Viewer Documentation

The code for the viewer has been documented in index.js in the reporting-bot project. A brief overview is also being provided here. For code related issues, refer to the cornerstone documentation <https://www.cornerstonejs.org/>. Please thoroughly read the cornerstone documentation as it is comprehensive. Do not make any changes to the viewer code before you have understood how cornerstone3D works and how the current viewer code works.

To Do:

* Solve memory related issues: Write some code to refresh the page without running the function **this.cornerstone()** before the page is redirected after reporting. This will ensure that no new images are loaded into cache and previously cached images are freed up. This has been discussed with Aman sir. **MUST DO BEFORE DEPLOYMENT TO PREVENT FREQUENT CRASHING**
* Add a tool that will allow syncing of scrolling for viewports
* Add a magnify tool option that will allow radiologist to magnify images in both stack and volume viewports
* Add a crosshair cursor so that in different orientations, positioning can be determined
* Clean up the function **server\_data** in views.py, just need to make it cleaner and easier to read

Future:

* Shift the viewer code from index.js and make it as a React component for cleanliness

Index.js

Important variables

**toolGroupId –** This is the id name of the toolGroup which keeps a store of all the tools being added

**toolGroup** – This is the actual toolGroup object. A global reference to it is required for any functions that deal with adding viewports to the toolGroup.

**renderingEngineId –** This is the id name of the renderingEngine which maintains all of the viewports.

**renderingEngine –** This is the actual renderingEngine object. A global reference to it is required for any functions that deal with changing the layout of the viewports.

**viewportIds** – This is the list of Ids used for the four viewports. A global reference is required as a lot of cornerstone.js functions related to viewports work by referencing the viewport Id. The viewport Ids are also used as data-values for the main viewport divs as a sort of mapping between the divs and the actual viewports

**first\_viewport, second\_viewport, third\_viewport, fourth\_viewport –** These variables will be used to define the viewport input objects in componentDidMount and a global reference is required

**viewport\_list** - This is a dictionary which will contain references to the objects first\_viewport, second\_viewport, etc after they have been defined in componentDidMount. A global reference is required for the drag and drop functionality of the series previews.

**indexMap –** This is a map for storing the ids of the divs below the viewer that display the image index in the viewport. It maps a viewportId to the HTML id of the index div.

**selected\_viewport** – This is a variable for storing the currently selected viewport (which gets a red border on click). A global reference is required as most viewport specific functions like tools functions rely on this variable.

**prev\_selected\_element** – This is a variable that stores the HTML div associated with each viewport. It requires a global reference for the screenshot function/

**curr\_tool –** This is a variable that stores the current cornerstone tool that is active. A global reference is required as in order to make a new tool active; the current tool has to be made passive

**prev\_layout** – This is a variable that stores the current layout of the viewports (‘one’, ‘two’, or ‘four’). A global reference is required as conditional logic is written in the layout function using this variable.

**Tools** – This is a dictionary containing all the tools that get added to the toolGroup.

Important Functions

**this.allowDrop –** This is a standard function that prevents the default event behaviour for drag and drop functionality. It is attached to the onDragOver event for every viewport div in viewport-container

**this.drop** – This function is called when dropping a preview image of a series onto a viewport. It is attached to the onDrop event for every viewport div in viewport-container. It checks whether the modality of the series is CT/MRI or non-CT/MRI and manipulates the viewport into a stack or a volume viewport accordingly using asynchronous functions. Please refer to code.

**this.fullScreen –** This function is called when pressing the Full Screen button. It simply manipulates the CSS grid properties of the outer div (id is ‘page-content’) which contains the CKEditor and the viewports. It changes the button’s value attribute to ‘small’ or ‘full’ to toggle the full screen. It uses the resize property of the renderingEngine to scale the images in the viewports.

**this.capture –** This function is called when pressing the Capture button. It uses the html2canvas library and the **prev\_selected\_element** variable to capture the canvas of the viewport div and downloads it as a png image.

**this.cornerstone –** This is an asynchronous function that is called in ComponentDidMount. It uses fetch api to create a POST request to the view server\_data to get the details of the study. It then uses a loop to go through each series in the study and populates the preview tab. It uses the function **createImageIdsAndCacheMetadata** to cache the metadata of the series and create the imageIds. If the series is a CT/MRI, it creates a volume using the imageIds and caches it and passes the volumeId as a data-value attribute for the preview image. If it is non-CT/MRI, it adds the imageIds to the list **nonCT\_ImageIds** and passes the corresponding index as a data-value attribute for the preview image.

