

Dokumentasi API



Disusun Oleh:

221511012

Gian Vilcan Patra

221511016

M Naufal Fadli Aziz

221511044

Arya Putra Kusumah

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA PROGRAM STUDI DIPLOMA III TEKNIK INFORMATIKA POLITEKNIK NEGERI BANDUNG

2023

Daftar Isi

| | |
|------------------------------------|----|
| 1. Register | 2 |
| 2. Login..... | 2 |
| 3. Booking Appointment | 4 |
| 4. Retrieve Booking Details..... | 5 |
| 5. Store Rating Details..... | 6 |
| 6. Store Favorite Lawyer | 7 |
| 7. Check Availability Lawyer | 9 |
| 8. Cancel Appointment | 10 |
| 9. Reschedule Appointment | 11 |

1. Register

- Endpoint: **POST** 'api/register'
- Deskripsi: Membuat akun untuk User
- Request Parameter:
 1. 'name': username
 2. 'email': email
 3. 'password': password
- Response:
 1. Success:


```
final userRegistration = await DioProvider().registerUser(
    _nameController.text,
    _emailController.text,
    _passController.text);
Navigator.of(context).pushReplacementNamed('/');
showSuccessDialog();
```
 2. Failed:


```
showValidationErrorDialog();
```
- Contoh Penggunaan:
Pada folder providers, file dio_provider.dart

```
//register new user
Future<dynamic> registerUser(
  String username, String email, String password) async {
  try {
    var user = await Dio().post('http://127.0.0.1:8000/api/register',
      data: {'name': username, 'email': email, 'password': password});
    if (user.statusCode == 201 && user.data != '') {
      return true;
    } else {
      return false;
    }
  } catch (error) {
    return error;
  }
}
```

2. Login

- Endpoint:
POST 'api/login'

GET 'api/user'

- Deskripsi: User melakukan login ke aplikasi
- Request Parameter:
 1. 'email': email
 2. 'password': password
- Response:
 1. Success:

```
final token = await DioProvider()
    .getToken(_emailController.text, _passController.text);

if (token == true) {
    //auth.loginSuccess(); //update login status
    //rediret to main page

    //grab user data here
    final SharedPreferences prefs =
        await SharedPreferences.getInstance();
    final tokenValue = prefs.getString('token') ?? '';

    if (tokenValue.isNotEmpty && tokenValue != '') {
        //get user data
        final response = await DioProvider().getUser(tokenValue);
        if (response != null) {
            setState(() {
                //json decode
                Map<String, dynamic> appointment = {};
                final user = json.decode(response);

                //check if any appointment today
                for (var lawyerData in user['lawyer']) {
                    //if there is appointment return for today

                    if (lawyerData['appointments'] != null) {
                        appointment = lawyerData;
                    }
                }

                auth.loginSuccess(user, appointment);
                MyApp.navigatorKey.currentState!.pushNamed('main');
            });
        }
    }
}
```

2. Failed

```

    } else {
      //login gagal, tampilkan dialog kesalahan
      _showErrorDialog('Incorrect email or password');
    }
  }
}

```

- Contoh Penggunaan:
Pada folder providers, file dio_provider.dart

```

//get token
Future<dynamic> getToken(String email, String password) async {
  try {
    var response = await Dio().post('http://127.0.0.1:8000/api/login',
      data: {'email': email, 'password': password});

    if (response.statusCode == 200 && response.data != '') {
      final SharedPreferences prefs = await SharedPreferences.getInstance();
      await prefs.setString('token', response.data);
      return true;
    } else {
      return false;
    }
  } catch (error) {
    return false; // Kembali nilai false jika terjadi kesalahan
  }
}

//get user data
Future<dynamic> getUser(String token) async {
  try {
    var user = await Dio().get('http://127.0.0.1:8000/api/user',
      options: Options(headers: {'Authorization': 'Bearer $token'}));
    if (user.statusCode == 200 && user.data != '') {
      return json.encode(user.data);
    }
  } catch (error) {
    return error;
  }
}

```

3. Booking Appointment

- Endpoint:
POST 'api/book'
- Deskripsi: Melakukan jadwal booking pada lawyer oleh user
- Request Parameter:
 1. 'date': date

2. 'day': day
3. 'time': time
4. 'lawyer_id': lawyer

- Response:

1. Success

```
final booking = await DioProvider().bookAppointment(
  getDate, getDay, getTime, lawyer['lawyer_id'], token!);
if (booking == 200) {
  MyApp.navigatorKey.currentState!
    .pushNamed('success_booking');
}
```

