

二次开发资源构件开发指南

一、什么是资源构件？

在开发自定义窗体或是封装二次开发控件时，往往会遇到以下一些场景：

A、抽提公用的逻辑 B、引入第三方资源（js 库、样式库等）等等

为了处理这类需求，我们提出了资源构件。

二、创建资源构件

1. 创建构件模板

使用 npm 方式安装 v3-cli

```
npm i v3-cli --save-dev
```

待安装完 v3-cli 插件后，将会出现安装向导，根据安装向导完成创建构件模板。

2. 命令说明

- i. **v3:package**
使用 webpack 打包资源构件
- ii. **v3:pack**
使用 webpack 打包资源构件，并将打包后结果封装成 vstore 构件
- iii. **v3:publish**
部署构件到 vstore 仓库
- iv. **v3:apply**
将资源构件安装到指定的执行系统中
- v. **v3:install**
安装 vstore 构件（二次开发控件、资源构件）
- vi. **v3:uninstall**
卸载 vstore 构件（二次开发控件、资源构件）

三、样例

(一) 引入第三方 ui (element-ui)

1. 创建 element-ui-lib 构件资源

(1) 创建 element-ui-lib 文件夹，在该文件夹下执行命令：npm i v3-cli --save-dev

```
D:\Workspace\nodejs\element-ui-lib>npm i v3-cli --save-dev
```

(2) 安装完成后，会出现向导，按照需求填入相关信息。

注意：插件类型请选择 resource：

```
? 插件类型( widget:二次开发控件 resource:资源构件 ):  
  widget  
> resource
```

(3) 在 element-ui-lib 中安装 element-ui 插件：npm i element-ui --save

```
D:\Workspace\nodejs\element-ui-lib>npm i element-ui --save
```

(4) 在 element-ui-lib/index.js 添加对 element-ui 的使用，此处使用方式参考 element ui 官网：

```
import 'element-ui';  
import 'element-ui/lib/theme-chalk/index.css';
```

此时，element-ui-lib 资源构件封装完成。

2. 部署到 vstore 仓库

使用命令将资源构件部署到 vstore 仓库中：

```
D:\Workspace\nodejs\element-ui-lib>npm run v3:publish
```

3. 自定义窗体中使用 element-ui-lib 构件资源

A. 执行系统处理

在执行系统中安装 element-ui-lib 资源构件（将来此步可以省略，目前正在处理）。
构件管理-》构件动态安装(动态升级)-》选择对应的库-》选择批量安装列表

同望V3基础平台 - 构件动态安装(动态升级)

