到起点;
2 表示找不到终点;
3 表示规划不出线路;
4 表示起终点距离200 米以内, 不返回线路;
5 表示起终点距离500 米内, 返回线路;
6 表示输入参数错误

2) 公交路线的搜索

在地图 API 中, `TBusLineSearch` 类表示公交路线的搜索类, 它的构造函数是 `TBusLineSearch(map:TMap ,opts: TBusLineSearchOptions)`, `map` 是地图对象, `opts` 是它的可选参数, 可以设置公交列表查询后的回调函数 (`onGetBusListComplete`) 和设置公交线路查询后的回调函数 (`onGetBusLineComplete`)。除了通过参数设置回调函数, 还可以使用一下方法和实现。

方法	返回值	说明
<code>getBusList(keyword: String)</code>	<code>none</code>	检索公交信息。
<code>getBusLine(busLstItem:BusListItem)</code>	<code>none</code>	检索公交线路。
<code>setGetBusListCompleteCallback(callback:Function)</code>	<code>none</code>	设置公交列表查询后的回调函数。
<code>setGetBusLineCompleteCallback(callback:Function)</code>	<code>none</code>	设置公交线路查询后的回调函数。

a 公交列表的回调函数

`TBusListResult` 类表示公交列表的检索结果。可以通过调用 `getNumBusList()` 方法得到公交列表的个数, `getBusListItem(i: Number)` 获取某一个具体的公交列表中的对象。

数据结构如下:

```
{  
  "stationNum": "42", //站数  
  "name": "850 路东直门外-马坡站", //线路名称  
  "uuid": "23084" //线路的id  
}
```

b 公交线路的回调函数

`TBusLine` 类公交线路, 此类提供了丰富的属性和方法, 见下面两表。

属性表:

属性	类型	说明
<code>lineName</code>	<code>string</code>	线路名称, 如: 68 路。
<code>lineType</code>	<code>number</code>	线路类型, 1 表示公交; 2 表示地铁; 3 表示磁悬浮。
<code>length</code>	<code>number</code>	线路的长度, 单位米。
<code>linePoint</code>	<code>string</code>	线路的详细点信息, 格式: <code>x,y;x,y;</code>

<i>startTime</i>	<i>string</i>	始发车时间格式为: hh:mm 24 小时制。
<i>endTime</i>	<i>string</i>	末班车时间格式为: hh:mm 24 小时制。
<i>totalTime</i>	<i>number</i>	公交线路的全程运营总时间, 单位分钟。
<i>stationCount</i>	<i>number</i>	该线路所具有的站点总数量。
<i>interval</i>	<i>number</i>	发车间隔, 单位秒。
<i>ticketcal</i>	<i>number</i>	计费模式, 0 表示单一; 1 表示按距离; 2 表示按站。
<i>totalPrice</i>	<i>number</i>	全程票价, 单位分。
<i>startPrice</i>	<i>number</i>	起步票价, 单位分。
<i>increasedPrice</i>	<i>number</i>	递增距离票价, 单位千米。
<i>increasedStep</i>	<i>number</i>	车站递增票价, 按站。
<i>ismonTicket</i>	<i>number</i>	是否支持月票, 0 表示不支持; 1 表示支持。
<i>isBidirectional</i>	<i>number</i>	是否双向行驶, 0 表示单项; 1 表示双向。
<i>isManual</i>	<i>number</i>	是否人工售票, 0 表示有人; 1 表示无人。
<i>status</i>	<i>number</i>	状态, 0 表示使用中; 1 表示非使用中
<i>company</i>	<i>string</i>	所属公交公司, 公交线路所属公司。

方法表:

方法	返回值	说明
<i>getNumBusStations()</i>	<i>number</i>	返回公交站点个数。
<i>getBusStation(i: Number)</i>	<i>JSON</i>	返回某一个具体的公交站信息。 数据结构如下: { "lonlat": "116.430522,39.939852", //站点坐标 "name": "东直门外", //站点名称 "uuid": "122805" //站的id 信息 }
<i>getPath()</i>	<i>array</i>	返回公交线地理坐标点数组。

