

Web Framework Programming

Interactive UI Modal & Ajax with JQuery

WEEK 07

Informatics Engineering
Universitas Surabaya



Outline

1

Modals UI

2

Ajax Concept

3

Ajax in Laravel

4

Implementing Ajax in a
Dynamic Modal

Modals UI

What is Modal

A **modal window** is an element that **sits on top of an application's main window**. It creates a mode that disables the main window but keeps it visible with the modal window as a child window in front of it. Users must interact with the modal window before they can return to the parent application. — Wikipedia

Example: <https://getbootstrap.com/docs/4.0/components/modal/>

When we use modal ?

Grab the user's attention

- If you want to interrupt a user's current task to catch the user's attention to something more substantial.

Need user input

- If you want to get information from the user. Ex. sign up or login form.

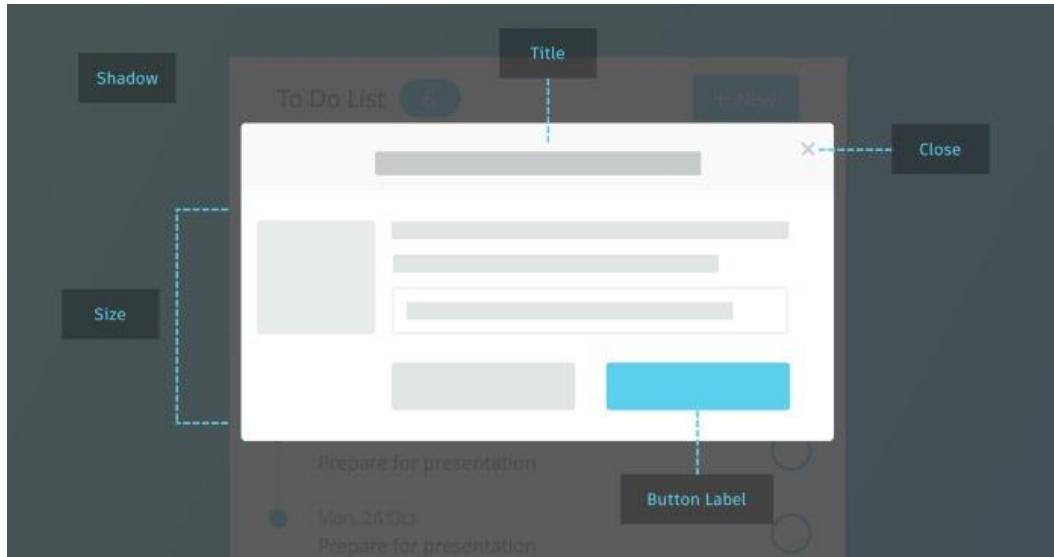
Show additional information in context.

- If you want to show additional information without losing the context of the parent page. Ex. showing larger images or videos.

Show additional information (not in context)

- If you want to show information that is not directly related to the parent page or other options that are "independent" from other pages. Ex. notifications.

Modal Structure



**Modal Content can
be Static or
Dynamic**

For AdminLTE4, we use this link:

- <https://adminlte.io/themes/v3/pages/UI/modals.html>
- <https://getbootstrap.com/docs/5.3/components/modal/>

We need additional javascript or CSS based on child layout

For additional JavaScript or CSS, we can use @stack and @push syntax on Blade Template Laravel.

@stack('name'): This directive in your layout view defines a named stack, which is essentially an array where you can **push multiple pieces of content**. It will render all the items that have been **pushed onto this stack**.

@push('name') ... @endpush: This directive in a child view adds content onto the specified named stack. You can push individual strings or blocks of HTML/Blade code. Importantly, you **can @push multiple times to the same stack from different parts of your views**.

layouts/app.blade.php (Layout View):

```
Blade

<!DOCTYPE html>
<html>
<head>
    <title>My App</title>
    @stack('scripts')
</head>
<body>
    <div class="container">
        @yield('content')
    </div>
</body>
</html>
```



resources/views/product/show.blade.php (Child View):

Blade

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

@section('content')
    <h1>Product Details</h1>
    @push('scripts')
        <script src="{{ asset('js/product_details.js') }}"></script>
    @endpush

    @push('scripts')
        <script>
            console.log('Additional script for product details');
        </script>
    @endpush
@endsection
```



For information:

- After Title tag, there are 2 scripts: (product_detail.js & console log)
- We can put @push inside @section or after @endsection

What the differences between @yield/@section and @stack/@push

Feature	@section / @yield	@push / @stack
Purpose	Define and inject a single block of content.	Define a stack to accumulate and render multiple items .
Cardinality	One section in the child corresponds to one yield .	Multiple pushes can contribute to a single stack .
Replacement	Child section content replaces the yield's default.	Pushed content is added to the stack.
Use Cases	Defining layout parts (title, sidebar, main content).	Including multiple CSS/JS files or inline scripts.

Make sure you have bootstrap JS and CSS before execution

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

CDN links

As reference, here are our primary CDN links.

Description	URL
CSS	https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css
JS	https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle.min.js

CDN via jsDelivr

Skip the download with [jsDelivr](#) to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4d6tcA065dIqgDZJtC07d8H+gjBkT+hpuabZD9F7eGm3wzD8WZJ/8q/gPQXWZfA==" crossorigin="anonymous" referrerpolicy="no-referrer">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/js/bootstrap.bundle.min.js" integrity="sha384-FO0sLjEPA033cj45AtmcI4qW1Q+H+NlEYD95aRf/FjxOOjm+S3P89aKqFf8Uw=" crossorigin="anonymous" referrerpolicy="no-referrer">
```

Copied!

If you're using our compiled JavaScript and prefer to include Popper separately, add Popper before our JS, via a

Check your layout (adminlte4)

```
59      crossorigin="anonymous"
60  />
61
62  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4d6tcA065dIqgDZJtC07d8H+gjBkT+hpuabZD9F7eGm3wzD8WZJ/8q/gPQXWZfA==" crossorigin="anonymous" referrerpolicy="no-referrer">
63 </head>
64 <!--end::Head-->
65
66  <!--end::Required Plugin(popper.js for Bootstrap 5)--><!--begin::Required Plugin(Bootstrap 5)-->
67  <script
68    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
69    integrity="sha384-0pUGzbkm6XF6gxjEnlmugrJXVbNuzT9qBBavbLwCsOGabyfzo0T0to5eqruptLy"
70    crossorigin="anonymous"
71  ></script>
```

Exercise #1

Showing Disclaimer - a simple static Modal in one page view

Two section for modals construction

1. Button to trigger modals
2. Hidden Modal elements

Modal Structure: Button

```
<button type="button" class="btn btn-primary" data-bs-toggle="modal"
       data-bs-target="#exampleModal">
    Launch demo modal
</button>

3 @section('content')
4 <!-- Button trigger modal -->
5 <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
6     Launch demo modal
7 </button>
```

Modal Structure: Form Modal

```
<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">Modal title</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        ...
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>
```

*put anywhere in the section content. Please try inside `List of Categories(Index of Category)` view.

```
42 @push ('modals')
43
44 <!-- Modal -->
45 <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
46   <div class="modal-dialog">
47     <div class="modal-content">
48       <div class="modal-header">
49         <h1 class="modal-title fs-5" id="exampleModalLabel">Modal title</h1>
50         <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
51     </div>
52     <div class="modal-body">
53       ...
54     </div>
55     <div class="modal-footer">
56       <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
57       <button type="button" class="btn btn-primary">Save changes</button>
58     </div>
59   </div>
60 </div>
61 </div>
62
63 @endpush
```

Launch demo modal

Hover Rows

The .table-hover class enables a hover state on table rows:

#	Name	Number of Food	List of Food Name
1	Appetizer	0	
2	Main Course	2	<ul style="list-style-type: none">• Nasi Merah dengan Ayam Panggang Kecap & Tumis Kangkung• Nasi Hitam dan Tumis Ca Kailan
3	Snack	0	
4	Dessert	0	
5	Coffee	0	
6	Non Coffee	0	
7	Healthy Juice	0	



3

15



over Rows

table-hover class enables a hover

Name	Number
Appetizer	0
Main Course	2
Snack	0
Dessert	0
Coffee	0
Non Coffee	0
Healthy Juice	0

Modal title



...

