

Università degli Studi di Messina



Dipartimento di Scienze matematiche e informatiche,
scienze fisiche e scienze della terra

Corso di Laurea Triennale in Informatica

Progetto di

PROGRAMMAZIONE 3

Realizzato da

SERGIO SICLARI

Anno Accademico

2020/2021

SOMMARIO

Capitolo 1: Introduzione	5
Tecnologie.....	5
Capitolo 2: Attori del sistema	6
Specifiche delle operazioni.....	6
Lista degli attori.....	6
Admin	6
Cameriere.....	6
Chef	6
Fornitore	7
Capitolo 3: Database	8
Elenco delle entità	8
Admin	8
Cameriere.....	9
Chef	9
Fornitore	9
Ristorante.....	9
Ambiente	10
Tavolo	10
Prenotazione	10
Ingrediente.....	10
Menù	10
Comanda.....	11
Dettagli_comanda	11
Segnalazione	11
Ordine.....	11
Dettagli_ordine.....	11
Modello E-R.....	12
Modello logico.....	12
Database design	13
Capitolo 4: RESTful API	14
Autenticazione	14
POST Login	14
POST Validate_Token.....	15
Ristorante.....	16
POST Create	16
GET Read.....	16

PUT Update	17
DEL Delete.....	17
GET Read_One.....	18
Ambiente	19
POST Create	19
GET Read.....	19
PUT Update	20
DEL Delete.....	20
GET Read_One.....	21
Tavolo	22
POST Create	22
GET Read.....	22
PUT Update	23
DEL Delete.....	23
GET Read_One.....	24
Prenotazione	25
POST Create	25
GET Read.....	25
PUT Update	26
DEL Delete.....	26
GET Read_One.....	27
Ingrediente.....	28
POST Create	28
GET Read.....	28
PUT Update	29
DEL Delete.....	29
GET Read_One.....	30
Comanda.....	31
POST Create	31
GET Read.....	31
PUT Update	32
DEL Delete.....	32
GET Read_One.....	33
Dettagli_comanda	34
POST Create	34
GET Read.....	34
PUT Update	35

DEL Delete.....	35
GET Read_One.....	36
PUT Ready.....	36
Ordine.....	37
POST Create	37
GET Read.....	37
PUT Update	38
DEL Delete.....	38
GET Read_One.....	39
GET Read_Max	39
Dettagli_ordine.....	40
POST Create	40
GET Read.....	40
PUT Update	41
DEL Delete.....	41
GET Read_One.....	42

CAPITOLO 1: INTRODUZIONE

Si vuole progettare un sito web per il proprietario di uno o più ristoranti, sul quale sarà possibile effettuare diverse operazioni. L'utilizzo di questo sistema è destinato esclusivamente ai dipendenti del ristorante e ai fornitori affiliati; un esempio delle funzionalità che saranno implementate sono:

- gestione delle comande (camerieri)
- gestione del magazzino (chef)
- gestione degli ordini (fornitori)

TECNOLOGIE

Le tecnologie utilizzate per la realizzazione del progetto mirano ad ottimizzare la comunicazione tra client e server, e a prevenire attacchi informatici abbastanza diffusi come ad esempio l'*SQL injection*.

Sono state utilizzate:

- Client-side:
 - HTML
 - CSS
 - jQuery
 - AJAX
- Server-side:
 - PHP

CAPITOLO 2: ATTORI DEL SISTEMA

E' importante definire tutti gli attori che interagiscono con il sistema poiché ognuno di essi ha un set specifico di operazioni che può svolgere.

SPECIFICHE DELLE OPERAZIONI

L'acronimo **CRUD** è riferito alle operazioni effettuate sul database ovvero: *create, read, update e delete*. Di seguito sono elencate le operazioni che il sistema permette agli utenti di eseguire:

- Admin: **CREATE, READ, UPDATE e DELETE**
- Cameriere: **CREATE, READ, UPDATE e DELETE**
- Chef: **CREATE, READ e UPDATE**
- Fornitore: **READ e UPDATE**

LISTA DEGLI ATTORI

Gli utenti che potranno accedere e usufruire del servizio web sono:

- Proprietario
- Camerieri
- Chef
- Fornitori

Ognuno di essi avrà un account, che può essere creato esclusivamente dall'admin, ovvero il proprietario del ristorante. Ogni account è caratterizzato da un **ruolo**, che viene richiesto al momento del log in; e che impedisce ad account con un certo ruolo, di accedere a pagine destinate ad altri utenti. A scopo esemplificativo, l'account di un cameriere non può in alcun modo accedere alla pagina riservata all'admin (scrivendo manualmente l'URL, ad esempio).

ADMIN

L'account di amministratore è riservato al proprietario del ristorante. Questo ruolo dà loro la possibilità di:

- creare, eliminare o aggiornare le informazioni dei **ristoranti**, in questo modo è possibile gestire al meglio un eventuale trasferimento dell'attività o anche un'espansione della catena di ristoranti
- creare o eliminare **account** per camerieri, chef e fornitori
- creare, eliminare o aggiornare le informazioni degli **ambienti** (ovvero le sale o i magazzini), per ogni ristorante
- creare, eliminare o aggiornare le informazioni dei **tavoli**, per ogni ambiente

CAMERIERE

L'account di un cameriere ha la possibilità di:

- creare, eliminare o aggiornare le informazioni delle **comande**
- creare, eliminare o aggiornare le informazioni riguardanti le **prenotazioni**

CHEF

L'account di uno chef ha la possibilità di:

- leggere e aggiornare le informazioni delle **comande**, più precisamente può aggiornare unicamente lo **stato** da 0 ("In preparazione") a 1 ("Pronto")
- creare **segnalazioni**, che verranno visualizzate dai fornitori per poterle accettare come **ordini**

FORNITORE

I fornitori hanno la possibilità di:

- leggere le informazioni sulle **segnalazioni**
- creare **ordini** (gli ordini vengono creati nel momento in cui vengono “presi in carico”)

CAPITOLO 3: DATABASE

Questo capitolo contiene tutte le informazioni relative al database dal quale il sistema trae informazioni.

Si vogliono memorizzare uno o più ristoranti, ognuno dei quali è **caratterizzato** da un codice che li identifica univocamente, il nome, il telefono e l'indirizzo civico.

