**Dentre**

**Fejlesztői dokumentáció**

**2023**

**Cél**

Egy étteremláncnak a nyilvántartás készítése, mely biztosítja az étterem és a vendégek számára a zökkenőmentes adminisztrációs munkákat.

A backend-t Laravel php keretendszerben, az adatbázist MYSQL szerver biztosítja.

**Felhasznált technológiák**

Backend

Laravel API(php keretrendszer)

* laravel -sanctum (Autentikációs kiegészítő csomag)

**Frontend**

**Mobil alkalmazás**

**Kódolási konvenciók**

A kódot git verziókezelővel használjuk.

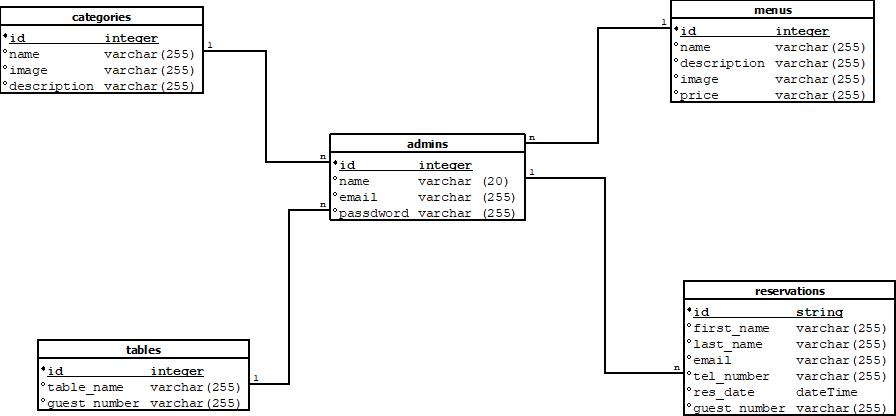
**Alapkönyvtárak**

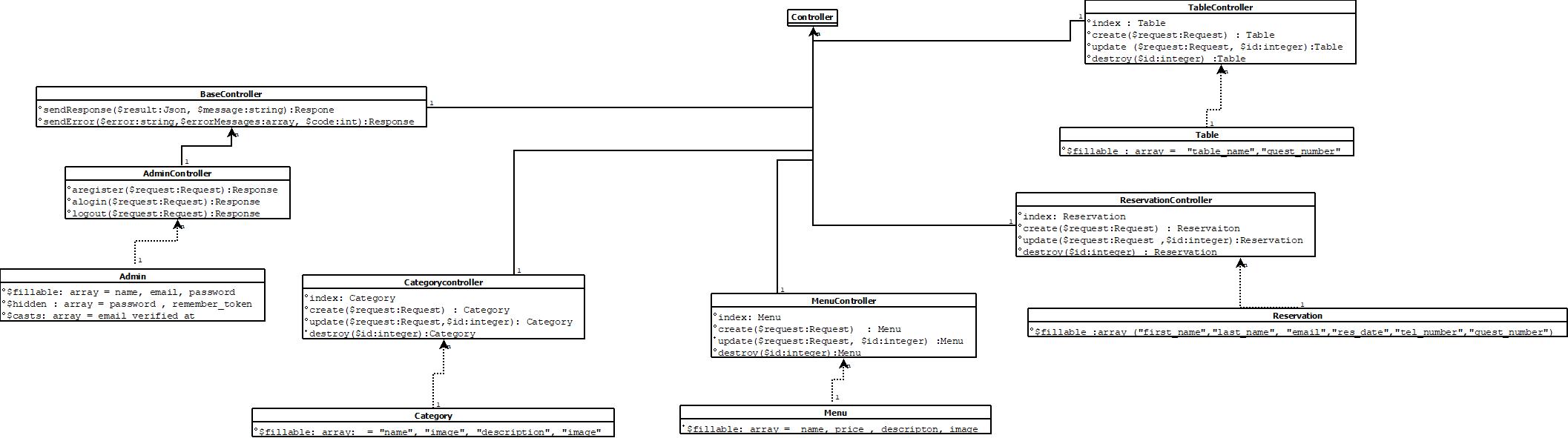
* api
* doc
* insomnia
* mobil
* web

Az api backendet, a web a webes frontend, doc a dokumentációt , az insomnia pedig a REST API lekérdezéséhez tartalmazó segítséget tartalmazza