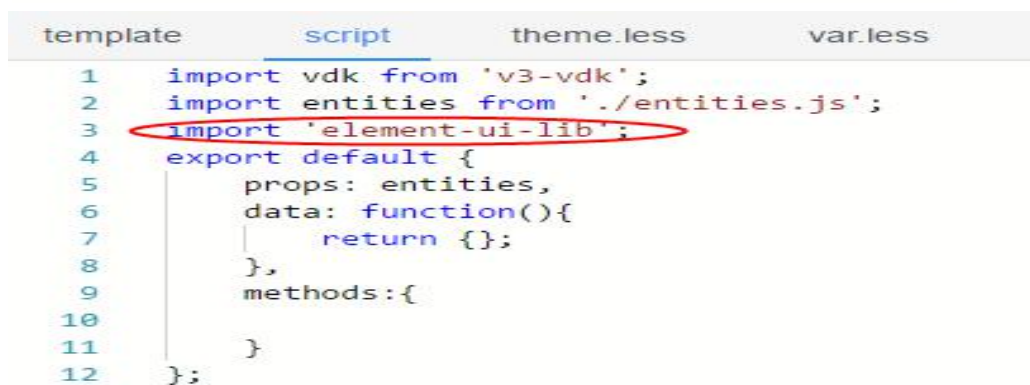


B. 开发系统处理

在开发系统中安装 element-ui-lib 资源构件。



在自定义窗体中使用资源构件。



部署预览即可。

4. 二次开发控件中使用 **element-ui-lib** 构件资源

以开发 `element-ui-button` 标签为例。

使用 `v3-cli` 创建二次开发控件模板。

使用 `v3:install` 安装 `element-ui-lib` 资源构件

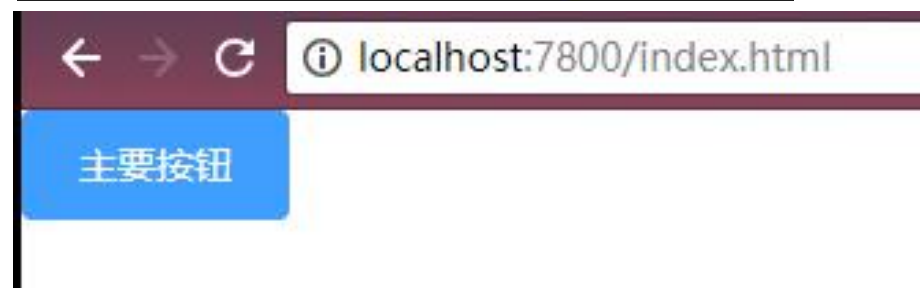
```
D:\Workspace\nodejs\element-ui-button> npm run v3:install element-ui-lib
```

在标签中引入 `element-ui-lib` 资源构件。

```
<template>
  <el-button type="primary">主要按钮</el-button>
</template>
<script>
  import 'element-ui-lib';
  export default {
    name : "element-ui-button",
    props : {
    },
    data : function(){
      return {};
    },
    methods : {
    }
  }
</script>
<style scoped lang="less" src="./theme.less"></style>
```

使用命令 `v3:test` 命令进行测试预览：

```
D:\Workspace\nodejs\element-ui-button> npm run v3:test
```



（二）引入第三方工具方法（lodash）

1. 创建 lodash-lib 构件资源

(1) 创建 lodash-lib 文件夹，在该文件夹下执行命令：npm i v3-cli --save-dev

```
PS D:\Workspace\nodejs\lodash-lib> npm i v3-cli --save-dev
```

(2) 安装完成后，会出现向导，按照需求填入相关信息

注意：插件类型请选择 resource：

```
? 插件类型( widget:二次开发控件 resource:资源构件 ):  
widget  
> resource
```

(3) 在 lodash-lib 中安装 lodash 插件：npm i lodash --save

```
D:\Workspace\nodejs\lodash-lib> npm install lodash --save
```

(4) 在 lodash-lib/index.js 添加对 lodash 的使用：

```
import _ from 'lodash';  
  
export default _;
```

此时，lodash-lib 资源构件封装完成。

2. 部署到 vstore 仓库

使用命令将资源构件部署到 vstore 仓库中：

```
D:\Workspace\nodejs\lodash-lib> npm run v3:publish
```

3. 自定义窗体中使用 lodash-lib 构件资源

A. 执行系统处理

在执行系统中安装 lodash-lib 资源构件（将来此步可以省略，目前正在处理）。
构件管理-》构件动态安装(动态升级)-》选择对应的库-》选择批量安装列表

同望V3基础平台 - 构件动态安装(动态升级)