Per avere una visione completa di ogni ristorante è necessario sottolineare che ogni ristorante è suddiviso in **ambienti**, intesi come spazi all'interno dei ristoranti. E' necessario memorizzare nel database ogni ambiente, di conseguenza l'ambiente sarà immagazzinato con un codice identificativo univoco, un nome ed una tipologia. Quest'ultima è introdotta per identificare facilmente di che tipo di ambiente si tratta (es. bagno, magazzino, sala, cucina).

Gli ambienti con tipologia **sala**, dispongono di diversi **tavoli**; ogni tavolo è caratterizzato da un numero identificativo e dal numero di posti; inoltre ogni tavolo può essere prenotato.

All'interno del database è presente la tabella **menu**, che contiene l'elenco dei prodotti in vendita dal ristorante, questi possono essere sia cibi che bevande. Sono anch'essi caratterizzati da un codice univoco oltre che da un nome ed un prezzo.

Inoltre, si vuole tenere traccia degli **ingredienti** presenti in magazzino che saranno memorizzati con un identificativo univoco, un identificativo di categoria, un identificativo di ambiente, un nome ed una quantità. La **categoria** e l'**ambiente** vengono utilizzate per identificare, rispettivamente, di che ingrediente si tratta e in quale magazzino si trova.

Per quanto riguarda il **personale**, è caratterizzato da un codice univoco che identifica ogni individuo (che differisce tra camerieri e chef), un codice che identifica in quale ristorante lavorano, nome, cognome, email e password (criptata).

ELENCO DELLE ENTITÀ

ADMIN

L'entità *Admin* rappresenta l'amministratore, ovvero il proprietario dei ristoranti. L'attributo più significativo è *id_admin*, che identifica in modo unico l'amministratore. E' necessario qualora il proprietario decidesse di affidare la gestione del ristorante ad un altro individuo, avendo quindi la necessità di creare diversi admin.

Attributi:

- **id_admin**: identifica in modo univoco ciascun admin
- **nome**
- **cognome**
- **email**
- **password**

CAMERIERE

L'entità *Cameriere* contiene informazioni sui membri del personale aventi tale ruolo.

Attributi:

- **id_cameriere:** identifica in modo univoco ciascun cameriere
- **id_ristorante:** specifica in quale ristorante lavora il cameriere
- **nome**
- **cognome**
- **email**
- **password**
- **data_nascita**

CHEF

L'entità *Chef* contiene informazioni sul personale avente tale ruolo.

Attributi:

- **id_chef:** identifica in modo univoco ogni chef
- **id_ristorante:** specifica in quale ristorante lavora lo chef
- **nome**
- **cognome**
- **email**
- **password**
- **data_nascita**

FORNITORE

L'entità *Fornitore* contiene informazioni riguardanti i fornitori affiliati ai ristoranti.

Attributi:

- **id_fornitore:** identifica univocamente ogni fornitore
- **id_genere:** specifica il genere della merce trattata dal fornitore
- **nome**
- **email**
- **password**

RISTORANTE

Ristorante è l'entità che rappresenta la struttura fisica dell'attività commerciale. Quest'ultima è caratterizzata, tra gli altri, dall'attributo **id_ristorante** poiché, oltre che identificarlo univocamente, rappresenta una scelta progettuale per una eventuale espansione dell'attività consentendo di aggiungere nuovi ristoranti al database.

Attributi:

- **id_ristorante:** identifica univocamente il ristorante
- **nome**
- **telefono**
- **indirizzo:** composto dai campi *via*, *numero civico*, *città*

AMBIENTE

L'entità *Ambiente* rappresenta i diversi locali, intesi come stanze, all'interno dei ristoranti.

Attributi:

- **id_ambiente:** identifica univocamente ogni ambiente
- **id_ristorante:** specifica a quale ristorante è collegato l'ambiente
- **id_tipologia:** definisce la tipologia di un ambiente (*sala, magazzino, ecc..*)
- **nome:** nome dell'ambiente (es. *Sala Gialla, Magazzino 1, ecc..*)

TAVOLO

Tavolo è l'entità necessaria ai camerieri per distinguere le comande. Un tavolo non può corrispondere a più di una comanda contemporaneamente, e viene identificato dal numero del tavolo (*id_tavolo*).

Attributi:

- **id_tavolo:** identifica univocamente ogni tavolo
- **id_ambiente:** specifica in quale ambiente si trova il tavolo
- **posti:** indica di quanti posti a sedere dispone il tavolo

PRENOTAZIONE

Prenotazione è un'entità necessaria per riservare un tavolo, se disponibile, ad un cliente.

Attributi:

- **id_prenotazione:** identifica univocamente ogni prenotazione
- **id_tavolo:** indica il tavolo interessato dalla prenotazione
- **nome:** nome del cliente
- **cognome:** cognome del cliente
- **data_prenotazione**

INGREDIENTE

Ingrediente rappresenta gli alimenti utilizzati nei ristoranti.

Attributi:

- **id_ingrediente:** identifica l'ingrediente univocamente
- **id_categoria:** indica a quale categoria appartiene l'ingrediente (es. *carne, pesce, ecc..*)
- **id_ambiente:** indica in quale magazzino si trova l'ingrediente
- **nome**
- **quantita:** specifica quanti pezzi di quell'ingrediente sono presenti in magazzino

MENÙ

Menù è l'entità che contiene tutti i piatti e le bevande che offre il ristorante.

Attributi:

- **id_menu:** identifica univocamente ciascun elemento del menu
- **nome:** indica il nome del piatto o della bevanda
- **prezzo**

COMANDA

Comanda è l'entità che rappresenta ogni ordinazione presa ai tavoli.

Attributi:

- **id_comanda:** identifica la comanda
- **id_tavolo:** indica a quale tavolo si riferisce la comanda

DETTAGLI_COMANDA

Quest'entità contiene le informazioni riguardanti le comande.

Attributi:

- **id_det_comanda:** identifica il record all'interno della tabella
- **id_comanda:** specifica a quale comanda è collegato il record
- **id_menu:** rappresenta il singolo piatto
- **stato:** indica in che stato si trova il piatto, 0 ("In preparazione") o 1 ("Pronto")

SEGNALAZIONE

L'entità *Segnalazione* rappresenta le segnalazioni create dagli chef per indicare la necessità di rifornire il magazzino di un determinato ingrediente.

