



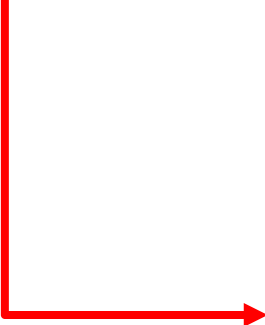
Create data

Now let's create a form for new members.

But before that, let's modify the members form by adding Add Member button

```
<a href="/members/create" class="btn btn-primary">Add new member +</a>
```

This is a sozolab members page



Add new member +

Name: Christina

Hobby: gardening

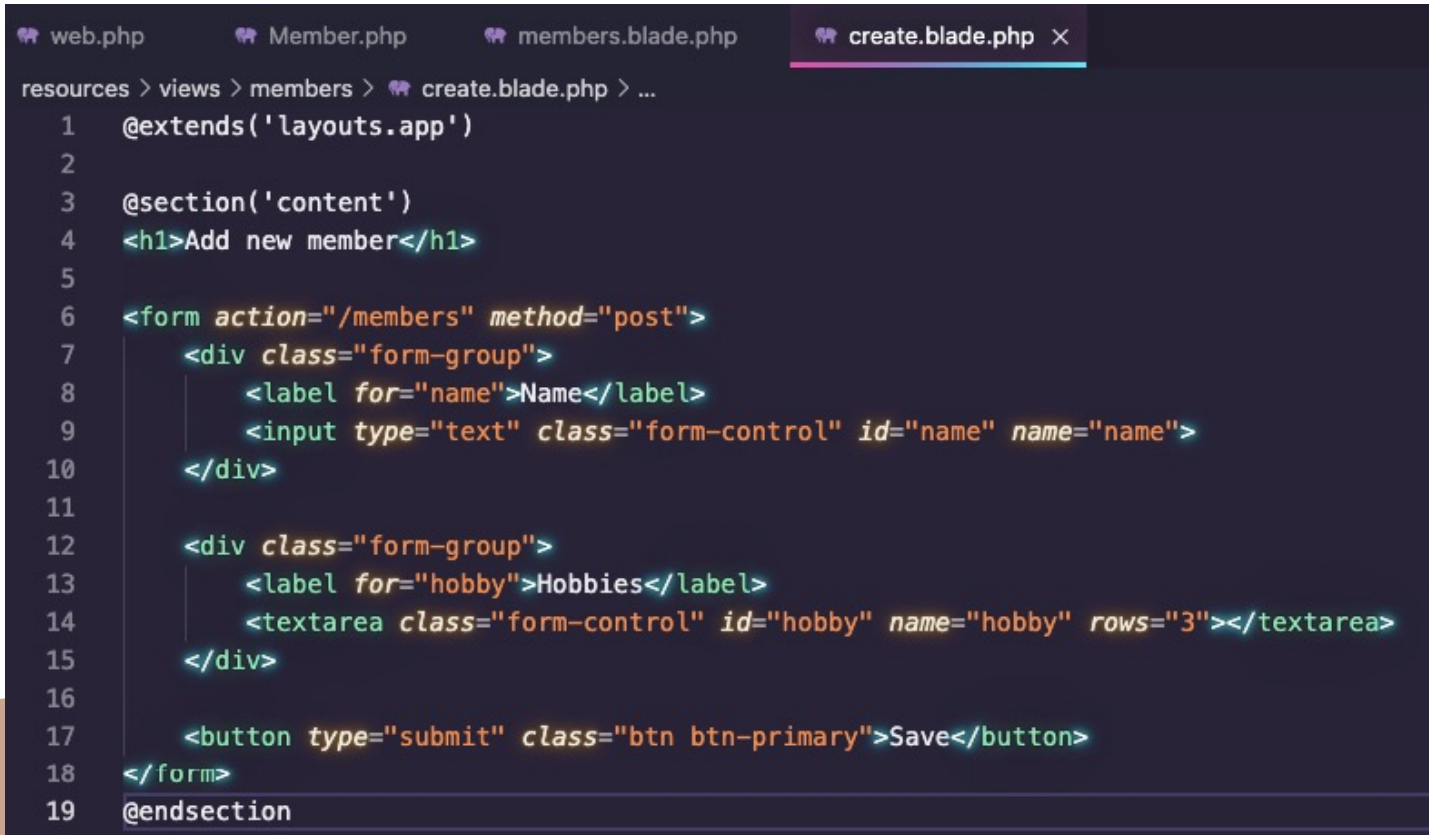
In order to Add new member + to work, we have to add the routing to a new form called create.

```
Route::get('/members', 'MemberController@index');  
Route::get('/members/create', 'MemberController@create');  
Route::get('/members/{slug}', 'MemberController@show');
```