3) 简单示例

a 公交搜索规划方案

代码如下:

```
<!DOCTYPE html>

<html>

<head>

  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>

  <meta name="keywords" content="天地图"/>

  <title> 公交搜索规划方案</title>

  <script type="text/javascript" src="http://api.tianditu.com/js/maps.js"></script>
```

```
<script type="text/javascript" src="http://api.tianditu.com/js/service.js"></script>
```

```
<script>
```

```
var map;          //地图对象
```

```
var zoom = 12;    //地图级别
```

```
var transitRoute; //公交搜索对象
```

```
var obj;          //公交搜索结果
```

```
var startLngLat;  //起点经纬度
```

```
var endLngLat;    //终点经纬度
```

```
var startTool;    //起点标注工具
```

```
var endTool;      //终点标注工具
```

```
var startIcon = "../images/bus/start.png"; //起点图标
```

```
var endIcon = "../images/bus/end.png";    //终点图标
```

```
var map_bus = "../images/bus/map_bus.png";
```

```
var map_metro = "../images/bus/map_metro.png";
```

```
function onLoad()
```

```
{
```

```
    //初始化地图对象
```

```
    map=new TMap("mapDiv");
```

```
    //设置显示地图的中心点和级别
```

```
    map.centerAndZoom(new TLatLng(118.79125,32.061),zoom);
```

```
    //允许鼠标滚轮缩放地图
```

```
    map.enableHandleMouseScroll();
```

```
    //允许双击地图放大
```

```
    map.enableDoubleClickZoom();
```

```
    //鼠标标起点
```

```
    var config = {
```

```
        icon : new TIcon(startIcon,new TSize(24,24),{anchor:new TPixel(14,28)})
```

```
    };
```



```

startTool = new TMarkTool(map,config);

TEvent.addListener(startTool,"mouseup",mouseUpStartMaker);

//鼠标标终点

var config = {

    icon : new TIcon(endIcon,new TSize(24,24),{anchor:new TPixel(14,28)})

};

endTool = new TMarkTool(map,config);

TEvent.addListener(endTool,"mouseup",mouseUpEndMaker);

var config = {

    policy:1, //公交导航的策略参数

    onSearchComplete:busSearchResult //检索完成后的回调函数

};

//创建公交搜索对象

transitRoute = new TTransitRoute(map,config);

}

//公交搜索

function searchBus()

{

    //清空显示列表

    document.getElementById("resultDiv").innerHTML = "";

    //清空地图

    map.clearOverLays();

    //起点经纬度

    var startVal = document.getElementById("start").value.split(",");

    startLngLat = new TLngLat(startVal[0],startVal[1]);

    //终点经纬度

    var endVal = document.getElementById("end").value.split(",");

    endLngLat = new TLngLat(endVal[0],endVal[1]);

```

```
//设置公交策略

transitRoute.setPolicy(getRadioValue());

//公交搜索

transitRoute.search(startLngLat,endLngLat);

}

//显示公交搜索结果

function busSearchResult(result)

{

    if(transitRoute.getStatus()==0)

    {

        document.getElementById("resultDiv").style.display="block";

        //添加起始点

        createStartMarker();

        obj = result;

        var resultList = document.createElement("div");

        //获取方案个数

        var plans = result.getNumPlans();

        for(var i=0;i<plans;i++)

        {

            //获得单条公交结果对象

            var plan = result.getPlan(i);

            //显示单个方案面板

            var div = document.createElement("div");

            div.style.cssText = "font-size:12px; cursor:pointer; border:1px solid #999999;";

            //闭包

            (function(i){

                div.onclick = function(){

                    //清空地图
```

```

        map.clearOverLays();

        //添加起始点

        createStartMarker();

        //显示线路

        createSegments(obj.getPlan(i));

    };

})(i);

//显示方案内容

var describeStr = "<strong> 方案"+(i+1)+"": "+plan.getLineName().join("→")+"", 总时间: "+plan.getDuration()+"分, 总距离: "+Math.round(plan.getDistance())+"米</strong>";

describeStr += "<div><img src='"+startIcon+"'>"+document.getElementById("start").value+"</div>";

//显示每个方案的详细信息

var segmentNum = plan.getNumSegments();

for(var m=0;m<segmentNum;m++)

{

    var line = plan.getDetails(m);

    var segmentLine = line.getSegmentLine()[0];

    //经过的公交或地铁的站数

    var stationCount = (segmentLine.segmentStationCount != "")?", 经过 "+segmentLine.segmentStationCount+"站:";

    //线路类型1, 步行; 2, 公交; 3, 地铁; 4, 地铁站内换乘

    if(line.getSegmentType()==1)

    {

        describeStr += "步行约"+segmentLine.segmentDistance+"米, 到达 "+line.getStationEnd().name+", ";

    }

    else if(line.getSegmentType()==2)

```

```

        {

            describeStr += "乘坐"+segmentLine.direction+stationCount+", 到/
达"+line.getStationEnd().name+", ";

        }

        else if(line.getSegmentType()==3)

        {

            describeStr += "乘坐"+segmentLine.direction+stationCount+", 到/
达"+line.getStationEnd().name+", ";

        }

        else if(line.getSegmentType()==4)

        {

            describeStr += "站内换乘, ";

        }

    }

    //去掉最后的逗号

    describeStr = describeStr.substring(0,describeStr.length-1);

    describeStr += "<div><img

src='"+endIcon+"'>"+document.getElementById("end").value+"</div>";

    div.innerHTML = describeStr;

    resultList.appendChild(div);

    //在地图上默认显示方案一的线路

    if(i==0)

    {

        createSegments(result.getPlan(0));

    }

}

//显示公交搜索结果

document.getElementById("resultDiv").appendChild(resultList);

