

Api MeetMe

- [Login](#)
 - [Login tramite credenziali](#)
 - [Login tramite token](#)
- [Richieste GET](#)
 - [Parametri delle richieste GET](#)
 - [Filtri per azioni index](#)
 - [Relazioni per azioni index](#)
 - [Parametri opzionali](#)
 - [Page e size](#)
 - [Esempi completi di richieste](#)
- [Richieste POST](#)

Tutte le api di MeetMe sono disponibili previa autenticazione.

A tutte le richieste, viene restituito un json con due indici: "success"=true e "data" contenente i dati.

Login

Il login può essere eseguito tramite la normale procedura di login, nel caso non si abbia a disposizione un token valido, oppure si può eseguire un qualunque richiesta inserendo l'header authorization con l'indice "token". Il token si può recuperare eseguendo una procedura di login con credenziali.

Il token ha sempre una lunghezza di 40 caratteri.

Login tramite credenziali

Il login viene eseguito tramite POST alla url "user/login-api" e con i parametri LoginForm[email] e LoginForm[password].

Nel caso di login errato viene restituito un json con indice "success" = false.

Risposta Server Login con credenziali

```
{
  "success": true,
  "data": {
    "id": 1,
    "name": "John",
    "surname": "Doe",
    "email": "user@example.org",
    "username": "marko@infordata.it",
    "token": "9xx50xx4xx0xx72xx29a89xxdcaxx9cxxdxx67xx",
    "creation_date": "2015-06-03 00:00:00",
    "update_date": "2018-04-11 17:50:24",
    "active": 1,
    "deleted": 0,
    "number": null,
    "bith_date": "0000-00-00",
    "tax_code": "",
    "pin": null,
    "hybridauth_provider_name": "",
    "hybridauth_provider_uid": "",
    "qualification": null,
    "customer_id": 19,
    "tipo": null,
    "language": "it",
    "default_event": 444,
    "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "contact_id": null,
    "url_redirect": "/dashboard",
    "logo": "ah7PAZIEVwHDr60ZeSs8Is7sV8NLPOLK.png",
    "notify": 1,
    "frequency": 24,
    "use_chat": 1,
    "auth_key": "",
    "registration_ip": "",
    "confirmed_at": null,
    "blocked_at": null,
    "hash": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "type": 1
  }
}
```

Login tramite token

Per eseguire il login tramite token basta accedere direttamente alla route interessata inserendo però l'header authorization con valore "token=xxxxxxxxxxxxxx".

Richieste GET

Di seguito sono elencate le entità per le quali è disponibile una chiamata GET:

Entità	Route di base	Azione	Parametri opzionali	Dati restituiti
Badge	badge	index		

BadgeTemplate	badge-template	index		
ContactForm	contact-form	index		Ritorna i dati del contact form più un elemento "form_fields" in cui sono inseriti i campi del form. La chiave è il nome stesso del campo e all'interno ci sono altre due chiavi: type e order. Esempio <pre> "form_fields": { "lista": { "type": "text", "order": 0 }, "level": { "type": "dropdown", "data": { "Ospite": "Ospite", "Staff": "Staff", "Giornalista": "Giornalista", "Tecnico": "Tecnico", "test": "test" }, "order": 4 } ... } </pre> Type può essere: checkbox, date, dropdown, textarea, text.
Contatti	contact	index	page size	
Eventi	event	get-app-events		Ritorna tutti gli eventi figli attivi.
	event	get-options? event_id=event_id		Ritorna tutte le opzioni per l'evento specificato.
Liste	list	index		
Luoghi	hall	index		
Mailing	mailing	index		
Partecipanti	participant	index	page size	
ParticipantSession	participant-session	index		
Postazioni	location	index		
Timbratori	timestamp	index		&relations[]=location &TimestampSearch[account_id]=xxxx
Traduzioni	translation	index		Ritorna un array con le traduzioni dei campi dei modelli previsti. I modelli previsti sono: Partecipanti, Contatti. La struttura del messaggio è la seguente: <pre> "Contact": { "id": "ID", "title": "Titolo", "name": "Nome" ... }, "Participant": { "id": "ID", "contact_id": "Contatto ID", "event_id": "ID Evento", ... } </pre>

ParticipantAccount	participant-account		account_id	
registrations	registrations	online	mac= (mac del timestamp) timbratura=status, date,time,serial, 1,0,00 (status 0-5) (date yyyy-mm-dd) (time hh:mm:ss)	https://app.meetme.pro/registrations/online?mac=000424BBA61B6FEF&timbratura=2,2020-12-03,10:53:58,9B5FEAC3,1,0,00

Parametri delle richieste GET

Le richieste GET devono specificare il formato di risposta desiderato. Il parametro deve avere indice "type" e deve essere passato in GET. Per ora è disponibile solo il formato json.

