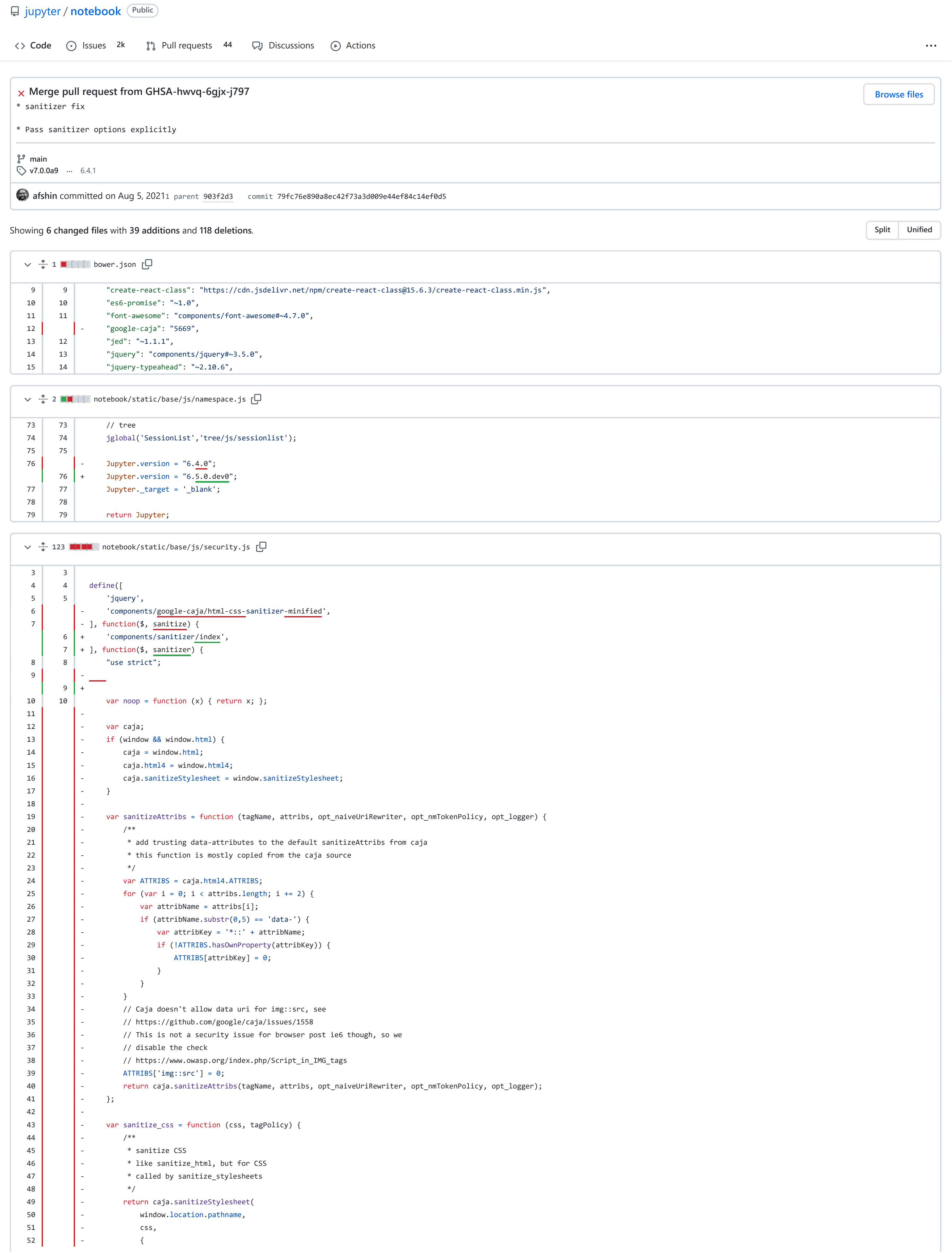


Merge pull request from GHSA-hwvq-6gjx-j797

```
* sanitizer fix

* Pass sanitizer options explicitly
```

main
v7.0.0a9 ... 6.4.1

afshin committed on Aug 5, 20211 parent 903f2d3 commit 79fc76e890a8ec42f73a3d009e44ef84c14ef0d5

Showing **6 changed files** with **39 additions** and **118 deletions**.

