YonghongTech
Talk with Data

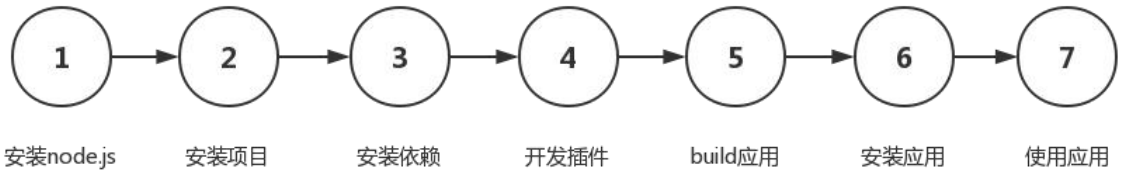
应用开发说明

Yonghong Z-Suite — V8.5.1

目录

应用开发说明文档

1. 开发流程说明



（1）安装 node.js：

教程详见：https://nodejs.org/en/download/

（2）安装项目：

新建任意一个文件夹，运行命令：npm install yh-plugin-project@latest

版本说明：如基于 8.5.0.X 版本插件开发，请使用 yh-plugin-project 版本为 1.0.6

（3）安装依赖：

在 yh-plugin-project 目录运行命令：npm install

模块说明：

① yh-plugin-manager 是必须引用的模块，不可以删除，并需要运行命令：

npm install yh-plugin-manager@latest 将其安装到最新版本；

② 如基于 8.5.0.X 版本开发插件，使用的 yh-plugin-manager 版本为 0.0.4；

③ ECharts 已默认添加，如果不需要的可以手动删除

（4）开发插件：

开发时用到的 API 以及示例详见“开发 API”介绍

（5）build 应用：

在 webpack.config.js 和 webpack.config.production.js 两个文件里配置文件名称和结果。其中 exportsMap 是配置应用文件的数组，开发者新建一个应用只需要在后面添加即可。build 步骤如下：

①运行命令： npm run plugin-build 将在 result 目录下生成需要的文件：配置的文件名称.js；

②运行命令： npm plugin-build-p 生成用于发布的结果；

③将 build 之后的 jar 目录下的以 jar 为后缀的文件上传的应用市场：

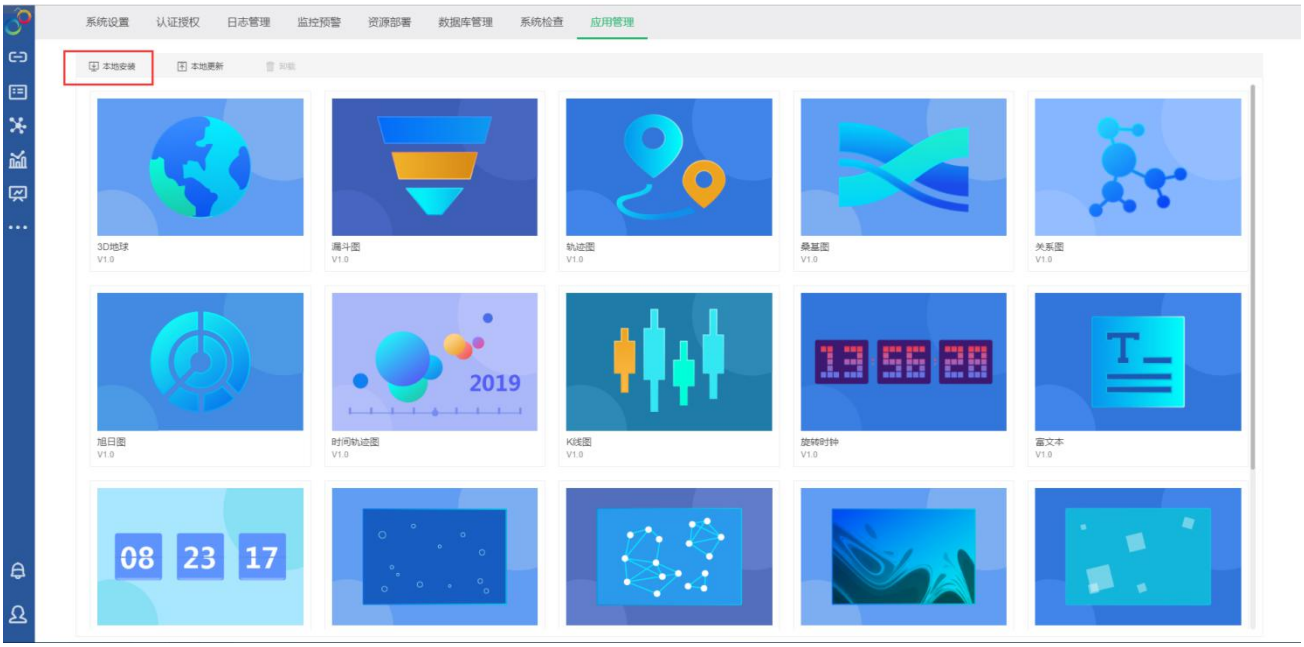
A.注册登录应用市场 http://plugins.yonghongtech.com/

