

# Introduzione

---

Questo documento descrive gli risorse che costituiscono l'API offerta dal backend della nostra applicazione web.

1. Versione corrente
2. Protocollo per la comunicazione
3. Host URL
4. Schema
5. Autenticazione
6. Modelli per la documentazione

## Versione corrente

---

Attualmente l'API è alla versione 0.2.0

## Protocollo per la comunicazione

---

Tutti i trasferimenti di dati sono conformi a HTTP/1.1 e tutti gli endpoint richiedono, al momento, http.

## Host URL

---

L'accesso all'API avviene tramite l'URL localhost o 127.0.0.1

## Schema

---

Tramite estensione dell'URL base (descritto sopra) è possibile accedere alle funzionalità offerte dalla API. Tutti i dati sono inviati e restituiti in formato JSON. Il server controlla che tutti i campi richiesti siano presenti nelle richieste; per i campi non richiesti ma possibili (es: nelle PATCH non è obbligatorio modificare tutti i campi), il server controlla il loro formato solo se sono presenti. Se nel body vengono inseriti dei campi non richiesti e non utili, il server li ignora. I codici di errore ritornati dal server dipendono dal tipo di errore, in alcuni casi il messaggio è "fisso" (es: 401-"Unauthorized: user is not a desk"), in altri esso dipende da errori ritornati da componenti interne del server (es: DBMS), in questi casi il messaggio di errore ritornato conterrà una stringa **reason** che conterrà il messaggio di errore generato dal componente interno. Nella tabella sottostante verranno elencati, per ogni tipo di richiesta, alcuni degli errori possibili ma non tutti; saranno presenti solo gli errori più specifici e riguardanti l'integrità dei dati (es: "tavolo già occupato" o "quest'azione è effettuabile solo dai cuochi"). Altri tipi di errori (più generici) potranno essere ritornati nelle response (es: "invalid\_token" o "parse failed").

## Autenticazione

---

## Gestione degli utenti

---

E' possibile ottenere informazioni riguardo il personale del ristorante ed effettuare operazioni di inserimento/modifica/cancellazione.

## Login personale

Titolo	Login
URL	/login
Metodo	GET
Parametri	
Corpo	{ username, password }
Successo	Codice: 200 Contenuto: { error: false, errorMessage: "", token: nuovo_token }
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: {statusCode: 500, error: true, errorMessage: response}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: {statusCode: 500, error: true, errorMessage:"Invalid user"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: {statusCode: 500, error: true, errorMessage:"Invalid password"}
Esempio	/login Body: { username: "prova1", password: "a" }

## Renew del JWT

Titolo	Renew del JWT
URL	/renew
Metodo	GET
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: { error: false, errorMessage: "", token: token_signed }
Esempio	/renew

## Lista del personale

Titolo	Fornisci lista utenti
URL	/users
Metodo	GET
Parametri	Richiesti: role=[ruolo]
Corpo	
Successo	Codice: 200 Contenuto: array di oggetti di tipo: {username, role}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: false, errorMessage: "Unauthorized: user is not a desk"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+ reason }
Esempio	/users?role="waiter"

## Inserimento utente

Titolo	Inserisci un utente
URL	/users
Metodo	POST
Parametri	
Corpo	{username, password, role}
Successo	Codice: 200 Contenuto:{ error: false, errorMessage: "", id: id_utente_inserito }
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errorMessage: "Wrong format"}
Errore	Codice: 409 CONFLICT Contenuto: {statusCode:409, error:true, errorMessage: "User already exists"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+reason }
Esempio	/users Body: { username: "prova1", password: "a", role: "waiter" }

## Rimozione utente

Titolo	Rimozione utente
URL	/users/:username
Metodo	DELETE
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: {error:false, errorMessage:""}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: " + reason }
Esempio	/users?username=prova

## Modifica totale dell'utente

Titolo	Modifica totale utente
URL	/users/:username
Metodo	PUT
Parametri	
Corpo	{ username, password, role }
Successo	Codice: 200 Contenuto: {username, role}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errorMessage: "Wrong format"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: " + reason }
Esempio	/users/1 Body: { username: "ciao1", password: "pwd",
	, role: "waiter" }

