

# Post

## Overview

Endpoint per le operazioni sui post

## Version information

*Version* : 1.0.0

## Paths

### Creazione di un nuovo post

POST /posts/

#### Description

Gestisce l'inserimento di un post. Un utente per pubblicare un post deve avere fatto l'accesso. Richiede permessi ["post\_write", "admin"]

#### Body parameter

Dati del post

*Name* : body

*Flags* : optional

| Name                                     | Schema                             |
|--|------------------------------------|
| <b>content</b><br><i>required</i>        | string                             |
| <b>topic</b><br><i>optional</i>          | string                             |
| <b>tag_users_id</b><br><i>optional</i>   | < <a href="#">ObjectId</a> > array |
| <b>tag_animals_id</b><br><i>optional</i> | < <a href="#">ObjectId</a> > array |

#### Responses

| HTTP Code | Description           | Schema                                 |
|-----------|-----------------------|--|
| 201       | Creato con successo   | <a href="#">Post</a>                   |
| 400       | Input malformato      | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato       | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti     | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error | No Content                             |

## Consumes

- `application/json`

## Produces

- `application/json`

## Tags

- Post

# Ricerca dei post secondo determinati criteri

GET /posts/

## Description

Gestisce la ricerca dei post dati alcuni criteri. Di default ordina in senso decrescente di data. Gestisce la paginazione.

## Parameters

| Type  | Name  | Description                          | Schema  |
|-------|---|--------------------------------------|---------|
| Query | <code>page_size</code><br><i>required</i>   | Numero di post per pagina            | number  |
| Query | <code>page_number</code><br><i>required</i> | Numero di pagina                     | number  |
| Query | <code>username</code><br><i>optional</i>    | Username dell'autore                 | string  |
| Query | <code>topic</code><br><i>optional</i>       | Categoria del post                   | string  |
| Query | <code>oldest</code><br><i>optional</i>      | Visualizzare prima i post più vecchi | boolean |

## Responses

| HTTP Code | Description           | Schema                                 |
|-----------|-----------------------|--|
| 200       | Trovato con successo  | < <a href="#">Post</a> > array         |
| 400       | Input malformato      | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato       | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti     | <a href="#">ErrorMessage</a>           |
| 404       | Non trovato           | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error | No Content                             |

## Consumes

- `application/json`

## Produces

- `application/json`

## Tags

- Post

# Inserimento di topic

```
POST /posts/topics/
```

## Description

Creazione di un nuovo topic. Richiede permessi [ [] , [{"admin"}] ]

## Body parameter

Dati del topic

*Name* : body

*Flags* : optional

*Type* : [Topic](#)

## Responses

| HTTP Code | Description                            | Schema                                 |
|-----------|--|--|
| 201       | Creato con successo                    | No Content                             |
| 400       | Input malformato                       | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato                        | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti                      | <a href="#">ErrorMessage</a>           |
| 409       | Topic con lo stesso nome già esistente | <a href="#">FieldErrorMessage</a>      |
| 500       | Internal server error                  | No Content                             |

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Topic

# Lista di tutti i topic

```
GET /posts/topics/
```

## Description

Restituisce la lista di tutti i topic disponibili.

## Responses

| HTTP Code | Description           | Schema                          |
|-----------|-----------------------|---------------------------------|
| 200       | Ok                    | < <a href="#">Topic</a> > array |
| 500       | Internal server error | No Content                      |

## Produces

- [application/json](#)

## Tags

- Topic

# Aggiornamento di topic

```
PUT /posts/topics/{topic}
```

## Description

Aggiornamento di un topic. Richiede permessi [ [] , [{"admin"}] ]

## Parameters

| Type | Name                                  | Schema |
|------|---------------------------------------|--------|
| Path | <code>topic</code><br><i>required</i> | string |

## Body parameter

Dati da aggiornare del topic

*Name* : body

*Flags* : optional

*Type* : [Topic](#)

## Responses

| HTTP Code | Description             | Schema                                 |
|-----------|-------------------------|--|
| 200       | Aggiornato con successo | No Content                             |
| 400       | Input malformato        | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato         | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti       | <a href="#">ErrorMessage</a>           |
| 404       | Topic non esistente     | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error   | No Content                             |

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Topic

# Cancellazione di topic

```
DELETE /posts/topics/{topic}
```

## Description

Cancellazione di un topic. Richiede permessi [ [] , [{"admin"}] ]

## Parameters

| Type | Name                                  | Schema |
|------|---------------------------------------|--------|
| Path | <code>topic</code><br><i>required</i> | string |

## Responses

| HTTP Code | Description             | Schema                                 |
|-----------|-------------------------|--|
| 200       | Cancellato con successo | No Content                             |
| 400       | Input malformato        | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato         | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti       | <a href="#">ErrorMessage</a>           |
| 404       | Topic non esistente     | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error   | No Content                             |

## Consumes

- `application/json`

## Produces

- `application/json`

## Tags

- Topic

# Ricerca di un post

```
GET /posts/{post_id}
```

## Description

Gestisce la ricerca di un post tramite il suo id. Un utente per ricercare un post deve avere fatto l'accesso. Richiede permessi ["post\_write", "admin"]