1 bower.json

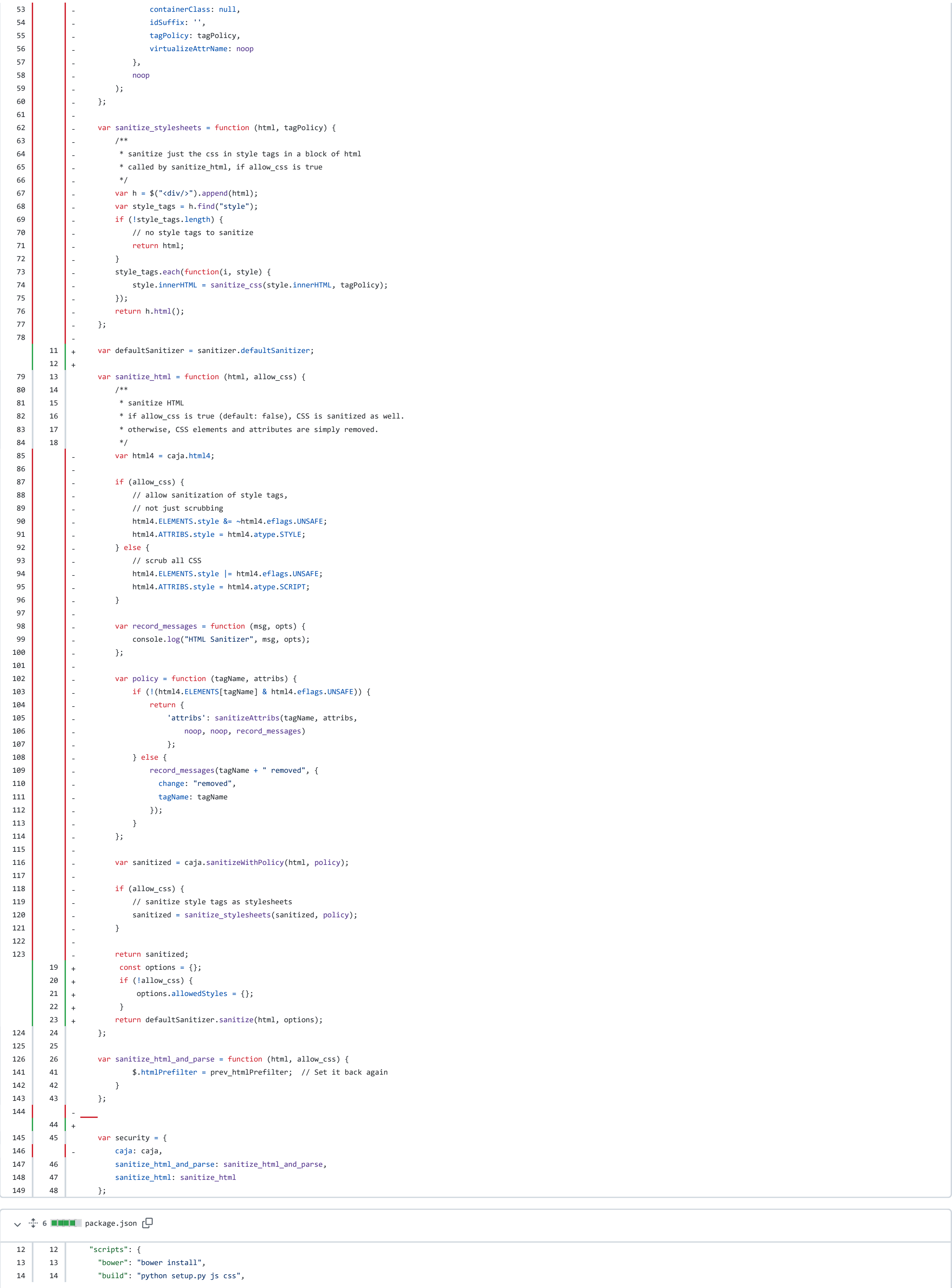
```
      "create-react-class": "https://cdn.jsdelivr.net/npm/create-react-class@15.6.3/create-react-class.min.js",
      "es6-promise": "~1.0",
      "font-awesome": "components/font-awesome#~4.7.0",
-     "google-caja": "5669",
      "jed": "~1.1.1",
      "jquery": "components/jquery#~3.5.0",
      "jquery-typeahead": "~2.10.6",
```