```

```

    }

}

//显示公交线路

function createSegments(plan,planNum)

{

    var segmentNum = plan.getNumSegments();

    for(var m=0;m<segmentNum;m++)

    {

        var line = plan.getDetails(m);

        var segmentLine = line.getSegmentLine()[0];

        //显示线路

        createRoute(segmentLine.linePoint,line.getSegmentType(),line.getStationStart().lonlat,line.getStationEnd()

        .lonlat);

        //显示换乘图标

        createMarker(line.getStationStart().lonlat,line.getStationEnd().lonlat,line.getSegmentType());

    }

}

//显示公交换乘图标,lnglatStartStr 表示该线路的起始点, lnglatEndStr 表示该线路的终点,

type 表示线路类型

function createMarker(lnglatStartStr;lnglatEndStr;type)

{

    if(type == 2) //公交

    {

        //公交标注

        var icon = new TIcon(map_bus,new TSize(23,23),{anchor:new TPixel(12,12)});

    }

}

```

```

else if(type == 3)//地铁
{
    //地铁标注
    var icon = new TIcon(map_metro,new TSize(23,23),{anchor:new TPixel(12,12)});
}

else    //地铁站内换乘
{
    //地铁标注
    var icon = new TIcon(map_metro,new TSize(23,23),{anchor:new TPixel(12,12)});
}

if(type != 1)
{
    var lnglatStartArr = lnglatStartStr.split(",");
    var lnglatStart = new TLatLng(lnglatStartArr[0],lnglatStartArr[1]);
    var lnglatEndArr = lnglatEndStr.split(",");
    var lnglatEnd = new TLatLng(lnglatEndArr[0],lnglatEndArr[1]);
    var startMarker = new TMarker(lnglatStart,{icon:icon});
    map.addOverlay(startMarker);
    var endMarker = new TMarker(lnglatEnd,{icon:icon});
    map.addOverlay(endMarker);
}
}

//公交线路, pointsStr 表示经纬度字符串, type 表示线路类型1, 步行; 2, 公交; 3, 地
铁; 4, 地铁站内换乘, lnglat 表示显示公交或地铁图标的经纬度
function createRoute(pointsStr,type,lnglatStartStr,lnglatEndStr)
{
    //去掉经纬度字符串最后一个分号, 并存储在一个数据中。
    var points = pointsStr.substring(0,pointsStr.length-1).split(";");

```

```
//存储经纬度的数组

var lnglatArr = [];

for(var i=0;i<points.length;i++)

{

    var lnglat = points[i].split(",");

    lnglatArr.push(new TLatLng(lnglat[0],lnglat[1]));

}

//步行

if(type == 1)

{

    var lineColor = "#2E9531"; //线的颜色

    var lineStyle = "dashed"; //线的样式

}

else if(type == 2) //公交

{

    var lineColor = "#2C64A7"; //线的颜色

    var lineStyle = "solid"; //线的样式

}

else if(type == 3) //地铁

{

    var lineColor = "#2C64A7"; //线的颜色

    var lineStyle = "solid"; //线的样式

}

else //地铁站内换乘

{

    var lineColor = "#2E9531"; //线的颜色

    var lineStyle = "dashed"; //线的样式

}
```

```

//创建线对象

var line = new TPolyline{lnglatArr:{strokeColor:lineColor, strokeWeight:5,
strokeOpacity:1, strokeStyle:lineStyle}});

//向地图上添加线

map.addOverLay(line);

}

//添加起始点

function createStartMarker()
{
//向地图上添加起点

var icon = new TIcon(startIcon,new TSize(24,24),{anchor:new TPixel(12,12)})

var startMarker = new TMarker(startLngLat,{icon:icon});

map.addOverLay(startMarker);

//向地图上添加终点

var icon = new TIcon(endIcon,new TSize(24,24),{anchor:new TPixel(12,12)})

var endMarker = new TMarker(endLngLat,{icon:icon});

map.addOverLay(endMarker);

}

function mouseUpStartMaker(point)
{
//向地图上添加起点

var icon = new TIcon(startIcon,new TSize(24,24),{anchor:new TPixel(14,28)})

var startMarker = new TMarker(point,{icon:icon});

map.addOverLay(startMarker);

//关闭标点工具

startTool.close();

//设置公交搜索起点

document.getElementById("start").value = point.getLng()+" "+point.getLat();

```



```

    }

    function mouseUpEndMaker(point)

    {

        //向地图上添加起点

        var icon = new TIcon(endIcon,new TSize(24,24),{anchor:new TPixel(14,28)})

        var endMarker = new TMarker(point,{icon:icon});

        map.addOverLay(endMarker);

        //关闭标点工具

        endTool.close();

        //设置公交搜索终点

        document.getElementById("end").value = point.getLng()+","+point.getLat();

    }

    //获得公交策略

    function getRadioValue()

    {

        var obj = document.getElementsByName("planType");

        for(var i=0;i<obj.length;i++)

        {

            if(obj[i].checked)

            {

                return obj[i].value;

            }

        }

    }

</script>

</head>

<body onLoad="onLoad()" style="position:relative;margin: 0px">

<div id="mapDiv" style="position:absolute;width: 100%; height:860px;z-index: -1"></div>