## Body parameter

Id del post

*Name* : post\_id  
*Flags* : required  
*Type* : [ObjectId](#)

## Responses

| HTTP Code | Description           | Schema                                 |
|-----------|-----------------------|--|
| 200       | Trovato con successo  | <a href="#">Post</a>                   |
| 400       | Input malformato      | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato       | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti     | <a href="#">ErrorMessage</a>           |
| 404       | Non trovato           | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error | No Content                             |

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Post

## Example HTTP request

Request body

```
"61338f5b47bfc65136b5de30"
```



# Modifica di un post

```
PUT /posts/{post_id}
```

## Description

Gestisce la modifica di un post tramite il suo id. Un utente per modificare un post deve avere fatto l'accesso. Richiede permessi ["post\_write", "admin"]

## Body parameter

Dati del post

*Name* : body

*Flags* : optional

| Name                                     | Schema                             |
|--|------------------------------------|
| <b>post_id</b><br><i>required</i>        | <a href="#">ObjectId</a>           |
| <b>content</b><br><i>optional</i>        | string                             |
| <b>topic</b><br><i>optional</i>          | string                             |
| <b>tag_users_id</b><br><i>optional</i>   | < <a href="#">ObjectId</a> > array |
| <b>tag_animals_id</b><br><i>optional</i> | < <a href="#">ObjectId</a> > array |

## Responses

| HTTP Code | Description             | Schema                                 |
|-----------|-------------------------|--|
| 200       | Modificato con successo | No Content                             |
| 400       | Input malformato        | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato         | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti       | <a href="#">ErrorMessage</a>           |
| 404       | Non trovato             | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error   | No Content                             |

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Post

# Cancellazione di un post

```
DELETE /posts/{post_id}
```

## Description

Gestisce la cancellazione di un post tramite il suo id. Un utente per cancellare un post deve avere fatto l'accesso. Richiede permessi ["post\_write", "admin"]

## Body parameter

Id del post

*Name* : post\_id  
*Flags* : required  
*Type* : [ObjectId](#)

## Responses

| HTTP Code | Description             | Schema                                 |
|-----------|-------------------------|--|
| 200       | Cancellato con successo | No Content                             |
| 400       | Input malformato        | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato         | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti       | <a href="#">ErrorMessage</a>           |
| 404       | Non trovato             | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error   | No Content                             |

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Post

## Example HTTP request

Request body

```
"61338f5b47bfc65136b5de30"
```

# Creazione di un nuovo commento

```
POST /posts/{post_id}/comments/
```

## Description

Gestisce l'inserimento di un commento sotto un post. Un utente per pubblicare un commento deve avere fatto l'accesso. Richiede permessi ["comment\_write", "admin"]

## Body parameter

Dati del post

*Name* : body

*Flags* : optional

| Name                       | Schema                   |
|----------------------------|--------------------------|
| post_id<br><i>required</i> | <a href="#">ObjectId</a> |
| content<br><i>required</i> | string                   |

## Responses

| HTTP Code | Description           | Schema                                 |
|-----------|-----------------------|--|
| 201       | Creato con successo   | No Content                             |
| 400       | Input malformato      | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato       | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti     | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error | No Content                             |

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Commento

# Ricerca dei commenti

```
GET /posts/{post_id}/comments/
```

## Description

Gestisce la ricerca dei commenti di un post tramite il suo id. Un utente per ricercare dei commenti deve avere fatto l'accesso. Richiede permessi ["comment\_write", "admin"]

## Body parameter

Id del post

*Name* : post\_id  
*Flags* : required  
*Type* : [ObjectId](#)

## Responses

| HTTP Code | Description           | Schema                                 |
|-----------|-----------------------|--|
| 200       | Trovato con successo  | < <a href="#">Response 200</a> > array |
| 400       | Input malformato      | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato       | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti     | <a href="#">ErrorMessage</a>           |
| 404       | Non trovato           | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error | No Content                             |

### Response 200

| Name                            | Schema                   |
|---------------------------------|--------------------------|
| user_id<br><i>required</i>      | <a href="#">ObjectId</a> |
| content<br><i>required</i>      | string                   |
| creationDate<br><i>required</i> | string (date-time)       |
| updateDate<br><i>optional</i>   | string (date-time)       |

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Commento

## Example HTTP request

### Request body

```
"61338f5b47bfc65136b5de30"
```

## Ricerca di un commento

```
GET /posts/{post_id}/comments/{comment_index}
```

### Description

Gestisce la ricerca di un commento di un post tramite l'id del post e l'indice del commento nell'array. Un utente per ricercare dei commenti deve avere fatto l'accesso. Richiede permessi ["comment\_write", "admin"]

### Body parameter

Dati del commento

*Name* : body

*Flags* : optional

| Name  | Schema                   |
|---|--------------------------|
| <code>post_id</code><br><i>required</i>       | <a href="#">ObjectId</a> |
| <code>comment_index</code><br><i>required</i> | number (int)             |

