**Web programozás**

**12.Laborgyakorlat**

**Téma:**

* Laravel CRUD műveletek (*sapistudents* projekt folytatása)
* REST API Laravel-ben

**Feladat:**

1. Hallgató létrehozása (“C”)
   1. Űrlap létrehozása a *create.blade.php* fájlban

<form method="post" action="">

@csrf

<div class="form-group">

<label for="name">Name:</label>

<input type="text" class="form-control" name="name"/>

</div>

<div class="form-group">

<label for="email">Email:</label>

<input type="text" class="form-control" name="email"/>

</div>

<button type="submit" class="btn btn-primary">Add</button>

</form>

* 1. A *store* metódus létrehozása *StudentController.php* fájlban

public function store(Request $request)

{

$request->validate([

'name' => 'required',

'email' => 'required|email',

]);

Student::create($request->all());

return redirect()->route('students.index')

->with('status','Student created successfully');

}

* 1. Útvonal beállítása a *web.php*-ben
     1. Űrlap megjelenítés

Route::get('students/create', [StudentController::class, 'create'])->name('students.create');

* + 1. Űrlap küldés

Route::post('/students/store', [StudentController::class, 'store'])->name('students.store');

* 1. Az „action” beállítása az űrlapban

<form method="post" action="{{ route('students.store') }}">

* 1. Űrlap megjelenítése: create metódus a StudentController.php-ben

public function create()

{

return view('students.create');

}

* 1. Tesztelés
     1. <http://localhost/sapistudents/public/students/create>

1. Módosítsuk a táblázatunkat, hozzáadva a szerkesztés és törlés lehetőségekkel (Edit, Delete). Példa:



1. Hallgató szerkesztése („U”)
   1. Az *edit* metódus létrehozása a *StudentController*-ben

public function edit(Student $student)

{

return view('students.edit', compact('student'));

}

* 1. Útvonal beállítása a *web.php*-ben

Route::get('students/edit/{student}',[StudentController::class, 'edit'])->name('students.edit');

* 1. A *href* (Edit) beállítása az index.blade.php-ben

href="{{ route('students.edit', $student->id) }}"

* 1. A szerkesztési űrlap létrehozása az *edit.blade.php* fájlban

<form method="post" action="" >

@csrf

<div class="form-group">

<label for="title">Név:</label>

<input type="text" class="form-control" name="name" value="{{$student->name}}" />

</div>

<div class="form-group">

<label for="title">E-mail cím:</label>

<input type="email" class="form-control" name="email" value="{{$student->email}}" />

</div>

<button type="submit" class="btn btn-primary">Update</button>

</form>

* 1. Az *update* metódus létrehozása a *StudentController*-ben

public function update(Request $request, Student $student)

{

$request->validate([

'name' => 'required',

'email' => 'required|email',

]);

$student->update($request->all());

return redirect()->route('students.index')

->with('status','Student updated successfully');

}

* 1. Útvonal beállítása a *web.php*-ben

Route::post('students/edit/{student}',[StudentController::class, 'update'])-> name('students.update');

* 1. Az „action” beállítása az űrlapban (*edit.blade.php*)

action="{{ route('students.update', $student->id) }}"

1. Hallgató törlése (“D”)
   1. A törlés módosítása az index.blade.php-ben

<td>

<form action="" method="post">

@csrf

@method("DELETE")

<button class="" type="submit">Delete</button>

</form>

</td>

* 1. A *destroy* metódus létrehozása a *StudentController*-ben

public function destroy(Student $student)

{

//$stud = Student::find($id);

$student->delete();

return redirect()->route('students.index')

->with('status', 'Student deleted successfully');

}

* 1. Útvonal beállítása a *web.php*-ben

Route::delete('/students/{student}', [StudentController::class, 'destroy'])->name('students.destroy');

* 1. Az „action” beállítása az űrlapban (*index.blade.php*)

action="{{ route('students.destroy',$student->id) }}"

1. REST API Laravel-ben
   1. Erőforrás létrehozása (app\Http\Resources)

php artisan make:resource StudentResource

* 1. Kontroller kétrehozása (API mappa)

php artisan make:controller API\\StudentController

* 1. Kontroller index metódus létrehozása

public function index()

{

$students = Student::all();

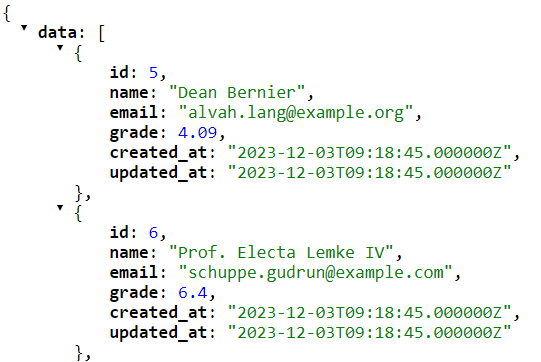
return StudentResource::collection($students);

}

* 1. Route létrehozása (routes\api.php)

Route::get('students', [App\Http\Controllers\API\StudentController::class, 'index']);

* 1. Tesztelés
     1. <https://localhost/sapistudents/api/students>



* 1. *StudentResource* módosítása

return [

'id' => $this->id,

'name' => $this->name,

'email' => $this->name,

'created\_at' => $this->created\_at,

'updated\_at' => $this->updated\_at,

]

* 1. Teszteljed az API-t Postman-ben