

JOBSHEET 13
“Flutter Laravel API”

Mata Kuliah [Pemrograman Mobile]

Dosen Pengampu: Ade Ismail, S.Kom., M.TI.



Nama : Wardanadira Pinkan Dwiyuwanda
NIM : 2241760116
Kelas : SIB-3D
No Absen : 26

PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

How to Build an API with Laravel Breeze in Laravel 11

Link github : <https://github.com/WardanadiraPinkan/PEMROGRAMAN-MOBILE>

FETCHING THE PACKAGES

Langkah 1: Install Laravel

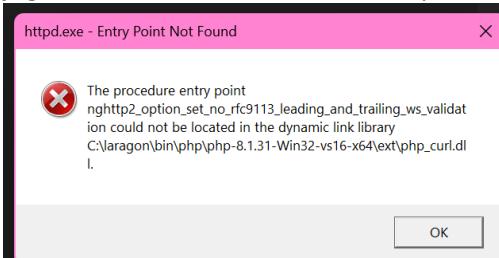
- Saya menginstall Laravel 10.3 dikarenakan saat saya amencoba menginstall Laravel 11 terjadi error dan tidak bisa dilanjutkan untuk menginstall Laravel 11

```
PS C:\Users\Wardendina\Pliko\OneDrive\Documents\SEMESTER 5\PHP_MDB_SIB\projectB1\laravel\11> composer create-project laravel/laravel="^11.0" api-breeze
Warning: The requested package laravel/laravel could not be found in any version
that satisfies your platform.
Cannot use laravel/laravel's latest version v21.0.8 as it requires php >=8.2 which is not satisfied by your platform.

In CreateProjectCommand.php line 421:
  Could not find package laravel/laravel with version 11.0 in a version installable using your PHP version, PHP extensions and Composer version.

create-project [-s] --stability STABILITY [-prefer-source] [-prefer-dist] [-prefer-install PREFER-INSTALL] [-repository REPOSITORY] [-repository-url URL] [-add-repository] [-dev] [-no-dev] [-no-custom-installers] [-no-scripts]
[-progress] [-no-secure-http] [-keep-vcs] [-remove-vcs] [-no-install] [-no-audit] [-audit-format AUDIT-FORMAT] [-ignore-platform-req IGNORE-PLATFORM-REQ] [-ignore-platform-reqs] [-ask] [-] [-package] [-directory] [-version]]
```

Seperti berikut errornya, Ketika saat saya mencoba mengganti php versi laragonnya ke 8.3 juga tidak bisa dan muncul error seperti ini



- Tetapi saat saya gunakan Laravel 10.3 bisa dikarenakan laptop saya hanya bisa menggunakan 8.2 dan saat mencoba di 8.3 tetep tidak bisa, dan hasil installan menggunakan 8.2 dengan Laravel 10.3 seperti ini

```
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termw3nd ..... DONE
spatie/laravel-ignition ..... DONE
```

Langkah 2: Install Laravel Breeze

```

PS C:\Users\Wardanadira_Pinkan\OneDrive\Documents\SEMESTER 5\PEMR_MOB_SIB\project13\laravel_11\api-breeze> composer require laravel/breeze --dev
Cannot use laravel/breeze's latest version v2.2.5 as it requires php ^8.2.0 which is not satisfied by your platform.
./composer.json has been updated
Running composer update laravel/breeze
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking laravel/breeze (v1.29.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing laravel/breeze (v1.29.1): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoLoadDump
> @php artisan package:discover --ansi

INFO Discovering packages.

laravel/breeze ..... DONE
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

82 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

INFO No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
Using version ^1.29 for laravel/breeze

```

Langkah 3: Konfigurasikan Basis Data dan Jalankan Migrasi

```

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel10.3_api
DB_USERNAME=root
DB_PASSWORD=

PS C:\Users\Wardanadira_Pinkan\OneDrive\Documents\SEMESTER 5\PEMR_MOB_SIB\project13\laravel_11\api-breeze> php artisan migrate
[info] The database 'laravel10.3_api' does not exist on the 'mysql' connection.
Would you like to create it? (yes/no) [no]
> yes

INFO Preparing database.

Creating migration table ..... 12ms DONE
INFO Running migrations.

2014_10_12_000000_create_users_table ..... 22ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 6ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 19ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 31ms DONE