## Gestione tavoli

### Lista tavoli

Titolo	Fornire lista tavoli
URL	/tables
Metodo	GET
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: array di oggetti di tipo: {number, max_people, state, associated_ticket}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk or a waiter"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: " + reason }
Esempio	/tables
Note:	Codice di esempio

## Inserimento tavolo

Titolo	Inserire un nuovo tavolo
URL	/tables
Metodo	POST
Parametri	
Corpo	{number, max_people}
Successo	Codice: 200 Contenuto: { number, max_people }
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errorMessage: "Wrong format"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: " + reason }
Esempio	/tables/1 Body: {number:1, max_people:5}
Note:	Codice di esempio

## Tavolo al dettaglio

Titolo	Fornire un tavolo specifico
URL	/tables/:number
Metodo	GET
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: array di oggetti di tipo: {number, max_people, state, associated_ticket}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: false, errormessage: "Unauthorized: user is not a desk or a waiter"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errormessage: "DB error: "+ reason }
Esempio	/tables/1
Note:	Codice di esempio

## Modifica di un attributo di un tavolo

Titolo	Modificare i campi di un tavolo
URL	/tables/:number
Metodo	PATCH
Parametri	
Corpo	{max_people?, status?}
Successo	Codice: 200 Contenuto: {number, max_people, state: state_table, associated_ticket}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errormessage: "Unauthorized: user is not a desk"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errormessage: "Wrong format"}
Esempio	/tables/1 Body: {number:1, max_people: 5 }
Note:	Codice di esempio

## Gestione del menu

### Fornire la lista dei prodotti servibili

Titolo	Fornire la lista dei prodotti
URL	/items
Metodo	GET
Parametri	type=[tipo]
Corpo	
Successo	Codice: 200 Contenuto: array di oggetti di tipo: {name, type, price, required_time, ingredients}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk or a waiter"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: " + reason }
Esempio	/items?type=dish
Note:	Codice di esempio

## Inserire un prodotto

Titolo	Inserire un prodotto specifico
URL	/items
Metodo	POST
Parametri	
Corpo	{name, type, ingredients, description, price}
Successo	Codice: 200 Contenuto:{ error: false, errorMessage: "", id: id_nuovo_item }
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errorMessage: "Wrong format"}
Errore	Codice: 409 CONFLICT Contenuto: {statusCode:409, error:true, errorMessage: "Item already exists"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+reason }
Esempio	/items/1 Body: { name: Spaghetti al pomodoro, type: piatto, ingredients: [{Spaghetti}, {Pomodoro}], description: "Spaghetti al pomodoro, price: 6.00 }
Note:	Codice di esempio

## Fornire un prodotto specifico

Titolo	Fornire un prodotto specifico
URL	/items/:id
Metodo	GET
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: { _id, name, type, price, ingredients, required_time, description }
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk or a waiter" }
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+reason }
Esempio	/items/1
Note:	Codice di esempio

## Modificare un prodotto specifico

Titolo	Modificare un prodotto specifico
URL	/items/:id
Metodo	PUT
Parametri	
Corpo	{name, type, ingredients, description, price}
Successo	Codice: 200 Contenuto: { _id, name, type, price, ingredients, required_time, description }
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk" }
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errorMessage: "Wrong format" }
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+reason }
Esempio	/items/1 Body: {name: Spaghetti al pomodoro, type: piatto, ingredients: [{Spaghetti}, {Pomodoro}], description: "Spaghetti al pomodoro, price: 6.00 }
Note:	Codice di esempio

## Eliminare un prodotto



Titolo	Eliminare un prodotto specifico
URL	/items/:id
Metodo	DELETE
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: {error:false, errorMessage:""}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+reason }
Esempio	/items/1
Note:	Codice di esempio

