Frontend és backend összeépítése

# BACKEND

|  |  |
| --- | --- |
| Laravel projekt elkészítése | composer create-project laravel/laravel BackendFrontend |
| Migráció + modell | php artisan make:model Writers --migration |
| Controller | php artisan make:controller WritersController |
| **Legalább 2 adattal töltsd fel a táblád!** | |

## MIGRÁCIÓ:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

## Controller: index, store, update, destroy metódusok

A képen szöveg, képernyőkép, Betűtípus látható

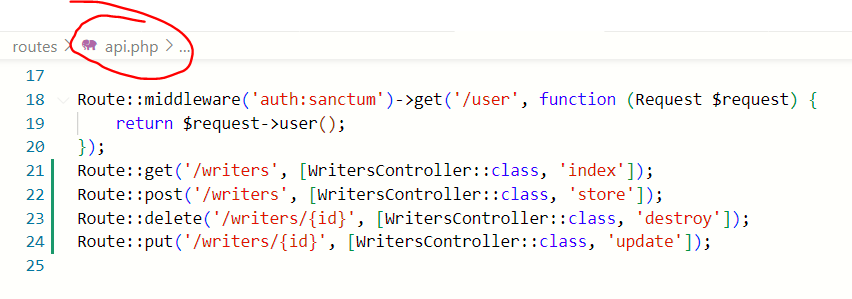
Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

## Végpontok az API.PHP-ban!

A végpontokat **az api.php**-ba írd!



# Teszteld a végpontokat!

# Backend oldalon a CORS beállítása

## Mi az a CORS?

A CORS a "Cross-Origin Resource Sharing" rövidítése, és egy biztonsági mechanizmus a webböngészőkben, amely segít megakadályozni a kereszt eredetű HTTP (HTTP Cross-Origin) kérések által okozott biztonsági problémákat. A kereszt eredetű HTTP kérés olyan kérés, amikor a weboldal egy forrás (pl. JavaScript vagy kép) elérésekor a kérés különbözik a forrás eredeti eredetétől. Az "eredeti eredet" itt azt az URL-t jelenti, amelyről a weboldal származik.

A CORS lényege, hogy a webböngészők ellenőrzik, hogy a kérésre engedélyt ad-e a célszerver, és csak akkor engedélyezik a kérést, ha a célszerver kifejezetten engedélyezi azt. A CORS felettébb fontos a webbiztonság szempontjából, mivel megakadályozza, hogy weboldalak káros kódokat futtassanak más weboldalakon, és megvédi a felhasználók a kereszt eredetű támadásoktól.

A CORS-t a HTTP fejlécek segítségével konfigurálják. Például a szerver a Access-Control-Allow-Origin fejléccel adhat engedélyt egy vagy több eredetnek, hogy kéréseket indítsanak. A böngésző pedig ellenőrzi, hogy a kérő oldal eredete szerepel-e a szerver engedélyezett eredetei között, mielőtt elküldené a kérést. Ha nincs megfelelő engedély, a kérés nem lesz végrehajtva.

A CORS mechanizmus segít megvédeni a weboldalakat és a felhasználókat a biztonsági fenyegetésektől, és lehetővé teszi, hogy a webes alkalmazások biztonságosan kommunikáljanak egymással, anélkül, hogy kihasználhatók lennének a kereszt eredetű támadások által.

## app/config/cors.php

"allowed\_origins" bejegyzés tartalmazza a megengedett domaineket, amelyekről engedélyezzük a végpontokhoz való hozzáférést!

Engedélyezni kell azokat a domaineket, ahonnan elérhetjük a végpontokat és a tokeneket.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

# Frontend oldal

Készítsd el az MVC szerkezetet a szokásos módon.

A Model réteghez tartozik a DataService osztály. Ez fogja kezelni az aszinkron kéréseket, ezúttal axios-t használunk ehhz.

Írd meg benne a getAxios, postAxios, putAxios, deleteAxios függvényeket!

Az osztály constructorában beállíthatod az alap fejléc beállításokat, például az alap URL-t

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus, dokumentum látható

Automatikusan generált leírás

Hívd meg a controllerben a megfelelő osztály metódusokat.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A megjelenit metódusban példányosíthatod a view-t.

Illetve a törlés, új adat, módosítás hívásokat majd a megfelelő eseménykere való feliratkozáskor kell meghívni.

# Mit jelent a CSRF token?

A CSRF token (Cross-Site Request Forgery token) egy biztonsági mechanizmus, amelyet webalkalmazások használnak a CSRF támadások elleni védekezéshez. A CSRF egy olyan támadásfajta, amely során egy rosszindulatú harmadik fél (például egy támadó) manipuláló kéréseket küld egy felhasználó nevében egy olyan webalkalmazás felé, amelyhez a felhasználó bejelentkezve van. Ezek a kérések akkor is végrehajtódnak, ha a felhasználó nincs tisztában velük, és komoly biztonsági veszélyt jelenthetnek.

A CSRF token azonosító vagy token, amit a webalkalmazás a felhasználóval kommunikálva generál, és minden olyan kéréshez vagy űrlaphoz hozzárendel, amely az alkalmazásban történik. A token általában rejtett mezőként vagy HTTP fejlécben szerepel a kérésekben. Amikor a felhasználó az alkalmazásban akciót végez, például egy űrlapot küld, a CSRF token is elküldődik az alkalmazásnak. Az alkalmazás ellenőrzi, hogy a kérésben lévő token egyezik-e a felhasználóval kapott tokennel. Ha nem egyezik, az alkalmazás elutasítja a kérést, mivel valószínűleg egy külső támadó próbált meg manipulálni.

A CSRF token használata nagymértékben hozzájárul az alkalmazások biztonságához, mivel megakadályozza a rosszindulatú kérések végrehajtását a felhasználói számlákban. Az alkalmazások fejlesztőinek és tervezőinek fontos figyelembe venniük a CSRF védelmet az alkalmazások tervezése során, és gondoskodni arról, hogy minden művelethez megfelelően használják a CSRF tokeneket.

A Laravel automatikusan generál CSRF tokent és elküldi a kliens böngészőjének, amikor a felhasználó bejelentkezik az alkalmazásba vagy űrlapot használ, vagy kérést indít. A CSRF token általában a csrf\_token néven található a HTTP fejlécben.