Attributi:

- **id_segnalazione:** identifica la segnalazione univocamente
- **id_chef:** permette di risalire a chi ha effettuato la segnalazione
- **ingrediente:** nome dell'ingrediente segnalato

ORDINE

Ordine è l'entità che rappresenta gli ordini presi in carico dai fornitori.

Attributi:

- **id_ordine:** identifica univocamente ogni ordine
- **id_fornitore:** specifica l'identificativo del fornitore che ha preso l'ordine in carico
- **stato:** rappresenta lo stato dell'ordine, 0 ("Preso in carico") o 1 ("Consegnato")

DETTAGLI_ORDINE

Analogamente a quanto succede per le comande, contiene le informazioni riguardanti gli ordini.

Attributi:

- **id_det_ordine:** identifica il record all'interno della tabella
- **id_ordine:** specifica a quale ordine è collegato il record
- **id_ingrediente:** Indica quale ingrediente appartiene all'ordine
- **quantita:** specifica quanti pezzi devono essere consegnati

MODELLO E-R

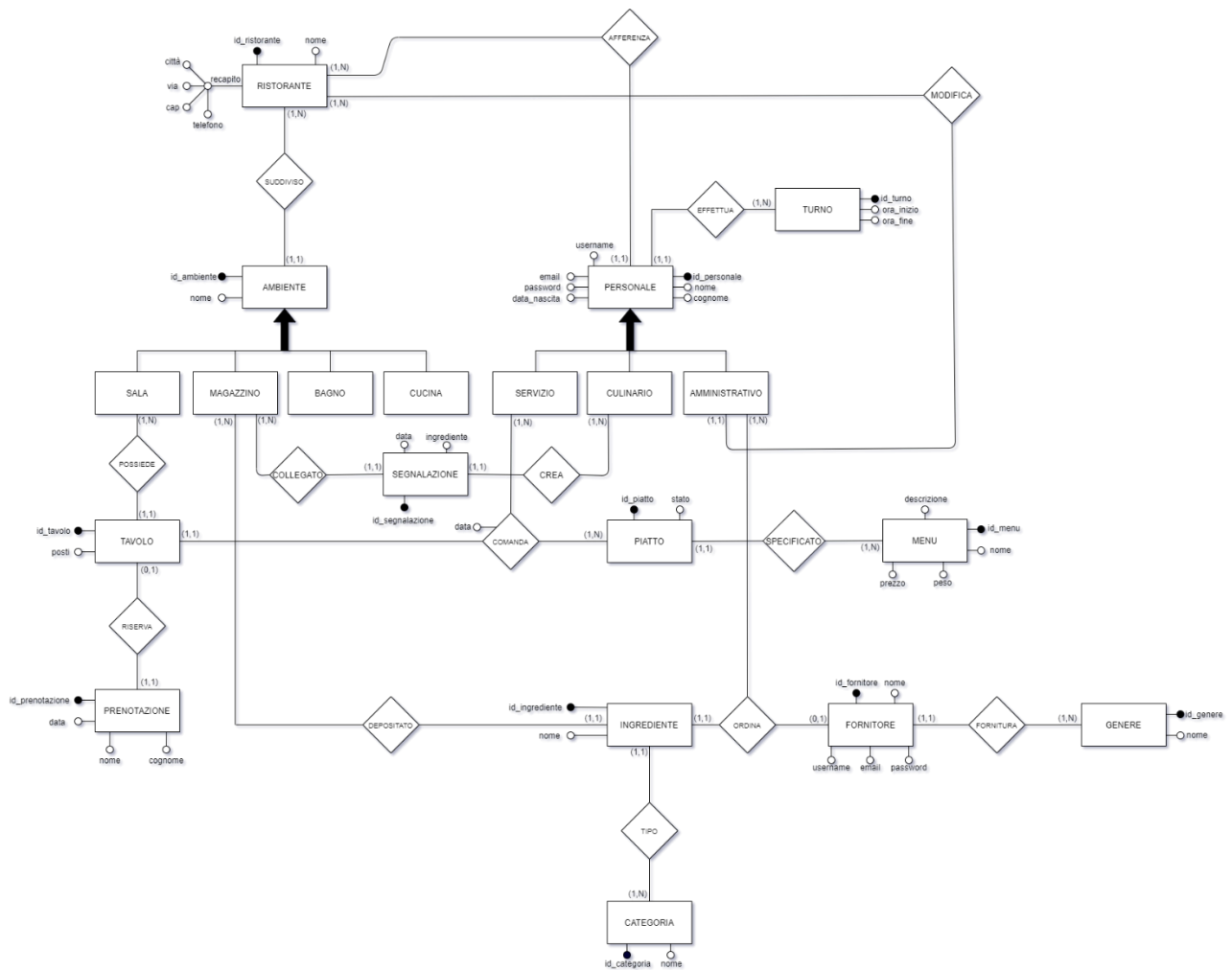


Figura 1: Modello E-R

MODELLO LOGICO

Si definisce, a questo punto, il **modello logico**, necessario per la definizione degli attributi e dei campi di collegamento:

Attributo sottolineato → PRIMARY KEY

Attributo in grassetto → FOREIGN KEY

RISTORANTE(id_ristorante, nome, recapito)

SERVIZIO(id_servizio, **id_ristorante**, **id_turno**, nome, cognome, email, username, password, data_nascita)

AMMINISTRATIVO(id_amministrativo, **id_ristorante**, **id_turno**, nome, cognome, email, username, password, data_nascita)

CULINARIO(id_culinario, **id_ristorante**, **id_turno**, nome, cognome, email, username, password, data_nascita)

TURNO(id_turno, ora_inizio, ora_fine)

AMBIENTE(id_ambiente, **id_ristorante**, **id_tipologia**, nome)

TAVOLO(id_tavolo, id_ambiente, posti)

TIPOLOGIA(id_tipologia, nome)

PRENOTAZIONE(id_prenotazione, id_tavolo, nome, cognome, data)

SEGNALAZIONE(id_segnalazione, id_culinario, id_ambiente, data, ingrediente)

COMANDA(id_comanda, id_tavolo, id_servizio, data)

PIATTO(id_piatto, id_comanda, id_menu, stato)

MENU(id_menu, nome, prezzo, peso)

INGREDIENTE(id_ingrediente, id_categoria, id_ambiente, id_ordine, nome)

CATEGORIA(id_categoria, nome)

