

# element无法enable问题

vue 中定义方法

```
    },
    testLoadRoiInfo() {
      // console.log("zoomValue", this.zoomValue, "wwwcValue", this.zoomValue, "renderTime", this.renderTime);
      // console.log("事件的细节-----", this.evt);

      // 往toolmanager里面添加状态
      // addToolState(element, toolName, measurementData)
      // 触发render函数, 进行图形绘制
      // onImageRendered(evt);
      console.log("testLoadRoiInfo-----");
      const element = document.querySelector('.cornerstone-element');
      window.cornerstone.enable(element);
      // addEnabledElement
      this.$imageRenderedEventDispatcher.renderData(this.evt);
      console.log("testLoadRoiInfo-----");
    },
  },
}
```

mounted注册监听事件

```
mounted() {
  this.element = this.$refs.viewer; // 获取element
  this.element.addEventListener('cornerstoneimagerendered', this.getZoomRate); // 监听事件
}
```

div

```
<div class="cornerstone-element" ref="viewer" data-index="0" oncontextmenu="return false"></div>
<!-- <a href="#" class="save-btn" data-index="1">Save</a>
```

定义button 触发事件

```
<input type="button" v-on:click="testHttp()" value="测试http请求">
<input type="button" v-on:click="testLoadRoiInfo()" value="测试加载图像数据">
```

进入页面 无操作 直接点击 加载图像数据, 调用renderData () → 就是吧  
onImageRendered复制了变成函数暴露出去

```

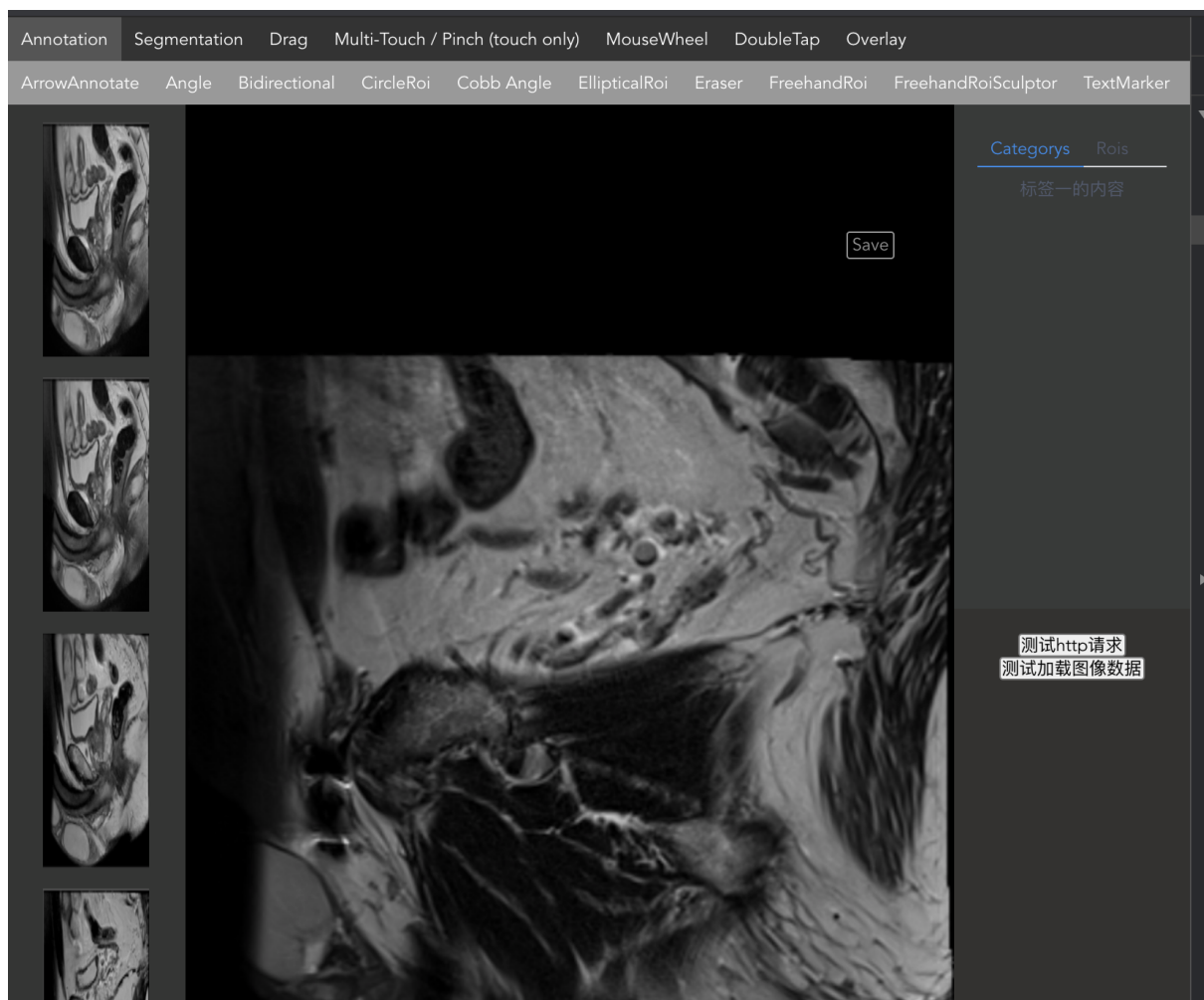
42 | });
43 | };
44 | //加载
45 | function renderData(evt) {
46 |
47 |     const eventData = evt.detail;
48 |     const element = eventData.element;
49 |     //添加允许的元件
50 |     external.cornerstone.addEnabledElement(e
51 |     // Render Annotation Tools
52 |     const toolsToRender = state.tools.filt
53 |     tool =>
54 |         tool.element === element &&
55 |         (tool.mode === 'active' ||
56 |         tool.mode === 'passive' ||

