Tick it easy

(Erdősi Péter, Hires Krisztián)
/https://github.com/epeter95/tick-it-easy/

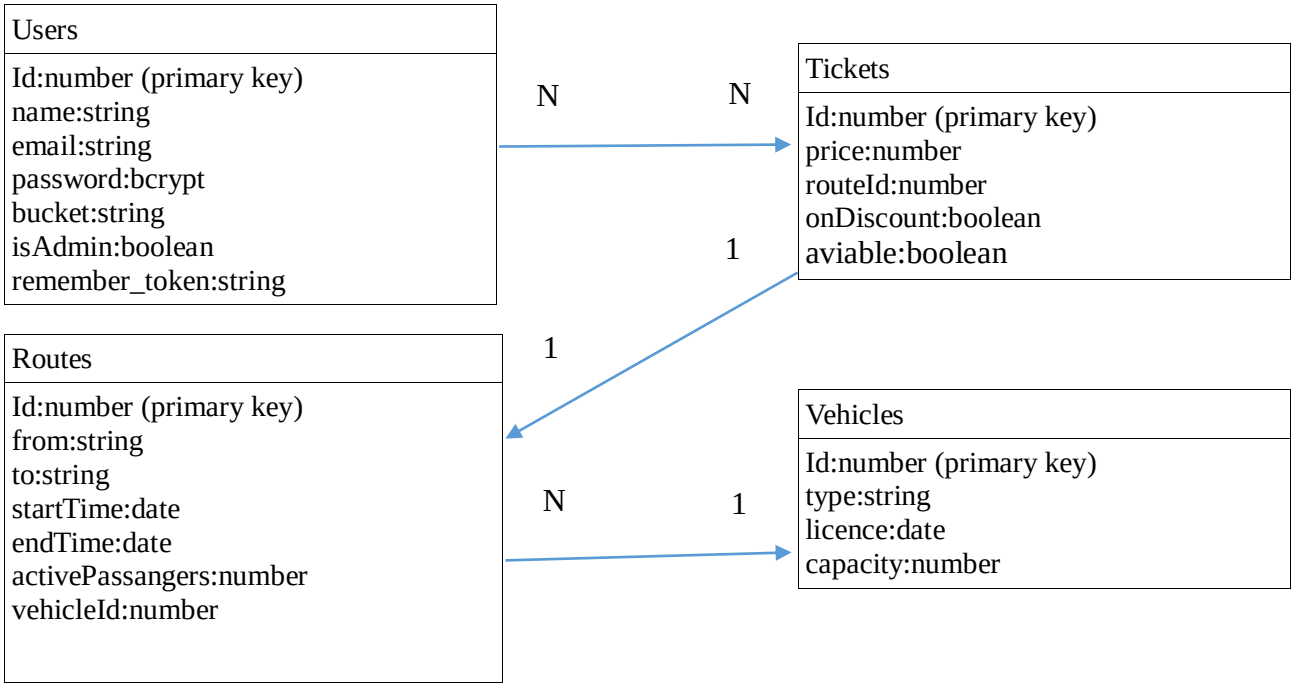
FE: https://tick-it-easy-fe.herokuapp.com/

BE: https://tick-it-easy-backend.herokuapp.com/

A commitált kód CI környezetben tesztelve van, majd sikeres tesztelést követően közzé van téve a fent említett domain címeken.
CI-tool: Travis CI
CD-tool: Heroku

1. **fejlesztői környezet bemutatása, beállítása, használt technológiák:**
   A fejlesztői környezet Laravel 8 és Angular 10.
   Backend futtatása: php artisan serve (lokálba a localhost:8000 porton)
   Frontend futtatása: ng serve (lokálba localhost:4200) / npm run start (lokálba localhost:8080)
   Használt technologiák:
   - adatbázis: (travis környezetben MySQL, Produkciós környezetben PostgreSQL)
   - laravelen belül: ORM megvalósítása Eloquent
   - adatbázis szerkezet létrehozása illetve módisítása migrációval, kezdetleges adatfeltöltése seederekkel történik
   - backend oldali tesztelés: PHPunit, tesztek feature tesztek a REST API tesztelésére
   - későbbiekben az authentication-t a laravelben a passport package fogja végezni illetve a custom middleware-ek

