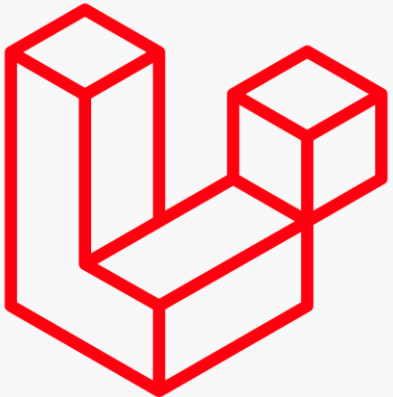


# Laravel Training d2

Tarmizi Sanusi

Khirulnizam Abd Rahman

<http://fstm.kuis.edu.my/blog/laravel>



# Module Outcome

## Day 2

- Relationship
- Show, Edit, Update & Delete
- Using Named Route
- Auth Middleware
- Handling Error Page
- Pagination

## 2. Route to appropriate Laravel Controller

## 1. Submit User Request



**Routing**

**routes/web.php**

**app/Http/Controllers**

**Controller**

## 3. Interact with Data Model

## 4. Controller invokes results View

**resources/views**

**View**

## 5. Render view in users browser

**Model**

**app/**



**Database**

# URL Named Route

The route helper may be used to generate URLs to named routes.

Named routes allow you to generate URLs without being coupled to the actual URL defined on the route.

Therefore, if the route's URL changes, no changes need to be made to your route function calls.

```
36 + <li class="nav-item">
37 +     <a class="nav-link" href="{{ route('training:index') }}">{{ __('Trainings') }}</a>
38 + </li>
```

<https://github.com/samtarmizi/laravel-spr/commit/fe5e375c4305294b817d8234fdb21bbb0877fe4>

# Routing

php artisan route:list

<https://laravel.com/docs/6.x/routing>.

GET HEAD	trainings	training:index	App\Http\Controllers\TrainingController@index	web,auth
GET HEAD	trainings/create	training:create	App\Http\Controllers\TrainingController@create	web,auth
POST	trainings/create	training:store	App\Http\Controllers\TrainingController@store	web,auth

# Model

```
protected $fillable = [  
    'title', 'description', 'trainer', 'user_id', 'attachment'  
];
```

```
public function user()  
{  
    return $this->belongsTo('App\User');  
}
```

```
protected $table= 'articles';
```

```
//define $article->submitted_date  
public function getSubmittedDateAttribute(){  
    return $this->created_at->format('d/m/Y');  
}
```

```
public function scopePublished($query){  
    return $query->where('published',1);  
}
```

Model used to define fillable attributes.

Model also can be used to define relationship

Model also can be used to to specify custom table.

Model also can be used to define mutator

Model also can be used to define local scope

# Relationship 1:M

A one-to-many relationship is used to define relationships where a single model owns any amount of other models.

**User has many trainings**

**Training belongs to User**

<https://laravel.com/docs/6.x/eloquent-relationships#one-to-many>

<https://github.com/samtarmizi/laravel-spr/commit/fa0e6673b4c73b13d2e8f2a33ef03ea22dfa50c4>

## On Training.php

```
public function user()  
{  
    return $this->belongsTo('App\User');  
}
```

## On User.php

```
public function trainings()  
{  
    return $this->hasMany('App\Training');  
}
```

# Relationship – Lazy Load Attribute

A one-to-many relationship is used to define relationships where a single model owns any amount of other models.

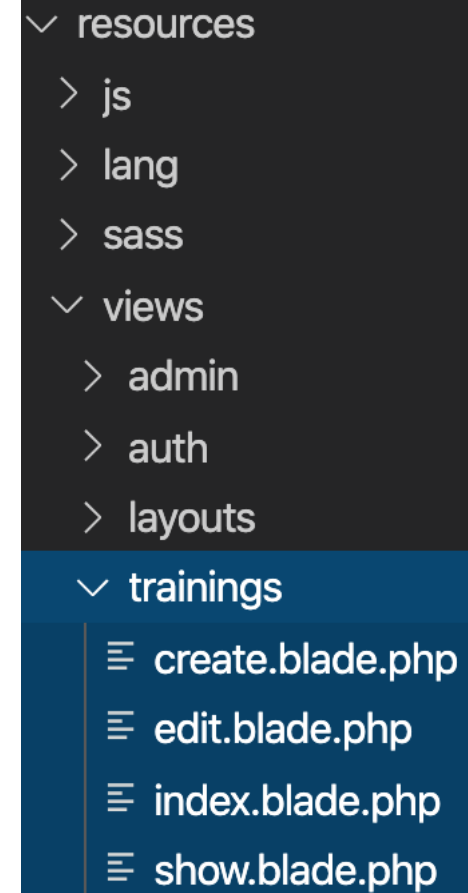
```
<td>{{ $training->user_id}}</td>  
<td>{{ $training->user->name}}</td>
```



