

Product to Queue

The Product to Queue app is a browser-based tool which helps determine the appropriate team/queue to route a case based on product names or buzzwords.

Using the App

To run the web app locally for development, you must have [Node](#) installed. With Node installed, use the command line to navigate to the project directory and type:

```
npm install
npm start
```

To build this app for deployment, type:

```
npm run build
```

Changing the Data

This application reads directly from a [spreadsheet](#). The spreadsheet must be a csv file named 'product_queue'.

The csv file must be in the following format:

```
Product, Tech_1, Tech_1 Buzzwords, ..., Tech_n, Tech_n Buzzwords, Support Method,
Reference
```

Data that can be edited without changes to code

- The text value of any cell
 - Changing the name of a Technology/Unit
 - Changing which queue a product belongs in
 - changing the reference, etc
- Adding rows (Products)
- Removing rows
- Adding Technology/Buzzword pairs (maintaining the format above)
- Removing T/B pairs

Data that requires changes to code after editing

- Adding new column (other than T/B pairs)
- Removing columns (other than T/B pairs)
- Changing the order of columns (other than T/B pairs)

Developer Doc

App.js

- Loads and decodes the CSV
 - `getCsvData(path)`
- Parses the CSV into the objects/arrays needed for the app
 - `parseData(results)`
- Creates a search index from the parsed JSON
 - `createIndex(documents)`
- The index is passed to the `Search` child component along with the other objects
- `Navbar` is also rendered

Search.js

- The child UI components are rendered
 - `Product`
 - `Technology`
 - `Result`
- The input change handler functions for the child components mutate state
 - `handleProductChange(newProd)` - searches
 - `handleBuzzwordsChange(newBuzz)` - searches
 - `handleTechnologyChange(newTech)` - searches
 - `handleFilterClick(item)` - finds item properties
- Searching creates a Lunr query string using the `createSearchString` static method. It then performs the search and displays the matches in a `VirtualScroll` in the `Product` component
 - `doSearch()`
- If there is only one match, or a match is clicked on: the static method `findResult` is used to retrieve the product info for display in `Result`

Reporting Issues

Issues with the application can be reported to Ben Elan or Lingtao Xie. If the spreadsheet needs to be edited in a way that requires changes to the code, feel free to reach out to us as well. We would be happy to help get the application working with the necessary changes.

Built With

- [React](#) - Frontend Framework
- [Reactstrap](#) - Bootstrap for React
- [React-Select](#) - React Dropdown
- [Lunr](#) - Search Engine
- [PapaParse](#) - CSV Parser