

前端测试文档

2020 年 05 月 07 日

目录

Android 测试——221701414 黎家泽..... 3

一、单元测试..... 3

二、使用 LeakCanary 分析内存泄漏..... 4

Web 后台——221701420 程伟行..... 6

一、单元测试..... 6

二、性能测试..... 6

Web 前台——221701418 王肃南..... 8

一、单元测试..... 8

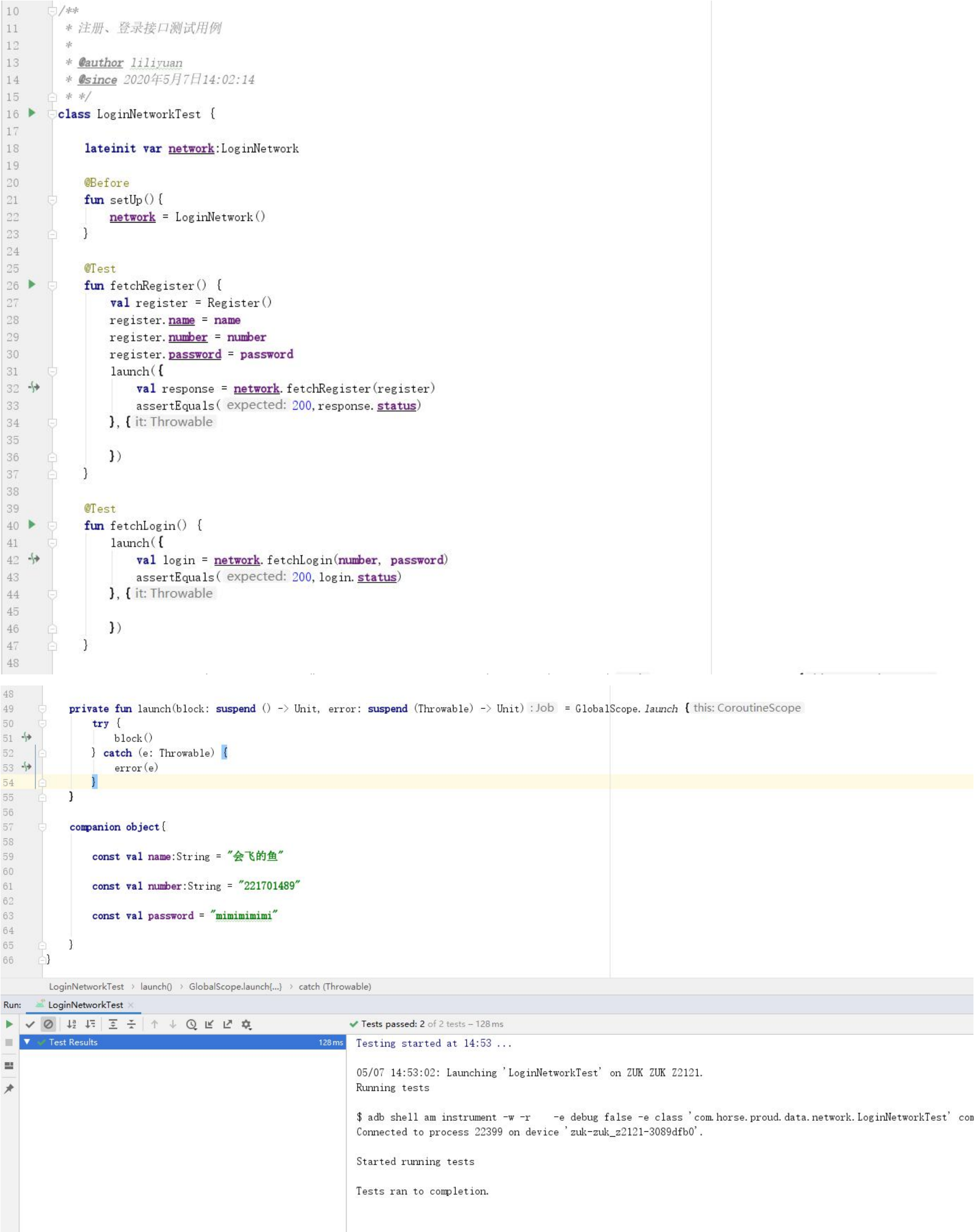
二、性能分析..... 8

一、单元测试