2 notebook/static/base/js/namespace.js

```
      // tree
      jglobal('SessionList','tree/js/sessionlist');

-     Jupyter.version = "6.4.0";
+     Jupyter.version = "6.5.0.dev0";
      Jupyter._target = '_blank';

      return Jupyter;
```

123 notebook/static/base/js/security.js

```

  define([
      'jquery',
-     'components/google-caja/html-css-sanitizer-minified',
- ], function($, sanitize) {
+     'components/sanitizer/index',
+ ], function($, sanitizer) {
      "use strict";
- 
+
      var noop = function (x) { return x; };
-
-     var caja;
-     if (window && window.html) {
-         caja = window.html;
-         caja.html4 = window.html4;
-         caja.sanitizeStylesheet = window.sanitizeStylesheet;
-     }
-
-     var sanitizeAttribs = function (tagName, attribs, opt_naiveUriRewriter, opt_nmTokenPolicy, opt_logger) {
-         /**
-          * add trusting data-attributes to the default sanitizeAttribs from caja
-          * this function is mostly copied from the caja source
-          */
-         var ATTRIBS = caja.html4.ATTRIBS;
-         for (var i = 0; i < attribs.length; i += 2) {
-             var attribName = attribs[i];
-             if (attribName.substr(0,5) == 'data-') {
-                 var attribKey = '*::' + attribName;
-                 if (!ATTRIBS.hasOwnProperty(attribKey)) {
-                     ATTRIBS[attribKey] = 0;
-                 }
-             }
-         }
-         // Caja doesn't allow data uri for img::src, see
-         // https://github.com/google/caja/issues/1558
-         // This is not a security issue for browser post ie6 though, so we
-         // disable the check
-         // https://www.owasp.org/index.php/Script_in_IMG_tags
-         ATTRIBS['img::src'] = 0;
-         return caja.sanitizeAttribs(tagName, attribs, opt_naiveUriRewriter, opt_nmTokenPolicy, opt_logger);
-     };
-
-     var sanitize_css = function (css, tagPolicy) {
-         /**
-          * sanitize CSS
-          * like sanitize_html, but for CSS
-          * called by sanitize_stylesheets
-          */
-         return caja.sanitizeStylesheet(
-             window.location.pathname,
-             css,
-             {
```



```diff
-                containerClass: null,
-                idSuffix: '',
-                tagPolicy: tagPolicy,
-                virtualizeAttrName: noop
-            },
-            noop
-        );
-    };
-
-    var sanitize_stylesheets = function (html, tagPolicy) {
-        /**
-         * sanitize just the css in style tags in a block of html
-         * called by sanitize_html, if allow_css is true
-         */
-        var h = $("<div/>").append(html);
-        var style_tags = h.find("style");
-        if (!style_tags.length) {
-            // no style tags to sanitize
-            return html;
-        }
-        style_tags.each(function(i, style) {
-            style.innerHTML = sanitize_css(style.innerHTML, tagPolicy);
-        });
-        return h.html();
-    };
-
+    var defaultSanitizer = sanitizer.defaultSanitizer;
+
     var sanitize_html = function (html, allow_css) {
         /**
          * sanitize HTML
          * if allow_css is true (default: false), CSS is sanitized as well.
          * otherwise, CSS elements and attributes are simply removed.
          */
-        var html4 = caja.html4;
-
-        if (allow_css) {
-            // allow sanitization of style tags,
-            // not just scrubbing
-            html4.ELEMENTS.style &= ~html4.eflags.UNSAFE;
-            html4.ATTRIBS.style = html4.atype.STYLE;
-        } else {
-            // scrub all CSS
-            html4.ELEMENTS.style |= html4.eflags.UNSAFE;
-            html4.ATTRIBS.style = html4.atype.SCRIPT;
-        }
-
-        var record_messages = function (msg, opts) {
-            console.log("HTML Sanitizer", msg, opts);
-        };
-
-        var policy = function (tagName, attribs) {
-            if (!(html4.ELEMENTS[tagName] & html4.eflags.UNSAFE)) {
-                return {
-                    'attribs': sanitizeAttribs(tagName, attribs,
-                        noop, noop, record_messages)
-                    };
-            } else {
-                record_messages(tagName + " removed", {
-                  change: "removed",
-                  tagName: tagName
-                });
-            }
-        };
-
-        var sanitized = caja.sanitizeWithPolicy(html, policy);
-
-        if (allow_css) {
-            // sanitize style tags as stylesheets
-            sanitized = sanitize_stylesheets(sanitized, policy);
-        }
-
-        return sanitized;
+         const options = {};
+         if (!allow_css) {
+             options.allowedStyles = {};
+         }
+        return defaultSanitizer.sanitize(html, options);
     };

     var sanitize_html_and_parse = function (html, allow_css) {
            $.htmlPrefilter = prev_htmlPrefilter;  // Set it back again
        }
    };
-
+
    var security = {
-        caja: caja,
        sanitize_html_and_parse: sanitize_html_and_parse,
        sanitize_html: sanitize_html
    };
```

