

# API-dokumentation

---

## AccountController

---

### POST: api/Account/register

---

Registrerar en ny användare.

#### Body:

Ett `RegisterViewModel` -objekt:

```
{
  "email": "user@example.com",
  "password": "string",
  "confirmPassword": "string",
  "accountName": "string"
}
```

#### Svar:

Ett meddelande om att registreringen lyckades.

#### Felkoder:

- 400: Ogiltiga data
- 400: Användaren finns redan

---

## ApiController

---

### POST: api/Account/login

---

Loggar in en befintlig användare.

#### Body:

Ett `LoginViewModel` -objekt:

```
{
  "Email": "användare@example.com",
  "Password": "lösenord123"
}
```

#### Svar:

Ett `LoginResponse` -objekt:

```
{
  "tokenType": "Bearer",
  "accessToken": "CfDJ8Ccxja12MX5Lv6lREZQmLY4...",
  "expiresIn": 3600,
  "refreshToken": "CfDJ8Ccxja12MX5Lv6lREZQmLa..."
}
```

#### Felkoder:

- 400: Ogiltigt inloggningsförsök
-

## POST: api/Account/refresh

---

Förnyar JWT-token.

### Body:

Ett `RefreshTokenViewModel` -objekt:

```
{
  "RefreshToken": "d45f46ad-4812-415d-8c22-6f37bb61c1e2"
}
```

### Svar:

Ett `LoginResponse` -objekt:

```
{
  "tokenType": "Bearer",
  "accessToken": "CfDJ8Ccxja12MX5Lv6lREZQmLY4...",
  "expiresIn": 3600,
  "refreshToken": "CfDJ8Ccxja12MX5Lv6lREZQmLa..."
}
```

### Felkoder:

- 400: Ogiltig token
- 401: Obehörig

## PostController

---

### GET: api/Post

Returnerar en paginerad lista med inlägg.

#### Parametrar:

- `pageNumber` : Sidnumret att hämta. Standard är 1.
- `pageSize` : Antalet inlägg per sida. Standard är 10.

### Svar:

En lista med `PostResponseDto` -objekt:

```
[
  {
    "Id": 1,
    "Caption": "Caption text",
    "ImageUrl": "http://example.com/image.jpg",
    "Username": "username",
    "LikesCount": 5,
    "Comments": [
      {
        "Id": 1,
        "CommentText": "Comment text",
        "Username": "username",
        "LikesCount": 2,
        "CreatedDate": "2022-01-01T00:00:00"
      }
    ],
    "CategoryName": "Category name",
    "CreatedDate": "2022-01-01T00:00:00"
  }
]
```

#### Felkoder:

- 404: No posts found
- 

#### GET: api/Post/{id}

Returnerar ett enkelt inlägg med ID.

#### Parametrar:

- `id` : ID för inlägget att hämta.

#### Svar:

Ett enda `PostResponseDto` -objekt:

```
{
  "Id": 1,
  "Caption": "Caption text",
  "ImageUrl": "http://example.com/image.jpg",
  "Username": "username",
  "LikesCount": 5,
  "Comments": [
    {
      "CommentText": "Comment text",
      "Username": "username",
      "LikesCount": 2
    }
  ],
  "CategoryName": "Category name",
  "CreatedDate": "2022-01-01T00:00:00"
}
```

#### Felkoder:

- 404: No post found
- 

#### GET: api/Post/Category/{categoryId}

Returnerar alla inlägg som tillhör en specifik kategori.

#### Parametrar:

- `categoryId` : ID för kategorin att hämta inlägg från.

#### Svar:

En lista med `PostResponseDto` -objekt:

```
[
  {
    "Id": 1,
    "Caption": "Caption text",
    "ImageUrl": "http://example.com/image.jpg",
    "Username": "username",
    "LikesCount": 5,
    "Comments": [
      {
        "Id": 1,
        "CommentText": "Comment text",
        "Username": "username",
        "LikesCount": 2,
        "CreatedDate": "2022-01-01T00:00:00"
      }
    ],
    "CategoryName": "Category name",
    "CreatedDate": "2022-01-01T00:00:00"
  }
]
```

#### Felkoder:

- 404: No posts found for this category

---

### POST: api/Post

Skapar ett nytt inlägg.

#### Body:

Ett `PostDto`-objekt:

```
{
  "Caption": "Caption text",
  "ImageUrl": "http://example.com/image.jpg",
  "CategoryId": 1
}
```

#### Svar:

Det skapade `Post`-objektet.

#### Felkoder:

- 401: Unauthorized
- 400: Invalid category

---

### PUT: api/Post/{id}

Uppdaterar ett befintligt inlägg.