规范测试一直编写代码的过程中容易被忽视的一环，平时写代码不是没有测试，只是测试的方法五花八门，比如：直接打印日志、PostMan 测试、Toast 输出、甚至肉眼观察等，一直都没有养成规范测试的习惯——包括这一次 Alpha 冲刺。

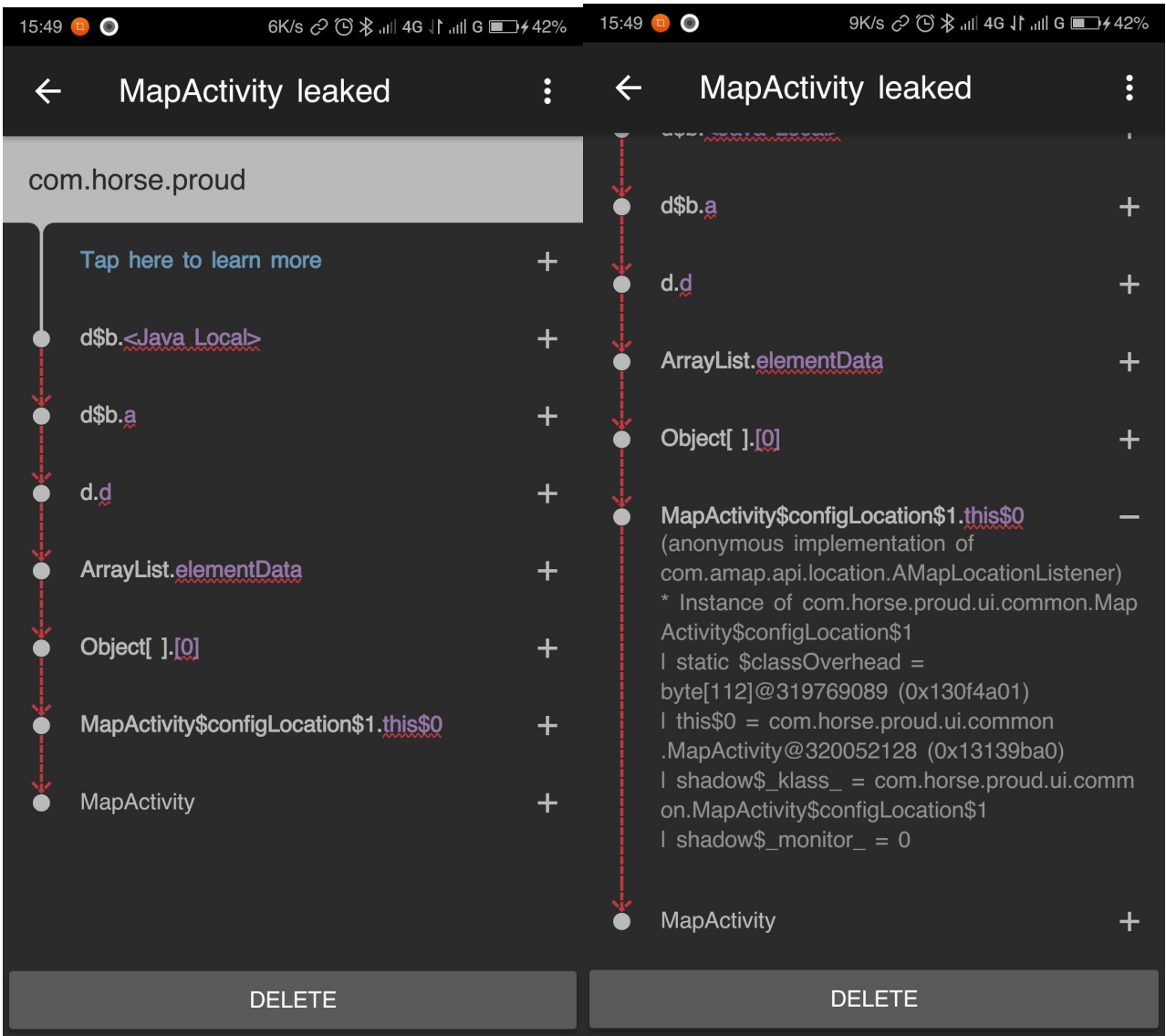
这一次 Alpha 冲刺时间紧、任务重，测试在 deadline 的最后一天才开始。在学习如何测试时我才意识到，Android 的测试和 Java 的测试有很多不同的地方，因为 Android 的测试需要真机的支持，所以我马上阅读文档入门了如何构建仪器单元测试。