Close

Save changes

- Nasi Merah dengan Ayam Panggang Kecap & Tumis Kangkung
- Nasi Hitam dan Tumis Ca Kailan

Exercise #2

Show Photo Product Category

We use “storage” for images

<https://laravel.com/docs/10.x/filesystem#the-public-disk>

Pre-requirements

Make a migration update for add column image in category table.

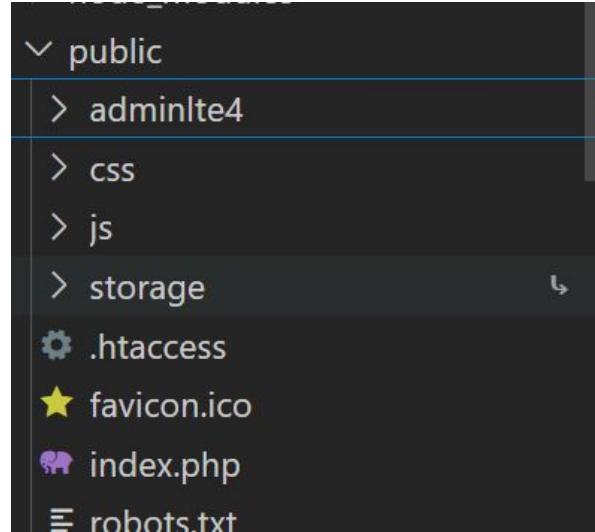
*php artisan make:migration
update_category_table*

```
● PS C:\laragon\www\example-app> php artisan make:migration update_category_table
INFO | Migration [C:\laragon\www\example-app\database\migrations\2025_04_09_172212_update_category_table.php] created successfully.
```

```
9  /**
10  * Run the migrations.
11  */
12 public function up(): void
13 {
14     Schema::table('categories', function (Blueprint $table) {
15         $table->string('image', 1000);
16     });
17 }
18
19 /**
20 * Reverse the migrations.
21 */
22 public function down(): void
23 {
24     Schema::table('categories', function (Blueprint $table) {
25         $table->dropColumn(['image']);
26     });
27 }
28 };
```

Pre-requirements (2)

- Make a shortcut folder (symlink) from storage to public folder with:
php artisan storage:link
- Make new folder inside storage.
Ex: category to store some images for each category data.
- Don't forget to update data in your database.

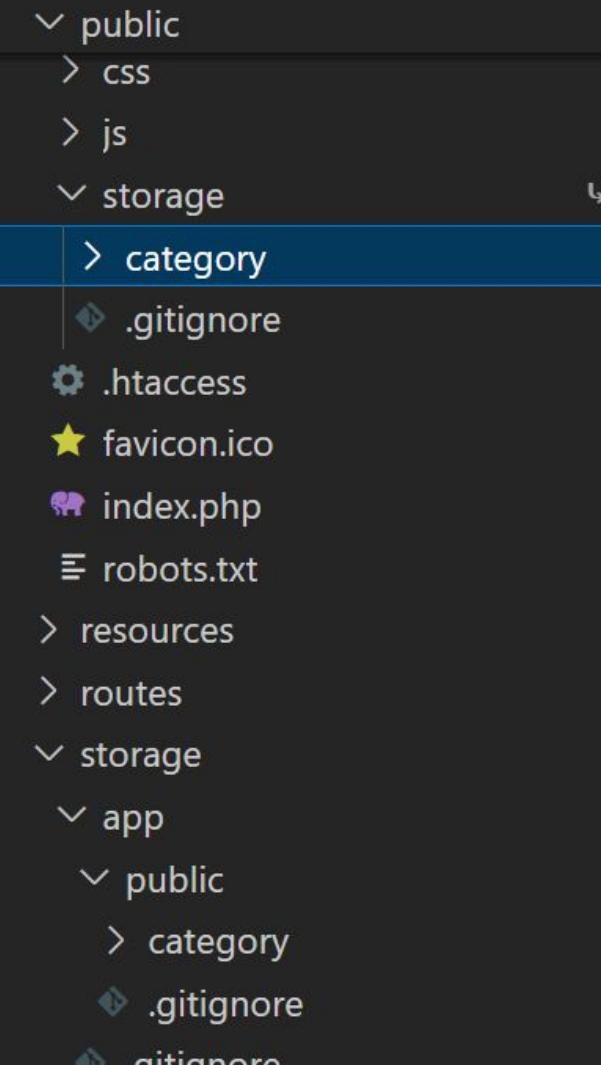


name	description	address	latitude	longitude	type_id	image
Taman Bungkul	gSUz5ELRiMpd	YtSyMslys	76.128	-84.584	1	images/places/tamanbungkul.jpg
Taman Pelangi	mlly0TOAqPqtO	2JdPFt9wl	51.649	-59.963	1	images/places/tamanpelangi.jpg

```
PS C:\laragon\www\example-app> php artisan storage:link
```

INFO The [C:\laragon\www\example-app\public\storage] link has been connected to [C:\laragon\www\example-app\storage\app\public].

```
PS C:\laragon\www\example-app> █
```



Symbolic Link (Symlink) is shortcut symbol that shared a references between public folder and storage folder.

If you make a directory inside public/storage, your new folder will be show up in your storage/app/public, and vice versa

Pick 2 category first and download 1 image of it

Google appetizer

All Images Shopping News Videos Books Maps : More Tools Saved

Finger food Easy Healthy Menu Fine dining Sandwich Hors d oeuvres Dinner Smoked salmon Christmas

Size Color Creative Commons licenses Type Any time Advanced Search Clear

DONGHUA DONGHUA

Delicious Assorted... vegetables | PickPik Plate of Appetizers... File:Seafood Appetizer (68172445).jp... PICRYL cheese. Appetizer ... Power Thesaurus 424 APPETIZER Synonyms PickPik Royalty-Free photo...

Google main course

All Images Shopping Videos Books Maps News : More Tools Saved

Menu Chicken Dinner Fine dining Vegetarian Food Fish Seafood Restaurant Healthy American

Size Color Creative Commons licenses Type Any time Advanced Search Clear

Licensable Needpix.com main course,gourmet,food,appet... Wikimedia Commons File:The Main Course.JPG - Wiki... PxHere restaurant, dish, meal, food, ... Flickr Lobster & Fish Main Course, Fri... Collections - GetArchive Main course menu vegetables ... itoldya test1 - GetArchive Kale eat main course ...

EXPLORER

EXAMPLE-APP

- public
 - css
 - js
- storage
 - category
 - appetizer.jpg
 - maincourse.jpg
 - .gitignore
 - .htaccess
 - favicon.ico
 - index.blade.php

... adminlte4.blade.php index.blade.php ...\\category appetizer.jpg maincourse.jpg

public > storage > category > appetizer.jpg

A photograph of a platter filled with various appetizers. The platter includes several small bowls containing dips like tzatziki and feta cheese, a variety of sliced fruits including apples and berries, and a selection of vegetables like cucumbers and carrots. There are also some meat slices and crackers.