# Views

Views are stored in the resources/views directory

1. Go to resources/views
2. Go to trainings folder, create:
  1. show.blade.php
  2. edit.blade.php



```
✓ resources
  > js
  > lang
  > sass
  ✓ views
    > admin
    > auth
    > layouts
  ✓ trainings
    ≡ create.blade.php
    ≡ edit.blade.php
    ≡ index.blade.php
    ≡ show.blade.php
```

# Controller

7 method in controllers

## Make Controller

```
php artisan make:controller TrainingController
```

### Action

~~index~~

~~create~~

~~store~~

show

edit

update

destroy

SHOW

# Controller - Show

Description	URL	Controller Function	View File
Show one of the trainings.	GET example.com/trainings/{id}	show()	app/views/trainings/show.blade.php

```
public function show(Training $training)
{
    return view('trainings.show')->with(compact('training'));
}
```

# Views – Show View

Views are stored in the resources/views directory

1. Go to resources/views/trainings/show.blade.php

```
@extends('layouts.app')

@section('content')
<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-8">
      <div class="card">
        <div class="card-header">Show Training</div>

        <div class="card-body">
          <div class="form-group">
            <label for="title">Title</label>
            <input type="text" class="form-control" name="title" value="{{ $training->title }}" readonly>
          </div>
          <div class="form-group">
            <label for="title">Description</label>
            <textarea name="description" cols="20" rows="10" class="form-control" readonly>{{ $training->title }}</textarea>
          </div>
          <div class="form-group">
            <label for="title">Trainer</label>
            <input type="text" class="form-control" name="trainer" value="{{ $training->trainer }}" readonly>
          </div>
          <div class="form-group">
            <a href="{{ route('training:index') }}" class="btn btn-link">Back to Training Index</a>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
@endsection
```

<https://github.com/samtarmizi/laravel-spr/blob/f024cb1035331aa1e845db71e5321ba154016140/resources/views/trainings/show.blade.php>

EDIT

# Controller - Edit

Description	URL	Controller Function	View File
Pull a training from the database and allow editing.	GET example.com/trainings/{id}/edit	edit()	app/views/trainings/edit.blade.php

```
public function edit(Training $training)
{
    return view('trainings.edit')->with(compact('training'));
}
```

# Views – Edit View

Views are stored in the resources/views directory

1. Go to resources/views/trainings/edit.blade.php

```
@extends('layouts.app')

@section('content')
<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-8">
      <div class="card">
        <div class="card-header">Edit Training</div>

        <div class="card-body">
          <form action="" method="POST">
            @csrf
            <div class="form-group">
              <label for="title">Title</label>
              <input type="text" class="form-control" name="title" value="{{ $training->title }}">
            </div>
            <div class="form-group">
              <label for="title">Description</label>
              <textarea name="description" cols="20" rows="10" class="form-control">{{ $training->title }}</textarea>
            </div>
            <div class="form-group">
              <label for="title">Trainer</label>
              <input type="text" class="form-control" name="trainer" value="{{ $training->trainer }}">
            </div>
            <div class="form-group">
              <button type="submit" class="btn btn-primary">Update Training</button>
              <a href="{{ route('training:index') }}" class="btn btn-link">Cancel</a>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
</div>
@endsection
```

<https://github.com/samtarmizi/laravel-spr/blob/f024cb1035331aa1e845db71e5321ba154016140/resources/views/trainings/edit.blade.php>



UPDATE

# Controller - Update

Description	URL	Controller Function	View File
Process the create form submit and save the training to the database.	POST example.com/trainings/{id}	update()	NONE

```
public function update(Request $request, Training $training){  
  
    $training->update($request->only('title','description','trainer'));  
  
    return redirect('/trainings');  
}
```