B.点击开发者应用打包。

根据要求填写信息，上传 build 后的 jar 包，相关人员审核后会打包通过邮件发送应用包。
注：如果为个人开发者，无部署 ID，可以填写 20190302003DEPT 代替。

（6）安装应用：

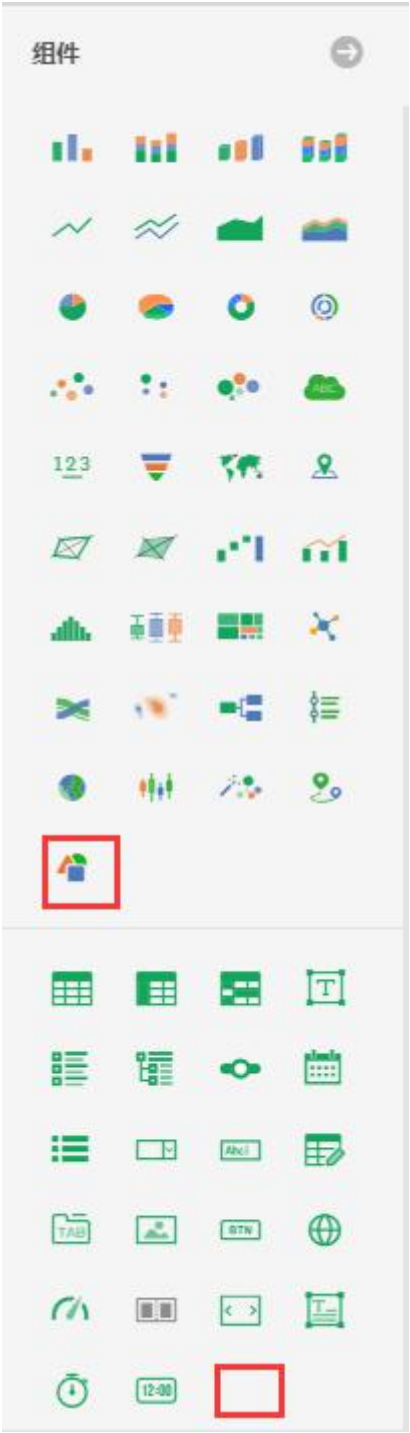
进入产品-管理系统-应用管理，点击本地安装，选择应用宝安装即可。



（7）使用应用：

安装成功后，刷新编辑报告即可。

A. 如果是组件类应用，不用重启 tomcat 就能看到新添加的应用显示在新建组件区域。



上半部分是 chart，下半部分是 element。新安装的应用会在列表尾部。

B. 如果是其他类别应用，则根据具体应用功能而定，目前仅支持后台功能应用，暂不支持前端功能应用开发。

2. 开发 API

（1）Hello World：

开发插件的示例：

```
import {register} from 'yh-plugin-manager';
import echarts from "echarts";

const plugin = {
    getProps: () => {}, //定义插件所需的属性
    getLocalView: (key) => {}, //定义本地化信息
    getBindInfo: () => {}, //定义插件所需的绑定信息
    render: (domName, props, grid, qinfo) => {}, //渲染组件

    clearAll: (domID) => {} //释放资源
}
register(plugin); // 注册组件
```