测试用例以注册和登录为例说明测试过程。以下是代码截图：



二、使用 LeakCanary 分析内存泄漏

无论是在服务器还是在电脑、在手机，内存泄漏都是一件很严重的事情。因为我们的目标是做一个能上线发布的产品，所以这方面也尤为重要。我这次第一次尝试使用 LeakCanary 进行内存泄漏分析，以下展示一个用 LeakCanary 工具进行内存泄漏分析的过程。
首先是获取到内存泄漏报告：



通过分析初步猜测，是 MapActivity 中的 configLocation 方法发生了内存泄漏，阅读源代码：

```
95 private fun configLocation() {
96     mLocationClient = AMapLocationClient(Proud.getContext())
97     mLocationListener = AMapLocationListener { it: AMapLocation!
98         if(it.errorCode == 0) {
99             //定位成功
100             latitude = it.latitude
101             longitude = it.longitude
102
103             aMap.addMarker(MarkerOptions()
104                 .position(LatLng(latitude, longitude))
105                 .draggable(true)
106                 .alpha(0.75f)
107                 .icon(BitmapDescriptorFactory.fromBitmap(BitmapFactory.decodeResource(resources, R.drawable.local)))
108                 .setFlat(true))
109
110             aMap.moveCamera(CameraUpdateFactory.changeLatLng(LatLng(latitude, longitude)))
111         } else {
112             //定位失败
113             showToast( content: "定位失败: ${it.errorInfo}")
114             logWarn(TAG, msg: "${it.errorInfo}")
115         }
116     }
117     mLocationClient.stopLocation()
118     map.visibility = View.VISIBLE
119     mLocationListener = null
120 }
121 mLocationClient.setLocationListener(mLocationListener)
122
123 mLocationOption = AMapLocationClientOption()
124 mLocationOption.locationPurpose = AMapLocationClientOption.AMapLocationPurpose.SignIn
125 mLocationOption.locationMode = AMapLocationClientOption.AMapLocationMode.High_Accuracy
126 mLocationOption.isOnceLocationLatest = true
127 //设置请求超时时间
128 mLocationOption.httpTimeOut = 80000
129
130 mLocationClient.setLocationOption(mLocationOption)
131 mLocationClient.stopLocation()
132 mLocationClient.startLocation()
133 }
```

分析代码：在代码当中，设置了一个用于定位的回调监听器，但是这个监听器很特殊，因为只需要回调一次，即：定位成功后，将不再使用。百度发现，监听器还真的有创建和销毁的说法！我们知道，**Java** 的回收机制很神奇，它会自动回收没有用的对象！所以，我们只需要在获取定位后，将这个监听器设置为空就可以了！！！