```

```
<div style="position:fixed;top: 50px;right: 50px;z-index: 1">

    <!-- 查询面板 -->

    <div style="font-size:13px; border:1px solid #999999; line-height:27px;
padding-left:7px;background:white">

        <input type="radio" name="planType" value="1" checked="checked"/>较快捷

        <input type="radio" name="planType" value="2"/>少换乘

        <input type="radio" name="planType" value="4"/>少步行

        <input type="radio" name="planType" value="8"/>不坐地铁

        <br/>

        起点: <input type="text" id="start" value="118.79365,32.08950"/>

        <input type="button" value="起点" onClick="startTool.open()"/>

        <br/>

        终点: <input type="text" id="end" value="118.63752,32.07751"/>

        <input type="button" value="终点" onClick="endTool.open()"/>

        <br/>

        <input type="button" onClick="searchBus()" value="公交搜索"/>

    </div>

    <br/>

    <!-- 结果面板 -->

    <div id="resultDiv" style="display: none;height:250px; width:300px;
overflow-y:scroll;background:white"></div>

</div>

</body>

</html>
```

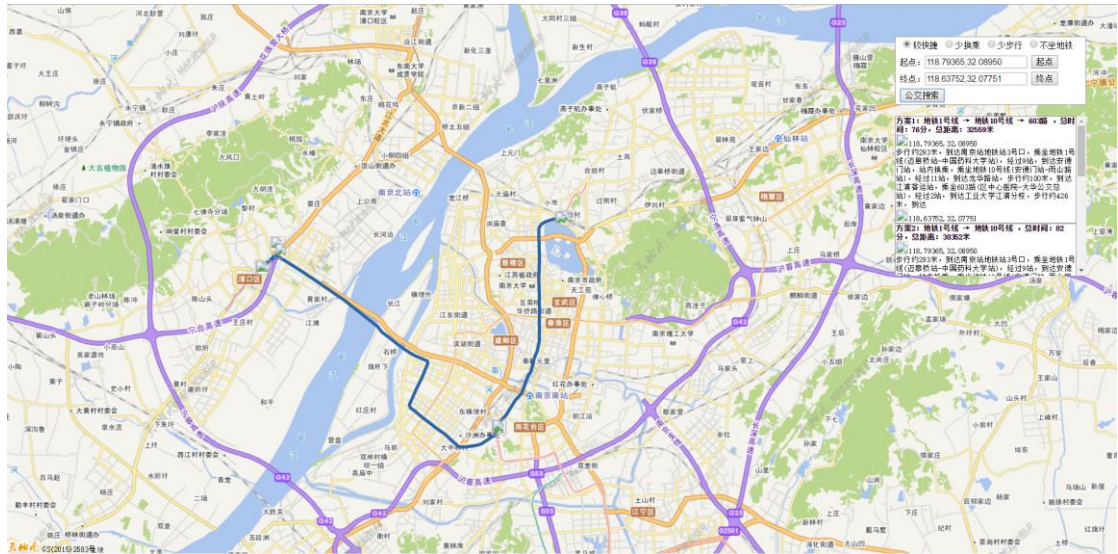


图9-4 公交搜索规划方案

b 公交线路查询

代码如下:

```
<!DOCTYPE html>

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=utf-8"/>

<meta name="keywords" content="天地图"/>

<title> 公交地铁线路查询</title>

<style type="text/css">

.bus{font-size:13px;background:white}

.busline{display:none; font-size:13px; border:1px solid #999999; height:250px; width:300px;
overflow-y:scroll;background:white}

</style>

<script type="text/javascript" src="http://api.tianditu.com/js/maps.js"></script>

<script type="text/javascript" src="http://api.tianditu.com/js/service.js"></script>

<script>

var map,zoom = 12;

var busLineSearch;
```

```

var lineIcon = "../images/bus/linePoi.png";

function onLoad()

{

    //初始化地图对象

    map=new TMap("mapDiv");

    //设置显示地图的中心点和级别

    map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);

    //允许鼠标滚轮缩放地图

    map.enableHandleMouseScroll();

    //允许双击地图放大

    map.enableDoubleClickZoom();

    var config = {

        onGetBusListComplete:busListSearchResult, //公交名称检索完成后的回调函数

        onGetBusLineComplete:busLineSearchResult //公交线路检索完成后的回调函
数

    };

    //创建搜索对象

    busLineSearch = new TBusLineSearch(map,config);

}

//搜索公交

function busSearch()

{

    //清空地图及查询列表

    map.clearOverLays();

    //清空公交线路列表

    document.getElementById("busDiv").innerHTML = "";

    //清空公交线路详细列表

    document.getElementById("buslineDiv").innerHTML = "";