Migrate first

```
PS C:\laragon\www\example-app> php artisan migrate
```

```
INFO Running migrations.
```

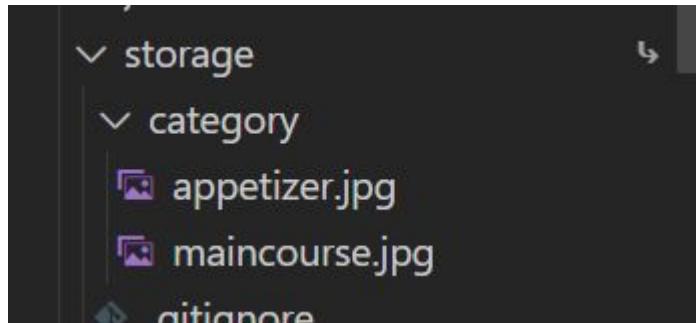
```
2025_04_09_172212_update_category_table .....
```

```
PS C:\laragon\www\example-app>
```

The screenshot shows the MySQL Workbench interface. On the left, there's a tree view of databases: 'myresto' is expanded, showing tables like 'categories', 'failed_jobs', 'foods', 'migrations', 'password_reset_tokens', 'personal_access_tokens', and 'users'. The 'categories' table is selected. On the right, the 'categories' table is shown in a detailed view with the following columns:

Columns:			
#	Name	Datatype	Length/Set
1	id	BIGINT	20
2	created_at	TIMESTAMP	
3	updated_at	TIMESTAMP	
4	name	VARCHAR	50
5	image	VARCHAR	1000

Insert your data based on your uploaded file



Host: 127.0.0.1 Database: myresto Table: categories Data

myresto.categories: 7 rows total (approximately)

		id	created_at	updated_at	name	image
	1	1	(NULL)	(NULL)	Appetizer	appetizer.jpg
	2	2	(NULL)	(NULL)	Main Course	maincourse.jpg
	3	3	(NULL)	(NULL)	Snack	
	4	4	(NULL)	(NULL)	Dessert	
	5	5	(NULL)	(NULL)	Coffee	
	6	6	(NULL)	(NULL)	Non Coffee	
	7	7	(NULL)	(NULL)	Healthy Juice	

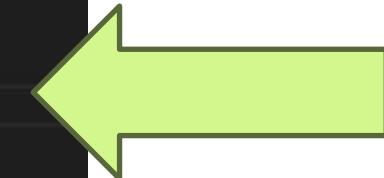
Using Static Modal

Create a dynamic modal form to access internal source (inside <td> data with a different ID). The modal content filled with images of the hotel and other information. Put the toggler button on each hotel record when looping the data.

First, add new column and give an action to trigger modal “imageModal”

```
17    <th>Show Image</th>
18    <th>Name</th>
19    <th>Number of Food </th>
20    <th>List of Food Name </th>
21  </tr>
22  <thead>
23  body>
24  @foreach($data as $d)
25  <tr>
26    <td>{{$d->id}}</td>
27    <td>
28      <button type="button" class="btn btn-primary" data-bs-toggle="modal"
29        data-bs-target="#imageModal-{{ $d->id }}">
30        Show
31      </button>
```

```
<button type="button" class="btn btn-primary" data-bs-toggle="modal"
       data-bs-target="#imageModal-{{ $d->id }}">
  Show
</button>
```



[Launch demo modal](#)

Hover Rows

The `.table-hover` class enables a hover state on table rows:

#	Show Image	Name	Number of Food	List of Food Name
1	Show	Appetizer	0	
2	Show	Main Course	2	<ul style="list-style-type: none">• Nasi Merah dengan Ayam F• Nasi Hitam dan Tumis Ca K
3	Show	Snack	0	
4	Show	Dessert	0	
5	Show	Coffee	0	
6	Show	Non Coffee	0	
7	Show	Healthy Juice	0	

Second, implement Modal HTML



```
@push ('modals')
    <!-- Modal {{ $d->id }} -->
    <div class="modal fade" id="imageModal-{{ $d->id }}" tabindex="-1" aria-labelledby="imageModalLabel"
        aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h1 class="modal-title fs-5" id="imageModalLabel">Gambar untuk Kategori {{$d->id}} </h1>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    {{ $d->id }} - {{ $d->name }}
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
                    <button type="button" class="btn btn-primary">Save changes</button>
                </div>
            </div>
        </div>
    </div>
@endpush
```

Notes

1. Use @push blade syntax
2. Put your @push and html modal near your button tag html
3. Modify your code based on your case study

```
26      <td>{{ $d->id }}</td>
27      <td>
28          <button type="button" class="btn btn-primary" data-bs-toggle="modal"
29              data-bs-target="#imageModal-{{ $d->id }}"
30          Show
31          </button>
32          @push ('modals')
33              <!-- Modal {{ $d->id }} -->
34              <div class="modal fade" id="imageModal-{{ $d->id }}" tabindex="-1"
35 >                  | aria-labelledby="imageModalLabel" ...
36          </div>
37          @endpush
38      </td>
39      <td>{{ $d->name }}</td>
40      <td>{{ count( $d->foods ) }}</td>
41      <td>
```

Don't forget to insert @stack('modals') in Your Layout

```
729      <!--begin::App Main-->
730      <main class="app-main">
731          | @yield('content')
732          | @stack('modals')
733      </main>
734      <!--end::App Main-->
```

Hover Rows

The .table-hover class enables a hover

#	Show Image	Name	
1	<button>Show</button>	Appetizer	0
2	<button>Show</button>	Main Course	2
3	<button>Show</button>	Snack	0
4	<button>Show</button>	Dessert	0

Gambar untuk Kategori 1



1 - Appetizer

CloseSave changes

localhost:8000/listkategori

.. Jual (FREE BUBBLE WR... New Tab S3-Admission Tools for Research

≡ Home Contact

Hover Rows

The .table-hover class enables a hover

#	Show Image	Name	
1	<button>Show</button>	Appetizer	0
2	<button>Show</button>	Main Course	2
3	<button>Show</button>	Snack	0
4	<button>Show</button>	Dessert	0

Gambar untuk Kategori 2



2 - Main Course

CloseSave changes

- Nasi Merah dengan Ayam Panggang Kecap & Tumis
- Nasi Hitam dan Tumis Ca Kailan

Modify your modals html with

Key point is “how to call image from storage?”. We use “asset” function and directly into public/storage

```
public
  - adminlte4
  - css
  - js
storage
  - category
    - appetizer.jpg
    - maincourse.jpg
  .gitignore
  .htaccess
  favicon.ico
  index.php
  robots.txt
resources
```

```
13      <table class="table table-hover">
23        <tbody>
25          <tr>
27            <td>
28              <div class="modal-dialog">
29                <div class="modal-content">
30                  <div class="modal-header">
31                    <h1 class="modal-title fs-5" id="imageModalLabel">Gambar untuk Kateg...
32                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria...
33                  </div>
34                  <div class="modal-body">
35                    
36                  </div>
37                  <div class="modal-footer">
38                    <button type="button" class="btn btn-secondary" data-bs-dismiss="mo...
39                    <button type="button" class="btn btn-primary">Save changes</button>
40                  </div>
41                </div>
42              </div>
43            </td>
44          </tr>
45        </tbody>
46      </table>
47
48
```

Result

Notes:

1. Don't forget to remove "save button"
2. Make your image adaptive inside modal"

You can try this trick:

```

```

≡ Home Contact

Hover Rows

The .table-hover class enables a hover effect on table rows.