ORDINE(id_ordine, id_amministrativo, id_fornitore, data, stato)

FORNITORE(id_fornitore, id_genere, nome, username, email, password)

GENERE(id_genere, nome)

DATABASE DESIGN

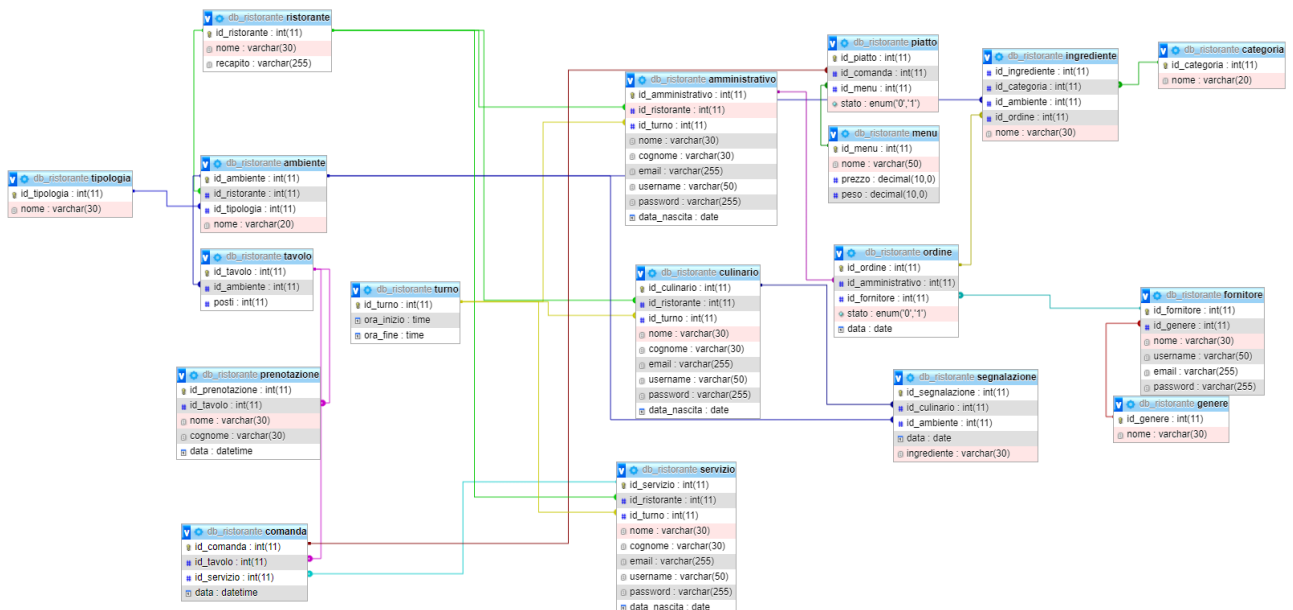


Figura 2: Database design

CAPITOLO 4: RESTFUL API

Il primo passo per la realizzazione di un'architettura di tipo RESTful è quello di mappare i metodi CRUD seguendo una logica ben precisa.

Metodo HTTP	Operazione	Descrizione
POST	Create	Crea una nuova risorsa
GET	Read	Ottiene una risorsa esistente
PUT	Update	Aggiorna una risorsa esistente
DELETE	Delete	Elimina una risorsa

Seguendo la tabella, si può affermare che ad ogni protocollo HTTP viene associata una determinata operazione.

Di seguito vengono mappate tutte le operazioni, seguendo la tabella.

AUTENTICAZIONE

Per accedere alla piattaforma, è stato adottato il sistema **JWT** (*JSON Web Token*). JWT è uno standard che definisce in modo compatto e autonomo per trasmettere in modo sicuro le informazioni tra le parti come un oggetto JSON. Le principali operazioni sono due nell'utilizzo di JWT.

POST LOGIN

Questa operazione permette di inviare al server il ruolo, l'email e la password dell'utente. In caso di esito positivo, il server restituisce un token JWT necessario per completare l'autenticazione.

api/authentication/login_utente.php

Body request (application/json)

Chiave	Richiesto	Descrizione
ruolo	Obbligatorio	Ruolo dell'utente che vuole accedere
email	Obbligatorio	E-mail dell'utente che vuole accedere
password	Obbligatorio	Password associata alla e-mail

Body response (application/json)

200 - OK

Chiave	Valore
message	Login effettuato con successo
jwt	Stringa che identifica univocamente l'accesso
token	Contiene tutte le informazioni riguardanti l'accesso (dati utente, ora di generazione del token, tempo di scadenza del token)

401 – Unauthorized

Chiave	Valore
message	I dati inseriti non sono corretti, riprova

POST VALIDATE_TOKEN

Operazione che invia al server il token JWT generato dopo il login. Essa garantisce o meno l'accesso all'utente in base alla validità del token, che scade dopo un certo lasso di tempo.

api/authentication/validate_token.php

Body request (application/json)

Chiave	Richiesto	Descrizione
jwt	Obbligatorio	Token generato al momento del login

Body response (application/json)

200 – OK

Chiave	Valore
message	Accesso accordato
data[]	Array contenente le informazioni sull'utente

401 – Unauthorized

Chiave	Valore
message	Accesso negato
error	Signature verification failed

RISTORANTE

POST CREATE

Crea un nuovo ristorante caratterizzato dai parametri inseriti.

api/ristorante/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
nome	Obbligatorio	Nome del ristorante
telefono	Obbligatorio	Recapito telefonico del ristorante
indirizzo	Obbligatorio	Indirizzo civico del ristorante

Body response (application/json)

201 – Created

Chiave	Valore
message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
message	Errore nei dati

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ

Legge tutti i ristoranti presenti nel database.

api/ristorante/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_ristorante	Codice identificativo del ristorante
nome	Nome del ristorante
telefono	Recapito telefonico del ristorante
indirizzo	Indirizzo civico del ristorante

404 – Not found

Chiave	Valore
message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di uno specifico ristorante.

api/ristorante/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ristorante	Obbligatorio	Codice identificativo del ristorante
nome	Obbligatorio	Nome del ristorante
telefono	Obbligatorio	Recapito telefonico del ristorante
indirizzo	Obbligatorio	Indirizzo civico del ristorante

Body response (application/json)

200 – OK

Chiave	Valore
message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

DEL DELETE

Elimina un ristorante dal database.

