

Run it: `java -jar ripout-0.0.1-SNAPSHOT.jar`

1) Landing page of the simulation app.

## **Ripout UI demo with React and Material-UI.**

All UI components, page logic, web server logic, and Tomcat web server hosting this demo is packaged as a Spring Boot application in a single Java JAR file.

A real Ripout UI app would be built as a bundle of required React and Material-UI Javascript components contained in a single static JS file hosted on a web server.

The bundled JS logic would communicate over the network with the Ripout workflow API running on Weblogic.

### Supporting Technologies

- Node.js
- Node Package Manager
- Webpack for compiling the static dependencies into one bundle.

**RIPOUT**

- 1) The app demonstrates [React + MaterialUI + Java workflow simulation] running on a Tomcat web server.

The simulated workflow contains multiple Ripout records at different routing steps.

Demonstration landing page.



- 2) Real app would display only records associated with the user's AD role.  
For demonstration, the simulation presents a list of open routings.



- 3) The simulation is pre-loaded with several dummy Ripouts at various stages in the workflow.



4) Selecting Originator reveals two open routings which Originator has not yet completed (signed off).

The screenshot shows the 'Ripout' application interface. At the top, there is a blue header bar with a home icon and the text 'Ripout'. On the right side of the header, it says 'Logged User: jslick9'. Below the header, there is a section titled 'Routing: Originator'. This section contains a table with the following columns: 'XREF', 'Originator', 'Created', and 'Nuclear'. The table has two rows of data:

	XREF	Originator	Created	Nuclear
OPEN	1	J. Siczpak	03-01-2020	Yes
OPEN	2	H. Walker	03-01-2020	No

At the bottom right of the table, there is a pagination control that says 'Rows per page: 5' and '1-2 of 2'.

5) Select first row of Originator's open routings. The Ripout input form appears. The sim implements only a small portion of the entire Ripout form.

The screenshot shows the 'Ripout' application interface with the input form for the first routing. The header bar is the same as in the previous screenshot. Below the header, there is a form with the following fields:

XREF	SEQ	Nuclear	Ripout Num	Originator
1	1001	Yes		J. Siczpak

Below this table, there are two more rows of fields:

Badge	Originator Date	Status	Work Order	WPPN
387128	03-01-2020	ORIG		w1234

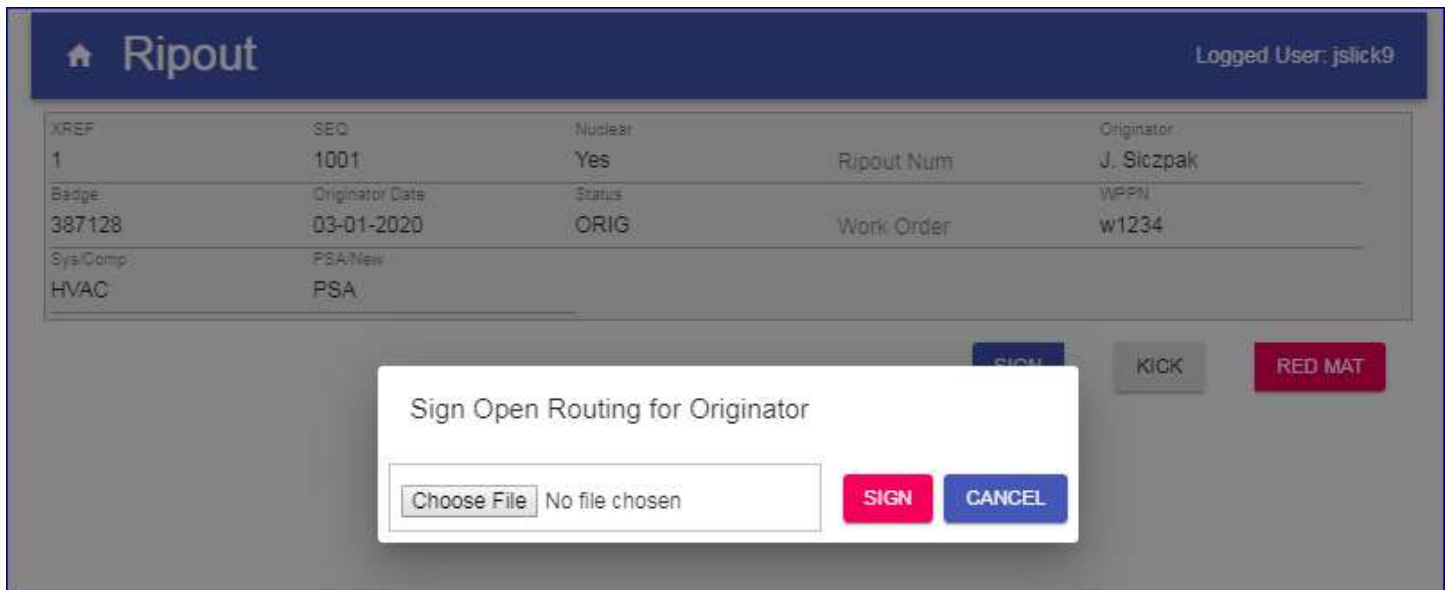
Below these fields, there are two more rows of fields:

Sys/Comp	PSA/New
HVAC	PSA

At the bottom right of the form, there are three buttons: 'SIGN' (blue), 'KICK' (grey), and 'RED MAT' (red).

6) User fills in required fields, then clicks button "Sign".

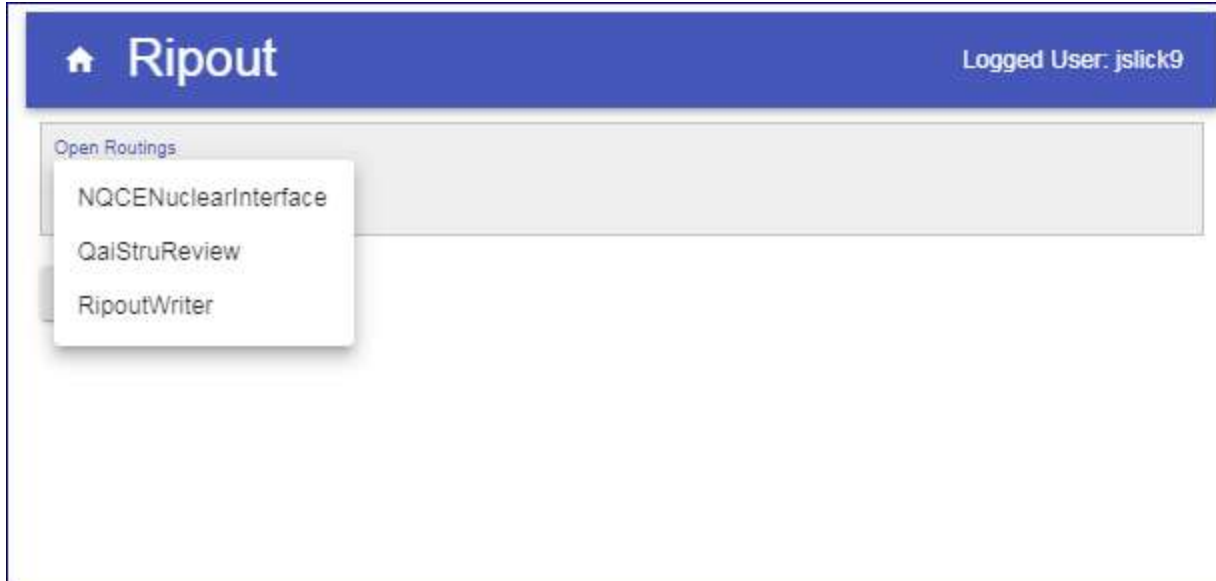
On the signing dialog, button "Choose File" presents a file navigation dialog for attaching files to the Ripout. Attachment function is not implemented in the demonstration app.



7) When user signs, the Ripout workflow is progressed to the next step.

When originator signs, the ripout record is progressed to RipoutWriter.

After both Originator open records have been signed, the Open Routings list has no Originator records. Both records have been progressed to RipoutWriter.



8) Ripout Writer now has xref's 1 & 2 to work on.

Routing: *RipoutWriter*

	XREF	Originator	Created	Nuclear
OPEN	1	J. Siczpak	03-01-2020	Yes
OPEN	2	H. Walker	03-01-2020	No
OPEN	3	J. Siczpak	03-01-2020	Yes
OPEN	4	J. Siczpak	03-01-2020	Yes
OPEN	5	H. Walker	03-01-2020	No

Rows per page: 5 ▾ 1-5 of 5 < >