#	Show Image	Name	Count
1	Show	Appetizer	0
2	Show	Main Course	0
3	Show	Snack	0
4	Show	Dessert	0

Gambar untuk Kategori 2

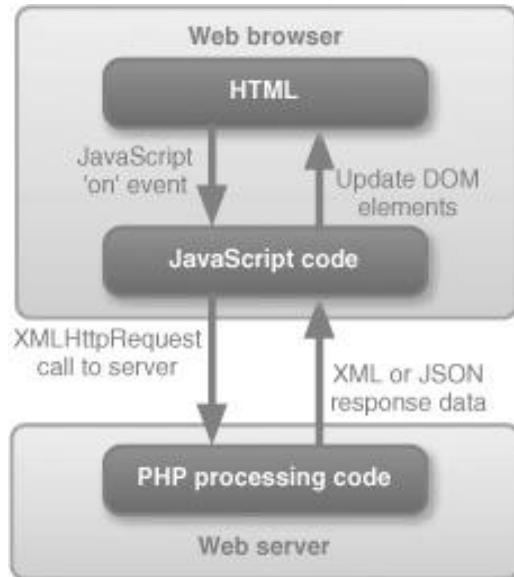
Close

Mathematics Engineering | Universitas Surabaya

AJAX Concept

AJAX

- **AJAX (Asynchronous JavaScript and XML)** is a set of web development techniques utilizing many web technologies used on the client-side to create asynchronous Web applications.
- Import jquery library in your view file to use ajax functions of jquery which will be used to send and receive data using ajax from the server.
- On the server side you can use the `response()` function to send response to client and to send response in JSON format you can chain the `response()` function with `json()` function



AJAX in View & Controller

1. Script JQuery AJAX is **implemented** in a view.
2. In a certain event handler , JQuery Ajax will request to spesific URL and will be accepted by Controller which next it will returning a json
3. The callback of JQuery ajax result will get the json, may be process it, and show it in a defined user interface

```
$.ajax({  
    url: '/api/posts'  
    type: 'POST',  
    data: {},  
    success: function () {},  
    error: function () {}  
});
```



Exercise #3

Using AJAX

Exercise #3 : Show a simple message with AJAX

Category with Hover Rows

The `.table-hover` class enables a hover state on table rows

#	Show Image	Name	Number of Food
1	<button>Show</button>	Appetizer	0
2	<button>Show</button>	Main Course	2
3	<button>Show</button>	Snack	0

Welcome

Home > Welcome

Did you know?
This message is sent by a controller

DOLAN BAKERY
[Disclaimer](#)

Step

1. Add the link and its event which calling jquery AJAX
2. Code the above Javascript function, Define command HTTP, suggested a POST method (more secure)
3. Define route and the used Controller
4. Create the Controller

1. EXAMPLE: Add the bulb icon and its event which calling jquery AJAX function in your view.

```
<h2>Category with Hover Rows</h2>
<p>The <a href="#"  
onclick="showinfo()".table-hover</a>  
class enables a hover state on table  
rows</p>
```

The [.table-hover](#) class enables a hover state on table rows

#	Show Image	Name	Number of Food
1	Show	Appetizer	0
2	Show	Main Course	2
3	Show	Snack	0

2. Define Route and Modify Controller

```
57     Route::post("/category/showInfo",[CategoryController::class, 'showInfo'])->name("category.showInfo");  
  
Route::post("/category/showInfo",[CategoryController::class, 'showInfo'])->name("category.showInfo");
```

After this, we are going to use “category.showInfo” routes as the target url of our AJAX.

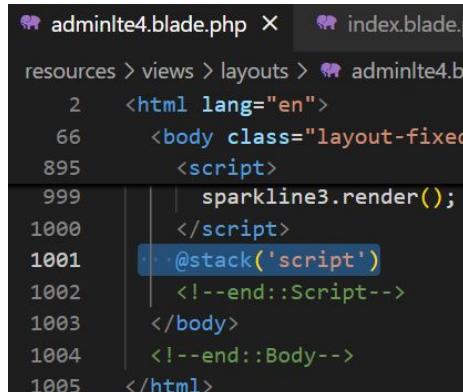
We will create the “showInfo” function in the CategoryController later.

3. Code the javascript function

Code the function in the **category.index** blade, put the codes below in a section. Since we are using section with different name, so we need to prepare a **yield** inside the head element of the layout.

```
@push("script")
<script>
    function showInfo() {
        $.ajax({
            type: 'POST',
            url: '{{ route("category.showInfo") }}',
            data: '_token=<?php echo csrf_token(); ?>',
            success: function(data) {
                $('#showinfo').html(data.msg);
            }
        });
    }
</script>
@endpush
```

Add this in the parent layout at the end of Script section



```
resources > views > layouts > adminlte4.blade.php
2   <html lang="en">
66     <body class="layout-fixed">
895       <script>
999         |   sparkline3.render();
1000        |   </script>
1001        |   @stack('script')
1002        |   <!--end::Script-->
1003     </body>
1004     <!--end::Body-->
1005   </html>
```

3. Code the javascript function : Explanation

```
@push("script")
<script>
    function showInfo() {
        $.ajax({
            type: 'POST',
            url: '{{ route("category.showInfo") }}',
            data: '_token=<?php echo csrf_token(); ?>',
            success: function(data) {
                $('#showinfo').html(data.msg);
            }
        });
    }
</script>
@endpush
```

JQuery ajax syntax is using **\$.ajax** followed by these parameters:

type □ command HTTP,
url □ routing destination,
data □ parameter/data that parsed to the route
success □ callback/action function that will execute when success

As default, Data transferred in the callback function is in a form of JSON

'data.msg' is a json object which will be generated by the Controller
It is a mandatory for using **_token** when using POST method

3. Code the javascript function : Implementation

```
88 @push("script")
89 <script>
90     function showInfo() {
91         $.ajax({
92             type: 'POST',
93             url: '{{ route("category.showInfo") }}',
94             data: '_token=<?php echo csrf_token(); ?>',
95             success: function(data) {
96                 $('#showinfo').html(data.msg);
97             }
98         });
99     }
100 </script>
101 @endpush
```

```
5 <div class="container">
6     <h2>Category with Hover Rows</h2>
7     <p>The <a href="#" onclick="showinfo()">tab</a>
8     <div id="showinfo"></div>
9
10    <table class="table table-hover">
11        <thead>
12            <tr>
13                <th>#</th>
14                <th>Show Image</th>
```

Don't forget to **implement** this section

Exercise #4

Implement AJAX

CategoryController.php X

adminlte4.blade.php

index.blade.php ...\\category

dem

app > Http > Controllers > CategoryController.php > CategoryController > showInfo

```
8 class CategoryController extends Controller
61     /*
62     public function destroy(Category $category)
63     {
64         //
65     }
66
67     public function showInfo()
68     {
69         return response()->json(array(
70             'status' => 'oke',
71             'msg' => "<div class='alert alert-info'>
72             Tutorial membuat bootstrap stripe ada di
73             https://getbootstrap.com/docs/4.0/content/tables/
74         ), 200);
75
76     }
```

Don't forget to add JQUERY JS before @stack('script')

Please insert this code inside your Parent View (Layout)

```
751    <!--begin::Script-->
752    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
753    <!--begin::Third Party Plugin(OverlayScrollbars)-->
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
```

4. Implement showInfo function inside Controller

```
return response()->json(array(  
    'status' => 'oke',  
    'msg' => "<div class='alert alert-info'>  
Tutorial membuat bootstrap stripe ada di  
https://getbootstrap.com/docs/4.0/content/ta  
bles/.</div>"  
, 200);
```

The screenshot shows a web browser window with the URL `localhost:8000/listkategori`. The page title is "Category with Hover Rows". Below the title, a sub-instruction reads: "The .table-hover class enables a hover state on table rows". A red arrow points from this text to the word ".table-hover". The main content is a table with the following data:

#	Show Image	Name	Number of Food
1	Show	Appetizer	0

Place it in the CategoryController.
Use method **response()->json()** to create json data.
Data in array with some pairs of 'key' and

Issue error □ Check your Javascript FIRST (Missing function name)

1. If you have an unresponsive action, please open: Right Click > Inspect Element > Console
2. If you have an error like this, please check your script code is already loaded or not. Use Ctrl+U instead