6 package.json

```diff
   "scripts": {
     "bower": "bower install",
     "build": "python setup.py js css",
+    "build:webpack": "webpack --mode development",
     "build:watch": "npm run watch",
     "watch": "onchange 'notebook/static/**/!(*.min).js' 'notebook/static/**/*.less' 'bower.json' -- npm run build"
   },
   "devDependencies": {
+    "@jupyterlab/apputils": "^3.1.3",
     "bower": "^1.8.8",
     "less": "~2",
     "onchange": "^6.0.0",
     "po2json": "^0.4.5",
-    "requirejs": "^2.3.6"
+    "requirejs": "^2.3.6",
+    "webpack": "^5.46.0",
+    "webpack-cli": "^4.7.2"
   }
 }
```

15 🟩🟩🟩🟩🟥 setupbase.py

```diff
         pjoin(components, "font-awesome", "css", "*.css"),
         pjoin(components, "es6-promise", "*.js"),
         pjoin(components, "font-awesome", "fonts", "*.*"),
-        pjoin(components, "google-caja", "html-css-sanitizer-minified.js"),
         pjoin(components, "jed", "jed.js"),
         pjoin(components, "jquery", "jquery.min.js"),
         pjoin(components, "jquery-typeahead", "dist", "jquery.typeahead.min.js"),
         pjoin(components, "requirejs", "require.js"),
         pjoin(components, "requirejs-plugins", "src", "json.js"),
         pjoin(components, "requirejs-text", "text.js"),
+        pjoin(components, "sanitizer", "index.js"),
         pjoin(components, "underscore", "underscore-min.js"),
         pjoin(components, "moment", "moment.js"),
         pjoin(components, "moment", "min", "*.js"),

     bower_dir = pjoin(static, 'components')
     node_modules = pjoin(repo_root, 'node_modules')
+    sanitizer_dir = pjoin(bower_dir, 'sanitizer')

     def should_run(self):
         if self.force:
             return True
         if not os.path.exists(self.bower_dir):
             return True
-
-        return mtime(self.bower_dir) < mtime(pjoin(repo_root, 'bower.json'))
+        if not os.path.exists(self.sanitizer_dir):
+            return True
+
+        bower_stale = mtime(self.bower_dir) < mtime(pjoin(repo_root, 'bower.json'))
+        if bower_stale:
+            return True
+
+        return mtime(self.sanitizer_dir) < mtime(pjoin(repo_root, 'webpack.config.js'))

     def should_run_npm(self):
         if not which('npm'):
             print("You can install js dependencies with `npm install`", file=sys.stderr)
             raise
         # self.npm_components()
+        if not os.path.exists(self.sanitizer_dir):
+            run(['npm', 'run', 'build:webpack'], cwd=repo_root, env=env)
         os.utime(self.bower_dir, None)
         # update package data in case this created new files
         update_package_data(self.distribution)
```

10 🟩🟩🟩🟩🟩 webpack.config.js

```diff
@@ -0,0 +1,10 @@
+const path = require('path');
+
+module.exports = {
+  entry: '@jupyterlab/apputils/lib/sanitizer',
+  output: {
+    filename: 'index.js',
+    path: path.resolve(__dirname, 'notebook/static/components/sanitizer'),
+    libraryTarget: "amd"
+  }
+}
```

1 comment on commit 79fc76e

meeseeksmachine commented on 79fc76e on Aug 10, 2021

This commit has been mentioned on **Jupyter Community Forum**. There might be relevant details there:

https://discourse.jupyter.org/t/can-not-use-jupyterlab-or-jupyter-notebooks-with-unsafe-eval-turned-off-in-content-security-policy/10321/1

Please sign in to comment.