```

```

document.getElementById("buslineDiv").style.display = "none";

//搜索公交线路

busLineSearch.getBusList(document.getElementById('keyWord').value)

}

//显示公交查询列表

function busListSearchResult(result)

{

    if(result)

    {

        //显示查询列表

        var busDiv = document.createElement("div");

        //循环公交列表个数，显示公交线路

        var busList = result.getNumBusList();

        for(var i=0;i<busList;i++)

        {

            //闭包

            (function(i){

                var busListItem = result.getBusListItem(i);

                //名称

                var name = busListItem.name;

                //在页面上显示查询的列表

                var a = document.createElement("a");

                a.href = "javascript://";

                a.innerHTML = name;

                a.onclick = function(){

                    //清空地图及查询列表

                    map.clearOverLays();

                    //获取公交线路的详细信息

```

```

        busLineSearch.getBusLine(busListItem);

    }

    busDiv.appendChild(document.createTextNode((i+1)+"."));

    busDiv.appendChild(a);

    busDiv.appendChild(document.createElement("br"));

    })(i);

}

document.getElementById("busDiv").appendChild(busDiv);

}

else

{

    alert("没有搜索到相关公交信息! ");

}

}

//显示公交线路查询列表

function busLineSearchResult(result)

{

    //显示查询列表

    var buslineDiv = document.createElement("div");

    //循环公交列表个数，显示公交线路

    var busStations = result.getNumBusStations();

    for(var i=0;i<busStations;i++)

    {

        //闭包

        (function(i){

            var busStation = result.getBusStation(i);

            //名称

            var name = busStation.name;

```

```

//经纬度坐标

var lnglatArr = busStation.lonlat.split(",");

var lnglat = new TLngLat(lnglatArr[0],lnglatArr[1]);

//向地图上添加公交站点

var icon = new TIcon(lineIcon,new TSize(8,8),{anchor:new TPixel(8,8)})

var marker = new TMarker(lnglat,{icon:icon});

marker.name = name;

marker.setTitle(name);

map.addOverLay(marker);

TEvent.bind(marker,"click",marker,function(){this.openInfoWinHtml(this.name)});

//在页面上显示公交站的列表

var a = document.createElement("a");

a.href = "javascript://";

a.innerHTML = name;

a.onclick = function(){

    showPosition(marker);

}

buslineDiv.appendChild(document.createTextNode((i+1)+"."));

buslineDiv.appendChild(a);

buslineDiv.appendChild(document.createElement("br"));

})(i);

}

//创建公交线对象

var line = new TPolyline(result.getPath(),{strokeColor:"#2C64A7", strokeWeight:5,

strokeOpacity:1});

//向地图上添加线

map.addOverLay(line);

//显示最佳级别

```

```

        map.setViewport(result.getPath());

        //清空公交线路详细列表

        document.getElementById("buslineDiv").innerHTML = "";

        document.getElementById("buslineDiv").style.display = "block";

        document.getElementById("buslineDiv").appendChild(document.createTextNode("沿线公
交站点:"));

        document.getElementById("buslineDiv").appendChild(buslineDiv);

    }

    //定位公交站

    function showPosition(marker)

    {

        marker.openInfoWinHtml(marker.name);

        map.panTo(marker.getLngLat());

    }

</script>

</head>

<body onLoad="onLoad()" style="position: relative; margin: 0px">

<div id="mapDiv" style="position: absolute; width: 100%; height: 860px; z-index: -1"></div>

<div style="position: fixed; top: 50px; right: 50px; z-index: 1">

    <!-- 搜索面板 -->

    <div class="search">

        <input type="text" id="keyWord" value="1" style="border: 0px solid #999999; font-size: large"/>

        <input type="button" onClick="busSearch();" value="搜索"/>

    </div>

    <br/>

    <!-- 公交提示面板 -->

    <div id="busDiv" class="bus"></div>

    <!-- 公交线路详细面板 -->

```



```

<div id="buslineDiv" class="busline"></div>

</div>

</body>

</html>