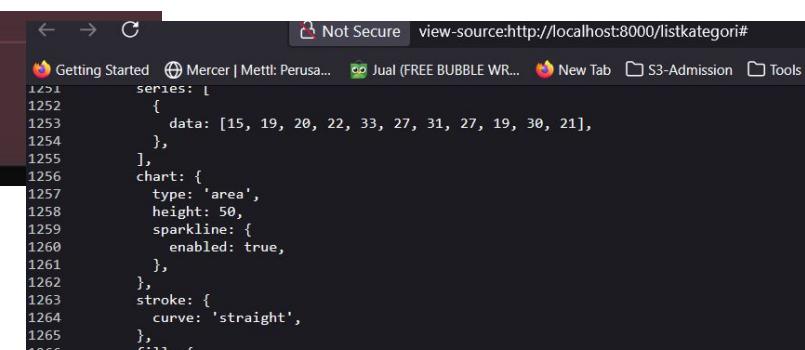
The screenshot shows a browser developer tools console with the following error message:

```
Uncaught TypeError: (intermediate value).showinfo is not a function
  onclick http://localhost:8000/listkategori:1
  [Learn More]
```

Below the error message, the page source code is displayed:

```
5 <div class="container">
6   <h2>Category with Hover Rows</h2>
7   <p>The <a href="#" onclick="showinfo()">.t
```

The line of code with the error is highlighted: `.`



The screenshot shows a browser developer tools script tab with the following code:

```
1251 series: [
1252   {
1253     data: [15, 19, 20, 22, 33, 27, 31, 27, 19, 30, 21],
1254   },
1255 ],
1256 chart: {
1257   type: 'area',
1258   height: 50,
1259   sparkline: {
1260     enabled: true,
1261   },
1262   stroke: {
1263     curve: 'straight',
1264   },
1265   fill: {
1266     opacity: 0.3,
1267   },
1268   yaxis: {
1269     min: 0,
1270   },
1271   colors: ['#DCE6EC'],
1272 };
1273
1274 const sparkline3 = new ApexCharts(document.querySelector('#sparkline-3'), option_sparkline3);
1275 sparkline3.render();
1276 </script>
1277 <script>
1278 function showInfo() {
1279   $.ajax({
1280     type: 'POST',
1281     url: 'http://localhost:8000/category/showInfo',
1282     data: '_token=3FnKpkfIeCgFPXhaVt0eYbHGCUulbhBQ0Bj',
1283     success: function(data) {
1284       $('#showinfo').html(data.msg);
1285     }
1286   });
1287 }
1288 </script>
1289 <!--end:Script-->
1290 </body>
1291 </--end:Body-->
1292 <!--end:Body-->
1293 </html>
```

In this case, our code is wrong, because
function `showInfo` uses 's' character lowercase and 'I' uppercase
But, programmer call with "showinfo" in onclick action

Issue error □ Check your Javascript FIRST (Missing JQuery to understand \$ syntax)

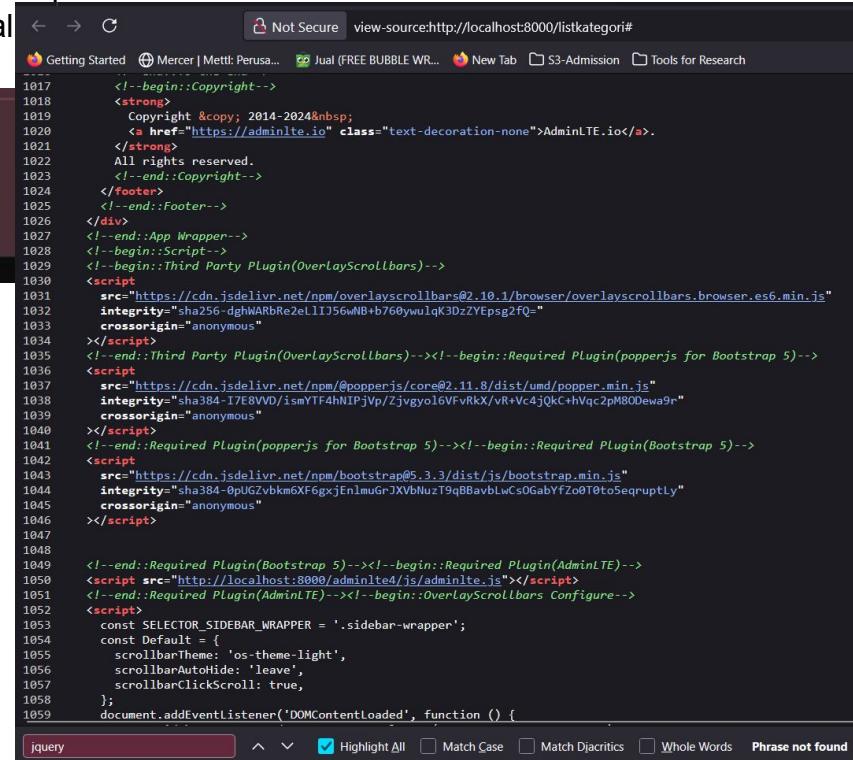
1. If you have an unresponsive action, please open: Right Click > Inspect Element > Console
2. If you have an error like this, please check your script code is all

```
! ▶ Uncaught ReferenceError: $ is not defined
    showInfo http://localhost:8000/listkategori:1280
    onclick http://localhost:8000/listkategori:1
    [Learn More]
```

```
5 <div class="container">
6   <h2>Category with Hover Rows</h2>
7   <p>The <a href="#" onclick="showinfo()">t
```

In this case, we have to check

1. JQuery JS is already loaded before we call ajax syntax
2. Use Find and type “Jquery”.
3. If there is no Jquery JS, please insert first inside your Layout Blade

