

Pokecoin Projektabgabe

Einführung

Der **Pokecoin Server** ist ein in Node JS geschriebener Webserver (<https://github.com/bykof/pokecoin-server>). Dieser Server bietet 5 Hauptgruppen als Open API konforme Endpunkte an.

- Users
 - Stellt die Authentifizierung und Erstellung von Benutzern zur Verfügung
- Blockchain
 - Stellt eine Blockchain zur Verfügung
- Wallet
 - Stellt eine Art "Portemonnaie" eines Benutzers zur Verfügung
- Cards
 - Stellt diverse Abfragen für Pokemon Karten zur Verfügung
- Views
 - Stellt Oberflächen zur Verfügung, um aktuelle Status einzusehen

Die API Endpunkte können über eine **Swagger UI** angezeigt werden:

<https://webeng.mi.hs-rm.de/docs/static/index.html>

Und die **swagger.json** kann über eine URL abgerufen werden:

<https://webeng.mi.hs-rm.de/docs/json>

Sollte sich die Adresse ändern, werde ich sie benachrichtigen.

User Story

Benutzer können sich beim Pokecoin Server mit ihrem Benutzernamen und Passwort registrieren. Beim Login bekommt der Benutzer einen Sicherheitstoken, der im Header bei POST- und GET-Requests mitgesendet werden muss. *Es gibt aber auch API Endpunkte, die nicht durch einen Token verifiziert werden müssen (bspw. /cards).*

Da der Pokecoin Server eine Blockchain besitzt, können sogenannte "**Pokecoins**" gefarmt werden. Wird ein korrekter Block an den Endpunkt "/blockchain/blocks" gePOSTet, gibt es eine Belohnung. Diese Belohnung ist ein Coin pro gefundenem Block.

Das aktuelle Guthaben der gefarmten Coins eines Benutzers, kann über "/wallet/balance" abgerufen werden. Sammelt der Benutzer genügend Pokecoins, kann er diese ausgeben für CardPackages "/cards/packages/{cardPackName}/buyDefaultPackage". Dabei wählt der Benutzer ein "CardPack" aus verschiedenen CardPackages "/cards/packages" aus. Wird ein CardPack gekauft, werden die dafür benötigten Coins automatisch abgezogen und der Benutzer

erhält seine erworbenen Cards. Die erworbenen Cards können jederzeit mit `"/cards/usercards"` abgerufen werden.

Grundsätzlich können sich die User alle Karten mit `"/cards/"` in einer Listenansicht anschauen und mit `"/cards/{cardId}"` in einer Detailansicht die Card genauer anschauen (in der auch weitere Detailinformationen angegeben werden).

Aufgabe / Anforderungen

Implementieren Sie eine React Single Page Application (SPA), die den Pokecoin Server konsumiert und folgende Funktionen erfüllt.

1. Ich als Benutzer kann mich mit meinem Benutzernamen und Passwort registrieren.
2. Ich als Benutzer kann mich mit meinem Benutzernamen und Passwort einloggen
3. Ich als Benutzer kann auf einer dedizierten Seite in der Applikation Pokecoins farmen. Ich kann diese Pokecoins **nur** farmen, wenn diese Seite **offen** ist und **angezeigt** wird. Auch das Wechseln in ein anderes Tab verhindert das Farmen.
4. Ich als Benutzer sehe zu jeder Zeit meine aktuellen Coins.
5. Ich als Benutzer kann auf einer Seite aus einer Anzahl an Cardpackages eins auswählen und es mit meinen Pokecoins kaufen.
6. Ich als Benutzer sehe auf einer Seite meine gekauften Cards als Listenansicht.
7. Ich als Benutzer kann in der Listenansicht meiner gekauften Cards auf eine Card klicken und komme in eine Detailansicht dieser Card. Auf der Detailansicht werden **alle** Informationen einer Karte angezeigt, die der Server zurückgibt.
8. Bekomme ich einen Fehler vom Backend, möchte ich als Benutzer wissen, was das Problem ist. Somit sollen mir aussagekräftige Fehlertexte angezeigt werden.

Nutzen Sie jegliche Bibliotheken (Chakra, Material UI etc.) um ihre SPA zu implementieren. Überlegen Sie sich, wie solch eine Applikation aussehen sollte und wie eine sinnvolle Benutzung erfolgen kann.

Bewertung

Die Lösung wird an Anhand folgender Punkte bewertet:

- Erfüllung der Anforderungen
- Benennung und sinnvolle Implementierung der Komponenten
- saubere Separierung von Logik, Datenhaltung und Anzeige (View, Model, Controller)
- sinnvolle Nutzung von Hooks

Abgabe

Offizielle Abgabe ist am **31.01.2022**

Bitte **packen** Sie das **Projekt und alle weiteren Dokumente (ggf. Dokumentation, Skizzen, Mockups)** in ein **ZIP-Archiv** und senden Sie es per E-Mail an michael.bykovski@hs-rm.de zu.