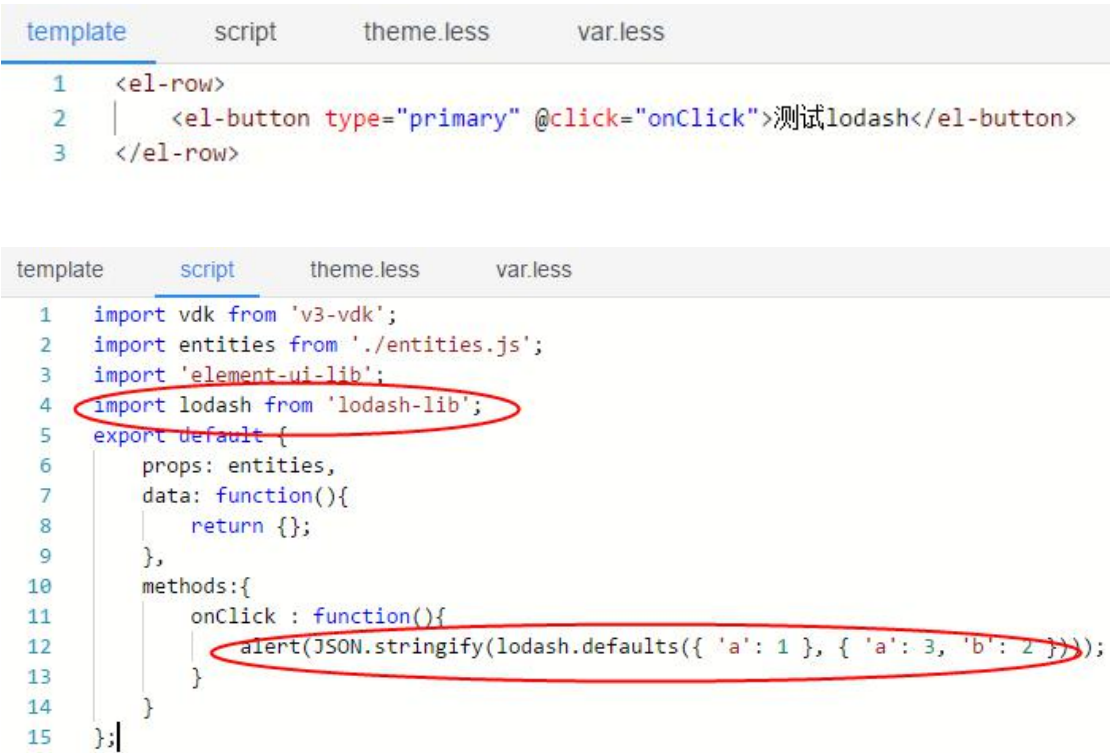


B. 开发系统处理

在开发系统中安装 lodash-lib 资源构件。



在自定义窗体中使用资源构件。



部署预览即可。

4. 二次开发控件中使用 lodash-lib 构件资源

以开发 element-ui-button 标签为例。

使用 v3-cli 创建二次开发控件模板。

使用 v3:install 安装 lodash-lib 资源构件

```
D:\Workspace\nodejs\element-ui-button> npm run v3:install lodash-lib
```

在标签中使用 lodash-lib 资源构件。

```
<template>
  <el-button type="primary" @click="onClick">主要按钮</el-button>
</template>
<script>
import 'element-ui-lib';
import lodash from 'lodash-lib';
export default {
  name : "element-ui-button",
  props : {
  },
  data : function(){
    return {};
  },
  methods : {
    onClick:function(){
      alert(JSON.stringify(lodash.defaults({ 'a': 1 }, { 'a': 3, 'b': 2 })));
    }
  }
}
</script>
<style scoped lang="less" src="./theme.less"></style>
```

使用命令 v3:test 命令进行测试预览：

```
D:\Workspace\nodejs\element-ui-button> npm run v3:test
```



（三）本地引入第三方 js 库（Swiper）

前往 swiper 官网下载，目前使用 swiper2 作为例子，下载网址：
<https://www.swiper.com.cn/download/index.html#file1>

得到两个资源文件：

- 1、swiper-4.3.5.min.css
- 2、swiper-4.3.5.min.js

1. 创建 swiper-ui-lib 构件资源

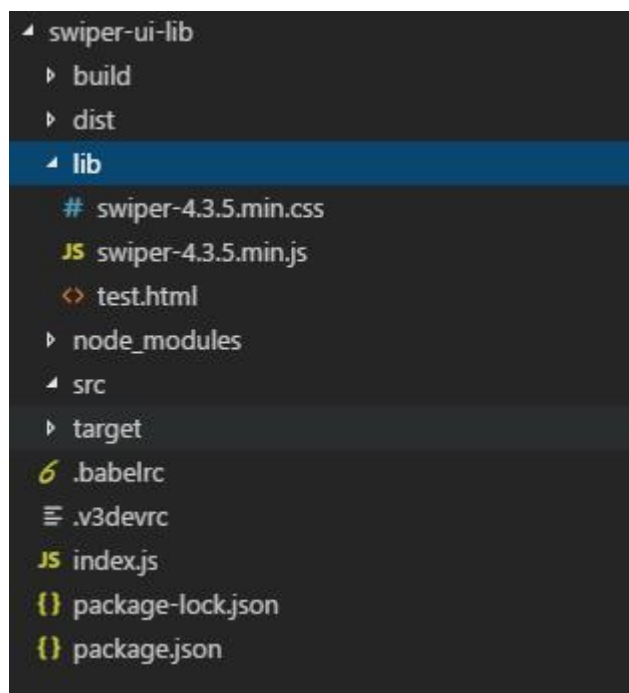
创建 swiper-ui-lib 文件夹，在该文件夹下执行命令：npm i v3-cli --save-dev

```
PS D:\Workspace\nodejs\swiper-ui-lib> npm i v3-cli --save-dev
```