DELETE

# Controller – Destroy/Delete

Description	URL	Controller Function	View File
Delete the training from DB.	GET example.com/trainings{id}/delete	delete()	NONE

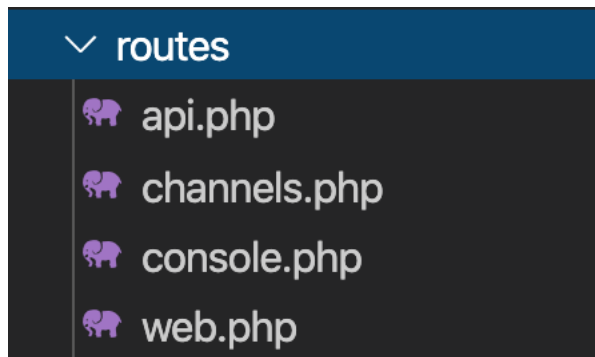
```
public function delete(Training $training)
{
    $training->delete();

    return redirect('/trainings');
}
```

# Routing – Group and Named Route

All Laravel routes are defined in your route files, which are located in the routes directory.

<https://laravel.com/docs/6.x/routing>.



```
Route::group([
    'middleware' => ['auth'],
    'prefix' => 'trainings',
    'as' => 'training:'
], function() {
    Route::get('/', 'TrainingController@index')->name('index');
    Route::get('/create', 'TrainingController@create')->name('create');
    Route::post('/create', 'TrainingController@store')->name('store');
    Route::get('/{training}', 'TrainingController@show')->name('show');
    Route::get('/{training}/edit', 'TrainingController@edit')->name('edit');
    Route::post('/{training}', 'TrainingController@update')->name('update');
    Route::get('/{training}/delete', 'TrainingController@delete')->name('delete');
});
```

# Routing – Group and Named Route

php artisan route:list

<https://laravel.com/docs/6.x/routing>.

GET HEAD	trainings/{training}	training:show	App\Http\Controllers\TrainingController@show	web,auth
POST	trainings/{training}	training:update	App\Http\Controllers\TrainingController@update	web,auth
GET HEAD	trainings/{training}/delete	training:delete	App\Http\Controllers\TrainingController@delete	web,auth
GET HEAD	trainings/{training}/edit	training:edit	App\Http\Controllers\TrainingController@edit	web,auth

# Routing – Named Route

Named route with model binding

```
<a href="{{ route('training:show',$training) }}" class="btn btn-primary">Show</a>
<a href="{{ route('training:edit',$training) }}" class="btn btn-success">Edit</a>
<a href="{{ route('training:delete',$training) }}" class="btn btn-danger"
    onclick="return confirm('Are you sure?')">Delete</a>
```

# Hacks

Creating Show - <https://github.com/samtarmizi/laravel-spr/commit/3ab27f716f0647077a0d524739d1b62d6bbae6f0>

Creating Edit - <https://github.com/samtarmizi/laravel-spr/commit/f024cb1035331aa1e845db71e5321ba154016140>

Creating Update - <https://github.com/samtarmizi/laravel-spr/commit/08c8b57b3e4b1e79577c236b41f6c689c8c2679b>

Creating Delete - <https://github.com/samtarmizi/laravel-spr/commit/f886bdc85de7b1eb217e364909efae38f71223b2>



# Middleware

Middleware provide a convenient mechanism for filtering HTTP requests entering your application.


For example, Laravel includes a middleware that verifies the user of your application is authenticated. If the user is not authenticated, the middleware will redirect the user to the login screen.

```
Route::get('/home', 'HomeController@index')->name('home')->middleware('auth');
```


```
public function __construct()
{
    $this->middleware('auth');
}
```

# Pagination

The paginate method automatically takes care of setting the proper limit and offset based on the current page being viewed by the user.

app/Http/Controllers/TrainingController.php 

```
$trainings = Training::all();  
$trainings = Training::paginate(5);
```

resources/views/trainings/index.blade.php 

```
@@ -54,6 +54,8 @@
```

```
@endforeach
```

```
</tbody>
```

```
</table>
```

```
+ {{ $trainings->links() }}
```

# Laravel Error Page

## Custom HTTP Error Pages

1. Create folder name **errors** in `resources/views/`
2. Example Http Error 404, create `404.blade.php`