- Contoh Penggunaan:

```
//store booking details
Future<dynamic> bookAppointment(
  String date, String day, String time, int lawyer, String token) async {
  try {
    var response = await Dio().post('http://127.0.0.1:8000/api/book',
      data: {'date': date, 'day': day, 'time': time, 'lawyer_id': lawyer},
      options: Options(headers: {'Authorization': 'Bearer $token'}));

    if (response.statusCode == 200 && response.data != '') {
      return response.statusCode;
    } else {
      return 'Error';
    }
  } catch (error) {
    return error;
  }
}
```

4. Retrieve Booking Details

- Endpoint:
GET 'api/appointments'
- Deskripsi: Melakukan schedule appointment
- Request Parameter:
 1. 'Authorization': 'Bearer \$token'
- Response:

Success:

```
//get appointments details
Future<void> getAppointments() async {
  final SharedPreferences prefs = await SharedPreferences.getInstance();
  final token = prefs.getString('token') ?? '';
  final appointment = await DioProvider().getAppointments(token);
  if (appointment != 'Error') {
    setState(() {
      schedules = json.decode(appointment);
    });
  }
}
```

- Contoh Penggunaan:

```
//retrieve booking details
Future<dynamic> getAppointments(String token) async {
  try {
    var response = await Dio().get('http://127.0.0.1:8000/api/appointments',
      options: Options(headers: {'Authorization': 'Bearer $token'}));

    if (response.statusCode == 200 && response.data != '') {
      return json.encode(response.data);
    } else {
      return 'Error';
    }
  } catch (error) {
    return error;
  }
}
```

5. Store Rating Details

- Endpoint:
POST 'api/reviews'
- Deskripsi: User memberikan rating pada lawyer
- Request Parameter:
 1. 'ratings': ratings,
 2. 'reviews': reviews,
 3. 'appointment_id': id,
 4. 'lawyer_id': lawyer
- Response:
Success:

```

final rating = await DioProvider()
    .storeReviews(
        response.comment,
        response.rating,
        widget.lawyer['appointments']
            ['id'], //this is appointment id
        widget.lawyer[
            'law_id'], //this is lawyer id
        token);

//if successful, then refresh
if (rating == 200 && rating != '') {
    MyApp.navigatorKey.currentState!
        .pushNamed('main');
}

```

- Contoh Penggunaan

```

//store rating details
Future<dynamic> storeReviews(
    String reviews, double ratings, int id, int lawyer, String token) async {
    try {
        var response = await Dio().post('http://127.0.0.1:8000/api/reviews',
            data: {
                'ratings': ratings,
                'reviews': reviews,
                'appointment_id': id,
                'lawyer_id': lawyer
            },
            options: Options(headers: {'Authorization': 'Bearer $token'}));

        if (response.statusCode == 200 && response.data != '') {
            return response.statusCode;
        } else {
            return 'Error';
        }
    } catch (error) {
        return error;
    }
}

```

6. Store Favorite Lawyer

- Endpoint:
POST 'api/fav'

- Deskripsi: User menambahkan data favorit lawyer
- Request Parameter:
 1. 'Authorization': 'Bearer \$token'
- Response:
- Success:

```
Provider.of<AuthModel>(context, listen: false).setFavList(list);

final SharedPreferences prefs =
  | await SharedPreferences.getInstance();
final token = prefs.getString('token') ?? '';

if (token.isNotEmpty && token != '') {
  // Update the favorite list into database
  final response = await DioProvider().storeFavLaw(token, list);
  // If insert successfully, then change the favorite status
  if (response == 200) {
    setState(() {
      isFav = !isFav;
    });
  }
}
```

- Contoh Penggunaan:

```
Future<dynamic> storeFavLaw(String token, List<dynamic> favList) async {
  try {
    var response = await Dio().post('http://127.0.0.1:8000/api/fav',
      data: {
        'favList': favList,
      },
      options: Options(headers: {'Authorization': 'Bearer $token'}));

    if (response.statusCode == 200 && response.data != '') {
      return response.statusCode;
    } else {
      return 'Error';
    }
  } catch (error) {
    return error;
  }
}
```


7. Check Availability Lawyer

- Endpoint:
POST 'api/check-slot'
- Deskripsi: User melihat lawyer yang tersedia
- Request Parameter:
 1. 'date': date
 2. 'time': time
 3. 'lawyer_id': lawyerId
- Response:
Success:

```
final getDate = DateConverted.getDate(_currentDay);
final getDay = DateConverted.getDay(_currentDay.weekday);
final getTime = DateConverted.getTime(_currentIndex!);

bool isAvailable = await DioProvider().checkSlotAvailability(
  getDate, getTime, lawyer['lawyer_id'], token!);

if (!isAvailable) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text('Booking Failed'),
        content: Text('Slot already booked!'),
        actions: <Widget>[
          TextButton(
            child: Text('OK'),
            onPressed: () {
              Navigator.of(context).pop();
            },
          ), // TextButton
        ], // <Widget>[]
      ); // AlertDialog
    },
  );
  return;
}
```