```

Langkah 4: Buat Create Authentication Endpoints

```

api-breeze > routes > api.php
1
2  use App\Http\Controllers\Auth\AuthenticatedSessionController;
3  use App\Http\Controllers\Auth\RegisteredUserController;
4  use Illuminate\Support\Facades\Route;
5
6
7  /*
8  |-----
9  | API Routes
10 |-----
11 |
12 | Here is where you can register API routes for your application. These
13 | routes are loaded by the RouteServiceProvider within a group which
14 | is assigned the "api" middleware group. Enjoy building your API!
15 |
16 */
17
18 Route::post('/register', [RegisteredUserController::class, 'store']);
19 Route::post('/login', [AuthenticatedSessionController::class, 'store']);
20 Route::post('/logout', [AuthenticatedSessionController::class, 'destroy'])->middleware('auth:sanctum');

```

Langkah 5: Update Controllers

- **RegisteredUserController.php**

```

api-breeze > app > Http > Controllers > RegisteredUserController.php > PHP Intelephense > RegisteredUserController
1  <?php
2
3  namespace App\Http\Controllers\Auth;
4
5  use App\Models\User;
6  use Illuminate\Auth\Events\Registered;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Hash;
9  use Illuminate\Validation\Rules;
10 use App\Http\Controllers\Controller;
11
12 class RegisteredUserController extends Controller
13 {
14     public function store(Request $request)
15     {
16         $request->validate([
17             'name' => ['required', 'string', 'max:255'],
18             'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
19             'password' => ['required', 'confirmed', Rules\Password::defaults()],
20         ]);
21
22         $user = User::create([
23             'name' => $request->name,
24             'email' => $request->email,
25             'password' => Hash::make($request->password),
26         ]);
27
28         event(new Registered($user));
29
30         $token = $user->createToken('auth_token')->plainTextToken;
31
32         return response()->json([
33             'access_token' => $token,
34             'token_type' => 'Bearer',
35             'user' => $user
36         ]);
37     }
38 }

```

- **AuthenticatedSessionController.php**

```

<?php

namespace App\Http\Controllers\Auth;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use App\Http\Controllers\Controller;

class AuthenticatedSessionController extends Controller
{
    public function store(Request $request)
    {
        $request->validate([
            'email' => ['required', 'string', 'email'],
            'password' => ['required', 'string'],
        ]);

        if (!Auth::attempt($request->only('email', 'password'))) {
            return response()->json([
                'message' => 'Invalid login credentials'
            ], 401);
        }

        $user = Auth::user();
        $token = $user->createToken('auth_token')->plainTextToken;

        return response()->json([
            'access_token' => $token,
            'token_type' => 'Bearer',
            'user' => $user,
            'status' => 'Login successful',
        ]);
    }

    public function destroy(Request $request)
    {
        $request->user()->currentAccessToken()->delete();
        return response()->json(['message' => 'Logout successful']);
    }
}

```

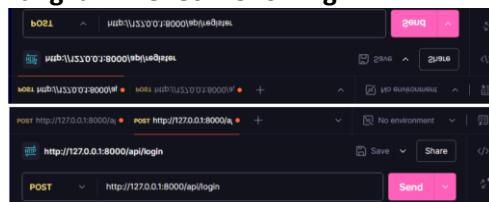
Langkah 6: Jalankan Aplikasi Laravel

```

PS C:\Users\Waphp artisan serve
OneDrive\Documents\SEMESTER 5\PEMR_MOB_SIB\project13\laravel 11\api-breeze>
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server

```

Langkah 7: Check following API



Pembuatan Aplikasi Mobile Flutter

Langkah 1: Persiapan Proyek Flutter

1. Buat Proyek Flutter Baru



Langkah 2:Mengatur Struktur Proyek

```
dependencies:  
  flutter:  
    sdk: flutter  
  http: ^0.13.3  
  shared_preferences: ^2.0.6  
  provider: ^6.0.0  
  flutter_secure_storage: ^5.0.2  
  flutter_secure_storage_windows: ^1.1.3
```

Langkah 3: Membuat Model Pengguna

```
class User {  
  final int id;  
  final String name;  
  final String email;  
  User({required this.id, required this.name, required this.email});  
  factory User.fromJson(Map<String, dynamic> json) {  
    return User(  
      id: json['id'],  
      name: json['name'],  
      email: json['email'],  
    );  
  }  
}
```

Langkah 4: Membuat Layanan API

```

import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:flutter_secure_storage/flutter_secure_storage.dart';
import '../models/user_model.dart';

class AuthService {
  final String apiUrl = "http://127.0.0.1:8000/api/login";
  final storage = const FlutterSecureStorage();
  Future<bool> login(String email, String password) async {
    final response = await http.post(
      Uri.parse('$apiUrl/login'),
      headers: {'Content-Type': 'application/json'},
      body: jsonEncode({'email': email, 'password': password}),
    );
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);
      await storage.write(key: 'token', value: data['token']);
      return true;
    } else {
      return false;
    }
  }

  Future<User?> getProfile() async {
    final token = await storage.read(key: 'token');
    final response = await http.get(
      Uri.parse('$apiUrl/profile'),
      headers: {
        'Content-Type': 'application/json',
        'Authorization': 'Bearer $token',
      },
    );
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);
      return User.fromJson(data['user']);
    } else {
      return null;
    }
  }

  Future<void> logout() async {
    await storage.delete(key: 'token');
  }
}