### Responses

| HTTP Code | Description           | Schema                                 |
|-----------|-----------------------|--|
| 200       | Trovato con successo  | <a href="#">Response 200</a>           |
| 400       | Input malformato      | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato       | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti     | <a href="#">ErrorMessage</a>           |
| 404       | Non trovato           | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error | No Content                             |

#### Response 200

| Name   | Schema                   |
|--|--------------------------|
| <code>user_id</code><br><i>required</i>      | <a href="#">ObjectId</a> |
| <code>content</code><br><i>required</i>      | string                   |
| <code>creationDate</code><br><i>required</i> | string (date-time)       |
| <code>updateDate</code><br><i>optional</i>   | string (date-time)       |

### Consumes

- `application/json`

## Produces

- `application/json`

## Tags

- Commento

# Modifica di un commento

```
PUT /posts/{post_id}/comments/{comment_index}
```

## Description

Gestisce la modifica di un commento di un post tramite l'id del post e l'indice del commento nell'array. Un utente per modificare un commento deve avere fatto l'accesso. Richiede permessi ["comment\_write", "admin"]

## Body parameter

Dati del commento

*Name* : body

*Flags* : optional

| Name  | Schema                   |
|---|--------------------------|
| <code>post_id</code><br><i>required</i>       | <a href="#">ObjectId</a> |
| <code>comment_index</code><br><i>required</i> | number (int)             |
| <code>content</code><br><i>optional</i>       | string                   |

## Responses

| HTTP Code | Description             | Schema                                 |
|-----------|-------------------------|--|
| 200       | Modificato con successo | No Content                             |
| 400       | Input malformato        | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato         | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti       | <a href="#">ErrorMessage</a>           |
| 404       | Non trovato             | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error   | No Content                             |

## Consumes

- `application/json`

## Produces

- `application/json`

## Tags

- Commento



# Cancellazione di un commento

```
DELETE /posts/{post_id}/comments/{comment_index}
```

## Description

Gestisce la cancellazione di un commento di un post tramite l'id del post e l'indice del commento nell'array. Un utente per cancellare un commento deve avere fatto l'accesso. Richiede permessi ["comment\_write", "admin"]

## Body parameter

Dati del commento

*Name* : body

*Flags* : optional

| Name                             | Schema                   |
|----------------------------------|--------------------------|
| post_id<br><i>required</i>       | <a href="#">ObjectId</a> |
| comment_index<br><i>required</i> | number (int)             |

## Responses

| HTTP Code | Description             | Schema                                 |
|-----------|-------------------------|--|
| 200       | Cancellato con successo | No Content                             |
| 400       | Input malformato        | <a href="#">ValidationErrorMessage</a> |
| 401       | Non autenticato         | <a href="#">ErrorMessage</a>           |
| 403       | Permessi mancanti       | <a href="#">ErrorMessage</a>           |
| 404       | Non trovato             | <a href="#">ErrorMessage</a>           |
| 500       | Internal server error   | No Content                             |

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Commento

# Definitions

## Topic

| Name                                 | Description  | Schema        |
|--------------------------------------|--|---------------|
| <code>name</code><br><i>optional</i> |  | string        |
| <code>icon</code><br><i>optional</i> | Pattern : <code>"^([A-Za-z0-9+/{4})*([A-Za-z0-9+/{2}== [A-Za-z0-9+/{3}=])?\$"</code> | string (byte) |

## Post

| Name   | Schema                                |
|--|---------------------------------------|
| <code>user_id</code><br><i>required</i>        | <code>ObjectId</code>                 |
| <code>content</code><br><i>required</i>        | string                                |
| <code>comments</code><br><i>optional</i>       | <code>&lt; comments &gt;</code> array |
| <code>topic</code><br><i>optional</i>          | string                                |
| <code>tag_users_id</code><br><i>optional</i>   | <code>&lt; ObjectId &gt;</code> array |
| <code>tag_animals_id</code><br><i>optional</i> | <code>&lt; ObjectId &gt;</code> array |
| <code>creationDate</code><br><i>required</i>   | string (date-time)                    |

### comments

| Name   | Schema                |
|--|-----------------------|
| <code>user_id</code><br><i>required</i>      | <code>ObjectId</code> |
| <code>content</code><br><i>required</i>      | string                |
| <code>creationDate</code><br><i>required</i> | string (date-time)    |
| <code>updateDate</code><br><i>optional</i>   | string (date-time)    |

## ObjectId

Type : string (ObjectId)

## ValidationErrorMessage

Type : `< FieldErrorMessage >` array

## FieldErrorMessage

| Name                                    | Description              | Schema |
|---|--------------------------|--------|
| <code>field</code><br><i>required</i>   | Il campo errato          | string |
| <code>message</code><br><i>optional</i> | Informazioni sull'errore | string |

## ErrorMessage

| Name                                    | Description              | Schema |
|---|--------------------------|--------|
| <code>message</code><br><i>optional</i> | Informazioni sull'errore | string |