api/ristorante/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ristorante	Obbligatorio	Codice identificativo del ristorante

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di un ristorante specifico, identificato tramite id.

api/ristorante/read_one.php?id_ristorante=<id_ristorante>

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ristorante	Obbligatorio	Codice identificativo del ristorante

Body response (application/json)

200 – OK

Chiave	Valore
id_ristorante	Codice identificativo del ristorante
nome	Nome del ristorante
telefono	Recapito telefonico del ristorante
indirizzo	Indirizzo civico del ristorante

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

AMBIENTE

POST CREATE

Crea un nuovo ambiente caratterizzato dai parametri inseriti.

api/ambiente/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ristorante	Obbligatorio	Codice identificativo del ristorante
id_tipologia	Obbligatorio	Codice identificativo della tipologia
nome	Obbligatorio	Nome dell'ambiente

Body response (application/json)

201 – Created

Chiave	Valore
message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
message	Errore nei dati

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ

Legge tutti gli ambienti presenti nel database.

api/ambiente/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_ambiente	Codice identificativo dell'ambiente
id_ristorante	Codice identificativo del ristorante
id_tipologia	Codice identificativo della tipologia
nome	Nome dell'ambiente
nome_ristorante	Nome del ristorante a cui appartiene l'ambiente
nome_tipologia	Nome della tipologia dell'ambiente

404 – Not found

Chiave	Valore
message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di uno specifico ambiente.

api/ambiente/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ambiente	Obbligatorio	Codice identificativo dell'ambiente
id_ristorante	Obbligatorio	Codice identificativo del ristorante
id_tipologia	Obbligatorio	Codice identificativo della tipologia
nome	Obbligatorio	Nome dell'ambiente

Body response (application/json)

200 – OK

Chiave	Valore
message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

DEL DELETE

Elimina un ambiente dal database.

api/ambiente/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ambiente	Obbligatorio	Codice identificativo dell'ambiente

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di un ambiente specifico, identificato tramite id.

api/ambiente/read_one.php?id_ambiente=<id_ambiente>

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ambiente	Obbligatorio	Codice identificativo dell'ambiente

Body response (application/json)

200 – OK

Chiave	Valore
id_ambiente	Codice identificativo dell'ambiente
nome	Nome dell'ambiente
id_ristorante	Codice identificativo del ristorante
nome_ristorante	Nome del ristorante
id_tipologia	Codice identificativo della tipologia
nome_tipologia	Nome della tipologia

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

TAVOLO

POST CREATE

Crea un nuovo tavolo caratterizzato dai parametri inseriti.

api/tavolo/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ambiente	Obbligatorio	Codice identificativo dell'ambiente
posti	Obbligatorio	Posti a sedere del tavolo

Body response (application/json)

201 – Created

Chiave	Valore
message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
message	Errore nei dati

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ

Legge tutti i tavoli presenti nel database.

api/tavolo/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_tavolo	Codice identificativo del tavolo
id_ambiente	Codice identificativo dell'ambiente
nome_ambiente	Nome dell'ambiente
posti	Posti a sedere del tavolo

404 – Not found

Chiave	Valore
message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di un tavolo specifico.

api/tavolo/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_tavolo	Obbligatorio	Codice identificativo del tavolo
id_ambiente	Obbligatorio	Codice identificativo dell'ambiente
posti	Obbligatorio	Posti a sedere del tavolo

Body response (application/json)

200 – OK

Chiave	Valore
message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

DEL DELETE

Elimina un tavolo dal database.

api/tavolo/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_tavolo	Obbligatorio	Codice identificativo del tavolo

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di un tavolo specifico, identificato tramite id.

```
api/tavolo/read_one.php?id_tavolo=<id_tavolo>
```

Body request (application/json)

Chiave	Richiesto	Descrizione
id_tavolo	Obbligatorio	Codice identificativo del tavolo

Body response (application/json)

200 – OK

Chiave	Valore
id_tavolo	Codice identificativo del tavolo
id_ambiente	Codice identificativo dell'ambiente
nome_ambiente	Nome dell'ambiente
posti	Posti a sedere del tavolo

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

PRENOTAZIONE

POST CREATE

Crea una nuova prenotazione caratterizzata dai parametri inseriti.

api/prenotazione/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_tavolo	Obbligatorio	Codice identificativo del tavolo
nome	Obbligatorio	Nome del cliente
cognome	Obbligatorio	Cognome del cliente
data_prenotazione	Obbligatorio	Data della prenotazione

Body response (application/json)

201 – Created

Chiave	Valore
message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
message	Errore nei dati

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ

Legge tutte le prenotazioni presenti nel database.

api/prenotazione/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_prenotazione	Codice identificativo della prenotazione
id_tavolo	Codice identificativo del tavolo
nome	Nome del cliente
cognome	Cognome del cliente
data_prenotazione	Data della prenotazione

404 – Not found

Chiave	Valore
message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di una specifica prenotazione.

api/prenotazione/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_prenotazione	Obbligatorio	Codice identificativo della prenotazione
id_tavolo	Obbligatorio	Codice identificativo del tavolo
nome	Obbligatorio	Nome del cliente
cognome	Obbligatorio	Cognome del cliente
data_prenotazione	Obbligatorio	Data della prenotazione

Body response (application/json)

200 – OK

Chiave	Valore
message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

DEL DELETE

Elimina una prenotazione dal database.

api/prenotazione/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_prenotazione	Obbligatorio	Codice identificativo della prenotazione

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di una prenotazione specifica, identificata tramite id.

```
api/prenotazione/read_one.php?id_prenotazione=<id_prenotazione>
```

Body request (application/json)

Chiave	Richiesto	Descrizione
id_prenotazione	Obbligatorio	Codice identificativo della prenotazione

Body response (application/json)

200 – OK

Chiave	Valore
id_prenotazione	Codice identificativo della prenotazione
id_tavolo	Codice identificativo del tavolo
nome	Nome del cliente
cognome	Cognome del cliente
data_prenotazione	Data della prenotazione

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

INGREDIENTE

POST CREATE

Crea un nuovo ingrediente caratterizzato dai parametri inseriti.

api/ingrediente/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_categoria	Obbligatorio	Codice identificativo della categoria
id_ambiente	Obbligatorio	Codice identificativo dell'ambiente
nome	Obbligatorio	Nome dell'ingrediente
quantita	Obbligatorio	Quantità presente in magazzino