一、单元测试

```
PASS tests/unit/example.spec.js
itemPage
  ✓ renders (26ms)

console.error node_modules/vue/dist/vue.runtime.common.dev.js:621
[Vue warn]: Unknown custom element: <el-amap> - did you register the component correctly? For recursive components, make sure to provide the "name" option.

found in

---> <Anonymous>
      <Root>

console.error node_modules/vue/dist/vue.runtime.common.dev.js:621
[Vue warn]: Unknown custom element: <el-amap-marker> - did you register the component correctly? For recursive components, make sure to provide the "name" option.

found in

---> <Anonymous>
      <Root>

console.log src/components/map.vue:26
undefined

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:       2.307s
Ran all test suites.
```

```
PASS tests/unit/example.spec.js
itemPage
  ✓ renders (12ms)

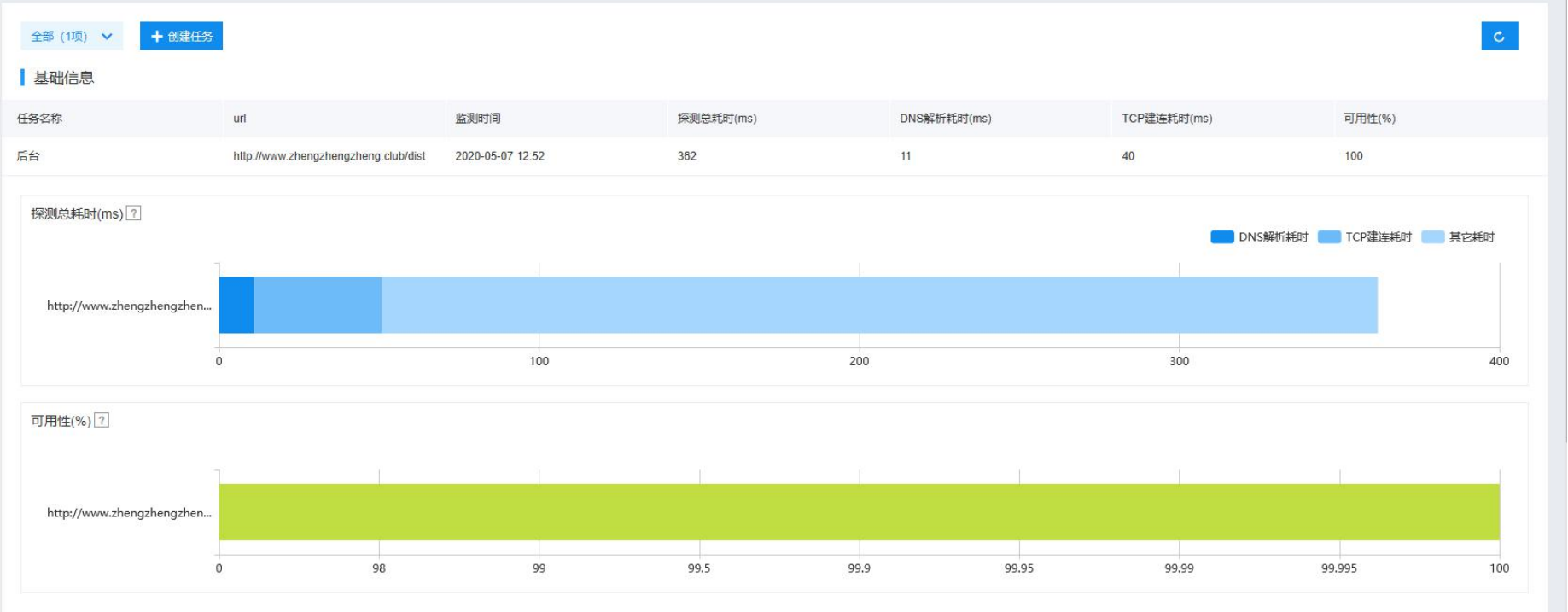
console.error node_modules/vue/dist/vue.runtime.common.dev.js:621
[Vue warn]: Unknown custom element: <router-view> - did you register the component correctly? For recursive components, make sure to provide the "name" option.

found in

---> <Anonymous>
      <Root>

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:       2.328s
Ran all test suites.
PS C:\Code\软工时间寒假作业\芥子空间\Mustard-space> █
```