```

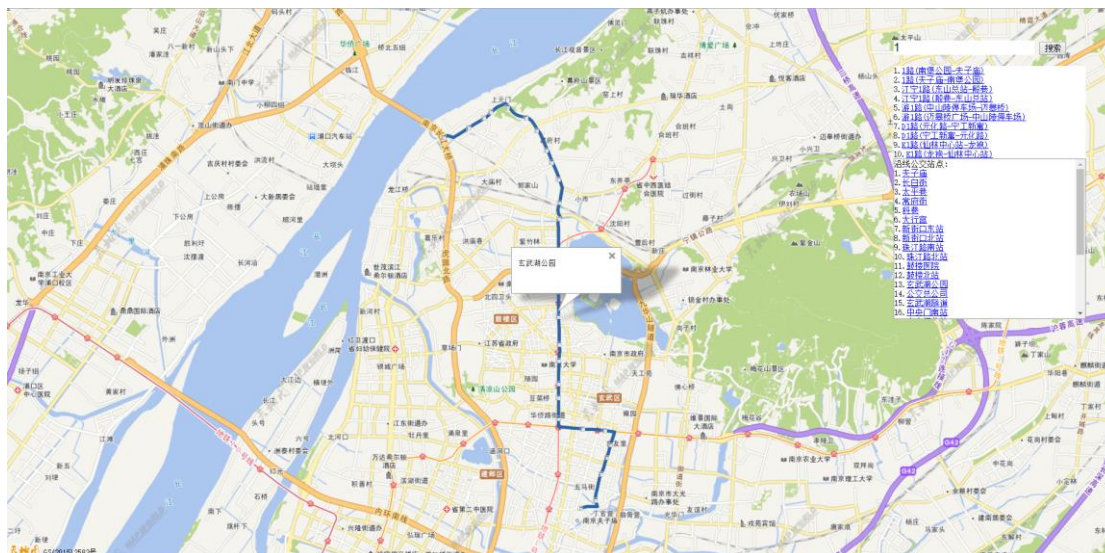


图9-5 公交线路查询

3. 驾车路线

在地图API 中, `TDrivingRoute` 类用于获取驾车路线的规划方案。它的构造函数是 `TDrivingRoute(map:TMap ,opts: TDrivingRouteOptions)`, `map` 是地图对象, `opts` 是此类的可选参数, 有驾车的策略参数 (`policy`) 和检索完成后的回调函数 (`onSearchComplete`)。

1) `TDrivingRoute` 的方法

a `search(start:TLngLat,end:TLngLat)`

发起检索, `start`: 起点坐标, `end`: 终点坐标。

b `getResults()`

返回最近一次检索的结果, 它的返回值是 `TDrivingRouteResult`, 下面会详细讲解。

c `clearResults()`

清除最近一次检索的结果。

d `setPolicy(policy:Number)`

设置路线规划策略, 参数为策略常量。

<p>参数说明:</p> <p>policy: 策略常量。常量如下:</p> <p>TMAP_DRIVING_POLICY_LEAST_TIME = 0 最少时间</p> <p>TMAP_DRIVING_POLICY_LEAST_DISTANCE = 1 最短距离</p> <p>TMAP_DRIVING_POLICY_AVOID_HIGHWAYS = 2 避开高速</p> <p>TMAP_DRIVING_POLICY_WALK = 3 步行</p>
--

e setSearchCompleteCallback(fun:Function)

设置检索结束后的回调函数。

2) TdrivingRouteResult 类

此类表示TDrivingRoute 的检索结果，没有构造函数，通过TDrivingRoute.getResults()方法或TDrivingRoute 的onSearchComplete 回调函数的参数得到。它的方法如下表所示。

方法	返回值	说明
getStart()	TLngLat	返回起点。
getEnd()	TLngLat	返回终点。
getNumPlans()	number	返回方案个数。
getPlan(i:Number)	TTransitRoutePlan	返回索引指定的方案。索引0 表示第一条方案。

TroutePlan 类此类表示一条驾车出行方案。它的方法有：

方法	返回值	说明
getNumRoutes()	number	返回该方案包含的线路的个数。
getRoute(i:Number)	TRoute	返回方案中索引指定的线路信息。索引0 表示第一条线路。
getDistance()	number	返回方案总距离，单位为公里。
getDuration()	number	返回方案总时间，单位为秒。
getWayPois()	Array<TLngLat >	返回途径点。
getPath()	Array<TLngLat >	返回路线的经纬度。

TRoute 类表示驾车路线，通过该类可以返回该段路线包含的关键点个数、索引指定的关键点、该段路线距离、该段路线在方案中的索引位置、该段路线的坐标。

关键点的用 TStep 类表示。可以返回关键点地理坐标、、本关键点在路线中的位置索引、关键点描述文本。

3) 简单示例

代码如下：

```

<!DOCTYPE html>

<html>

```

```

<head>

    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>

    <meta name="keywords" content="天地图"/>

    <title>根据起终点经纬度驾车导航</title>

    <style type="text/css">

        p {text-indent:20px;}

    </style>

    <script type="text/javascript" src="http://api.tianditu.com/js/maps.js"></script>

    <script type="text/javascript" src="http://api.tianditu.com/js/service.js"></script>

    <script>

        var map;           //地图对象

        var zoom = 12;     //地图级别

        var drivingRoute;  //驾车路线规划对象

        var obj;           //搜索结果

        var startLngLat;   //起点经纬度

        var endLngLat;     //终点经纬度

        var startTool;     //起点标注工具

        var endTool;       //终点标注工具

        var startIcon = "../images/bus/start.png"; //起点图标

        var endIcon = "../images/bus/end.png";    //终点图标

        var infoWin = null;

        function onLoad()

        {

            //初始化地图对象

            map=new TMap("mapDiv");

            //设置显示地图的中心点和级别

            map.centerAndZoom(new TLngLat(118.79125,32.061),zoom);

            //允许鼠标滚轮缩放地图