SOLO le azioni "index" permettono di utilizzare i filtri e le relazioni (spiegate di seguito).

Filtri per azioni index

Il filtro è un parametro GET che specifica il campo su cui applicare il filtro e il suo valore.

Per comporre il filtro, si usa il seguente formato: {route di base}Search[{campo}]=valore.

IMPORTANTE: la route di base deve essere con la prima lettera maiuscola.

Esempi:

ListSearch[id]=6 ritorna solo la lista con id = 6

ContactSearch[name]=paolo&ContactSearch[surname]=rossi ritorna solo i contatti con nome "paolo" e cognome "rossi".

Relazioni per azioni index

È possibile specificare le relazioni da ottenere insieme al modello richiesto. Il parametro deve avere indice "relations" e deve essere passato in GET.

Esempio:

relations[]=customer ritorna il modello cliente dell'entità ricercata

relations[]=customer&relations[]=contacts ritorna il modello cliente e i modelli contatti dell'entità ricercata.

Parametri opzionali

Page e size

Questi due parametri sono gestiti in coppia e permettono la gestione della paginazione. Sono parametri GET e hanno rispettivamente chiavi "page" e "size".

"page" indica il numero di pagina richiesto mentre "size" il numero di elementi per pagina.

Esempi completi di richieste

/list/index?accessToken=xxxxxxxxxxxxxxxxxxxx&type=json

/list/index?accessToken=xxxxxxxxxxxxxxxxxxxx&type=json&ListSearch%5Bid%5D=6

/list/index?accessToken=xxxxxxxxxxxxxxxxxxxx&type=json&ListSearch%5Bid%5D=6&relations[]=customer

/list/index?accessToken=xxxxxxxxxxxxxxxxxxxx&type=json&ListSearch%5Bid%5D=6&relations[]=customer&relations[]=contacts

Richieste POST

Le richieste POST devono essere inviate con due header http:

Content-Type: application/x-www-form-urlencoded

Authorization: token=xxxxxxxxxxxxxxxxxxxxxxxxxx

Di seguito sono elencate le url disponibili:

URL	Parametri	Azione eseguita	
customer /change- default-event	default_event_id (int)	Imposta l'evento di default per l'utente attualmente loggato.	
registrations /register	contact_id (int) event_id (int) registration_time (int) non obbligatorio timestamper_id (int) non obbligatorio direction (int) non obbligatorio session_id (int) non obbligatorio account_id (int) non obbligatorio	Accredita il contatto per l'evento. Utilizzato per impostare il registration_time del partecipante. Quindi questo parametro deve essere inviato solo nel caso si esegua un accredito quando il server non è raggiungibile. Il parametro corrisponde ai secondi dalla Unix Epoch. Utilizzato per creare una timbratura sul timbratore specificato. Se non viene specificato, si utilizza il timbratore associato all'evento, se esiste. Utilizzato per impostare il verso della timbratura. Se non specificato, viene messo come verso di default "ingresso". Utilizzato per il controllo della sessione e per la timbratura. Per il controllo della sessione, viene utilizzato per sapere se il partecipante può accedere a quella sessione. Per la timbratura, imposta la timbratura per quella sessione. Se non specificato, la timbratura non viene associata alla sessione.	
registrations /add? id=event_id& type=json	event_id (int) in GET type=json in GET Contact[campo] in POST registration_time (int) non obbligatorio in POST timestamper_id (int) non obbligatorio in POST direction (int) non obbligatorio session_id (int) non obbligatorio	Crea un contatto, il partecipante e lo accredita. I dati del contatto devono essere passati in POST e nessun dato è obbligatorio. I dati devono essere inviati con chiave Contact[campo]. Quindi, per inviare il nome ed il cognome, si invieranno i seguenti dati: Contact[name]=paolo Contact[surname]=rossi Per i dettagli dei campi, guardare registrations/register.	

registration /unregister	event_id (int) POST (obbligatorio) contact_id (int) POST (obbligatorio) return: JSON array(success : true/false message : 'motivazione errore')	Rimuove accreditato del partecipante settando registration_time e registration_code a NULL	
contact /update	id (int) GET (obbligatorio) - id contatto type (string) GET (obbligatorio se si vuole usare via Ajax, valorizzarlo con 'json') Contact[campo] in POST return: JSON array(success : true/false message : 'motivazione errore')	Permette di modificare i campi del contatto via ajax. I dati del contatto devono essere passati in POST e nessun dato è obbligatorio. I dati devono essere inviati con chiave Contact[campo]. Quindi, per inviare il nome ed il cognome, si invieranno i seguenti dati: Contact[name]=paolo Contact[surname]=rossi	
registrations	covid	contact_id , event_id, covid_test_date	example: Api url: https://meetme.andrea.smartforge.it/registrations/covid Api params: {event_id=1001, contact_id=848943, covid_test_date=2021-02-03 11:45:00}