Body response (application/json)

201 – Created

Chiave	Valore
message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
message	Errore nei dati

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ

Legge tutti gli ingredienti presenti nel database.

api/ingrediente/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_ingrediente	Codice identificativo dell'ingrediente
id_categoria	Codice identificativo della categoria
id_ambiente	Codice identificativo dell'ambiente
nome	Nome dell'ingrediente
quantita	Quantità presente in magazzino

404 – Not found

Chiave	Valore
message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di un ingrediente specifico.

api/ingrediente/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ingrediente	Obbligatorio	Codice identificativo dell'ingrediente
id_categoria	Obbligatorio	Codice identificativo della categoria
id_ambiente	Obbligatorio	Codice identificativo dell'ambiente
nome	Obbligatorio	Nome dell'ingrediente
quantita	Obbligatorio	Quantità presente in magazzino

Body response (application/json)

200 – OK

Chiave	Valore
message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

DEL DELETE

Elimina un ingrediente dal database.

api/ingrediente/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ingrediente	Obbligatorio	Codice identificativo dell'ingrediente

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di un ingrediente specifico, identificato tramite id.

```
api/ingrediente/read_one.php?id_ingrediente=<id_ingrediente>
```

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ingrediente	Obbligatorio	Codice identificativo dell'ingrediente

Body response (application/json)

200 – OK

Chiave	Valore
id_ingrediente	Codice identificativo dell'ingrediente
id_categoria	Codice identificativo della categoria
id_ambiente	Codice identificativo dell'ambiente
nome	Nome dell'ingrediente
quantita	Quantità presente in magazzino

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

COMANDA

POST CREATE

Crea una nuova comanda caratterizzata dai parametri inseriti.

api/comanda/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_tavolo	Obbligatorio	Codice identificativo del tavolo

Body response (application/json)

201 – Created

Chiave	Valore
message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
message	Errore nei dati

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ

Legge tutte le comande presenti nel database.

api/comanda/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_comanda	Codice identificativo della comanda
id_tavolo	Codice identificativo del tavolo
data_creazione	Data di creazione della comanda
ultima_modifica	Data dell'ultima modifica al record

404 – Not found

Chiave	Valore
message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di una specifica comanda.

api/comanda/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_comanda	Obbligatorio	Codice identificativo della comanda
id_tavolo	Obbligatorio	Codice identificativo del tavolo

Body response (application/json)

200 – OK

Chiave	Valore
message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

DEL DELETE

Elimina una comanda dal database.

api/comanda/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_comanda	Obbligatorio	Codice identificativo della comanda

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di una specifica comanda, identificata tramite id.

```
api/comanda/read_one.php?id_comanda=<id_comanda>
```

Body request (application/json)

Chiave	Richiesto	Descrizione
id_comanda	Obbligatorio	Codice identificativo della comanda

Body response (application/json)

200 – OK

Chiave	Valore
id_comanda	Codice identificativo della comanda
id_tavolo	Codice identificativo del tavolo
data_creazione	Data di creazione della comanda
ultima_modifica	Data dell'ultima modifica al record

404 – Not found

Chiave	Valore
Message	L'elemento cercato non è presente nel database

DETTAGLI_COMANDA

POST CREATE

Crea un record nella tabella dettagli_comanda caratterizzato dai parametri passati.

api/dettagli_comanda/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_comanda	Obbligatorio	Codice identificativo della comanda
id_menu	Obbligatorio	Codice identificativo dell'elemento del menu

Body response (application/json)

201 – Created

Chiave	Valore
Message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
Message	Errore nei dati

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

GET READ

Legge tutte le informazioni della tabella dettagli_comanda.

api/dettagli_comanda/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_det_comanda	Codice identificativo del record nella tabella dettagli_comanda
id_comanda	Codice identificativo della comanda
id_tavolo	Codice identificativo del tavolo
id_menu	Codice identificativo dell'elemento del menu
nome	Nome dell'elemento del menu
prezzo	Prezzo dell'elemento del menu
stato	Stato dell'elemento del menu

404 – Not found

Chiave	Valore
Message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di un record nella tabella dettagli_comanda.

api/dettagli_comanda/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_det_comanda	Obbligatorio	Codice identificativo del record nella tabella dettagli_comanda
id_comanda	Obbligatorio	Codice identificativo della comanda
id_menu	Obbligatorio	Codice identificativo dell'elemento del menu
stato	Obbligatorio	Stato dell'elemento del menu

Body response (application/json)

200 – OK

Chiave	Valore
Message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

DEL DELETE

Elimina un record dalla tabella dettagli_comanda.

api/dettagli_comanda/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_det_comanda	Obbligatorio	Codice identificativo del record nella tabella dettagli_comanda
id_comanda	Obbligatorio	Codice identificativo della comanda
id_menu	Obbligatorio	Codice identificativo dell'elemento del menu

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di un record specifico della tabella dettagli_comanda, identificato tramite id.

api/dettagli_comanda/read_one.php?id_det_comanda=<id_det_comanda>

Body request (application/json)

Chiave	Richiesto	Descrizione
id_det_comanda	Obbligatorio	Codice identificativo del record nella tabella dettagli_comanda

Body response (application/json)

200 – OK

Chiave	Valore
id_det_comanda	Codice identificativo del record nella tabella dettagli_comanda
id_comanda	Codice identificativo della comanda
id_tavolo	Codice identificativo del tavolo
id_menu	Codice identificativo dell'elemento del menu
stato	Stato dell'elemento del menu

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

PUT READY

Quest'operazione si occupa esclusivamente di cambiare lo stato del record interessato da 0 ("In preparazione") a 1 ("Pronto").

api/dettagli_comanda/ready.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_det_comanda	Obbligatorio	Codice identificativo del record nella tabella dettagli_comanda

Body response (application/json)

200 – OK

Chiave	Valore
Message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

ORDINE

POST CREATE

Crea un ordine caratterizzato dai parametri passati.

api/ordine/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_fornitore	Obbligatorio	Codice identificativo del fornitore

Body response (application/json)

201 – Created

Chiave	Valore
Message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
Message	Errore nei dati

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

GET READ

Legge tutti gli ordini presenti nel database.

api/ordine/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_ordine	Codice identificativo dell'ordine
id_fornitore	Codice identificativo del fornitore
stato	Stato dell'ordine