## Gestione dei ticket (scontrini) e degli ordini

### Inserimento di un ticket

Titolo	Inserimento di un ordine
URL	/tickets
Metodo	POST
Parametri	
Corpo	{waiter, table, people_number}
Successo	Codice: 200 Contenuto: { error: false, errormessage: "", _id: ticket_id }
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errormessage: "Unauthorized: user is not a desk or a waiter"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errormessage: "Wrong format"}
Errore	Codice: 409 CONFLICT Contenuto: {statusCode:409, error:true, errormessage: "Table associated hasn't enough seats"}
Errore	Codice: 409 CONFLICT Contenuto: {statusCode:409, error:true, errormessage: "Table associated doesn't exist"}
Errore	Codice: 409 CONFLICT Contenuto: {statusCode:409, error:true, errormessage: "Ticket already exists"}
Errore	Codice: 409 CONFLICT Contenuto: {statusCode:409, error:true, errormessage: "Table is already taken"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errormessage: "DB error: "+reason }
Esempio	/tickets Body: {id_wai: 1, tab_number: 1, people_number: 2 }

## Recupero info dei tickets

Titolo	Recupero info dei tickets
URL	/tickets
Metodo	GET
Parametri	waiter=[id_cam]&table=[id_tavolo]&state=[stato]
Corpo	
Successo	Codice: 200 Contenuto: [{_id, waiter, table, state, orders, total, start, people_number}]
Errore	Codice: 400 BAD REQUEST Contenuto: {error: true, errorMessage: "The state of orders accepted are ordered, preparation, ready, delivered and all"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+reason }
Esempio	/tickets?waiter=waiter1&table=1

## Recupero info di un ticket specifico

Titolo	Recupero info di un ticket specifico
URL	/tickets/:id
Metodo	GET
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: { _id, waiter, table, state, orders, total, start, people_number }
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk or a waiter" }
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+ reason }
Esempio	/tickets/1
Note:	/tickets/1

## Modifica info di un ticket specifico

Titolo	Modifica info di un ticket specifico
URL	/tickets/:id
Metodo	PATCH
Parametri	
Corpo	{end, state, total}
Successo	Codice: 200 Contenuto: {error:false, errorMessage:""}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errorMessage: "Unauthorized: user is not a desk"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errorMessage: "Wrong format"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+ reason }
Esempio	/tickets/1 Body: {end:timestamp(now), state: "close"}

## Recupero ordini relativi ad un ticket

Titolo	Recupero ordini relativi ad un ticket
URL	/tickets/:id/orders
Metodo	GET
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: [{added, _id, name_item, username_waiter, state, price, type_item, required_time}]
Errore	Codice: 500 INTERNAL SERVER ERROR { statusCode:500, error: true, errorMessage: "DB error: "+ reason }
Esempio	/tickets/1/orders

## Inserimento di un ordine in un ticket

Titolo	Inserimento ordine in un ticket specifico
URL	/tickets/:id/orders
Metodo	POST
Parametri	
Corpo	{name_item, price, added, type_item, required_time}
Successo	Codice: 200 Contenuto: {error:false, errormessage:""}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errormessage: "Unauthorized: user is not a desk or a waiter"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errormessage: "Wrong format"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errormessage: "DB error: "+ reason }
Esempio	/tickets/1/orders Body: { name_item: "Spaghetti", price: 7.00, added: ["doppio pomodoro"], required_item: 12, type: "dish" }

## Modifica di un ordine

Titolo	Modifica ordine
URL	/tickets/:id/orders/:id
Metodo	PATCH
Parametri	
Corpo	{state, username_executer }
Successo	Codice: 200 Contenuto: {error:false, errorMessage:""}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errorMessage: "Wrong format"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errorMessage: "Username_executer not required"}
Errore	Codice: 404 NOT FOUND Contenuto: { statusCode:404, error: true, errorMessage: "Order id not found" }
Errore	Codice: 409 CONFLICT Contenuto: { statusCode:409, error: true, errorMessage: "Conflict, orderd state change not coherent with the regular state changes flow" }
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errorMessage: "DB error: "+ reason }
Errore	Altri errori
Esempio	/tickets/1/orders Body: { state: "pronto" }
Note:	Codice di esempio

