

通过脚本自定义统计图表

通过脚本自定义统计图表

例如，“机器人任务”是一个定制业务对象，想统计成功失败率。

image.png

首先要定义和注册一个扩展图表组。首先在脚本启动时，用 addStatsGroup 注册新的统计图表组

```
function boot() {
```

```
    addStatsGroup(statsGroup)
```

```
}
```

statsGroup 变量是我们要定制的图表组，里面有一个图表项。定义如下： const statsGroup:
ChartGroup = { id: "Ext", label: "扩展", items: [

```
{
```

```
    id: STATS_EXT_PREFIX + "ExtRobotTask", label: "机器人任务最近三十日统计",
```

```
}
```

```
]
```

```
}
```

一个图表组里有一个或多个图表项。对于脚本定义的图表项，id 要加 **STATS_EXT_PREFIX** 前缀。让系统知道从脚本里取数据。界面需要数据时，会通过接口、服务层，最终调用脚本的 extStatsCalc 方法。这个方法主要传入图表项的 ID。返回 Echart 的配置对象。 function extStatsCalc(tc: TraceContext, id: string) {

// id 是图表项的 id，根据 id 决定计算什么数据

```
if (id != (STATS_EXT_PREFIX + "ExtRobotTask")) return
```

```
// 如果要统计最近 N 天数据，用这个辅助方法获取相关参数 const params =  
stats.buildLastDaysParams(30)
```

```
// 初始化数据，一个 Map，键是日志，值是一个对象，包含多个统计值。值对象的属性名需要与 dimensions 数组一致。 const data: MapToTyped<{ datetime: string, 创建数: number, 完成数: number, 异常数: number }> = {} for (const d of params.dates) data[d] = {datetime: d, 创建数: 0, 完成数: 0, 异常数: 0}
```

// 从辅助参数中获取开始结束时间，查询所有需要的数据

```
const evList = entity.findMany(tc, "ExtRobotTask", cq.and([cq.gte("createdOn", params.startInstant), cq.lte("createdOn", params.endInstant)]), entity.buildFindOptions(["createdOn", "status"], null, null, null)
```

```
)
```

// 进行汇总

```
for (const ev of evList) {
```

```
const dt = utils.formatDate(ev["createdOn"], "yyyy-MM-dd")
```

```
const sv = data[dt]
```

```
if (!sv) continue
```

```
+ +sv["创建数"]
```

```
const status = ev["status"]
```

```
if (status == "Done") + +sv["完成数"]
else if (status == "Cancelled" || status == "Failed") + +sv["异常数"]
```

```
}
```

```
base.logDebug(tc, data: ${JSON.stringify(data)})
```

// 将 Map 转化为数组，按日期排序

```
const source = Object.values(data)
```

```
source.sort((a, b) => {
```

```
return a.datetime > b.datetime ? 1 : a.datetime < b.datetime ? -1 : 0
```

```
}
```

```
// 一下就是 echart 的选项了，根据需要构造
```

```
const option = {
```

```
title: {
```

```
    text: "机器人任务最近三十日统计", x: "left", y: "top",  
},
```

```
legend: {
```

```
x: "right", y: "top"
```

```
},
```

```
dataset: {
```

```
    dimensions: ["datetime", "创建数", "完成数", "异常数"],  
    source,  
},
```

```
xAxis: {
```

```
type: "category"
```

```
},
```

```
yAxis: {
```

```
type: "value"
```

```
},
```

series: [

```
{type: "bar", label: {show: true, position: "top"}},  
{type: "bar", label: {show: true, position: "top"}},  
{type: "bar", label: {show: true, position: "top"}},
```

]

}

return JSON.stringify(option)

}
