

# Code Ordering Style

Import Ordering:

1. React
  2. Packages
  3. EFC packages
  4. Internal references
  5. CSS
- 

1. Expanding of Props
  - `const { recentNodes, maxRowsToDisplay } = props;`
2. React Hooks (except `useEffect`)
  - `useState`, `useContext`, etc
3. Custom Hooks
  - `useSomeApi`, `useForm`, etc
4. Other Variables and Methods
  - can be intermixed
5. Effects (`useEffect`)
6. return statement

The following is an example of how properties are ordered in a component function

```
export const RecycleBin = (props: RecycleBinProps) => {
```

```
  const { active } = props;
```

```
  const [showConfirmation, setShowConfirmation] = useState(false);
```

```
  const [recycleBinItems, setRecycleBinItems] = useState<RecycleBinItem[]>([]);
```

```
  const [selectedItems, setSelectedItems] = useState<RecycleBinItem[]>([]);
```

```
  const { getItemIcon, restoreRecycleBinIcon, recycleBinIcon } = useIcons();
```

```
  const { getRecycleBinFiltered, purgeFromRecycleBin, restoreFromRecycleBin } = useRecycleBinApi();
```

```
  const { t } = useTranslation('RecycleBin');
```

```
  const openPurgeConfirmation = () => setShowConfirmation(true);
```

```
  const closePurgeConfirmation = () => setShowConfirmation(false);
```

```
  const getRecycleBinItems = () => {
```

```
    getRecycleBinFiltered(initFilterProps).then((items: RecycleBinItem[]) => {
```

```
      setRecycleBinItems(items);
```

```
});
```

```
};
```

```
const onSelectItems = (items: RecycleBinItem[]) => {
```

```
  setSelectedItems(items);
```

```
};
```

```
const getIcon = (item: RecycleBinItem): EfcIconProps => {
```

```
  const fileExt = item.node.name.slice(item.node.name.lastIndexOf('.'));
```

```
  return getItemIcon(item.node.systemType, fileExt);
```

```
};
```

```
const purgeSelectedItems = () => {
```

```
  openPurgeConfirmation();
```

```
};
```

```
const purgeItem = (item: RecycleBinItem) => {
```

```
  if (!!item) {
```

```
    setSelectedItems([item]);
```

```
    openPurgeConfirmation();
```

```
  }
```

```
};
```

```
const confirmPurge = (confirmed: boolean) => {
```

```
  closePurgeConfirmation();
```

```
  if (!!confirmed) {
```

```
    purgeFromRecycleBin(selectedItems.map((item) => item.node)).then(refreshList);
```

```
  }
```

```
};
```

```
const restoreSelectedItems = () => {
```

```
  restoreFromRecycleBin(selectedItems.map((item) => item.node)).then(refreshList);
```

```
};
```

```
const restoreItem = (item: RecycleBinItem) => {
```

```
  restoreFromRecycleBin([item.node]).then(refreshList);
```

```
};
```

```
const refreshList = () => {
```

```
  getRecycleBinItems();
```

```
  setSelectedItems([]);
```

```
};
```

```
const columns: (string | EfcTableColumnInfo)[] = [
```

```
  { name: 'node.name', searchBy: true, displayName: t(RecycleBinKeys.Name), useIcon: getIcon },
```

```
  { name: 'dateDeleted', searchBy: true, displayName: t(RecycleBinKeys.DateDeleted), isDate: true },
```

```
  { name: 'path', searchBy: true, displayName: t(RecycleBinKeys.Path) },
```

```
];
```

```
const bulkActions: EfcCallToActionInfo[] = [
```

```
  { text: t(RecycleBinKeys.Purge), icon: recycleBinIcon, onClick: purgeSelectedItems },
```

```
  { text: t(RecycleBinKeys.Restore), color: 'primary', icon: restoreRecycleBinIcon, onClick: restoreSelectedItems },
```

```
];
```

```
const contextActions: EfcCallToActionInfo[] = [
```

```
  { text: t(RecycleBinKeys.Restore), icon: restoreRecycleBinIcon, onClick: restoreItem },
```

```
  { text: t(RecycleBinKeys.PurgeForever, 'Purge Forever'), icon: recycleBinIcon, onClick: purgeItem },
```

```
];
```

```
const confirmationCTAs: EfcCallToActionInfo[] = [
```

```
  { text: t(RecycleBinKeys.Cancel), color: 'primary', emphasis: 'med', onClick: () => confirmPurge(false) },
```

```
  { text: t(RecycleBinKeys.Purge), color: 'primary', emphasis: 'high', onClick: () => confirmPurge(true) },
```

```
];
```

```
useEffect(() => {
```

```
  if (!!active) {
```

```
    getRecycleBinItems();
```

```
  }
```

```
}, [active]);
```

```
return (
```

```
  <div className='my-2'>
```

```
    <Suspense>
```

```
      <FullTable data={recycleBinItems} columns={columns} bulkActions={bulkActions}
```

```
contextActions={contextActions} onSelect={onSelectItems} />
```

```
      {recycleBinItems.length == 0 &&
```

```
        <Text>
```

```
          {t(RecycleBinKeys.EmptyMsg)}
```

```
        </Text>
```

```
      }
```

```
    <Modal toggle={closePurgeConfirmation} title={t(RecycleBinKeys.AreYouSure)}
```

```
isOpen={showConfirmation} ctas={confirmationCTAs}>
```

```
      <Modal.Body>
```

```
        <Trans t={t} count={selectedItems.length} i18nKey={RecycleBinKeys.PurgeConfirmMsg}>
```

```
          {{ count: selectedItems.length }} files will be permanently removed from your account and will
```

```
not be able to be recovered.
```

```
        </Trans>
```

```
      </Modal.Body>
```

```
    </Modal>
```

```
  </Suspense>
```

```
</div>
```

```
);
```

```
};
```

Revision #5

Created 1 November 2022 21:23:20 by Quinn Godfrey

Updated 22 December 2022 23:37:28 by Rachel Hicken