404 – Not found

Chiave	Valore
Message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di un record nella tabella ordine.

api/ordine/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ordine	Obbligatorio	Codice identificativo dell'ordine
id_fornitore	Obbligatorio	Codice identificativo del fornitore
stato	Obbligatorio	Stato dell'ordine

Body response (application/json)

200 – OK

Chiave	Valore
Message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

DEL DELETE

Elimina un record dalla tabella ordine.

api/ordine/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ordine	Obbligatorio	Codice identificativo dell'ordine

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di un record specifico della tabella ordine, identificato tramite id.

```
api/ordine/read_one.php?id_ordine=<id_ordine>
```

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ordine	Obbligatorio	Codice identificativo dell'ordine

Body response (application/json)

200 – OK

Chiave	Valore
id_ordine	Codice identificativo dell'ordine
id_fornitore	Codice identificativo del fornitore
stato	Stato dell'ordine

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

GET READ_MAX

Quest'operazione si occupa esclusivamente di selezionare dalla tabella ordine il record con il valore 'id_ordine' maggiore, ovvero l'ultimo ordine generato.

```
api/ordine/read_max.php
```

Body response (application/json)

200 – OK

Chiave	Valore
id_ordine	Codice identificativo dell'ordine

404 – Not found

Chiave	Valore
Message	Non è stato trovato alcun risultato

DETTAGLI_ORDINE

POST CREATE

Crea un record nella tabella dettagli_ordine caratterizzato dai parametri passati.

api/dettagli_ordine/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_ordine	Obbligatorio	Codice identificativo dell'ordine
id_ingrediente	Obbligatorio	Codice identificativo dell'ingrediente
quantita	Obbligatorio	Quantità dell'ingrediente

Body response (application/json)

201 – Created

Chiave	Valore
Message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
Message	Errore nei dati

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

GET READ

Legge tutte le informazioni della tabella dettagli_ordine.

api/dettagli_ordine/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_det_ordine	Codice identificativo del record nella tabella dettagli_ordine
id_ordine	Codice identificativo dell'ordine
id_ingrediente	Codice identificativo dell'ingrediente
quantita	Quantità dell'ingrediente

404 – Not found

Chiave	Valore
Message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di un record nella tabella dettagli_ordine.

api/dettagli_ordine/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_det_ordine	Obbligatorio	Codice identificativo del record nella tabella dettagli_ordine
id_ordine	Obbligatorio	Codice identificativo dell'ordine
id_ingredient	Obbligatorio	Codice identificativo dell'ingrediente
quantita	Obbligatorio	Quantità dell'ingrediente

Body response (application/json)

200 – OK

Chiave	Valore
Message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

DEL DELETE

Elimina un record dalla tabella dettagli_ordine.

api/dettagli_ordine/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_det_ordine	Obbligatorio	Codice identificativo del record nella tabella dettagli_ordine

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di un record specifico della tabella dettagli_ordine, identificato tramite id.

api/dettagli_ordine/read_one.php?id_det_ordine=<id_det_ordine>

Body request (application/json)

Chiave	Richiesto	Descrizione
id_det_ordine	Obbligatorio	Codice identificativo del record nella tabella dettagli_ordine

Body response (application/json)

200 – OK

Chiave	Valore
id_det_ordine	Codice identificativo del record nella tabella dettagli_ordine
id_ordine	Codice identificativo dell'ordine
id_ingredient	Codice identificativo dell'ingrediente
quantita	Quantità dell'ingrediente

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

MENU

Come già specificato nella sezione [elenco delle entità](#), Menu contiene tutte le informazioni relative alle pietanze e alle bevande proposte dal ristorante.

POST CREATE

Crea un record nella tabella menu caratterizzato dai parametri passati.

api/menu/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
nome	Obbligatorio	Nome dell'elemento
prezzo	Obbligatorio	Prezzo dell'elemento

Body response (application/json)

201 – Created

Chiave	Valore
Message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
Message	Errore nei dati

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

GET READ

Legge tutti i record della tabella menu.

api/menu/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_menu	Codice identificativo dell'elemento del menu
nome	Nome dell'elemento del menu
prezzo	Prezzo dell'elemento del menu

404 – Not found

Chiave	Valore
Message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di un record nella tabella menu.

api/menu/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_menu	Obbligatorio	Codice identificativo dell'elemento del menu
nome	Obbligatorio	Nome dell'elemento del menu
prezzo	Obbligatorio	Prezzo dell'elemento del menu

Body response (application/json)

200 – OK

Chiave	Valore
Message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

DEL DELETE

Elimina un record dalla tabella menu.

api/menu/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_menu	Obbligatorio	Codice identificativo dell'elemento del menu

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di un record specifico della tabella menu, identificato tramite id.

```
api/menu/read_one.php?id_menu=<id_menu>
```

Body request (application/json)

Chiave	Richiesto	Descrizione
id_menu	Obbligatorio	Codice identificativo dell'elemento del menu

Body response (application/json)

200 – OK

Chiave	Valore
id_menu	Codice identificativo dell'elemento del menu
nome	Nome dell'elemento del menu
prezzo	Prezzo dell'elemento del menu

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

SEGNALAZIONE

POST CREATE

Crea una segnalazione caratterizzata dai parametri passati.

api/segnalazione/create.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_chef	Obbligatorio	Codice identificativo dello chef
ingrediente	Obbligatorio	Nome dell'ingrediente

Body response (application/json)

201 – Created

Chiave	Valore
Message	Operazione eseguita con successo

400 – Bad request

Chiave	Valore
Message	Errore nei dati

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

GET READ

Legge tutti i record della tabella segnalazione.

api/segnalazione/read.php

Body response (application/json)

200 – OK

Chiave	Valore
id_segnalazione	Codice identificativo della segnalazione
id_chef	Codice identificativo dello chef
id_ingrediente	Codice identificativo dell'ingrediente
ingrediente	Nome dell'ingrediente

404 – Not found

Chiave	Valore
Message	Non è stato trovato alcun risultato

PUT UPDATE

Aggiorna i dati di un record nella tabella segnalazione.

api/segnalazione/update.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_segnalazione	Obbligatorio	Codice identificativo dell'elemento del menu
id_chef	Obbligatorio	Codice identificativo dello chef
ingrediente	Obbligatorio	Nome dell'ingrediente