After that we move on to MemberController.php to create the “create” function.

```
public function create()  
{  
    return view('members.create');  
}
```

Now we create a new form named create.blade.php in views/members

A screenshot of a code editor with a dark theme. The editor has four tabs at the top: 'web.php', 'Member.php', 'members.blade.php', and 'create.blade.php' (which is the active tab). The breadcrumb navigation shows 'resources > views > members > create.blade.php > ...'. The code is a Blade template for creating a new member. It starts with '@extends('layouts.app')', followed by '@section('content')'. Inside the section, there's an h1 tag 'Add new member'. Then, a form is created with 'action="/members"' and 'method="post"'. The form contains two 'form-group' divs. The first group has a 'Name' label and a text input with 'id="name"' and 'name="name"'. The second group has a 'Hobbies' label and a text area with 'id="hobby"', 'name="hobby"', and 'rows="3"'. A 'Save' button with 'type="submit"' and 'class="btn btn-primary"' is at the bottom of the form. The code ends with '@endsection'.

```
resources > views > members > create.blade.php > ...
1  @extends('layouts.app')
2
3  @section('content')
4  <h1>Add new member</h1>
5
6  <form action="/members" method="post">
7      <div class="form-group">
8          <label for="name">Name</label>
9          <input type="text" class="form-control" id="name" name="name">
10     </div>
11
12     <div class="form-group">
13         <label for="hobby">Hobbies</label>
14         <textarea class="form-control" id="hobby" name="hobby" rows="3"></textarea>
15     </div>
16
17     <button type="submit" class="btn btn-primary">Save</button>
18 </form>
19 @endsection
```


To receive the members page, we will return to web.php to add a route using the post method.

```
Route::get('/members', 'MemberController@index');  
Route::get('/members/create', 'MemberController@create');  
Route::get('/members/{slug}', 'MemberController@show');  
Route::post('/members', 'MemberController@store');
```

Then we will add a “store” function in MemberController.php

```
public function store(Request $request)  
{  
    $member = new Member;  
    $member->name = $request->name;  
    $member->hobby = $request->hobby;  
    $member->save();  
}
```

According to Laravel, we have to create the **created_at** and **updated_at** columns in the database as timestamps.

Field	Type	Len...	Unsign...	Zero...	Bin...	Allow N...	Key	Def...	Extra	Encod...	Collation	Co...
id	INT	↕	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI		auto... ↕	↕	↕	↕
name	VARCHAR	↕ 20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None ↕	UT... ↕	utf8... ↕	↕
hobby	VARCHAR	↕ 50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None ↕	UT... ↕	utf8... ↕	↕
created_at	TIMESTAMP	↕	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None ↕	↕	↕	↕
updated_at	TIMESTAMP	↕	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None ↕	↕	↕	↕

Add new member

Name

Haru

Hobbies

programming



Save



id	name	hobby	created_at	updated_at
1	Christina	gardening	NULL	NULL
2	Defry	badminton	NULL	NULL
3	Fikry	football	NULL	NULL
4	Nazmun	reading	NULL	NULL
5	Tan	swimming	NULL	NULL
6	Haru	programming	2021-06-05 04:27:12	2021-06-05 04:27:12

How about if we save name and hobby fields in
a **blank condition**?

To avoid this, we need to add form validation.

Before the user saves the data, we have to make sure that the empty fields are filled. For that we need to add coding in MemberController → store function.

```
public function store(Request $request)
{
    $request->validate([
        'title' => 'required|unique:posts|max:20',
        'body' => 'required',
    ]);

    $member = new Member;
    $member->name = $request->name;
    $member->hobby = $request->hobby;
    $member->save();

    return redirect('members');
}
```

Add new member

- The title field is required.
- The body field is required.

Name

Hobbies

Save

Thank you 😊

