参考文档

|  |
| --- |
| 官网教程(<https://jsplumbtoolkit.com/demos.html>)  官网API(http://jsplumbtoolkit.com/apidocs/  )  官网源码(https://github.com/jsplumb/jsplumb  )  mxGraph(https://www.2cto.com/kf/201605/504591.html)  jsPlumb开发入门教程（实现html5拖拽连线）(http://www.cnblogs.com/dyllove98/archive/2013/07/28/3221763.html)  jsPlumb.jsAPI阅读笔记(官方文档翻译)(http://www.cnblogs.com/leomYili/p/6346526.html)  jsPlumb（1） - 开始使用(https://segmentfault.com/t/jsplumb/blogs)  jsPlumb（2）-配置  jsPlumb（3）-基本概念  jsPlumb（4）-元素拖放  jsPlumb之流程图项目总结及实例(http://www.cnblogs.com/leomYili/p/7145445.html)  jsPlumb 学习笔记(http://blog.csdn.net/dananhai381/article/details/38870615) |

参考效果图

|  |
| --- |
|  |

如何正确引用js插件

|  |
| --- |
| <script type="text/javascript" src="js/jquery-3.1.1.js"></script> <script type="text/javascript" src="js/jquery-ui-1.11.2.min.js"></script> <script type="text/javascript" src="js/jquery.jsPlumb-1.7.2.js"></script> |

如何正确使用内置对象实例

|  |
| --- |
| jsPlumb.ready(**function**() {  }); |

获取内置对象实例

|  |
| --- |
| **var** instance = jsPlumb.getInstance(); |

当前页面的html布局如下

|  |
| --- |
| <style>.cc {position: relative; width:100%;height: 600px;}  .item {  background-color: #ddddff;  width: 80px;  height: 40px;  position: absolute;  }   #mysql\_in {  left: 20px;  top: 20px;  }   #mysql\_out {  left: 20px;  top: 100px;  }   #state3 {  left: 100px;  top: 250px;  } </style>  <div class="cc">  <div id="mysql\_in" class="item">mysql数据源</div>  <div id="mysql\_out" class="item">mysql输出源</div> </div> |

效果如下

|  |
| --- |
|  |

1. 使mysql\_in和mysql\_out元素可被任意拖拽

|  |
| --- |
| **var** instance = jsPlumb.getInstance(); instance.draggable(jsPlumb.getSelector(".item"), { containment:".cc"}); |

1. 要使某一个元素可被拖拽,其父元素需要满足某些必备的样式

|  |
| --- |
| .cc {position: relative; width:100%;height: 600px;} |

1. 给mysql\_in和mysql\_out元素分别添加可连线的锚点

|  |
| --- |
| **var** color1 = "#316b31"; **var** exampleEndpoint1 = {  endpoint:["Dot", { radius:8 }],//设置连接点的形状为圆形  paintStyle:{ fillStyle:color1 },//设置连接点的颜色  isSource:**true**, //是否可以拖动（作为连线起点）  scope:"green dot",//连接点的标识符，只有标识符相同的连接点才能连接  connectorStyle:{ strokeStyle:color1, lineWidth:6 },//连线颜色、粗细  connector: ["Bezier" ,{ curviness:10 }],//设置连线为贝塞尔曲线  maxConnections:1,//设置连接点最多可以连接几条线  isTarget:**true** //是否可以放置（作为连线终点） };  instance.addEndpoint("mysql\_in", { anchor: "TopCenter" }, exampleEndpoint1);//将exampleEndpoint1类型的点绑定到id为state1的元素上 instance.addEndpoint("mysql\_out", { anchor: "RightMiddle" }, exampleEndpoint2);//将exampleEndpoint1类型的点绑定到id为state1的元素上 |

效果如下

|  |
| --- |
|  |

1. 给mysql\_in和mysql\_out元素再次分别添加可连线的锚点

|  |
| --- |
|  |