```

Langkah 5: Menyusun State Management dengan Provider

```

import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:flutter_secure_storage/flutter_secure_storage.dart';
import '../models/user_model.dart';

class AuthService {
  final String apiUrl = "http://127.0.0.1:8000/api/login";
  final storage = const FlutterSecureStorage();
  Future<bool> login(String email, String password) async {
    final response = await http.post(
      Uri.parse('$apiUrl/login'),
      headers: {'Content-Type': 'application/json'},
      body: jsonEncode({'email': email, 'password': password}),
    );
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);
      await storage.write(key: 'token', value: data['token']);
      return true;
    } else {
      return false;
    }
  }

  Future<User?> getProfile() async {
    final token = await storage.read(key: 'token');
    final response = await http.get(
      Uri.parse('$apiUrl/profile'),
      headers: {
        'Content-Type': 'application/json',
        'Authorization': 'Bearer $token',
      },
    );
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);
      return User.fromJson(data['user']);
    } else {
      return null;
    }
  }

  Future<void> logout() async {
    await storage.delete(key: 'token');
  }
}

```

Langkah 6: Membuat Halaman Login

```
children: [
    TextFormField(
        controller: emailController,
        decoration: const InputDecoration(labelText: 'Email'),
    ), // TextFormField
    TextFormField(
        controller: passwordController,
        decoration: const InputDecoration(labelText: 'Password'),
        obscureText: true,
    ), // TextFormField
    const SizedBox(height: 20),
    ElevatedButton(
        onPressed: () async {
            bool success = await authProvider.login(
                emailController.text,
                passwordController.text,
            );
            if (success) {
                if (context.mounted) {
                    Navigator.of(context).pushReplacementNamed('/profile');
                }
            } else {
                if (context.mounted) {
                    ScaffoldMessenger.of(context).showSnackBar(
                        const SnackBar(
                            content: Text('Login failed!'),
                        ), // SnackBar
                    );
                }
            }
        },
        child: const Text('Login'),
    ), // ElevatedButton
], // Column
), // Padding
); // Scaffold
```

Langkah 7: Membuat Halaman Profil

```
if (!mounted) return;

if (authProvider.user != null) {
    print("Navigating to profile");
    Navigator.of(context).pushReplacementNamed('/profile');
} else {
    print("Navigating to login");
    Navigator.of(context).pushReplacementNamed('/login');
}
);

@Override
Widget build(BuildContext context) {
    return const Scaffold(
        body: Center(
            child: Text('My Flutter App', style: TextStyle(fontSize: 24)),
        ), // Center
    ); // Scaffold
}
```

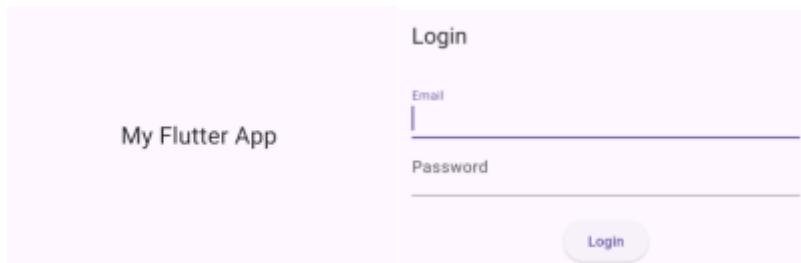
Langkah 8: Mengatur Routing dan Provider

```
debugShowCheckedModeBanner: false,
title: 'Flutter App',
theme: ThemeData(
    primarySwatch: Colors.blue,
), // ThemeData
initialRoute: '/',
routes: [
    RouteSettings(name: '/profile', pageBuilder: (context) => ProfileScreen()),
    RouteSettings(name: '/login', pageBuilder: (context) => LoginScreen()),
]
```

Langkah 9: Menyiapkan Splash Screen

```
@override
Widget build(BuildContext context) {
    return const Scaffold(
        body: Center(
            child: Text('My Flutter App', style: TextStyle(fontSize: 24)),
        ), // Center
    ); // Scaffold
}
```

Langkah 10: Menjalankan Aplikasi



Keterangan :

Mohon maaf apabila saya mengumpulkan terlambat dikarenakan kondisi saya sedang sakit (pusing, panas, batuk, pilek, asam lambung, maag) maka mohon maaf sekali saya mengumpulkan terlambat dikarenakan sakit. Sekian, Terima Kasih.