

ログイン API の動作確認

下記の内容でリクエストを送信する。この例では以前 seeder を用いて作成したユーザでログインしている。

項目	設定値
Method	POST
URI	http://localhost/api/login
Headers	Accept: application/json
Body	email: dio@example.com password: password

レスポンス (token を後で使用するのでメモしておく)

```
{
  "access_token": "2|rTz5BHdADnsVIZniDXj7q2gGYo0nc5dvnyDJkuM8292e53eb",
  "token_type": "Bearer"
}
```

email: dio@example.com
password: password

```

User.php app\Models\User.php
12 {
13     use HasFactory, Notifiable;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var array<int, string>
19      */
20     protected $fillable = [
21         'name',
22         'email',
23         'password',
24         'department',
25         'class',
26         'profile_image',
27         'number',
28         'hobby',
29         'business_experience',
30     ];
31

```

userテーブルの
編集したいカラム

```

UserController.php app\Http\Controllers\Api
LARAVEL-GW01
  app
    Http
      Controllers
        Api
          TweetLikeController.php
          UserController.php
        Auth

```

```

LARAVEL-GW01
  app
  bootstrap
  config
  database
  node_modules
  public
  resources
  routes
    api.php
    auth.php
    console.php

```

api.php routes\api.php

```
16 Route::middleware('auth:sanctum')->group(function(){
17     ...Route::apiResource('tweets', TweetController::class);
18     ...Route::post('/tweets/{tweet}/like', [TweetLikeController::class,
19     ...Route::delete('/tweets/{tweet}/like', [TweetLikeController::class
20     ...route::apiResource('tweets.comments', CommentController::class);
21     ...Route::get('user', [UserController::class, 'getUser']);
22     ...Route::get('users', [UserController::class, 'index']);
23     ...Route::put('user', [UserController::class, 'updateUser']);
24 });
25
```

UserController.php app\Http\Controllers\Api\UserController.php

```
7 use App\Models\User;
8 use Illuminate\Support\Facades\Hash;
9 use Illuminate\Support\Facades\Auth;
10
11 class UserController extends Controller
12 {
13     ...// ユーザー情報の更新
14     ...public function updateUser(Request $request)
15     ...{
16         ...$user = Auth::user();
17
18         ...$validatedData = $request->validate([
19         ...     'name' => 'sometimes|required|string|max:255',
20         ...     'email' => 'sometimes|required|string|email|max:255'
```

HTTP <http://localhost/api/user> Save No Environment

PUT <http://localhost/api/user> Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Code Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value
<input type="checkbox"/>	email	dio@example.com
<input type="checkbox"/>	password	password
<input checked="" type="checkbox"/>	number	1002
<input checked="" type="checkbox"/>	department	企画

Body Cookies Headers (7) Test Results Status: 200 OK Time: 66 ms Size: 540 B

Pretty Raw Preview JSON

```
3  {
4    "id": 2,
5    "name": "ディオ・ブランドー",
6    "email": "dio@example.com",
7    "email_verified_at": null,
8    "created_at": "2024-09-05T11:29:29.000000Z",
9    "updated_at": "2024-09-05T11:29:29.000000Z",
10   "department": null,
11   "class": null,
12   "profile_image": null,
13   "number": null,
14 }
```