- Contoh Penggunaan:

```

Future<bool> checkSlotAvailability(
  String date, String time, int lawyerId, String token) async {
  try {
    var response = await Dio().post('http://127.0.0.1:8000/api/check-slot',
      data: {'date': date, 'time': time, 'lawyer_id': lawyerId},
      options: Options(headers: {'Authorization': 'Bearer $token'}));

    return response.statusCode == 200;
  } catch (error) {
    return false;
  }
}

```

8. Cancel Appointment

- Endpoint:
POST 'api/ appointments/cancel/{id}'
- Deskripsi: User melakukan pembatalan appointment
- Request Parameter:
 1. 'Authorization': 'Bearer \$token'
- Response:
Success:
appointment_card.dart

```

final SharedPreferences prefs =
  await SharedPreferences.getInstance();
final token = prefs.getString('token') ?? '';

final response = await DioProvider().cancelAppointment(
  widget.lawyer['appointments']['id'], token);

if (response == 200) {
  // Refresh atau update UI setelah sukses membatalkan
  MyApp.navigatorKey.currentState!.pushNamed('main');
} else {
  // Tampilkan pesan error jika gagal
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text('Failed to cancel appointment'),
    ));
}

```

appointment_page.dart

```
Future<void> cancelAppointment(int appointmentId) async {
  final SharedPreferences prefs = await SharedPreferences.getInstance();
  final token = prefs.getString('token') ?? '';
  final response =
    await DioProvider().cancelAppointment(appointmentId, token);

  if (response == 200) {
    // Refresh appointment list after cancellation
    getAppointments();
  } else {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('Failed to cancel appointment')),
    );
  }
}
```

- Contoh Penggunaan:

```
Future<int> cancelAppointment(int appointmentId, String token) async {
  try {
    var response = await Dio().post(
      'http://127.0.0.1:8000/api/appointments/cancel/$appointmentId',
      options: Options(headers: {'Authorization': 'Bearer $token'}),
    );

    return response.statusCode ?? 500;
  } catch (error) {
    return 500;
  }
}
```

9. Reschedule Appointment

- Endpoint:
POST 'api/reschedule'
- Deskripsi: User melakukan penjadwalan ulang appointment
- Request Parameter:
 1. 'appointment_id': appointmentId,
 2. 'date': date,
 3. 'time': time,
 4. 'Authorization': 'Bearer \$token'
- Response:
Success:

```

if (isReschedule) {
  final response = await DioProvider().rescheduleAppointment(
    appointmentId!, getDate, getTime, token!);
  if (response == 200) {
    MyApp.navigatorKey.currentState!
      .pushNamed('success_booking');
  }
} else {
  final booking = await DioProvider().bookAppointment(
    getDate, getDay, getTime, lawyer['lawyer_id'], token!);
  if (booking == 200) {
    MyApp.navigatorKey.currentState!
      .pushNamed('success_booking');
  }
}
}

```

- Contoh Penggunaan:

```

Future<int> rescheduleAppointment(
  int appointmentId, String date, String time, String token) async {
  try {
    final response = await Dio().post(
      'http://127.0.0.1:8000/api/reschedule',
      data: {
        'appointment_id': appointmentId,
        'date': date,
        'time': time,
      },
      options: Options(
        headers: {'Authorization': 'Bearer $token'},
      ),
    );
    return response.statusCode!;
  } catch (e) {
    print(e);
    return 400; // Return an error code if request fails
  }
}

```

10.Update User

- Endpoint:
PUT 'api/update'
- Deskripsi: User melakukan update data user
- Request Parameter:
 1. 'name': name
 2. 'email': email

- Response:
Success:

```
Future<bool> updateUser(String? name, String? email) async {  
  SharedPreferences prefs = await SharedPreferences.getInstance();  
  String? token = prefs.getString('token');  
  if (token != null) {  
    bool success = await DioProvider().updateUser(token, name, email);  
    if (success) {  
      await fetchUser();  
      return true;  
    }  
  }  
  return false;  
}
```

- Contoh Penggunaan:

```
Future<bool> updateUser(String token, String? name, String? email) async {  
  try {  
    var response = await Dio().put(  
      'http://127.0.0.1:8000/api/update',  
      data: {'name': name, 'email': email},  
      options: Options(headers: {'Authorization': 'Bearer $token'}),  
    );  
    return response.statusCode == 200;  
  } catch (error) {  
    return false;  
  }  
}
```