```

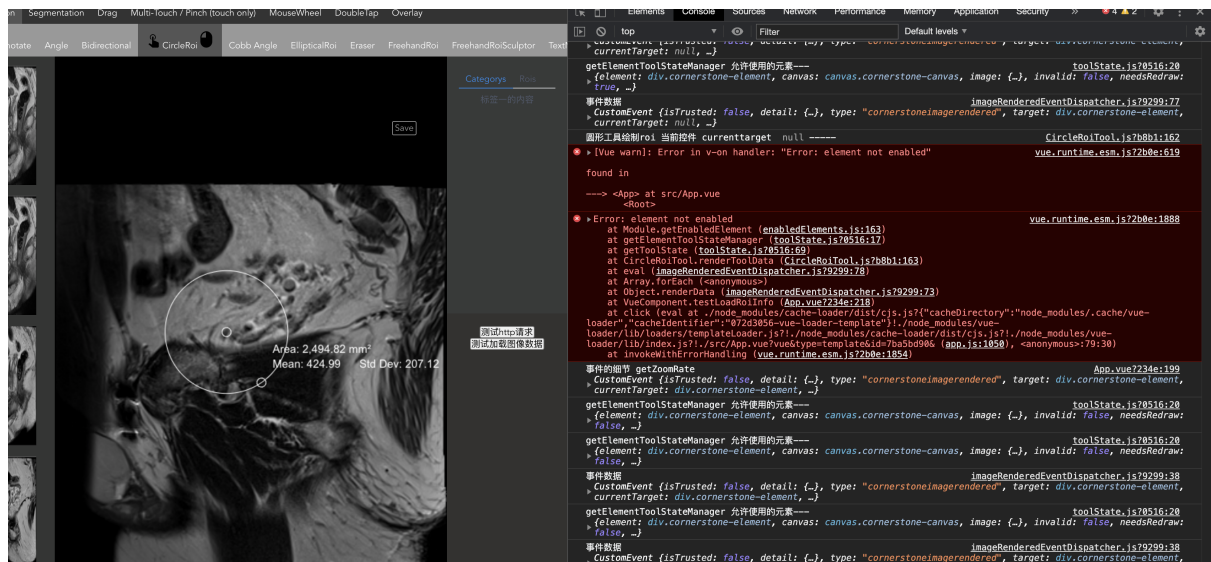
```

6 | const segmentationModule = getModule('segmentation
7 |
8 | const onImageRendered = function(evt) {
9 |     const eventData = evt.detail;
10 |     const element = eventData.element;
11 |
12 |     // Render Annotation Tools
13 |     const toolsToRender = state.tools.filter(
14 |         tool =>
15 |             tool.element === element &&
16 |             (tool.mode === 'active' ||

```



画个圆，再次点击测试加载图像数据，会报错。



## 调用流程

```
tool.renderToolData(evt);
```

```
//绘制图像的函数
renderToolData(evt) {
  console.log("圆形工具绘制roi 当前控件 currenttarget ",evt.currentTarget,"-----");
  const toolData = getToolState(evt.currentTarget, this.name);
```

```
function getToolState(element, toolName) {
  const toolStateManager = getElementToolStateManager(element);

  return toolStateManager.get(element, toolName);
}
```

```
* @returns {object} The toolstate.
function getElementToolStateManager(element) {
  const enabledElement = external.cornerstone.getEnabledElement(element);
  // If the enabledElement has no toolStateManager, create a default one for it
  // NOTE: This makes state management element specific
  console.log("getElementToolStateManager 允许使用的元素---",enabledElement.toolStateManager);
  if (enabledElement.toolStateManager === undefined) {
```

```

150  * @returns {EnabledElement} A Cornerstone Enabled Element
151  * @memberof EnabledElements
152  */
153  export function getEnabledElement (element) {
154    if (element === undefined) {
155      throw new Error('getEnabledElement: parameter element must not be undefined');
156    }
157    for (let i = 0; i < enabledElements.length; i++) {
158      if (enabledElements[i].element === element) {
159        return enabledElements[i];
160      }
161    }
162    throw new Error('element not enabled');
163  }
164 }

```

原因是这个element没有被enable，于是采用如下方式在调用renderdata前进行enable，无效还是报错

```

const element = document.querySelector('.cornerstone-element');
window.cornerstone.enable(element);

```

于是在renderdata () 里渲染图像时提前强行添加element，也无效

```

43  };
44  //加载
45  function renderData(evt) {
46
47    const eventData = evt.detail;
48    const element = eventData.element;
49    //添加允许的元件
50    external.cornerstone.addEnabledElement(element);
51    // Render Annotation Tools

```