2. **adatbázis-terv: táblák kapcsolati UML diagramja:**

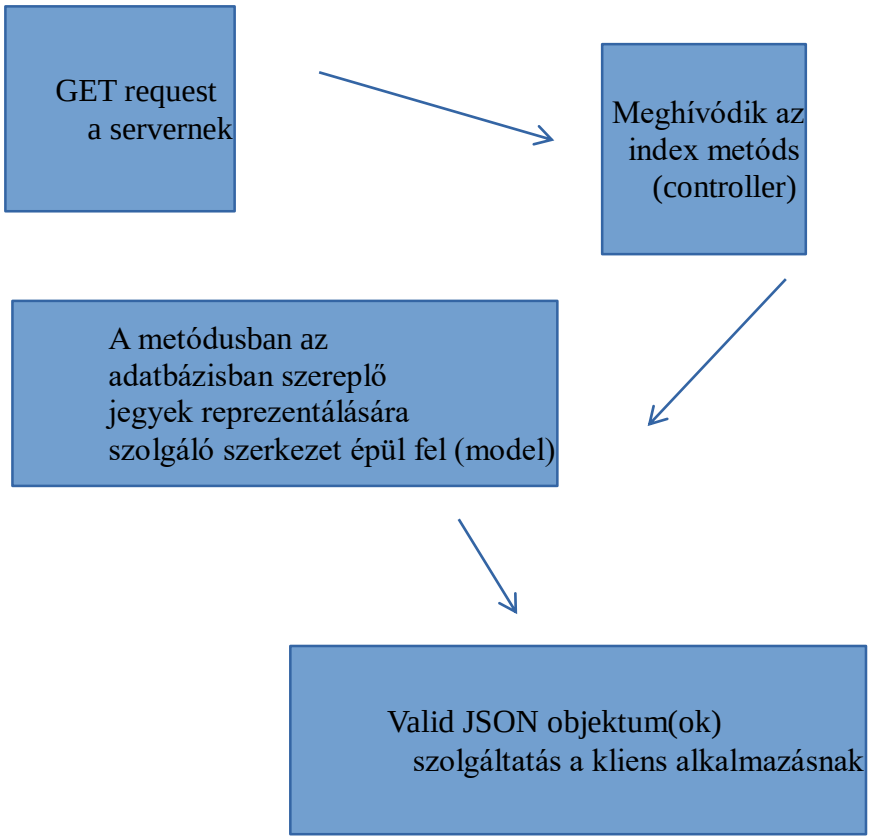


3. **alkalmazott könyvtárstruktúra bemutatása:**
   A mappastrukturában így található meg: Laravel 8 – tick-it-easy-backend, Angular 10 – frontend. A strukturában jelen van még egy ReadMe.md ahol ismertetjük az webalkalmazást, illetve egy CI/CD folyamatot vezérlő / levezető yml fájl a Travis CI számára.
4. **Végpont-tervek és leírások:**

Domain	Method	URI	Name	Action	Middleware
	GET|HEAD	/		Closure	web
	GET|HEAD	api/routes	routes.index	App\Http\Controllers\RoutesController@index	api
	POST	api/routes	routes.store	App\Http\Controllers\RoutesController@store	api
	GET|HEAD	api/routes/create	routes.create	App\Http\Controllers\RoutesController@create	api
	DELETE	api/routes/{route}	routes.destroy	App\Http\Controllers\RoutesController@destroy	api
	PUT|PATCH	api/routes/{route}	routes.update	App\Http\Controllers\RoutesController@update	api
	GET|HEAD	api/routes/{route}	routes.show	App\Http\Controllers\RoutesController@show	api
	GET|HEAD	api/routes/{route}/edit	routes.edit	App\Http\Controllers\RoutesController@edit	api
	GET|HEAD	api/tickets	tickets.index	App\Http\Controllers\TicketsController@index	api
	POST	api/tickets	tickets.store	App\Http\Controllers\TicketsController@store	api
	GET|HEAD	api/tickets/create	tickets.create	App\Http\Controllers\TicketsController@create	api
	DELETE	api/tickets/{ticket}	tickets.destroy	App\Http\Controllers\TicketsController@destroy	api
	PUT|PATCH	api/tickets/{ticket}	tickets.update	App\Http\Controllers\TicketsController@update	api
	GET|HEAD	api/tickets/{ticket}	tickets.show	App\Http\Controllers\TicketsController@show	api
	GET|HEAD	api/tickets/{ticket}/edit	tickets.edit	App\Http\Controllers\TicketsController@edit	api
	GET|HEAD	api/user		Closure	api
					auth:api
	POST	api/vehicles	vehicles.store	App\Http\Controllers\VehiclesController@store	api
	GET|HEAD	api/vehicles	vehicles.index	App\Http\Controllers\VehiclesController@index	api
	GET|HEAD	api/vehicles/create	vehicles.create	App\Http\Controllers\VehiclesController@create	api
	GET|HEAD	api/vehicles/{vehicle}	vehicles.show	App\Http\Controllers\VehiclesController@show	api
	PUT|PATCH	api/vehicles/{vehicle}	vehicles.update	App\Http\Controllers\VehiclesController@update	api
	DELETE	api/vehicles/{vehicle}	vehicles.destroy	App\Http\Controllers\VehiclesController@destroy	api
	GET|HEAD	api/vehicles/{vehicle}/edit	vehicles.edit	App\Http\Controllers\VehiclesController@edit	api

5. **1 db végpont működésének leírása, mi történik, milyen lépések követik egymást:**

<domain>/api/tickets működése:



6. **fontosabb specifikumok bemutatása (ha van ilyen):**
   Az összes az :)