**Felülettervek**

Adatmodell



API UML terv

**Végpontok**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Végpontok** | **Metódus** | **Azonosítás** | | **Leírás** | |
| /st | post | igen | | Kategória felvétel (admin) | |
| /categories/{id} | put | igen | | kategória frissítése | |
| /destroy/{id} | delete | igen | | kategória törlése | |
| /menus-create | post | igen | | menü létrehozása | |
| /destroy/{id} | delete | igen | | menü törlése | |
| /menus/{id} | put | igen | | menü frissítése | |
| /reservations | get | igen | | foglalások (csoportos lekérdezés) | |
| /reservations-create | post | nem | | foglalás igénylése | |
| /reservation/{id} | put | igen | | foglalás frissítése (ha a vendéknek megváltott az adata) | |
| /destroy/{id} | delete | igen | | foglalás törlése | |
| /table-create | post | igen | | asztal létrehozása (főasztal stb) | |
| /table/{id} | put | igen | | asztal frisstése | |
| /destroy/{id} | delete | igen | | asztal törlése | |
| /users | get | | igen | | az összes felhasználó megmutatása |
| /register | post | | nem | | regisztráció(admin) |
| /userreg | post | | nem | | felszanálói regisztálás |
| /login | post | | nem | | belépés |
| /categories | get | | nem | | összes felvett kategória |
| /menus | get | | nem | | összes felvett menü |
| /reservation-create | post | | nem | | foglalás |
| /tables | GET | | nem | | összes elérhető asztal |
| /logout | post | | nem | | admin kijelentkezés |
| /userlogout | post | | nem | | felhasználó kijelentkezése |

**Általános működés:**

A REST API http kéréseket fogad, melyek tartalmazzák a műveletekhez szükséges megfelelő adatokat. A titkos műveletek végpontjai védetted, autentikáció szükséges a használatukhoz.

Ilyenek az asztal, menük, kategóriák létrehozása-módosítása és a törlése is. A regisztráció, a be és kijelentkezés , a felhasználók elérhetik a menük, kategóriák és az elérhető szabad asztalokat , mivel nincs szükség autentikációra, mivel ezeknek a végpontjuk publikusak.

Az adatokat JSON formátumban fogadja és dolgozza fel. A vezérlést kontrollerek valósítják meg, minden adatkezelési csoportnak (autentikáció, kategória, menük, asztalok, foglalások) külön kontrollerük van és itt történik meg az adatfeldolgozás.

A kontrollerek modellekkel vannak kapcsolatban, amelyek az adatkezelésért felelősek. Minden adatkezelési csoportnak (felhasználó, admin, menük, kategóriák, foglalások, asztalok) külön modellje van, itt történik az adatok adatbázisból kiolvasása, illetve az adatok kiírása adatbázisba.

A modellek adatbázis táblákkal vannak kapcsolatban, melyek az adatok tárolásáért felelősek.

Az adatbázis táblák adatait a modellek kezelik.

**Osztályok**

**Admincontroller**

Feladata egy új admin felvétele, admin autentikációja, admin kijelentkeztetése.

**Metódusok**

1. **alogin()**

* Feladata a felhasználó azonosítása név és jelszó alapján. Sikeres autentikáció esetén generál
* egy tokent a felhasználó számára és bejegyzi a personal\_access\_tokens adatbázis tábla
* megfelelő mezőjébe, majd átadja a BaseController sendResponse() metódusának a saját üzenettel együtt.
* *bejövő paraméterek*: $request (a bejelentkezéshez szükséges adatok a kérésben, name, password ).
* *kimenő adatok*: token, saját üzenet.

1. **logout()**

* Feladata a felhasználó kijelentkeztetése és a token törlése az adatbázis táblából.
* *bejövő paraméterek:* $request (a kijelentkezéshez szükségek adatok a kérésben, token)
* *kimenő adatok:* saját üzenet.

1. **aregister()**

* *bejövő paraméterek:* $request (a regisztrációhoz szükséges adatok a kérésben, name, email, password, confirm\_password)
* Az adatokat validálja, majd sikeres érvényesítés után bejegyzi az admins adatbázis tábla megfelelő mezőibe.
* *kimenő adatok:* saját üzenet.

**BaseController**

Feladata a bejelentkezési műveletek válaszainak küldése. Sikeres művelet esetén a generált azonosító tokent visszaküldi, sikertelen művelet esetén hibaüzenetet küld.

**Metódusok:**

1. ***sendResponse()***

* *bejövő paraméterek:* $result (A generált token), $message (A felhasználónak szánt saját üzenet)
* kimenő adatok: Response (sikeres bejelentkezés esetén a token).