```

```

        map.enableHandleMouseScroll();

        //允许双击地图放大

        map.enableDoubleClickZoom();

        //鼠标标起点

        var config = {

            icon : new TIcon(startIcon,new TSize(24,24),{anchor:new TPixel(14,28)})

        };

        startTool = new TMarkTool(map,config);

        TEvent.addListener(startTool,"mouseup",mouseUpStartMaker);

        //鼠标标终点

        var config = {

            icon : new TIcon(endIcon,new TSize(24,24),{anchor:new TPixel(14,28)})

        };

        endTool = new TMarkTool(map,config);

        TEvent.addListener(endTool,"mouseup",mouseUpEndMaker);

        var config = {

            policy:0, //驾车策略

            onSearchComplete:searchResult //检索完成后的回调函数

        };

        //创建公交搜索对象

        drivingRoute = new TDrivingRoute(map,config);

    }

    //公交搜索

    function searchDrivingRoute()

    {

        //清空显示列表

        document.getElementById("resultDiv").innerHTML = "";

        //清空地图

```

```

        map.clearOverLays();

        //起点经纬度

        var startVal = document.getElementById("start").value.split(",");

        startLngLat = new TLngLat(startVal[0],startVal[1]);

        //终点经纬度

        var endVal = document.getElementById("end").value.split(",");

        endLngLat = new TLngLat(endVal[0],endVal[1]);

        //设置驾车策略

        drivingRoute.setPolicy(getRadioValue());

        //驾车路线搜索

        drivingRoute.search(startLngLat,endLngLat);

    }

    //显示公交搜索结果

    function searchResult(result)

    {

        document.getElementById("resultDiv").style.display="block";

        //添加起始点

        createStartMarker();

        obj = result;

        var resultList = document.createElement("div");

        //获取方案个数

        var routes = result.getNumPlans();

        for(var i=0;i<routes;i++)

        {

            //获得单条驾车方案结果对象

            var plan = result.getPlan(i);

            //显示单个方案面板

            var div = document.createElement("div");

```

```

div.style.cssText = "font-size:12px; cursor:pointer; border:1px solid #999999";

//显示方案内容

var describeStr = "<strong>总时间: "+plan.getDuration()+" 秒, 总距离: "+Math.round(plan.getDistance())+" 公里</strong>";

describeStr += "<div><img src='"+startIcon+"'>"+document.getElementById("start").value+"</div>";

//显示该方案每段的描述信息

var numRoutes = plan.getNumRoutes();

for(var m=0;m<numRoutes;m++)
{
    var route = plan.getRoute(m);

    describeStr += (m+1)+"<span>"+route.getDescription()+"</span><br/>"

    //显示该方案每段的详细描述信息

    var numStepsStr = "";

    var numSteps = route.getNumSteps();

    for(var n=0;n<numSteps;n++)
    {
        var step = route.getStep(n);

        numStepsStr += "<p>"+(n+1)+"<a href='javascript:/" onclick='showPosition('"+step.getPosition().getLng()+"\",""+step.getPosition().getLat()+"\",""+step.getDescription()+"\");'>"+step.getDescription()+"</a></p>";

    }

    describeStr += numStepsStr;

}

describeStr += "<div><img src='"+endIcon+"'>"+document.getElementById("end").value+"</div>";

div.innerHTML = describeStr;

resultList.appendChild(div);

```

```

        //显示驾车线路

        createRoute(plan.getPath());

        //显示最佳级别

        map.setViewport(plan.getPath());

    }

    //显示公交搜索结果

    document.getElementById("resultDiv").appendChild(resultList);

}

//公交线路, pointsStr 表示经纬度字符串, type 表示线路类型1, 步行; 2, 公交; 3, 地
铁; 4, 地铁站内换乘, lnglat 表示显示公交或地铁图标的经纬度

function createRoute(lnglats,lineColor)

{

    //创建线对象

    var line = new TPolyline(lnglats,{strokeColor:"#2C64A7", strokeWeight:5,
strokeOpacity:0.9});

    //向地图上添加线

    map.addOverLay(line);

}

//添加起始点

function createStartMarker()

{

    //向地图上添加起点

    var icon = new TIcon(startIcon,new TSize(24,24),{anchor:new TPixel(12,12)})

    var startMarker = new TMarker(startLngLat,{icon:icon});

    map.addOverLay(startMarker);

    //向地图上添加终点

    var icon = new TIcon(endIcon,new TSize(24,24),{anchor:new TPixel(12,12)})

    var endMarker = new TMarker(endLngLat,{icon:icon});