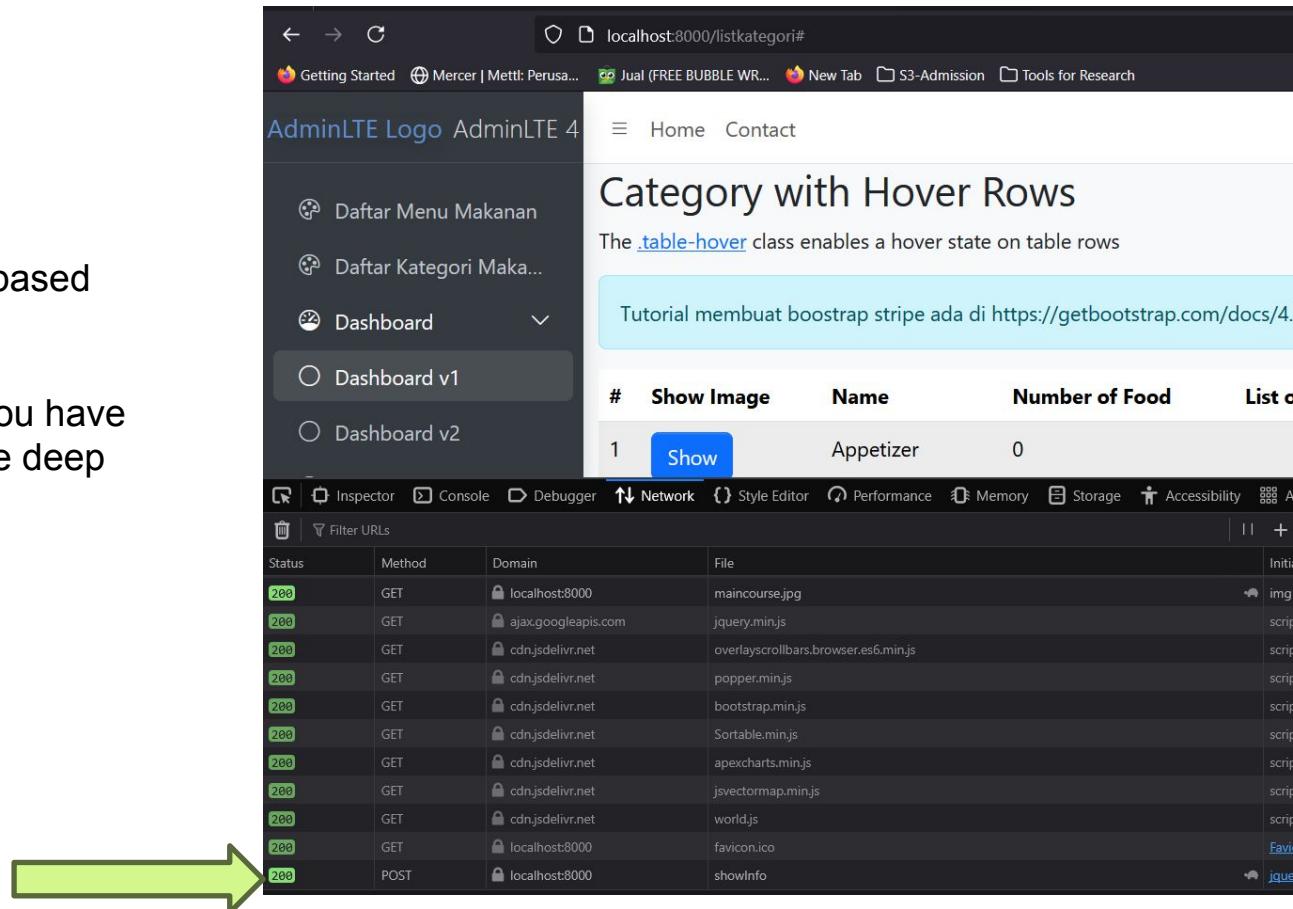


```
1017 <!--begin::Copyright-->
1018 <strong>
1019 Copyright © 2014-2024
1020 <a href="https://adminlte.io" class="text-decoration-none">AdminLTE.io</a>.
1021 </strong>
1022 All rights reserved.
1023 <!--end::Copyright-->
1024 </footer>
1025 <!--end::Footer-->
1026 </div>
1027 <!--end::App Wrapper-->
1028 <!--begin::Script-->
1029 <!--begin::Third Party Plugin(OverlayScrollbars)-->
1030 <script
src="https://cdn.jsdelivr.net/npm/overlayscrollbars@2.10.1/browser/overlayscrollbars.browser.es6.min.js"
integrity="sha256-dghWARbRe2eL1J56wNB+b760ywhulqk3DzZYEpsg2FQ="
crossorigin="anonymous">
1031 </script>
1032 <!--end::Third Party Plugin(OverlayScrollbars)--><!--begin::Required Plugin(popperjs for Bootstrap 5)-->
1033 <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper_min.js"
integrity="sha256-17E8VVD/ismYTF4HN1PjVp/7jvygoIz6VFvRX/Xr+Vc4jQkC+vVqCcPM8ODewa9r"
crossorigin="anonymous">
1034 </script>
1035 <!--end::Required Plugin(popperjs for Bootstrap 5)--><!--begin::Required Plugin(Bootstrap 5)-->
1036 <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
integrity="sha256-0pUGzbkmXF6gxjEnlmuGrJXbhluzT9qBavbluCsoGabYFzoT0to5eqrptly"
crossorigin="anonymous">
1037 </script>
1038 <!--end::Required Plugin(Bootstrap 5)--><!--begin::Required Plugin(AdminLTE)-->
1039 <script src="http://localhost:8000/adminlte4/js/adminlite.js"></script>
1040 <!--end::Required Plugin(AdminLTE)--><!--begin::OverlayScrollbars Configure-->
1041 <script>
1042 src="https://cdn.jsdelivr.net/npm/overlayscrollbars@2.10.1/browser/overlayscrollbars.browser.es6.min.js"
integrity="sha256-dghWARbRe2eL1J56wNB+b760ywhulqk3DzZYEpsg2FQ="
crossorigin="anonymous">
1043 </script>
1044 <!--end::OverlayScrollbars Configure-->
1045 <!--end::Required Plugin(Bootstrap 5)-->
```

```
751 <!--begin::Script-->
752 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
753 <!--begin::Third Party Plugin(OverlayScrollbars)-->
```

It's visible to check AJAX Process

1. Inspect Element > Network
2. Refresh Again until you see all processes
3. Click link with Ajax
4. There is one request network based on your code.
5. If status 200, your request is completely successful. But if you have error, you need to inspect more deep



The screenshot shows a browser window with the URL `localhost:8000/listkategori#`. The page content displays a sidebar with 'AdminLTE Logo' and 'AdminLTE 4' branding, followed by a list of links: 'Daftar Menu Makanan', 'Daftar Kategori Maka...', 'Dashboard' (with sub-options 'Dashboard v1' and 'Dashboard v2'), and a dropdown menu. Below the sidebar is a table titled 'Category with Hover Rows' with the following data:

#	Show Image	Name	Number of Food
1	Show	Appetizer	0

At the bottom of the page is a footer with links: 'Tutorial membuat bootstrap stripe ada di <https://getbootstrap.com/docs/4.0/components/card/#table-hovers>' and 'List c'. The browser's developer tools Network tab is open, showing a list of requests. A large green arrow points from the list number 5 in the left column to the 'Status' column in the Network tab table, highlighting the status codes. The table shows the following requests:

Status	Method	Domain	File
200	GET	localhost:8000	maincourse.jpg
200	GET	ajax.googleapis.com	jquery.min.js
200	GET	cdn.jsdelivr.net	overlayscrollbars.browser.es6.min.js
200	GET	cdn.jsdelivr.net	popper.min.js
200	GET	cdn.jsdelivr.net	bootstrap.min.js
200	GET	cdn.jsdelivr.net	Sortable.min.js
200	GET	cdn.jsdelivr.net	apexcharts.min.js
200	GET	cdn.jsdelivr.net	jsvectormap.min.js
200	GET	cdn.jsdelivr.net	world.js
200	GET	localhost:8000	favicon.ico
200	POST	localhost:8000	showInfo

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

10 | ⌂ ... ×

Filter URLs

Status	Met...	Domain	File	Initiator	Type	Transferred	Size	Headers	Cookies	Request	Response	Timings	Stack Trace
200	GET	localhost...	adminlte.js	script	js	29.61 kB	29.4...						
200	GET	localhost...	user1-128x128.jpg	script	img	2.90 kB	2.75 ...						
200	GET	localhost...	appetizer.jpg	script	img	137.78 kB	137...						
200	GET	localhost...	maincourse.jpg	script	img	11.81 kB	11.6...						
200	GET	ajax.googl...	jquery.min.js	script	js	cached	0 B						
200	GET	cdn.jsdelivr...	overlayscrollbars.browser.es6.min.js	script	js	cached	0 B						
200	GET	cdn.jsdelivr...	popper.min.js	script	js	cached	0 B						
200	GET	cdn.jsdelivr...	bootstrap.min.js	script	js	cached	0 B						
200	GET	cdn.jsdelivr...	Sortable.min.js	script	js	cached	0 B						
200	GET	cdn.jsdelivr...	apexcharts.min.js	script	js	cached	0 B						
200	GET	cdn.jsdelivr...	jsvectormap.min.js	script	js	cached	0 B						
200	GET	cdn.jsdelivr...	world.js	script	js	cached	0 B						
200	GET	localhost...	favicon.ico	FaviconLoade...	vnd....	cached	0 B						
200	POST	localhost...	showInfo	script	jquery.min.js:...	json	1.30 kB	189 B					

27 requests | 672.02 kB / 622.94 kB transferred | Finish: 9.49 s | DOMContentLoaded: 6.27 s | load: 6.33 s

Filter Headers | Block | Resend

▶ POST http://localhost:8000/category/showInfo

Status: 200 OK (?)
 Version: HTTP/1.1
 Transferred: 1.30 kB (189 B size)
 Referrer Policy: strict-origin-when-cross-origin
 Request Priority: Highest
 DNS Resolution: System

▼ Response Headers (1.106 kB)