## Cancellazione di una comanda

Titolo	Cancellazione comanda
URL	/tickets/:id/orders/:id
Metodo	DELETE
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: {error:false, errormessage:""}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errormessage: "Unauthorized: user is not a waiter"}
Errore	Codice: 404 NOT FOUND Contenuto: { statusCode:404, error: true, errormessage: "Order id not found" }
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errormessage: "DB error: "+reason }
Esempio	/tickets/1/orders
Note:	Codice di esempio

## Report

### Fornire i report

Titolo	Fornire un report
URL	/reports
Metodo	GET
Parametri	start=[date start]&end=[date end]
Corpo	
Successo	Codice: 200 Contenuto: { _id, date, total, total_customerts, total_orders, average_stay, users_reports}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: false, errormessage: "Unauthorized: user is not a desk"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errormessage: "DB error: "+ reason }
Esempio	/report?start=2019-06-06T19:36:59.722Z&end=2019-06-06T19:36:59.722Z

### Inserire un report

<b>Titolo</b>	<b>Inserire un report</b>
URL	/report
Metodo	POST
Parametri	
Corpo	{date, total, total_customerts, total_orders, average_stay, users_reports}
Successo	Codice: 200 Contenuto: { error: false, errormessage: "", id: report_id}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errormessage: "Unauthorized: user is not a desk"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errormessage: "Wrong format"}
Errore	Codice: 409 CONFLICT Contenuto: {statusCode:409, error:true, errormessage: "Report already exists"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errormessage: "DB error: "+reason }
Esempio	/report Body: "date": "2019-06-03T00:00:00.000Z", "total": 5600, "total_customers": 120, "total_orders": {"dish": 350, "beverage": 712 }, "average_stay": 120, "users_reports": { "waiters": [ { "username": "waiter1", "customers_served": 80, "orders_served": 912 }, {"username": "waiter2", "customers_served": 40, "orders_served": 305 }], "bartenders": [{"username": "bartender1", "items_served": 400 }, {"username": "waiter2", "items_served": 700 }], "cookers": [{ "username": "cook1", "items_served": 60},{ "username": "cook2", "items_served": 1110}]}

## Ottenere un report specifico



Titolo	Ottenere un report specifico
URL	/reports/:id
Metodo	GET
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: { _id, date, total, total_orders, total_customers, average_stay, users_reports }
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errormessage: "Unauthorized: user is not a desk" }
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errormessage: "DB error: "+reason }
Esempio	/report/1

## Cancellare un report

Titolo	Cancellare un report
URL	/reports/:id
Metodo	DELETE
Parametri	
Corpo	
Successo	Codice: 200 Contenuto: {error:false, errormessage:""}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errormessage: "Unauthorized: user is not a desk" }
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errormessage: "DB error: "+reason }
Esempio	/report/1

## Modifica di un report

Titolo	Modifica di un report
URL	/reports/:id
Metodo	PUT
Parametri	
Corpo	{date, total, total_orders, total_customers, average_stay, users_reports}
Successo	Codice: 200 Contenuto: {error:false, errormessage:""}
Errore	Codice: 401 UNAUTHORIZED Contenuto: { statusCode:401, error: true, errormessage: "Unauthorized: user is not a desk or a waiter"}
Errore	Codice: 400 BAD REQUEST Contenuto: { statusCode:400, error: true, errormessage: "Wrong format"}
Errore	Codice: 500 INTERNAL SERVER ERROR Contenuto: { statusCode:500, error: true, errormessage: "DB error: "+reason }
Esempio	/report/1 Body: {"total": 5600}

## Modelli per la documentazione

Come modelli per la stesura della documentazione, abbiamo utilizzato le seguenti risorse: - <https://developers.facebook.com/docs/graph-api/using-graph-api/v2.1> ; - <https://developer.github.com/v3/> ; - <https://gist.github.com/iroso/3426278> ;