```

```

        map.addOverLay(endMarker);
    }

    //定位关键点
    function showPosition(lng,lat,des)
    {
        if(infoWin)
        {
            map.removeOverLay(infoWin);

            infoWin = null;
        }

        var lnglat = new TLatLng(lng,lat);

        infoWin=new TInfoWindow(lnglat,new TPixal([0,0]));

        infoWin.setLabel(des);

        map.addOverLay(infoWin);

        //打开信息窗口时地图自动平移

        infoWin.enableAutoPan();
    }

    function mouseUpStartMaker(point)
    {
        //向地图上添加起点

        var icon = new TIcon(startIcon,new TSize(24,24),{anchor:new TPixal(14,28)})

        var startMarker = new TMarker(point,{icon:icon});

        map.addOverLay(startMarker);

        //关闭标点工具

        startTool.close();

        //设置公交搜索起点

        document.getElementById("start").value = point.getLng()+" "+point.getLat();
    }

```



```

function mouseUpEndMaker(point)
{
    //向地图上添加起点

    var icon = new TIcon(endIcon,new TSize(24,24),{anchor:new TPixel(14,28)})

    var endMarker = new TMarker(point,{icon:icon});

    map.addOverlay(endMarker);

    //关闭标点工具

    endTool.close();

    //设置公交搜索终点

    document.getElementById("end").value = point.getLng()+","+point.getLat();

}

//获得驾车路线策略

function getRadioValue()
{
    var obj = document.getElementsByName("planType");

    for(var i=0;i<obj.length;i++)
    {
        if(obj[i].checked)
        {
            return obj[i].value;
        }
    }
}

```

</script>

</head>

<body onLoad="onLoad()" style="position:relative;margin: 0px">

<div id="mapDiv" style="position:absolute;width: 100%; height:860px;z-index: -1"></div>

<div style="position:fixed;top: 50px;right: 50px;z-index: 1">

```
<!-- 查询面板 -->

<div style="font-size:13px; border:1px solid #999999; line-height:27px;
padding-left:7px;background:white">

    <input type="radio" name="planType" value="0" checked="checked"/> 最少时间
    <input type="radio" name="planType" value="1"/> 最短距离
    <input type="radio" name="planType" value="2"/> 避开高速
    <input type="radio" name="planType" value="3"/> 步行
    <br/>
    起点: <input type="text" id="start" value="118.79365,32.08950"/>
    <input type="button" value="起点" onClick="startTool.open()"/>
    <br/>
    终点: <input type="text" id="end" value="118.63752,32.07751"/>
    <input type="button" value="终点" onClick="endTool.open()"/>
    <br/>
    <input type="button" onClick="searchDrivingRoute()" value="驾车路线搜索"/>
</div>

<br/>

<!-- 结果面板 -->

<div id="resultDiv" style="display: none;height:250px; width:300px;
overflow-y:scroll;background:white"></div>

</div>

</body>

</html>
```

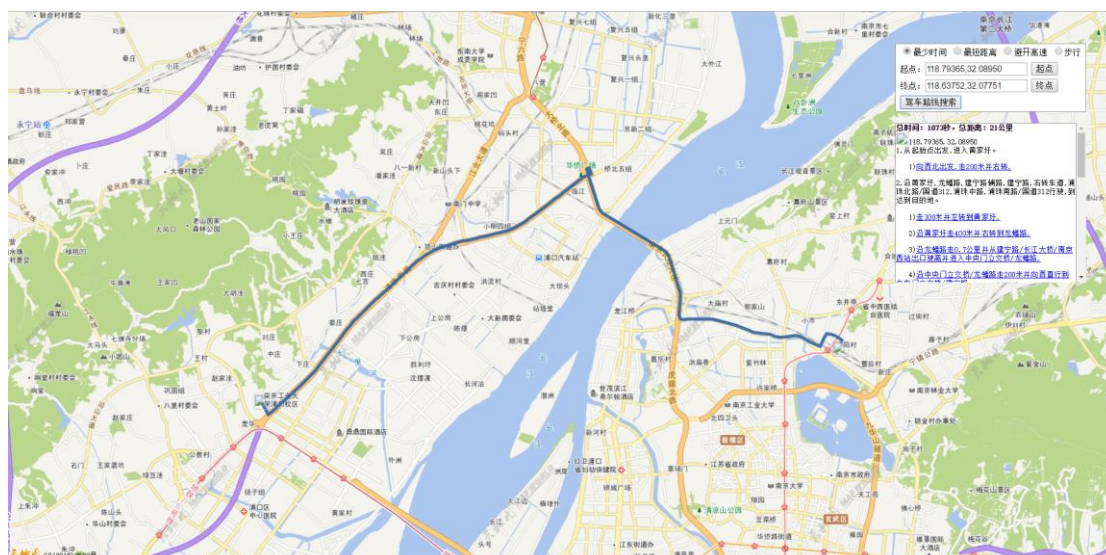


图9-6 驾车路线