#### Parametrar:

- `id`: ID för inlägget att uppdatera.

#### Body:

Ett `PostDto`-objekt:

```
{
  "Caption": "Updated caption text",
  "ImageUrl": "http://example.com/updated_image.jpg",
  "CategoryId": 2
}
```

**Svar:**

Det uppdaterade `Post` -objektet.

**Felkoder:**

- 404: Post not found
- 401: Unauthorized
- 400: Invalid category

---

## DELETE: `api/Post/{id}`

Raderar ett befintligt inlägg.

**Parametrar:**

- `id` : ID för inlägget att radera.

**Svar:**

No content.

**Felkoder:**

- 404: Post not found
- 401: Unauthorized

---

# LikeController

---

## POST: `api/Like/Post/{postId}`

Lägger till en gilla-markering till ett inlägg.

**Parametrar:**

- `postId` : ID för inlägget att gilla.

**Felkoder:**

- 404: Inlägget hittades inte
- 400: Du har redan gillat detta inlägg

---

## POST: `api/Like/Comment/{commentId}`

Lägger till en gilla-markering till en kommentar.

**Parametrar:**

- `commentId` : ID för kommentaren att gilla.

**Felkoder:**

- 404: Kommentaren hittades inte
- 400: Du har redan gillat denna kommentar

---

## DELETE: `api/Like/{postId}`

Tar bort en gilla-markering från ett inlägg.

#### Parametrar:

- `postId` : ID för inlägget att ogilla.

#### Felkoder:

- 404: Inlägget hittades inte
- 404: Inlägget är inte gillat
- 401: Unauthorized

## DELETE: api/Like/Comment/{commentId}

---

Tar bort en gilla-markering från en kommentar.

#### Parametrar:

- `commentId` : ID för kommentaren att ogilla.

#### Felkoder:

- 404: Kommentaren hittades inte
- 404: Kommentaren är inte gillad
- 401: Unauthorized

---

# CommentController

---

## POST: api/Comment

---

Skapar en ny kommentar till ett inlägg.

#### Parametrar:

- `postId` : ID för inlägget att kommentera.

#### Body:

Ett `CommentDto` -objekt:

```
{
  "Text": "Comment text"
}
```

#### Felkoder:

- 404: Post not found
- 401: Unauthorized

## GET: api/Comment/{id}

---

Hämtar en kommentar med ett specifikt ID.

#### Parametrar:

- `id` : ID för kommentaren att hämta.

#### Felkoder:

- 404: Comment not found

## PUT: api/Comment/{id}

---

Uppdaterar en befintlig kommentar.

#### Parametrar:

- `id` : ID för kommentaren att uppdatera.

### Body:

Ett `CommentDto` -objekt:

```
{
  "Text": "Updated comment text"
}
```

### Felkoder:

- 404: Comment not found
- 401: Unauthorized

## DELETE: `api/Comment/{id}`

---

Raderar en befintlig kommentar.

### Parametrar:

- `id` : ID för kommentaren att radera.

### Felkoder:

- 404: Comment not found
- 401: Unauthorized

---

# CategoryController

---

## GET: `api/Category`

---

Hämtar alla kategorier.

### Felkoder:

- 404: No categories found
- 500: An unexpected error occurred

## GET: `api/Category/{id}`

---

Hämtar en kategori med ett specifikt ID.

### Parametrar:

- `id` : ID för kategorin att hämta.

### Felkoder:

- 404: Category not found
- 500: An unexpected error occurred

## POST: `api/Category`

---

Skapar en ny kategori.

### Body:

Ett `CategoryDto` -objekt:

```
{
  "Name": "Category name"
}
```

### Felkoder:

- 400: Category data is required
- 401: You are not authorized to perform this action
- 500: An error occurred while updating the database
- 500: An unexpected error occurred

## PUT: api/Category/{id}

---

Uppdaterar en befintlig kategori.

### Parametrar:

- `id` : ID för kategorin att uppdatera.

### Body:

Ett `CategoryDto` -objekt:

```
{
  "Name": "Updated category name"
}
```

### Felkoder:

- 400: Category data is required
- 401: You are not authorized to perform this action
- 404: Category not found
- 500: An error occurred while updating the database
- 500: An unexpected error occurred

## DELETE: api/Category/{id}

---

Raderar en befintlig kategori.

### Parametrar:

- `id` : ID för kategorin att radera.

### Felkoder:

- 401: You are not authorized to perform this action
- 404: Category not found
- 500: An error occurred while updating the database
- 500: An unexpected error occurred