The React-based CONTENTdm site no longer supports custom JavaScript files at the collection level. This is due to the need for all custom, client-side scripts to tie in with the event lifecycle of the application and fire (or not fire) based on both page context and the readiness of the DOM.

If you want to create custom JavaScript functions that take effect only with specific collections, you can do so by adding an `if` clause to your custom JavaScript code. Any custom JavaScript that manipulates or reads the DOM will already need to have utilized `document.addEventListener` to make sure it executes at the appropriate time, e.g.:

```javascript
(function() {
    document.addEventListener('cdm-item-page:ready', function(e){
        var message = 'hello, world!';
        console.log (message);
    });
})();
```

The above script uses IIFE and the `ready` event to log a message to the console whenever a collection landing page’s DOM has fully rendered. If you want this same function to run only for a specific collection, you would add an `if` clause like this:

```javascript
(function() {
    document.addEventListener('cdm-item-page:ready', function(e){
        let globalScope = false;
        let collectionScope = ['oclcsample', 'otheralias1', 'otheralias2'];
        if (globalScope || collectionScope.includes(e.detail.collectionId)) {
            var message = 'hello, world!';
            console.log (message);
        }
    });
})();
```

The `collectionId` value can be inspected for any event class that has specific collection context. This is all of the page classes for collection and item pages. See [List of JavaScript lifecycle events](https://help.oclc.org/Metadata_Services/CONTENTdm/Advanced_website_customization/Customization_cookbook) for more information about the event page classes.

Also note that as with any JavaScript customization in CONTENTdm, you may need to add the `if` clause to both your `ready` page event as well as your `update` page event, depending what your code is trying to do. Since React applications will update parts of the DOM dynamically, you might need to execute your JavaScript code at the `update` page event to make sure you are acting on the current version of the DOM. You can see an example script that
executes only in a specific collection at the \texttt{ready, update} and \texttt{leave} page events by clicking the link below:

\textbf{View}: View an example script.