安装完成后，会出现向导，按照需求填入相关信息。**注意：插件类型请选择 resource：**

```
? 插件类型( widget:二次开发控件 resource:资源构件 ):  
  widget  
> resource
```

在 swiper-ui-lib 文件夹中创建 lib 文件，将 swiper-4.3.5.min.css、swiper-4.3.5.min.js 两个文件拷贝至该文件夹中。



打开.v3devrc 文件，调整 extraSources 节点配置。**注意：如果文件有加载顺序要求，则在此按照顺序配置。**


```

{
  "libType": "dev",
  "type": "resource",
  "pluginCode": "swiper-ui-lib",
  "groupId": "com.toone.v3.platform",
  "matchVersion": 1,
  "symbolicName": "com.toone.v3.platform-vjs.framework.extension.platform.custom.resource.swiper-ui-lib",
  "version": "1.0.0",
  "minMatchVersion": 1,
  "resourceCode": "vPlatform-resource-c80593740499f63f2f3c89a0538aa983",
  "out": "target",
  "sources": "dist",
  "extraSources": [
    "lib/swiper-4.3.5.min.css",
    "lib/swiper-4.3.5.min.js"
  ],
  "dependencies": [],
  "VjsList": [],
  "depVjsList": [],
  "require": [],
  "libs": [],
  "account": "xiedh",
  "pwd": "summer"
}

```

此时，swiper-ui-lib 资源构件封装完成。

2. 部署到 vstore 仓库

使用命令将资源构件部署到 vstore 仓库中：

```
PS D:\Workspace\nodejs\swiper-ui-lib> npm run v3:publish
```

3. 自定义窗体中使用 siwper-ui-lib 资源构件

A. 执行系统处理

在执行系统中安装 swiper-ui-lib 资源构件（将来此步可以省略，目前正在处理）。

构件管理-》构件动态安装(动态升级)-》选择对应的库-》选择批量安装列表

同望V3基础平台 构件动态安装(动态升级)

<

B. 开发系统处理

在开发系统中安装 swiper-lib 资源构件。



在自定义窗体中使用资源构件。参考：<https://www.swiper.com.cn/usage/index.html>

```

template      script      theme.less      var.less
1  <div class="swiper-container">
2      <div class="swiper-wrapper">
3          <div class="swiper-slide">Slide 1</div>
4          <div class="swiper-slide">Slide 2</div>
5          <div class="swiper-slide">Slide 3</div>
6      </div>
7      <!-- 如果需要分页器 -->
8      <div class="swiper-pagination"></div>
9  </div>

```

```

template      script      theme.less      var.less
1  import vdk from 'v3-vdk';
2  import entities from './entities.js';
3  import swiper from 'swiper-ui-lib';
4  export default {
5      props: entities,
6      data: function(){
7          return {};
8      },
9      methods:{
10     },
11     mounted:function(){
12         new Swiper('.swiper-container', {
13             direction: 'vertical',
14             loop: true,
15
16             // 如果需要分页器
17             pagination: {
18                 el: '.swiper-pagination',
19             }
20         })
21     }
22 };

