

# Post

## Overview

Endpoint per le operazioni sui post

## Version information

*Version* : 1.0.0

## Paths

# Ricerca dei post secondo determinati criteri

GET /posts/

## Description

Gestisce la ricerca dei post dati alcuni criteri. Gestisce la paginazione.

Di default ordina in senso decrescente di data.

## Parameters

Type	Name	Description	Schema
Query	<code>page_size</code> <i>required</i>	Numero di post per pagina	number (int)
Query	<code>page_number</code> <i>required</i>	Numero di pagina	number (int)
Query	<code>username</code> <i>optional</i>	Username dell'autore	< string > array
Query	<code>topic</code> <i>optional</i>	Topic del post	string
Query	<code>oldest</code> <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>
500	Internal server error	No Content

## Produces

- `application/json`

## Tags

- Post

# 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* : required

Name	Schema
content <i>required</i>	string
topic <i>required</i>	string
tag_users <i>optional</i>	< string > array
tag_animals_id <i>optional</i>	< <a href="#">ObjectId</a> > array

## Responses

HTTP Code	Description	Schema
201	Creato con successo Headers : <b>location</b> (string) : Endpoint da cui è possibile accedere al post.	<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	Topic inesistente	<a href="#">ErrorMessage</a>
500	Internal server error	No Content

## Consumes

- **application/json**

## Produces

- **application/json**

## Tags

- Post

# 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"] ]

## Parameters

Type	Name	Description	Schema
Path	<code>post_id</code> <i>required</i>	Id del post	string (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

## Produces

- `application/json`

## Tags

- Post

# 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"] ]

## Parameters

Type	Name	Description	Schema
Path	<code>post_id</code> <i>required</i>	Id del post da modificare	string (ObjectId)

## Body parameter

Dati del post

*Name* : body

*Flags* : required

Name	Schema
<code>content</code> <i>optional</i>	string
<code>topic</code> <i>optional</i>	string
<code>tag_users</code> <i>optional</i>	< string > array
<code>tag_animals_id</code> <i>optional</i>	< <a href="#">ObjectId</a> > array

## Responses

HTTP Code	Description	Schema
200	Modificato 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	Topic o post 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"] ]

## Parameters

Type	Name	Description	Schema
Path	<code>post_id</code> <i>required</i>	Id del post da cancellare	string (ObjectId)

## Responses

HTTP Code	Description	Schema
204	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

## Produces

- `application/json`

## Tags

- Post



# Ricerca dei commenti

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

## Description

Gestisce la ricerca con paginazione dei commenti di un post tramite il suo id.

## Parameters

Type	Name	Description	Schema
Path	<code>post_id</code> <i>required</i>	Id del post	string (ObjectId)
Query	<code>page_size</code> <i>required</i>	Numero di commenti per pagina	number (int)
Query	<code>page_number</code> <i>required</i>	Numero di pagina	number (int)

## Responses

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

## Produces

- `application/json`

## Tags

- Commento

# 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"] ]

## Parameters

Type	Name	Description	Schema
Path	<code>post_id</code> <i>required</i>	Id del post	string (ObjectId)

## Body parameter

Dati del commento

*Name* : body

*Flags* : required

Name	Schema
<code>content</code> <i>required</i>	string

## Responses

HTTP Code	Description	Schema
201	Creato con successo Headers : <code>location</code> (string) : Endpoint da cui è possibile accedere al commento.	<a href="#">Comment</a>
400	Input malformato	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi mancanti	<a href="#">ErrorMessage</a>
404	Post inesistente	<a href="#">ErrorMessage</a>
500	Internal server error	No Content

## Consumes

- `application/json`

## Produces

- `application/json`

## Tags

- Commento

## 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.

### Parameters

Type	Name	Description	Schema
Path	<code>post_id</code> <i>required</i>	Id del post	string (ObjectId)
Path	<code>comment_index</code> <i>required</i>	Indice del commento all'interno del post	number (int)

### Responses

HTTP Code	Description	Schema
200	Trovato con successo	<a href="#">Comment</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

### 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"] ]

## Parameters

Type	Name	Description	Schema
Path	<code>post_id</code> <i>required</i>	Id del post	string (ObjectId)
Path	<code>comment_index</code> <i>required</i>	Indice del commento all'interno del post	number (int)

## Body parameter

Dati del commento

*Name* : body

*Flags* : required

Name	Schema
<code>content</code> <i>optional</i>	string

## Responses

HTTP Code	Description	Schema
200	Modificato con successo	<a href="#">Comment</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

- 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"]

## Parameters

Type	Name	Description	Schema
Path	<code>post_id</code> <i>required</i>	Id del post	string (ObjectId)
Path	<code>comment_index</code> <i>required</i>	Indice del commento all'interno del post	number (int)

## Responses

HTTP Code	Description	Schema
204	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

## Tags

- Commento

# 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



# Inserimento di topic

POST /posts/topics/

## Description

Creazione di un nuovo topic.

Richiede permessi [ [] , [{"admin"}] ]

## Body parameter

Dati del topic

*Name* : body

*Flags* : required

*Type* : [Topic](#)

## Responses

HTTP Code	Description	Schema
201	Creato con successo	<a href="#">Topic</a>
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](#)

# Aggiornamento di topic

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

## Description

Aggiornamento di un topic.

Richiede permessi [ [] , [{"admin"}] ]

## Parameters

Type	Name	Description	Schema
Path	<code>topic</code> <i>required</i>	Nome del topic da aggiornare	string

## Body parameter

Dati da aggiornare del topic

*Name* : body

*Flags* : required

Name	Schema
<code>name</code> <i>optional</i>	string
<code>icon</code> <i>optional</i>	string (base64)

## Responses

HTTP Code	Description	Schema
200	Aggiornato con successo	<a href="#">Topic</a>
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	Description	Schema
Path	topic <i>required</i>	Nome del topic da cancellare	string

## Responses

HTTP Code	Description	Schema
204	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`

## Tags

- Topic

# Definitions

## Topic

Name	Schema
name <i>required</i>	string
icon <i>optional</i>	string (base64)

## Post

Name	Description	Schema
<b>id</b> <i>required</i>		<a href="#">ObjectId</a>
<b>author</b> <i>required</i>	Username dell'autore	string
<b>content</b> <i>required</i>		string
<b>topic</b> <i>required</i>		string
<b>tag_users</b> <i>required</i>		< string > array
<b>tag_animals_id</b> <i>required</i>		< <a href="#">ObjectId</a> > array
<b>creationDate</b> <i>required</i>		string (date-time)

## Comment

Name	Description	Schema
<b>index</b> <i>required</i>	Indice del commento rispetto al post a cui appartiene	number (int)
<b>author</b> <i>required</i>		string
<b>content</b> <i>required</i>		string
<b>creationDate</b> <i>required</i>		string (date-time)
<b>updateDate</b> <i>required</i>		string (date-time)

## ValidationErrorMessage

Type : < [FieldErrorMessage](#) > array

## FieldErrorMessage

Name	Description	Schema
<b>field</b> <i>required</i>	Il campo errato	string
<b>message</b> <i>optional</i>	Informazioni sull'errore	string

## ErrorMessage

Name	Description	Schema
<b>message</b> <i>optional</i>	Informazioni sull'errore	string

## ObjectId

*Type* : string (ObjectId)