

roi 数据增删改查接口

1、添加roi工具

MEASUREMENT_ADDED

```
eredEventDispatcher.js • JS CircleRoiTool.js JS addNewMeasurement.js JS moveAllHandles.js • JS
src > stateManagement > JS toolState.js > addToolState
28 * Adds tool state to the toolStateManager, this is done by tools as well
29 * as modules that restore saved state.
30 * @public
31 * @method addToolState
32 *
33 * @param {HTMLElement} element The element.
34 * @param {string} toolName The name of the tool the state belongs to.
35 * @param {Object} measurementData The data to store in the state.
36 * @returns {undefined}
37 */
38 function addToolState(element, toolName, measurementData) {
39   const toolStateManager = getElementToolStateManager(element);
40
41   measurementData.uuid = measurementData.uuid || uuidv4();
42   toolStateManager.add(element, toolName, measurementData);
43
44   const eventType = EVENTS.MEASUREMENT_ADDED;
45   const eventData = {
46     toolName,
47     toolType: toolName, // Deprecation notice: toolType will be replaced by toolName
48     element,
49     measurementData,
50   };
51   console.log('MEASUREMENT_ADDED---eventData', eventData);
52   triggerEvent(element, eventType, eventData);
53 }
```

2、修改roi工具

MEASUREMENT_MODIFIED

已有的数据修改过程中

```
JS moveNewHandle.js JS moveHandle.js × index.html
src > manipulators > JS moveHandle.js > _dragHandler > modifiedEventData
183 }
184
185 external.cornerstone.updateImage(element);
186
187 const activeTool = getActiveTool(element, buttons, interactionType);
188
189 if (activeTool instanceof BaseAnnotationTool) {
190   activeTool.updateCachedStats(image, element, annotation);
191 }
192
193 const eventType = EVENTS.MEASUREMENT_MODIFIED;
194 const modifiedEventData = {
195   toolName,
196   toolType: toolName, // Deprecation notice: toolType will be replaced by toolName
197   element,
198   measurementData: annotation,
199 };
200 console.log(
201   'MEASUREMENT_MODIFIED11111--eventData 已有数据修改',
202   modifiedEventData
203 );
204 triggerEvent(element, eventType, modifiedEventData);
205 }
206
207 function _cancelEventHandler(
208   toolName,
209   evtDetail,
210   annotation,
211   handle,
212   options = {},
213   interactionType,
214   { dragHandler, upOrEndHandler },
215   doneMovingCallback
216 ) {
```

新增数据移动过程中

```
JS moveNewHandle.js × JS moveHandle.js index.html
src > manipulators > JS moveNewHandle.js > _moveHandler
166 handle.x = targetLocation.x;
167 handle.y = targetLocation.y;
168
169 if (options && options.preventHandleOutsideImage) {
170   clipToBox(handle, image);
171 }
172
173 external.cornerstone.updateImage(element);
174
175 const activeTool = getActiveTool(element, buttons, interactionType);
176
177 if (activeTool instanceof BaseAnnotationTool) {
178   activeTool.updateCachedStats(image, element, annotation);
179 }
180
181 const eventType = EVENTS.MEASUREMENT_MODIFIED;
182 const modifiedEventData = {
183   toolName,
184   toolType: toolName, // Deprecation notice: toolType will be replaced by toolName
185   element,
186   measurementData: annotation,
187 };
188 console.log(
189   'MEASUREMENT_MODIFIED2222--modifiedEventData 数据新增',
190   '-----element-----',
191   element,
192   '-----eventType-----',
193   eventType,
194   '-----modifiedEventData-----',
195   modifiedEventData
196 );
197 _saveRoiInfo(modifiedEventData);
198
199 triggerEvent(element, eventType, modifiedEventData);
200 }
201 function _saveRoiInfo(eventData) {
```

3、数据测量完毕

MEASUREMENT_COMPLETED

```
JS triggerEvent.js JS addNewMeasurement.js X JS moveNewHandle.js JS touchMoveHandle
src > eventDispatchers > mouseEventHandlers > JS addNewMeasurement.js > default > handleMove
38     tool.name,
39     measurementData,
40     measurementData.handles.end,
41     tool.options,
42     'mouse',
43     success => {
44       if (measurementData.cancelled) {
45         return;
46       }
47
48       if (success) {
49         const eventType = EVENTS.MEASUREMENT_COMPLETED;
50         const eventData = {
51           toolName: tool.name,
52           toolType: tool.name, // Deprecation notice: toolType will be replaced by toolName
53           element,
54           measurementData
55         };
56         console.log('MEASUREMENT_COMPLETED---eventdata', eventData);
57         triggerEvent(element, eventType, eventData);
58       } else {
59         removeToolState(element, tool.name, measurementData);
60       }
61     }
62   );
}
```

4、数据删除

MEASUREMENT_REMOVED

```
JS toolState.js X
src > stateManagement > JS toolState.js > removeToolState
94   for (let i = 0; i < toolData.data.length; i++) {
95     if (toolData.data[i] === data) {
96       indexOfData = i;
97     }
98   }
99
100   if (indexOfData !== -1) {
101     toolData.data.splice(indexOfData, 1);
102
103     const eventType = EVENTS.MEASUREMENT_REMOVED;
104     const eventData = {
105       toolName,
106       toolType: toolName, // Deprecation notice: toolType will be replaced by toolName
107       element,
108       measurementData: data,
109     };
110     console.log('MEASUREMENT_REMOVED---eventData', eventData);
111     triggerEvent(element, eventType, eventData);
112   }
113 }
114
115 /**
```