**This.toggleTool** – This function is called whenever a cornerstone tool is selected. It makes the current tool passive, makes the selected tool active and updates the value of **curr\_tool**

**This.layoutSettings** – This function is used to change the layout of the viewports (1x1, 1x2, 2x2). It uses the **prev\_layout** variable to conditionally disable/enable viewports and then resizes the current viewports.

**This.slabThickness –** This function is used to change the slab thickness of a volume in a volume viewport. NOTE: if it is tried for stack viewport, cornerstone will log an error in the console.

**This.alignmentSettings –** This function is used to align the image within the viewport itself. Options are left, right, and center. Uses the viewport class’ displayArea property.

**This.volumeOrientation –** This function is used to switch the orientation of a volume in a volume viewport. Options are coronal, axial, sagittal. Uses the viewports class’ setOrientation method. NOTE: if it is tried for a stack viewport, cornerstone will log an error in the console.

**This.orientationSettings –** This function is to change 4 orientation settings of the image – rotate left, rotate right, horizontal flip, vertical flip. Rotation uses the viewport class’ rotate property whereas flipping is achieved using the viewport’s setCamera method.

**This.windowingSettings –** This function is used to change the window width and window center properties (contrast) of the image. Options are – invert, lungs, brain, bone, etc. Please crosscheck the exact voiRange values for each setting online or with a doctor.

**This.viewportSettings –** This function is for general viewport settings. Currently, only reset viewport is accomplished using this function.

**componentDidMount –** This function obviously runs everytime the component is loaded. All main cornerstone activity such as adding tools to toolGroup, defining each viewport, enabling the first viewport, calling this.cornerstone(), adding event listeners to the viewport Divs is done here.

HTML Documentation

All cornerstone related HTML is inside the div with className = ‘cornerstone-container’. Within this, there are three main divs – tools, viewport-container, details (rename this to something better)

**Tools Div Children**

Each child div has the className – ‘button-container’.

|  |  |  |
| --- | --- | --- |
| **Children** | **Purpose** | **Functions Used** |
| Child Div 1 | This div contains the button (value=’Zoom’)for enabling the zoom tool | toggleTool() |
| Child Div 2 | This div contains the button (value=’Pan’) for the pan tool and a select tag (id=’alignment’) for the alignment options – left, right, center | toggleTool()  alignmentSettings() |
| Child Div 3 | This div contains the button (value=’PlanarRotate’) for the planar rotate tool and a select tag (id=’orientation’) for the orientation options – rotate left, rotate right, horizontal flip, vertical flip. | toggleTool()  orientationSettings() |
| Child Div 4 | This div contains the button (value=’Probe’) for the probe tool | toggleTool() |
| Child Div 5 | This div contains the button (value=’Contrast’) for the contrast tool and a select tag (id=’windowing’) for all the windowing options – invert, bone, lungs, etc | toggleTool()  windowingSettings() |
| Child Div 6 | This div contains the select tag (id=’measurement’) for all the measurement tools. | toggleTool() |
| Child Div 7 | This div contains the button (value=’Eraser’) for the eraser tool. After activating it, click on any annotation to remove it | toggleTool() |
| Child Div 8 | This div contains the select tag (id=’mpr’) for volume orientations | volumeOrientation() |
| Child Div 9 | This div contains the input tag which acts as a slider to change the slab thickness | slabThickness() |
| Child Div 10 | This div contains the select tag (id=’layout’) for the layout settings | layoutSettings() |
| Child Div 11 | This div contains the button for resetting the viewport | viewportSettings() |
| Child Div 12 | This div contains the button for taking a screenshot of the viewport | Capture() |
| Child Div 13 | This div contains the button for toggling full screen option for the viewer | fullScreen() |

**Viewport-container div children**

|  |  |  |  |
| --- | --- | --- | --- |
| **Children** | **Purpose** | **HTML ID** | **Functions** |
| Child Div 1 | Div for the first viewport | viewport1 | allowDrop()  drop() |
| Child Div 2 | Div for the second viewport | viewport2 | allowDrop()  drop() |
| Child Div 3 | Div for the third viewport | viewport3 | allowDrop()  drop() |
| Child Div 4 | Div for the fourth viewport | viewport4 | allowDrop()  drop() |

**CSS Documentation**

Important Styles

**.page-content{} –** display: grid

.**viewport-container{} –** display: grid

**.viewport{} –** Positive: relative

**.cornerstone-canvas{}** – height: 100%, width: 100%

**.svg-layer{}** – positive: relative !important