1. ***sendError()***

* *bejövő paraméterek:* $error (A php által generált hibaüzenet), $errorMessages (Saját hibaüzenet), $code (A válaszban küldendő http kód).
* *kimenő adatok:* Response (sikertelen bejelentkezés esetén hibaüzenet és a hiba kódja).

**CategoryController**

Feladata egy új kategória felvétele, meglévő kategóriák adatainak módosítása, meglévő kategóriák megjelenítése, meglévő kategóriák törlése.

**Metódusok:**

1. **index()**

* Visszaadja az adatbázisban szereplő kategóriák listáját csoportonként.
* *bejövő paraméterek:* NINCS
* *kimenő adatai:* Category (kategória modell amely tartalmazza a kategóriák adatbázisból lekért adatait.

1. **create()**

* Eltárol az adatbázisban egy új kategóriát.
* *bejövő paraméterek:* request (Kategóriák adatait tartalmazó kérés).
* *kimenő adatok*: Category (A kategóriák adatait tartalmazó modell) , egyéni üzenet.

1. **update()**

* Frissíti az adatbázisban tárolt egy adott kategória adatait.
* *bejövő paraméterek:* $request, $id ( a kategóriák adatait tartalmazó kérés és a kategóriák azonosítója ).
* *kimenő adatok:* Category ( a kategóriák adatait tartalmazó modell) , egyéni üzenet.

1. **destroy()**
   * Feladata a kiválasztott katergória törlése az adatbázisból.
   * *bejövő paraméterek:* $id (A kiválasztott kategória azonosítója ).
   * *kimenő adatok:* Category (A kiválasztott kategória adatait tartalmazó modell) , egyéni üzenet.

**Menucontroller**

Feladata egy új menü felvétele, meglévő menük adatainak módosítása, meglévő menük megjelenítése, meglévő menük törlése.

**Metódusok:**

1. **index ()**

* Visszaadja az adatbázisban szereplő menük listáját csoportonként.
* *bejövő paraméterek:* NINCS
* *kimenő adatai:* Menu (menü modell amely tartalmazza a menük adatbázisból lekért adatait.

1. **create()**

* Eltárol az adatbázisban egy új menüt.
* *bejövő paraméterek:* request (Menük adatait tartalmazó kérés).
* *kimenő adatok*: Menu (A menük adatait tartalmazó modell), egyéni üzenet.

1. **update()**

* Frissíti az adatbázisban tárolt egy adott menü adatait.
* *bejövő paraméterek:* $request, $id ( a menük adatait tartalmazó kérés és a menük azonosítója ).
* *kimenő adatok:* Menu ( a menük adatait tartalmazó modell ), egyéni üzenet,.

1. **destroy()**
   * Feladata a kiválasztott menü törlése az adatbázisból.
   * *bejövő paraméterek:* $id (A kiválasztott menü azonosítója ).
   * *kimenő adatok:* Menu (A kiválasztott menü adatait tartalmazó modell) , egyéni üzenet.

**ReservationController**

Feladata egy új foglalás felvétele, meglévő foglalások adatainak módosítása, meglévő foglalások megjelenítése, meglévő foglalások törlése.

**Metódusok:**

1. **index ()**

* Visszaadja az adatbázisban szereplő foglalások listáját csoportonként.
* *bejövő paraméterek:* NINCS
* *kimenő adatai:* Reservation (foglalások modell amely tartalmazza a foglalások adatbázisból lekért adatait.

1. **create()**

* Eltárol az adatbázisban egy új foglalást.
* *bejövő paraméterek:* request (Foglalások adatait tartalmazó kérés).
* *kimenő adatok*: Reservation (A foglalások adatait tartalmazó modell, egyéni üzenet).

1. **update()**

* Frissíti az adatbázisban tárolt egy adott foglalás adatait.
* *bejövő paraméterek:* $request, $id ( a foglalások adatait tartalmazó kérés és a foglalások azonosítója ).
* *kimenő adatok:* Reservation (a foglalások adatait tartalmazó modell, egyéni üzenet).

1. **destroy()**
   * Feladata a kiválasztott foglalás törlése az adatbázisból.
   * *bejövő paraméterek:* $id (A kiválasztott foglalás azonosítója ).
   * *kimenő adatok:* Reservation (A kiválasztott menü adatait tartalmazó modell) , egyéni üzenet.

**TableController**

Feladata egy új asztal felvétele, meglévő asztalok adatainak módosítása, meglévő asztalok megjelenítése, meglévő asztalok törlése.

**Metódusok:**