Naziv dokumenta: Specifikacija REST API sučelja – Social Network

Sadržaj:

# Specifikacija REST API sučelja – Endpoints

## 1. Dohvaćanje podataka o trenutno ulogiranom korisniku

**URI:** GET /api/user **Autentikacija:** Da (JWT ili sesijska autentikacija) **Opis:** Dohvaća detalje trenutno ulogiranog korisnika.

**Zaglavlja zahtjeva:**

Content-Type: application/json  
Accept: application/json  
Authorization: Bearer {token}

**Primjer zahtjeva:**

curl -X GET "http://socialnetwork.test/api/user" \  
-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9..." \  
-H "Accept: application/json"

**Primjer odgovora (200 OK):**

{  
 "id": 1,  
 "name": "Boško Raguž",  
 "email": "bosko@example.com",  
 "role": "user",  
 "created\_at": "2025-08-29T12:00:00Z"  
}

**Primjer odgovora (401 Unaunthenticated):**

{  
 "message": "Unauthenticated."  
}

## 2. Kreiranje nove grupe

**URI:** POST /api/groups **Autentikacija:** Da **Opis:** Stvara novu grupu u sustavu.

**Primjer tijela zahtjeva:**

{  
 "name": "Fitness Enthusiasts",  
 "description": "Grupa za sve ljubitelje teretane"  
}

**Primjer odgovora (201 Created):**

{  
 "id": "3wuPF",  
 "name": "Fitness Enthusiasts",  
 "description": "Grupa za sve ljubitelje teretane",  
 "created\_at": "2025-08-29T12:30:00Z"  
}

## 3. Ažuriranje grupe

**URI:** POST /api/groups/{id\_string} **Autentikacija:** Da

**URL parametri:**

| Parametar | Tip | Opis | Primjer |
| --- | --- | --- | --- |
| id\_string | string | ID grupe za ažuriranje | 3wuPF |

**Primjer tijela zahtjeva:**

{  
 "name": "Fitness & Health Enthusiasts",  
 "description": "Grupa za sve koji vole fitness i zdravlje"  
}

**Primjer odgovora (200 OK):**

{  
 "id": "3wuPF",  
 "name": "Fitness & Health Enthusiasts",  
 "description": "Grupa za sve koji vole fitness i zdravlje",  
 "updated\_at": "2025-08-29T12:45:00Z"  
}

## 4. Brisanje grupe

**URI:** DELETE /api/groups/{id\_string} **Autentikacija:** Da

**URL parametri:**

| Parametar | Tip | Opis | Primjer |
| --- | --- | --- | --- |
| id\_string | string | ID grupe za brisanje | 3wuPF |

**Primjer zahtjeva:**

curl -X DELETE "http://socialnetwork.test/api/groups/3wuPF" \  
-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9..." \  
-H "Accept: application/json"

**Primjer odgovora (204 No Content):** *Nema sadržaja u tijelu odgovora*

## 5. Dohvaćanje postova za dashboard

**URI:** GET /api/dashboard/posts **Autentikacija:** Da

**Primjer zahtjeva:**

curl -X GET "http://socialnetwork.test/api/dashboard/posts" \  
-H "Authorization: Bearer {token}" \  
-H "Accept: application/json"

**Primjer odgovora (401 Unaunthenticated):**

{  
 "message": "Unauthenticated."  
}

## 6. Brisanje posta iz grupe

**URI:** DELETE /api/groups/{group\_id\_string}/posts/{id} **Autentikacija:** Da

**URL parametri:**

| Parametar | Tip | Opis | Primjer |
| --- | --- | --- | --- |
| group\_id\_string | string | ID grupe | 3wuPF |
| id | int | ID posta | 1 |

**Primjer zahtjeva:**

curl -X DELETE "http://socialnetwork.test/api/groups/3wuPF/posts/1" \  
-H "Authorization: Bearer {token}" \  
-H "Accept: application/json"

**Primjer odgovora (401 Unaunthenticated):**

{  
 "message": "Unauthenticated."  
}

## 7. Primjeri koda – Laravel kontroleri

// UserController.php  
public function currentUser(Request $request) {  
 return response()->json($request->user());  
}  
  
// GroupController.php  
public function store(Request $request) {  
 $group = Group::create($request->only(['name', 'description']));  
 return response()->json($group, 201);  
}  
  
public function update(Request $request, $id\_string) {  
 $group = Group::where('id\_string', $id\_string)->firstOrFail();  
 $group->update($request->only(['name', 'description']));  
 return response()->json($group);  
}  
  
public function destroy($id\_string) {  
 $group = Group::where('id\_string', $id\_string)->firstOrFail();  
 $group->delete();  
 return response()->json(null, 204);  
}  
  
// DashboardController.php  
public function getPosts() {  
 return response()->json(Post::latest()->get());  
}  
  
// GroupPostController.php  
public function destroy($group\_id\_string, $id) {  
 $post = Post::where('group\_id\_string', $group\_id\_string)->findOrFail($id);  
 $post->delete();  
 return response()->json(null, 204);  
}

## 8. Napomene

* Svi endpointovi zahtijevaju autentikaciju.
* Neautentificirani zahtjevi vraćaju **401 Unauthorized**.