二、性能测试



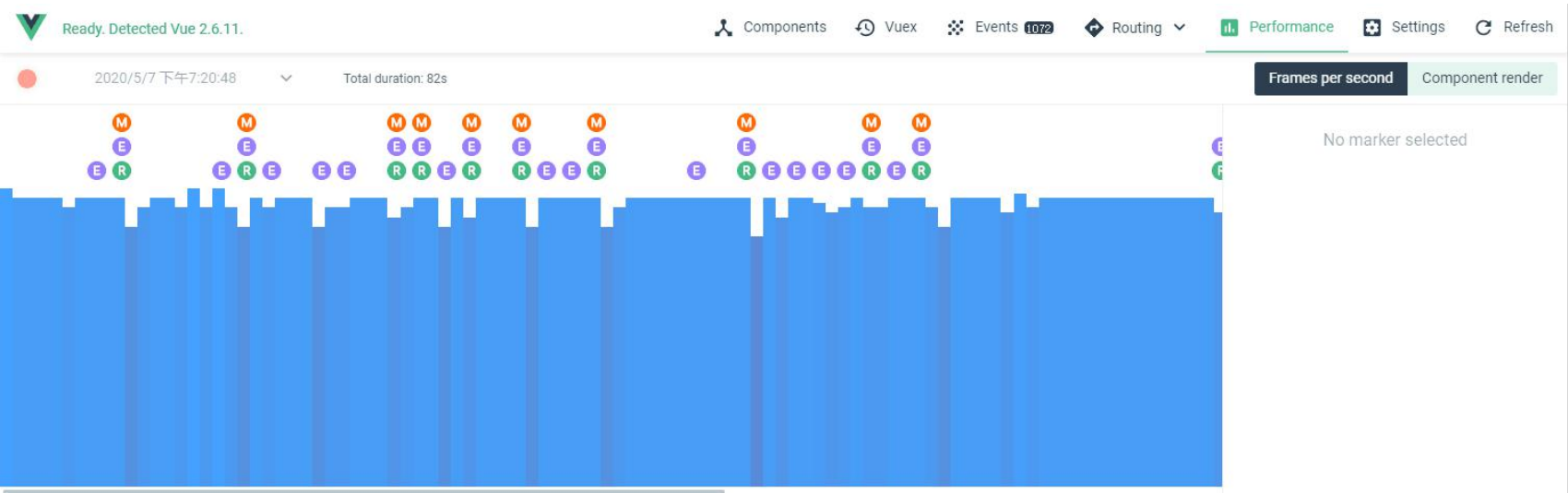
一、单元测试

```
Counter
  ✓ 包含一个span标签 (8ms)
  ✓ 有一个可点击按钮 (9ms)
  ✓ 点赞后数量+1 (3ms)

console.error node_modules\@vue\test-utils\dist\vue-test-utils.js:1735
[vue-test-utils]: contains is deprecated and will be removed in the next major version Use `wrapper.find`, `wrapper.findComponent` or `wrapper.get` instead

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.8s
Ran all test suites.
```

二、性能分析



2020/5/7 下午7:20:48	Total duration: 103s
Filter components	
<ElMenuItem>	6381 ms
<Transition>	2540 ms
<ElSubmenu>	2059 ms
<ElDropdownItem>	909 ms
<ElCol>	841 ms
<ElMenu>	726 ms
<ElButton>	576 ms
<App>	440 ms
<D2LayoutHeaderAside>	436 ms
<RouterLink>	416 ms
<D2LayoutHeaderAsideMenuHeader>	412 ms
<D2Tabs>	368 ms
<ElTabs>	366 ms

<ElInput>	174 ms	■
<ElTabPane>	158 ms	■
<Page1>	134 ms	■
<TabNav>	108 ms	■
<Page2>	90 ms	■
<D2Source>	50 ms	▮
<ElUpload>	49 ms	▮
<ElUploadList>	45 ms	▮
<TransitionGroup>	40 ms	▮
<Upload>	35 ms	▮
<ElDialog>	32 ms	▮
<D2Contextmenu>	23 ms	▮
<WorkList>	18 ms	▮