（2）API：

①getProps

getProps: () => {}，定义插件所需的属性。

参数：

无。

返回值：

Object[]，插件所需要的属性组。属性可支持以下字段：

字段	类型	描述
type（必须）	String	定义属性的类型，支持的类型有： ["number", "string", "color", "line", "integer", "boolean", "combo", "font", "param", "chartColor", "rotate", "alpha", "size"]，其中类型 combo 属性也需要数组选项。大小写敏感。
key（必须）	String	定义属性的 key
value（可选）	与 type 对应	定义属性的默认值

group（可选）	String	定义属性所属的组，多个属性作为组属性
option（可选）	Array	类型 combo 特有的属性，表示所有选项。
visible（可选）	Boolean	定义属性的可见性（8.7+）
position（可选）	pos	定义属性所在的位置（9.0+），可选值为：“setting”：对应设置页，“format”：对应格式页（默认值）“graph”：对应图形页

示例：

```
GetProps: () = {
   return
    [{
       type: ‘number’,
       key: number1,
       value: 1.222
    },
     {
       type: ‘string’,
       key: title,
       value: ‘标题’,
       group: ‘group1’
    },
     {
       type: ‘combo’,
       key: ‘titlePosition’,
       value:‘标题位置’,
       option: [‘top’, ‘bottom’],
       group:‘group1’
    },
     {
       type:‘rotate’,
       key: ‘xuanzhuan’,
       value: 0 (0, 1, 2, 3, 4, 5, 6, 7, 8)
    },
    {
       group: ‘titleOption’,
       type: COMBO,
       key: ‘titlePosition’,
       Value: ‘top’,
```

```
            position: GRAPH, //9.0+ support
            option: [ 'top' ,  'bottom' ]
        },
        {
            type: STRING,
            key:  '_PLUGIN_EFFECT' ,
            value:  'PARAMETER' ,
            visible: true //8.7+ support
        }]
    }
```

②getLocalView

getLocalView: (key) => {} 定义本地化信息。

参数:

key,需要本地化的内容。定义 local 信息方式为 key: object，根据 key 获取 object；key 对应于 props 中 key，可不完全定义，没有定义的表明为 null,不渲染说明信息；object 是包含四种语言 key 的对象。

返回值:

Object,本地化信息。

字段	类型	描述
key（必须）	string	需要本地化的内容

示例：

```
    getLocalView: (key) => {
        const LocalSetting = {
            "PluginName": {"zh_CN": "漏斗图", "en_US": "Funnel", "ja_JP": "漏斗图", "zh_TW": "漏斗图"},
            "title": {"zh_CN": "标题", "en_US": "Title", "ja_JP": "タイトル", "zh_TW": "标题"},
            "color": {"zh_CN": "颜色", "en_US": "Color", "ja_JP": "色彩", "zh_TW": "颜色"},
            "font": {"zh_CN": "字体", "en_US": "Font", "ja_JP": "文字", "zh_TW": "字体"},
            "line": {"zh_CN": "线", "en_US": "Line", "ja_JP": "线", "zh_TW": "线"},
            "border": {"zh_CN": "边框", "en_US": "Border", "ja_JP": "边框", "zh_TW": "边框"},
            "group1": {"zh_CN": "组 1", "en_US": "group1", "ja_JP": "组 1", "zh_TW": "组 1"},
        }
        return LocalSetting[key];
    }
```

③getBindInfo

getBindInfo: () => {} ，定义插件所需的绑定信息。

参数：

无。

返回值：

包含三组列绑定信息和名称说明信息（名称信息 V8.8 支持）：xcolumns， ycolumns，zcolumns，zcolumnName（8.8+，对应的 value 为第三列即 z 列的名称）；