Raw

- Cache-Control: no-cache, private
- Connection: close
- Content-Type: application/json
- Date: Thu, 10 Apr 2025 05:55:27 GMT
- Date: Thu, 10 Apr 2025 05:55:27 GMT
- Host: localhost:8000
- Set-Cookie: XSRF-TOKEN=eyJpdil6ImRoTmZmUjBHUmxAIaZXMS9yMTI0QWc9PSIsInZhHVljoQiJTMEZOMGhvYUINz2tBZ2ZDQUM4sVWMW401ZOTdLSDNkK2N4QH3K004-JVIEQUNTTVp4b0;o--IVaWQwTz

XML CSS JS XHR Fonts Images Media WS Other | Disable Cache | NO Throttling |

Headers Cookies Request Response Timings Stack Trace

Filter Request Parameters

Form data Raw

```
_token: "3FnKpkFfleCgFFPXhaVt0eEYbHGCXuIbhaBQQBj"
```

Headers Cookies Request Response Timings Stack Trace

Filter properties

JSON Raw

```
status: "oke"

msg: "<div class='alert alert-info'>\n Tutorial membuat bootstrap stripe ada di \n https://getbootstrap.com/docs/4.0/content/tables/'.</div>"
```

If you have 500 ERROR, so programmer must fix it.

Check deeper into “Request” and “Response”

In this case, programmer must check “Is there any wrong or typo between function name in Controller and function name inside Router?”

200	GET	localhost:8000	favicon.ico	FaviconLoader.sys.mjs:175
500	POST	localhost:8000	showInfo	jquery.min.js:2 (xhr)

Headers Cookies Request Response Timings Stack Tr

Filter Request Parameters

Form data

_token: "3FnKpkFfleCgFFPXdhaVt0eEYbHGXuulbhaBQQBj"

Headers Cookies Request Response Timings Stack Trace

Filter properties

JSON Raw

```
message: "Method App\\Http\\Controllers\\CategoryController::showInfo does not exist."
exception: "BadMethodCallException"
file: "C:\\laragon\\www\\example-app\\vendor\\laravel\\framework\\src\\Illuminate\\Routing\\Controller.php"
line: 68
▶ trace: (47)[{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, ...]
```

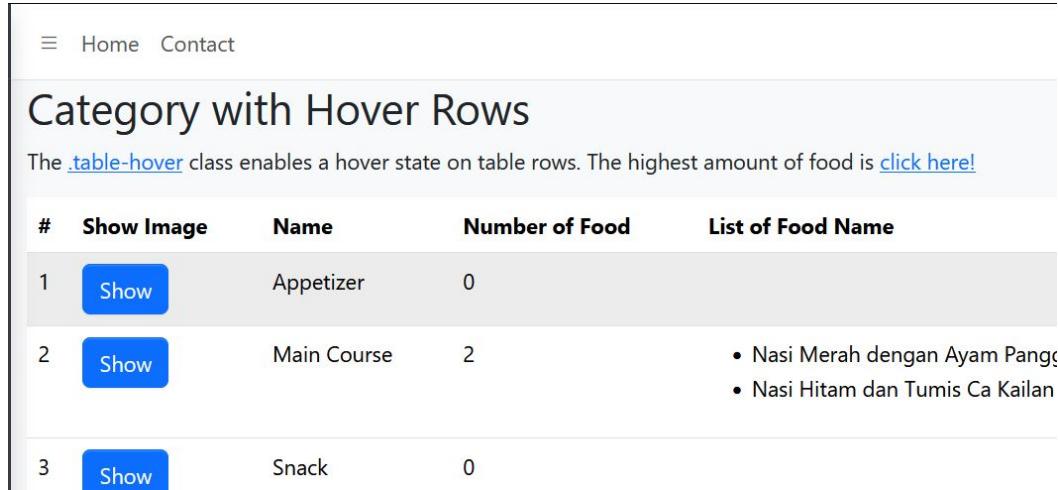
Category with Hover Rows

The `.table-hover` class enables a hover state on table rows

Tutorial membuat bootstrap stripe ada di <https://getbootstrap.com/docs/4.0/content/tables/> .

#	Show Image	Name	Number of Food	List of Food Name
---	------------	------	----------------	-------------------

Exercise #4 : Add new action by reuse/modify exercise #3 to show the Category name where has the highest number of food variant



The screenshot shows a web page with a header containing a menu icon, "Home", and "Contact". Below the header, the title "Category with Hover Rows" is displayed. A descriptive text states: "The `.table-hover` class enables a hover state on table rows. The highest amount of food is [click here!](#)". The main content is a table with the following data:

#	Show Image	Name	Number of Food	List of Food Name
1	<button>Show</button>	Appetizer	0	
2	<button>Show</button>	Main Course	2	<ul style="list-style-type: none">Nasi Merah dengan Ayam PanggangNasi Hitam dan Tumis Ca Kailan
3	<button>Show</button>	Snack	0	

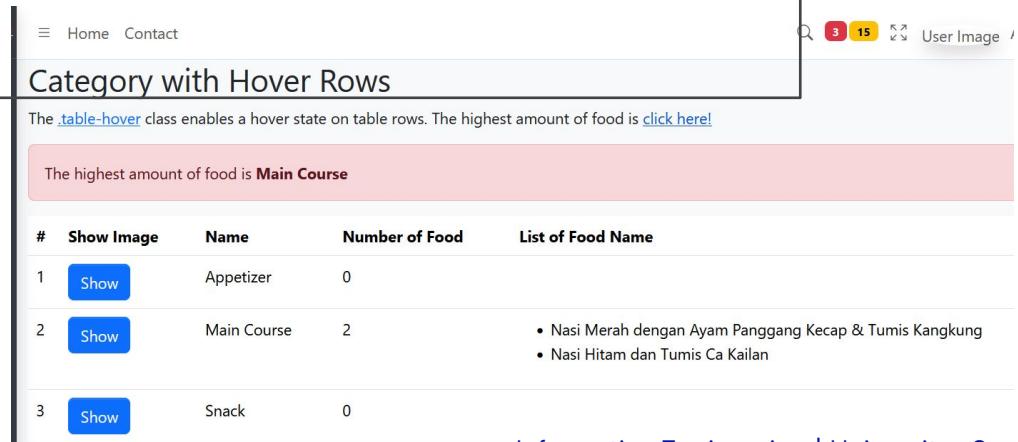
(in the future, when we already have a transaction, it can be changed with more useful information like the most selling product, etc)

```
public function showHighestFood()
{
    //https://laravel.com/docs/10.x/eloquent-relationships#aggregating-related-models
    $highestFoodCategory = Category::withCount('foods') // Eager load the count of related 'foods'
        ->orderByDesc('foods_count') // Order the categories by the count of foods in descending order
        ->first(); // Get the first category, which will have the highest count

    return response()->json(array(
        'status' => 'oke',
        'msg' => "<div class='alert alert-danger'>
            The highest amount of food is .'" . $highestFoodCategory->name . ".</div>"
    ), 200);
}
```

#4. Example for add query in new link

```
public function showInfo() {  
    $highestFoodCategory = Category::withCount('foods')  
        ->orderByDesc('foods_count')  
        ->first();  
  
    return response()->json(array(  
        'status' => 'oke',  
        'msg' => "<div class='alert alert-danger'>  
            The highest amount of food is <b>".$highestFoodCategory->name."</b></div>"  
    ), 200);  
}
```



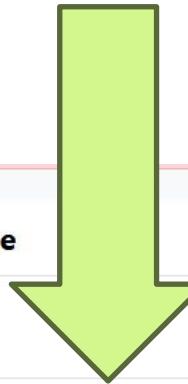
The screenshot shows a web application interface. At the top, there is a navigation bar with links for Home and Contact, and icons for search, notifications (3), messages (15), and user profile. Below the navigation bar is a header titled "Category with Hover Rows". A note below the header states: "The `.table-hover` class enables a hover state on table rows. The highest amount of food is [click here!](#)". The main content area displays a table with the following data:

#	Show Image	Name	Number of Food	List of Food Name
1	<button>Show</button>	Appetizer	0	
2	<button>Show</button>	Main Course	2	<ul style="list-style-type: none">Nasi Merah dengan Ayam Panggang Kecap & Tumis KangkungNasi Hitam dan Tumis Ca Kailan
3	<button>Show</button>	Snack	0	

Exercise #5

Make a Modal for showing foods in master category (replace bullet point)

We need to change this with modals for elevate the efficient of User Interface



#	Show Image	Name	Number of Food	List of Food Name
1	Show	Appetizer	0	
2	Show	Main Course	2	<ul style="list-style-type: none">• Nasi Merah dengan Ayam Panggang Kecap & Tumis Kangkung• Nasi Hitam dan Tumis Ca Kailan
3	Show	Snack	0	
4	Show	Dessert	0	
5	Show	Coffee	0	
6	Show	New Coffee	0	

1. Prepare : Blank Modal Form

Still in category/index.blade.php , we will create the modal header, modal body, and modal footer from a View, therefore we will clear the modal-content (or put with an animated loading gif)

```
@push ('modals')
<!-- Modal -->
<div class="modal fade" id="detailModal" tabindex="-1" aria-labelledby="detailModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="DetailModalLabel">List of </h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        ...
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
@endpush
```

```
73  @push ('modals')
74  <!-- Modal -->
75  <div class="modal fade" id="detailModal" tabindex="-1" aria-labelledby="detailModalLabel" aria-hidden="true">
76    <div class="modal-dialog">
77      <div class="modal-content">
78        <div class="modal-header">
79          <h1 class="modal-title fs-5" id="DetailModalLabel">List of </h1>
80          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
81        </div>
82        <div class="modal-body">
83          ...
84        </div>
85        <div class="modal-footer">
86          <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
87        </div>
```

2. Add the Link and its event which calling jquery AJAX Function

```
<td>
<button type="button" class="btn btn-info" data-bs-toggle="modal"
        data-bs-target="#detailModal" onclick="showDetail('{{ $d->id }}')>
    Details
</button>
...
</td>
```

Category with Hover Rows

The [.table-hover](#) class enables a hover state on table rows. The highest amount of food is [click here!](#)

#	Show Image	Name	Number of Food	List of Food Name
1	Show	Appetizer	0	Details
2	Show	Main Course	2	Details
3	Show	Snack	0	Details
4	Show	Dessert	0	Details
5	Show	Coffee	0	Details
6	Show	Non Coffee	0	Details
7	Show	Healthy Juice	0	Details

resources > views > category > index.blade.php

```
5   <div class="container">
10    <table class="table table-hover">
20      <tbody>
22        <tr>
23          <td>{{ $a->name }}</td>
24          <td>{{ count( $d->foods ) }}</td>
25          <td>
26            <button type="button" class="btn btn-info" data-bs-toggle="modal"
27                  data-bs-target="#detailModal" onclick="showDetail('{{ $d->id }}")">
28              Details
29            </button>
30
31            {{-- <ul>
32              @foreach($d->foods as $df)
33              <li>{{ $df->name }}</li>
34              @endforeach
35              </ul> --}}
36          </td>
37      </tr>
```

List of

[Close](#)

List of Food

[Show](#)[Show](#)[Show](#)[Show](#)[Show](#)[Show](#)

Main Course

2

[Details](#)

Snack

0

[Details](#)

Dessert

0

[Details](#)

Coffee

0

[Details](#)

Non Coffee

0

[Details](#)

3. Define Route and Controller for AJAX Data

```
Route::post("/category/showListFoods",
    [CategoryController::class, 'showListFoods'])
->name("category.showListFoods");
```

- In the next step, we are going to use AJAX and redirect the data by using route 'category.showListFoods'.
- We need 1 parameter to know specific category, but we parse inside the AJAX. So we don't have to make required parameter inside Routing.
- Don't forget to create the function 'showListFoods' in CategoryController later.

```
55  
56  
57  
58  
59  
60 Route::post("/category/showListFoods",  
61 | | | [CategoryController::class, 'showListFoods'])  
62 | | ->name("category.showListFoods");  
63
```

Check with: **php artisan route:list**

```
GET|HEAD  api/user .....  
POST      category/showHighestFood ..... category.showHighestFood > CategoryController@showHighestFood  
POST      category/showInfo ..... category.showInfo > CategoryController@showInfo  
POST      category/showListFoods ..... category.showListFoods > CategoryController@showListFoods
```

4. Code the javascript function

Code the function in the **category.index** blade, put the codes below in a section.

```
function showDetail(id) {
    $.ajax({
        type: 'POST',
        url: '{{ route("category.showListFoods") }}',
        data: {
            '_token': '<?php echo csrf_token(); ?>',
            'idcat': id,
        },
        success: function(data) {
            $('#detail-title').html(data.title);
            $('#detail-body').html(data.body);
        }
    });
}
```

```
73     @push ('modals')
74     <!-- Modal -->
75     <div class="modal fade" id="detailModal" tabindex="-1" aria-labelledby="detailModalLabel" aria-
76         <div class="modal-dialog">
77             <div class="modal-content">
78                 <div class="modal-header">
79                     <h1 class="modal-title fs-5" id="DetailModalLabel" id="detail-title">List of </h1>
80                     <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></but
81                 </div>
82                 <div class="modal-body" id="detail-body">
83                     ...
84                 </div>
85                 <div class="modal-footer">
86                     <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
87                 </div>
88             </div>

```

```
118     function showDetail(id) {
119         $.ajax({
120             type: 'POST',
121             url: '{{ route("category.showListFoods") }}',
122             data: {
123                 '_token': '<?php echo csrf_token(); ?>',
124                 'idcat': id,
125             },
126             success: function(data) {
127                 $('#detail-title').html(data.title);
128                 $('#detail-body').html(data.body);
129             }
130         });
131     }
132 </script>
```

5. Implement showListFoods inside CategoryController

```
public function showListFoods()
{
    $category = Category::find($_POST['idcat']);
    $name = $category->name;
    $data = $category->foods;
    return response()->json(array(
        'status' => 'oke',
        'title' => $name.' Food List',
        'body' => view('category.showListFood', compact('name', 'data'))->render()
    ), 200);
}
```

Make sure you are already make relationship of category and foods

Render make json can accept html code

```
public function showListFoods()
{
    $category = Category::find($_POST['idcat']);
    $name = $category->name;
    $data = $category->foods;
    return response()->json(array(
        'status' => 'oke',
        'title' => $name.' Food List',
        'body' => view('category.showListFood', compact('name', 'data'))->render(),
    ), 200);
}
```

6. Create a new View

Create in view\category\showListFood.blade.php

```
<ul>
@foreach ($data as $f)
<li>{{$f->name}}</li>
@endforeach
</ul>
```

You can try to run at this point to check whether the modal is running well and get the hotel name or not.

CategoryController.php

showListFood.blade.php ×

resources > views > category > showListFood.blade.php

```
1  <ul>
2  @foreach ($data as $f)
3  |  <li>{{$f->name}}</li>
4  @endforeach
5  </ul>
```

Category with

The `.table-hover` class enab

#	Show Image
1	<button>Show</button>
2	<button>Show</button>
3	<button>Show</button>

Main Course Food List



- Nasi Merah dengan Ayam Panggang Kecap & Tumis Kangkung
- Nasi Hitam dan Tumis Ca Kailan

Close

List of Food Name

<button>Details</button>
<button>Details</button>
<button>Details</button>

Find any error ?

Debug your codes by accessing the console on your browser.
You can access it through “Inspect Element”.

Check the Debugging Example to get more detailed tips on debugging.

Thank You