

# Blog API - Dokumentacja Testów

---

## Przegląd

Zestaw testów dla API bloga obejmujący główne scenariusze użycia, przypadki brzegowe, błędy autoryzacji i walidacji.

## Środowisko testowe

### Wymagania

- Java 21+
- PostgreSQL na localhost:5432
- Aplikacja na http://localhost:8080

## Kategorie testów

### 1. Testy uwierzytelniania

#### Test 1.1: Poprawne logowanie

```
curl -X POST http://localhost:8080/api/v1/auth/login \  
-H "Content-Type: application/json" \  
-d '{"email": "user@gmail.com", "password": "1234"}'
```

**Oczekiwany wynik:** 200 OK, token JWT

#### Test 1.2: Niepoprawne hasło

```
curl -X POST http://localhost:8080/api/v1/auth/login \  
-H "Content-Type: application/json" \  
-d '{"email": "user@gmail.com", "password": "zle_haslo"}'
```

**Oczekiwany wynik:** 401 Unauthorized

#### Test 1.3: Rejestracja nowego użytkownika

```
curl -X POST http://localhost:8080/api/v1/auth/signin \  
-H "Content-Type: application/json" \  
-d '{  
  "name": "Test User",  
  "email": "test@example.com",  
  "password": "password123"  
}'
```

**Oczekiwany wynik:** 201 Created

## 2. Testy kategorii

### Test 2.1: Pobieranie kategorii (publiczne)

```
curl -X GET http://localhost:8080/api/v1/categories
```

**Oczekiwany wynik:** 200 OK

### Test 2.2: Tworzenie kategorii (wymagana autoryzacja)

```
# Pobierz token
TOKEN=$(curl -s -X POST http://localhost:8080/api/v1/auth/login \
  -H "Content-Type: application/json" \
  -d '{"email": "user@gmail.com", "password": "1234"}' | \
  grep -o '"token": "[^"]*" | cut -d'"' -f4)

curl -X POST http://localhost:8080/api/v1/categories \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $TOKEN" \
  -d '{"name": "Technologia"}'
```

**Oczekiwany wynik:** 201 Created

### Test 2.3: Tworzenie kategorii bez autoryzacji

```
curl -X POST http://localhost:8080/api/v1/categories \
  -H "Content-Type: application/json" \
  -d '{"name": "Technologia"}'
```

**Oczekiwany wynik:** 401 Unauthorized

### Test 2.4: Walidacja - za krótka nazwa

```
curl -X POST http://localhost:8080/api/v1/categories \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $TOKEN" \
  -d '{"name": "Te"}'
```

**Oczekiwany wynik:** 400 Bad Request

### 3. Testy tagów

#### Test 3.1: Pobieranie tagów (publiczne)

```
curl -X GET http://localhost:8080/api/v1/tags
```

Oczekiwany wynik: 200 OK

#### Test 3.2: Tworzenie tagów

```
curl -X POST http://localhost:8080/api/v1/tags \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer $TOKEN" \  
-d '{"names": ["java", "spring", "api"]}'
```

Oczekiwany wynik: 201 Created

### 4. Testy postów

#### Test 4.1: Pobieranie opublikowanych postów (publiczne)

```
curl -X GET http://localhost:8080/api/v1/posts
```

Oczekiwany wynik: 200 OK

#### Test 4.2: Pobieranie szkiców (tylko zalogowani)

```
# Z autoryzacją  
curl -X GET http://localhost:8080/api/v1/posts/drafts \  
-H "Authorization: Bearer $TOKEN"  
  
# Bez autoryzacji  
curl -X GET http://localhost:8080/api/v1/posts/drafts
```

Oczekiwany wynik: 200 OK / 401 Unauthorized

#### Test 4.3: Tworzenie posta

```
# Pobierz ID kategorii  
CATEGORY_ID=$(curl -s -X GET http://localhost:8080/api/v1/categories | \  
grep -o '"id": "[^"]*' | head -1 | cut -d'"' -f4)
```

```
curl -X POST http://localhost:8080/api/v1/posts \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{
  "title": "Mój pierwszy post",
  "content": "To jest treść mojego pierwszego posta blogowego z
wystarczającą ilością tekstu.",
  "categoryId": "'$CATEGORY_ID'",
  "tagIds": [],
  "status": "PUBLISHED"
}'
```

**Oczekiwany wynik:** 201 Created

#### Test 4.4: Walidacja posta - za krótki tytuł

```
curl -X POST http://localhost:8080/api/v1/posts \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{
  "title": "Hi",
  "content": "Za krótki tytuł",
  "categoryId": "'$CATEGORY_ID'",
  "status": "PUBLISHED"
}'
```

**Oczekiwany wynik:** 400 Bad Request

#### Test 4.5: Filtrowanie postów po kategorii

```
curl -X GET "http://localhost:8080/api/v1/posts?categoryId=$CATEGORY_ID"
```

**Oczekiwany wynik:** 200 OK

### 5. Przypadki brzegowe

#### Test 5.1: Nieprawidłowy token JWT

```
curl -X GET http://localhost:8080/api/v1/posts/drafts \
-H "Authorization: Bearer nieprawidlowy.token"
```

**Oczekiwany wynik:** 401 Unauthorized

#### Test 5.2: Nieprawidłowy JSON

```
curl -X POST http://localhost:8080/api/v1/categories \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer $TOKEN" \  
-d '{"name": "Test"' # Brakuje zamykającego nawiasu
```

**Oczekiwany wynik:** 400 Bad Request

### Test 5.3: Nieistniejący post

```
curl -X GET http://localhost:8080/api/v1/posts/00000000-0000-0000-0000-  
000000000000
```

**Oczekiwany wynik:** 404 Not Found