字段	类型	描述
xcolumns	Array	定义 x 列，数值为列 Object 数组
ycolumns	Array	定义 y 列，数值为列 Object 数组
zcolumns	Array	定义 z 列，数值为列 Object 数组
zcolumnName	String	定义 z 列的名称（8.8+）

每个 coulmn 数据列结构：

字段	类型	描述
index	Int	序号
localView	String	本地化 key
isDiscreted	Boolean	绑定数据为离散，度量时转为聚合类型为空
isDim	Boolean	绑定的数据类型
isBindFree	Boolean	绑定列是否受重复列绑定的限制

示例：

```
getBindInfo: () => {
  return
    {xcolumns:
       [{
          index: 0,
          isDim: true
       }],
     ycolumns:
      [{
          index: 0,
          isDim: false
       }],
```

```
    zcolumns:
     [{
         index: 0,
         isDim: true
      }],
     zcolumnName: 'title' //8.8+ support
    };
}
```

④render

render: (domName, props, grid, qinfo) => {}，渲染组件。

参数：

domName - 插件渲染的 dom id

props - 插件的 props 信息

grid - 插件绑定生成的数据信息

qinfo - 插件具体绑定信息

返回值：

无。

props 中包含 getprops 中信息：

```
width: // 插件宽(v8.5+)
height: // 插件高(v8.5+)
hasBg: // 插件是否有背景，null 以及白色为不含有背景，图片与双色是有背景(v8.5+)
isFullScreen: // 是否全屏(v8.5+)
themeObj:// 主题信息(v8.7+)
themeColor:// 主题颜色(v8.7+)
isMobile:// 是否为移动端(v8.5+)
iOS: // 是不是 ios 操作系统(v8.8+)
elemName: // 组件名称(v8.7+)
model : // elem model 信息(v8.7+)
filter: (dimFilter, negative) => {}(v8.5+)
local: // 系统当前语言 (v8.5+)
firstMondy: // 是否一周从周一开始(v8.7+)
```

⑤clearAll

clearAll: (domID) => {}，用于手动释放资源。

参数:

domID - 插件渲染的 dom id

返回值:

无。

⑥register

register 用于在模块中注册插件。

示例：

```
register(plugin);
```

⑦isValid（仅用于 8.5）

isValid(plugin) {}，检查插件是否有效。

参数：

plugin - 生成的插件。

返回值：

插件是否有效。

⑧getRegistered（仅用于 8.5）

getRegistered(key) {}获取模块中的 register 插件。

（3）功能性说明

①filter 函数

插件可用过自身数据过滤影响其他组件，例如漏斗图通过图例选择进行数据过滤，props.filter 函数接收一个数组，数组成员为键值对（<columnView: selectedKey>）：

```
funnel.on('legendselectchanged', function (params) { //监听'legendselectchanged'事件
        const result = [];
        let view = qinfo.xcols[0].col.getView(false);

        if(params.selected) {
            for(let key in params.selected) {
                params.selected[key] && result.push([{[view]: key}]);
            }
        }

        if(result.length == 0) {
            result.push([{[view]: null}]);
        }

        props.filter(result); //过滤出选中的数据
    });
```

②插件绑定数据说明

插件根据绑定的数据生成 grid，grid 类似于 table 数据，根据指定的行列可以获取对应的数据。可通过绑定同样的数据展示在 table 组件内进行参考：

漏斗图

流程步骤	转化率_总和
加购	0.938
交易完成	0.852
下单	0.355
支付	0.88
浏览	1

③参数功能说明（V8.7 版本及以后可用）

BI 产品 8.7 及以后版本支持插件作为参数组件，传入数据。步骤为：

getProps()函数中添加特定属性内容，可将插件指定为参数插件：

```
{
    type: string,
    key: “_pluginEffect_”,
    value: “parameter”,
    visible: false
}
```

其中：type 为 string，key 为约定值_pluginEffect_，value 为约定值 parameter，visible 表明属性是否在右侧属性设置中可见，建议为 false.

参数插件可支持生成两个参数，且生成的参数名为 组件名+“_Start”、组件名+“_End”。