

Teammee API

Implementazione API

Creazione utente di sistema

Per interagire con le API server è opportuno creare un utente di sistema che abbia autorizzazione per le azioni che vogliamo andare ad implementare.

Creazione Client API

Una volta creato l'utente di sistema andando nella sua scheda profilo è necessario attivare una client API per avere le credenziali per ottenere il rilascio di un token Oauth2.

Nel campo "Redirect URI" andrà inserita la url dell'istanza teammee

Andranno poi selezionati gli scopi secondo le necessità implementative

Scopo	Descrizione
add_post	Permette l'inserimento di un post
add_document	Permette l'inserimento di un documento
add_event	Permette l'inserimento di un evento
delete_post	Permette la cancellazione di un post
delete_document	Permette la cancellazione di un documento
delete_event	Permette la cancellazione di un evento
modify_post	Permette la modifica di un post
modify_document	Permette la modifica di un documento
modify_event	Permette la modifica di un evento
read_post	Permette la lettura di un singolo post

read_document	Permette la lettura di un singolo documento
read_event	Permette la lettura di un singolo evento
read_livestream	Permette la lettura di una livestream (home, utente, gruppo)
impersonate_any	Permette al client API di interagire come se fosse un altro utente (specificandone l'id come descritto nella reference)

Richiesta token autenticazione

Per ottenere un token di autenticazione valido è necessario fare una richiesta secondo lo standard OAuth2 Client Credential (<http://tools.ietf.org/html/rfc6749#section-3.2>) all'endpoint <https://nomeistanza.teammee.com/oauth/token>.

La richiesta deve essere effettuata in POST con i seguenti parametri

Parametro	Valore Richiesto	Descrizione
grant_type	client_credentials	Richiede il rilascio di un token
client_id	Client ID della Client API	E' il valore del campo Client ID della Client API precedentemente generata
client_secret	Client Secret della Client API	E' il valore del campo Client Secret della Client API precedentemente generata
scope	Scopi separati da spazi	Sono gli scopi della Client API precedentemente generata separati da spazi

Il server (in caso di corretta richiesta) risponderà con un oggetto JSON del tipo:

```
{
  "access_token":"c0b9c9b2826f1d83f4be08018ea7fbcc979413e",
  "expires_in":3600,
  "token_type":"Bearer",
  "scope":"add_post delete_post read_post"
}
```

Chiamata API

Abbiamo usato una convenzione per quanto riguarda i metodi API: il metodo HTTP determina il comportamento dell'end-point.

PUT (multipart)-> API d'inserimento.

POST -> API di modifica.

DELETE -> API di cancellazione.

GET -> API di lettura.

I dettagli sulle funzionalità implementate ed i parametri valorizzati sono documentati nella Reference Guide.

Esempio d'inserimento post

Richiesta HTTP:

PUT /API/Post/ HTTP/1.1

Authorization: Bearer 2ada10bffd4095a1ea9aabc7e0eb9bf2981ffb3

Content-Type: multipart/form-data; boundary=__X_BOUNDARY__

Host: nomeistanza.teammee.com

Connection: close

Content-Length: 205

--__X_BOUNDARY__

Content-Disposition: form-data; name="payload.json"

{"id_group":1,"isPublic":null,"message":"<p>test da api su gruppo</p>","user_list":null}

--__X_BOUNDARY__--

Risposta:

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

Transfer-Encoding: chunked

Connection: close

{"result":"ok","id":"2"}

Esempio codice PHP

```
<?php

// Get cURL resource
$ch = curl_init();
// Set url
curl_setopt($ch, CURLOPT_URL, 'https://nomeistanza.teammee.com.locals/API/Post/');

// Set method
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT');

// Set options
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

// Set headers
curl_setopt($ch, CURLOPT_HTTPHEADER, [
    "Authorization: Bearer 2ada10bffd4095a1ea9aabc7e0eb9bf2981ffb3",
]);

// Create body
$body = [
    "payload.json" => '{"id_group":1,"isPublic":null,"message":"<p>test da api su gruppo<\p>","user_list":null}',
];

// Set body
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $body);

// Send the request & save response to $resp
$resp = curl_exec($ch);
if(!$resp) {
    die("Error: " . curl_error($ch) . " - Code: " . curl_errno($ch));
} else {
    echo "Response HTTP Status Code : " . curl_getinfo($ch, CURLINFO_HTTP_CODE);
    echo "\nResponse HTTP Body : " . $resp;
}

// Close request to clear up some resources
curl_close($ch); ?>
```