```

部署预览即可。

（四）在线请求第三方 js（Map）

前往百度地图官网，先注册百度账号，登陆 API 控制台；申请到密钥；使用百度地图 API。得到百度地图 api 的 js 文件路径。

1.创建 map-source 构件资源

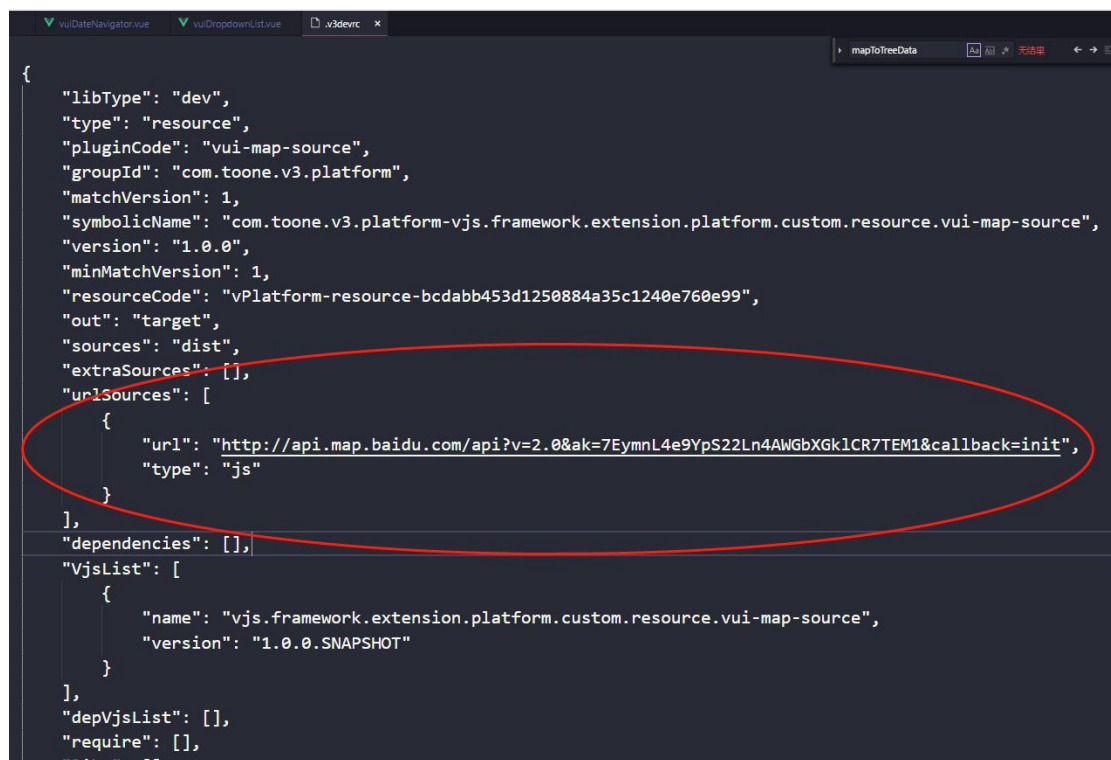
创建 map-source 文件夹，在该文件夹下执行命令：npm i v3-cli --save-dev

```
PS D:\work\v平台二次开发\vui\测试\map-source> npm i v3-cli --save-dev
```

安装完成后，会出现向导，按照需求填入相关信息。**注意：插件类型请选择 resource：**

```
? 插件类型( widget:二次开发控件 resource:资源构件 ):
widget
> resource
```

打开.v3devrc 文件，调整 urlSources 节点配置。



```
{
  "libType": "dev",
  "type": "resource",
  "pluginCode": "vui-map-source",
  "groupId": "com.toone.v3.platform",
  "matchVersion": 1,
  "symbolicName": "com.toone.v3.platform-vjs.framework.extension.platform.custom.resource.vui-map-source",
  "version": "1.0.0",
  "minMatchVersion": 1,
  "resourceCode": "vPlatform-resource-bcdabb453d1250884a35c1240e760e99",
  "out": "target",
  "sources": "dist",
  "extraSources": [],
  "urlSources": [
    {
      "url": "http://api.map.baidu.com/api?v=2.0&ak=7EymnL4e9YpS22Ln4AWGbXGk1CR7TEM1&callback=init",
      "type": "js"
    }
  ],
  "dependencies": [],
  "VjsList": [
    {
      "name": "vjs.framework.extension.platform.custom.resource.vui-map-source",
      "version": "1.0.0.SNAPSHOT"
    }
  ],
  "depVjsList": [],
  "require": [],
  "init": []
}
```

此时， map-source 资源构件封装完成。

2.部署到 vstore 仓库

使用命令将资源构件部署到 vstore 仓库中：

```
PS D:\work\v平台二次开发\vui\测试\map-source> npm run v3:publish
```

3. 二次开发控件中使用 map-source 构件资源

以开发 vui-m-map 标签为例。
使用 v3-cli 创建二次开发控件模板。
使用 v3:install 安装 map-source 资源构件

```
PS D:\work\移动端二次开发\svnDownload\mobile\vui-m-map> npm run v3:install map-source
```

