**BRE Editor – User Guide**

**Version:** April 29 2025  
**Author:** Revat Mehra & Kathryn George

**1  Introduction**

The **Business‑Rule Engine (BRE) Editor** is a web‑based tool that renders your BRE policy JSON in an easy‑to‑navigate, spreadsheet‑like UI. Product, risk‑ops, and engineering teams can view and edit business rules **without touching raw JSON**.  
Typical use‑cases include:

* Making quarterly or monthly changes to rules upon client requests
* Editing the policy due to market fluctuation or repo rate changes upon client request

**2  System Requirements**

| **Requirement** | **Details** |
| --- | --- |
| Node.js | ≥ v18 LTS |
| npm | ≥ v9 |
| Browser | Chrome 135+, Firefox 122+, Edge 123+ |
| OS | Windows 10+, macOS 12+, or any modern Linux distro |

**3  Installation & First Run**

1. **Clone the repository**

git clone https://github.com/rm1605/brew-ui/tree/version-1   
cd <file name>

1. **Install dependencies**

npm install

1. **Start the dev server**

npm start

The app launches automatically on [http://localhost:3000](http://localhost:3000/). Hot‑reload is enabled.

The editor is also deployed on Vercel: <https://brew-ui.vercel.app/>

**4  The UI at a Glance**

| **Area** | **What it does** |
| --- | --- |
| **Top bar** | Upload/download buttons, policy metadata, global search |
| **Rule Table** | Tree‑grid of every field in ruleUnitDtoList with expand/collapse handles |
| **Edit Dialog** | Appears when you click any editable cell; supports validation |
| **Allowed ⇆ Blocked switch** | Inline dropdown that flips the list you are editing |
| **Download Modal** | Allows you to record audit metadata before exporting |

**5  Loading a Policy**

1. Click **Upload Policy JSON**.
2. Select your policy.json file
3. The table populates

**6  Navigating & Editing Rules**

* **Rules and Deviations Tabs** – Access rules and deviations separately
* **Expand/Collapse** – Click ▸ triangles to drill into nested objects/arrays.
* **Editable fields** – Only a few designated fields can be edited. Double‑click or press ↵ to edit. Editable fields are highlighted in pale yellow and can be filtered through checking the box on top of the edit popup.
* **Allowed / Block lists**
  1. Click the field label.
  2. Use the inline dropdown to pick **Allowed** or **Blocked**.
  3. Enter comma‑separated values in the multi‑line input.
  4. Change deviation and nondeviation rule accordingly via dropdowns
* **Bulk copy/paste** – Standard Ctrl‑C / Ctrl‑V shortcuts work inside inputs.
* **Validation** – Red outline appears if the value violates basic type checks.

**7 Deleting Rules**

* **Delete** – Trash‑can icon at the row level → confirm in the prompt.

**8  Attaching Client Requests**

Every policy iteration must originate from a client request document, which is to be uploaded as proof.

1. Click **Attach Client Request**.
2. Select the PDF or Excel file.
3. The filename is displayed next to the button and is embedded in your export ZIP.

**9  Exporting & Audit Trail**

1. Hit **Download Export**.
2. In the modal, fill:
   * **Name of Author**
   * **Author** – your email or initials
   * **Change Note** – short description
3. Click **Generate**.

You receive policy\_export.zip containing:

| **File** | **Description** |
| --- | --- |
| old-rules.json | Pre‑edit snapshot |
| new-rules.json | Current table contents |
| diff-summary.xlsx | One row per changed field |
| audit-metadata.json | Your modal inputs + timestamp |
| (optional) client-request.pdf/xlsx | Original request |

**10  Rolling Back**

1. Unzip any previous export.
2. Click **Upload Policy JSON** and load old-rules.json to restore that version.
3. Make further tweaks and export again—each ZIP is immutable.

**11  Keyboard Shortcuts**

| **Action** | **Keys** |
| --- | --- |
| Save cell | **Enter** |
| Cancel edit | **Esc** |
| Expand / Collapse row | **→ / ←** |
| Global search | **Ctrl + F** |
| Download export | **Ctrl + Shift + S** |

**12  Best Practices**

* Always export a ZIP after *every* significant change.
* Keep the Rule ID stable—analytics may depend on it.
* Treat diff-summary.xlsx as the authoritative change‑log for audits.
* Validate against staging before pushing JSON to production services.

**13  Appendix – Policy JSON Overview**

{

"policyDto": {

"brePolicyName": "Personal Loan Policy",

"policyDescription": "…"

},

"ruleUnitDtoList": [

{

"ruleId": "AGE\_01",

"category": "Borrower Profile",

"ruleConfig": {

"operator": "BETWEEN",

"operatorValue": [21, 60],

"allowedList": [],

"blockList": []

},

"orderOfOccurence": [ "AGE\_01" ],

"decision": "REJECT"

}

// …

]

}

*All top‑level fields are mandatory. Unknown keys are preserved but ignored by the UI.*

**14  Troubleshooting**

| **Symptom** | **Likely Cause** | **Resolution** |
| --- | --- | --- |
| Upload button unresponsive | Browser blocked file‑picker | Check pop‑up settings |
| Red border after editing | Invalid type/format | Revert or correct value |
| ZIP missing diff‑summary | No changes detected | Modify at least one field |
| Download stalls >1 min | Large client request attached | Wait—ZIP builds in browser |