A Dynamic Card using a REST endpoint

Phillip G. Bradford[*]

March 30, 2025

Abstract

A card component is filled with data from a REST endpoint. This card component example uses a burn block from the Stacks chain.

1 Loading data into a dynamic card

The levels of abstraction from

- <CardHolderEndpointBurnBlock /> to
- leveraging CardHolderEndpointBurnBlock
- carrying data in a BurnBlockContainer

are designed to maximize flexibility. This allows us to easily create new cards when we want them.

Listing 1 shows a basic React hook useEffect. The hook useEffect creates side-effects during the loading of the function CardHolderEndpointBurnBlock.

```
import { useEffect, useState } from "react";

import RoutesAndEndpoints from "./endPoints/RoutesAndEndpoints";
import BurnBlockContainer from "./coreContainers/BurnBlockContainer";
import DynamicCardCollapsible from "./DynamicCardCollapsible";

export default function CardHolderEndpointBurnBlock() {

    const [container, setContainer] = useState(null);
```

[*] phillip.bradford@uconn.edu, phillip.g.bradford@gmail.com, University of Connecticut, Department of Computing, Stamford, CT USA

```

    useEffect(() => {
        const API_PROXY_URL = RoutesAndEndpoints.ROUTE_LAST_BURN_BLOCK;
        async function fetchData() {
            try {
                const response = await fetch(API_PROXY_URL, {
                    headers: { 'Content-Type':
                      'text/javascript; charset=utf-8'  }
                  });
                const jsonData = await response.json();

                const burnBlockContainer =
                 new BurnBlockContainer(jsonData['results'][0]);
                setContainer(burnBlockContainer.getData());
            } catch (error) {
                console.error("Error fetching data:", error);
            }
        }

        fetchData();
    }, []);

    return (
        <div>
            {container ? (
               <DynamicCardCollapsible props={container} />
            ) : (
                <p>Loading...</p>
            )}
        </div>
    );
}
```

Listing 1: Dynamic card

The function CardHolderEndpointBurnBlock in Listing 1

1. gets a BurnBlock from the endpoint defined in RoutesAndEndpoints

2. Puts the BurnBlock from the endpoint in a class instance of BurnBlockContainer and then sets this to the container in setContainer(burnBlockContainer.getData()).

3. Assuming the container is not **null**, its data is put in the props of <DynamicCardCollapsible props={container} />

References

[1] Ryan J. Blake: *Mastering Bootstrap 5*, Beginners Handbook, 2024.

[2] React Bootstrap: *React Bootstrap Cards*, *https://react-bootstrap.netlify.app/docs/components/cards/* , Accessed: 2025-03-15.