就可以进行百度地图相关的构件二次开发。

4. 自定义窗体中使用 map-source 资源构件

A. 执行系统处理

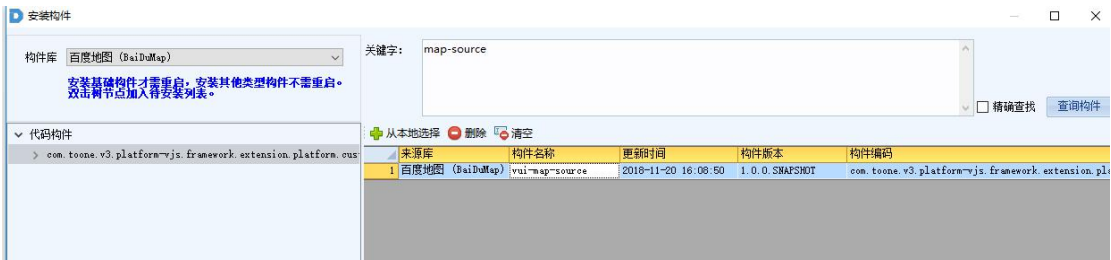
在执行系统中安装 map-source 资源构件（将来此步可以省略，目前正在处理）。
构件管理-》构件动态安装(动态升级)-》选择对应的库-》选择批量安装列表

同望V3基础平台 - 动力平台构件仓库

Main OSGI Status 构件管理 数据管理 移动端管理 系统维护 系统诊断 关于									
[代码类]构件信息: 总共有11个构件 百度地图(BaiduMap) [切换到配置类构件] 系统打包库: 测试库(内测) 注: 开发库或测试库的构件是不稳定的, 请慎重安装、升级!									
全部 已安装 未安装 可升级 批量升级 批量安装									
构件标识	构件名称	库中版本	已安装版本	库中版本时间	当前版本时间	构件说明	开发者	状态	操作
com.toone.v3.platform-vjs.framework.extension.platform.custom.widget.vui-m-map	vui-m-map	1.0.0.SNAPSHOT	1.0.0.SNAPSHOT	2018-11-21 13:52:28	2018-11-21 13:53:01	移动版地图	陈家宝	已安装	
com.toone.v3.platform-vjs.framework.extension.platform.custom.widget.vui-m-map	vui-m-map	1.0.0.SNAPSHOT	1.0.0.SNAPSHOT	2018-11-21 13:49:58	2018-11-21 13:52:42	web版地图	陈家宝	已安装	
com.toone.v3.platform-vjs.framework.extension.platform.custom.resource.vui-map-source	map-source	1.0.0.SNAPSHOT	1.0.0.SNAPSHOT	2018-11-20 16:08:50	2018-11-20 16:14:27	v3资源构件	陈家宝	已安装	
com.toone.v3.platform-vjs.framework.extension.platform.native.cordova.baidutracer	vjs.framework.extension.platform.native.cordova.baidutracer	3.4.0.SNAPSHOT		2018-11-17 16:37:51		cordova插件: 媒体捕获	vbase	未安装	
com.toone.v3.platform-Webfunc	Webfunc_GetDeviceTra	3.3.0.SNAPSHOT		2018-11-17 14:0		2018-11-16wangyue: 获取百度地图	vbase	未安装	

B. 开发系统处理

在开发系统中安装 map-source 资源构件。



在自定义窗体中使用资源构件。

template	script	theme.less	var.less
1	<vui-page>		
2	<!--设置驾车策略-->		
3	<h3>waypoints&policy策略</h3>		
4	<vui-m-map height="300px">		
5	<vui-m-map-route routeway="driving" start="同望公司" end="拱北口岸" startCity="珠海" endCity="珠海" :auto-viewport="true" :panel="false" color="#111" policy="BMAP_DRIVING_POLICY_LEAST_TIME">		
6	</vui-m-map-route>		
7	</vui-m-map>		
8			
9			
10	</vui-page>		

template	script	theme.less	var.less
1	import vdk from 'v3-vdk';		
2	import entities from './entities.js';		
3	import map from 'map-source'		
4	export default {		
5	props: entities,		
6	data: function(){		
7	return {};		
8	},		
9	methods:{		
10			
11	},		
12	mounted(){		
13	new BMap.ScaleControl({		
14	anchor: argus[scaleAnchor],		
15	offset: scaleOffset && createSize(BMap, scaleOffset)		
16	})		
17	}		
18	};		

部署预览即可。

waypoints&policy策略