Body response (application/json)

200 – OK

Chiave	Valore
Message	Modifiche applicate con successo

503 – Service unavailable

Chiave	Valore
Message	Servizio non disponibile

DEL DELETE

Elimina un record dalla tabella segnalazione.

api/segnalazione/delete.php

Body request (application/json)

Chiave	Richiesto	Descrizione
id_segnalazione	Obbligatorio	Codice identificativo della segnalazione

Body response (application/json)

200 – OK

Chiave	Valore
message	Operazione eseguita con successo

503 – Service unavailable

Chiave	Valore
message	Servizio non disponibile

GET READ_ONE

Legge i dati di un record specifico dalla tabella segnalazione, identificato tramite id.

```
api/segnalazione/read_one.php?id_segnalazione=<id_segnalazione>
```

Body request (application/json)

Chiave	Richiesto	Descrizione
id_segnalazione	Obbligatorio	Codice identificativo della segnalazione

Body response (application/json)

200 – OK

Chiave	Valore
id_segnalazione	Codice identificativo dell'elemento del menu
id_chef	Codice identificativo dello chef
ingrediente	Nome dell'ingrediente

404 – Not found

Chiave	Valore
message	L'elemento cercato non è presente nel database

CAPITOLO 5: INTERFACCIA GRAFICA

Per la realizzazione dell'interfaccia grafica sono stati utilizzati alcuni framework, che rendono facile l'implementazione e migliorano l'esperienza dell'utente finale.

BOOTSTRAP

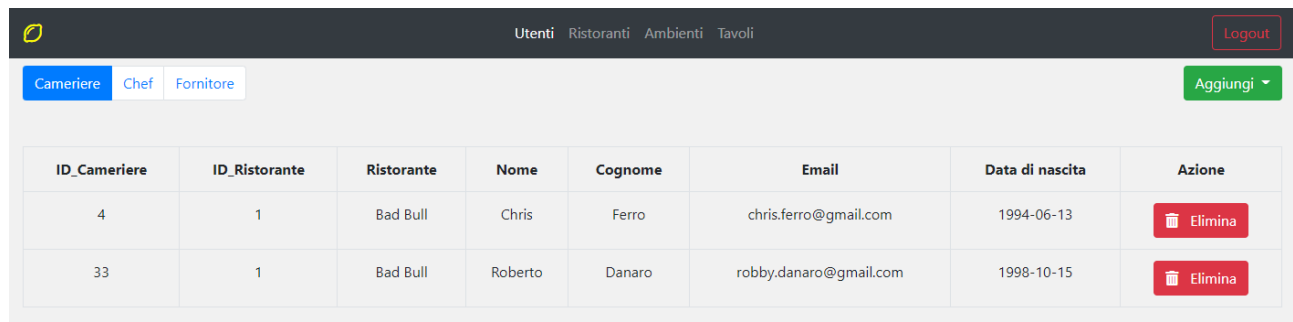
Bootstrap è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web. Essa contiene modelli di progettazione basati su HTML e CSS, sia per la tipografia, che per le varie componenti dell'interfaccia, come moduli, pulsanti e navigazione, così come alcune estensioni opzionali di JavaScript. Inoltre, l'utilizzo di Bootstrap ci dà la garanzia che il nostro servizio web sarà **responsive**, ovvero si presenta con un contenuto dinamico che cambia, le immagini si ridimensionano e gli spazi prendono la giusta forma in base alle esigenze.

Bootstrap è stato utilizzato seguendo lo stesso schema per ogni pagina, ovvero l'utilizzo di tabelle per rappresentare i dati forniti dall'API.

La versione utilizzata è Bootstrap 4.6.

Principalmente sono state utilizzate le classi:

- table
- table-responsive
- table bordered
- table-hover
- btn
- form-control





ID_Cameriere	ID_Ristorante	Ristorante	Nome	Cognome	Email	Data di nascita	Azione
4	1	Bad Bull	Chris	Ferro	chris.ferro@gmail.com	1994-06-13	 Elimina
33	1	Bad Bull	Roberto	Danaro	robby.danaro@gmail.com	1998-10-15	 Elimina

Figura 3: Interfaccia utente

In generale, la struttura delle pagine di ogni utente sarà la stessa ma la differenza sostanziale si trova nella rappresentazione dei dati e nell'implementazione delle operazioni che ciascuno di essi può utilizzare.

Attraverso la barra di navigazione nella parte superiore della finestra, l'utente può spostarsi tra le varie pagine a lui destinate. In ogni pagina, l'utente troverà diversi *button* coi quali può interagire con l'API, richiamando form per creare o aggiornare le informazioni o ancora eliminarle.

BOOTBOX.JS

Bootbox.js è una piccola libreria JavaScript che consente di creare finestre di dialogo utilizzando i moduli di Bootstrap. Nella piattaforma, viene utilizzato unicamente per la gestione degli alert.

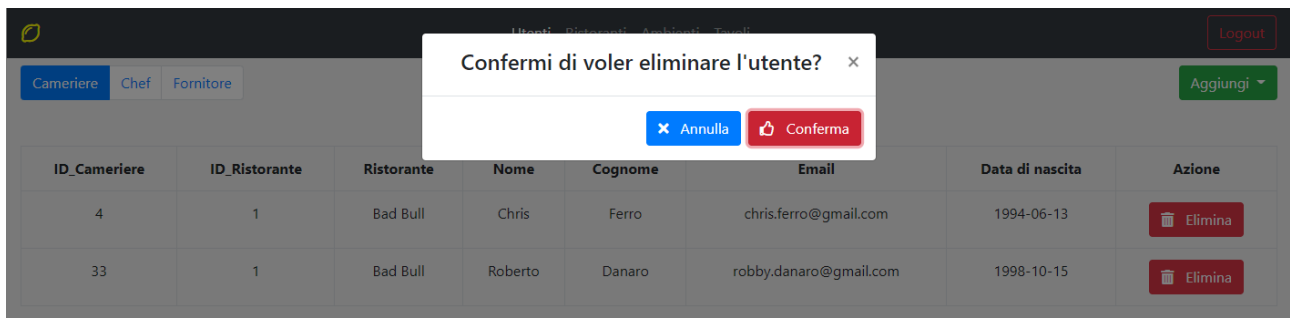


Figura 4: Bootbox alert

Ad esempio, dopo aver fatto clic sul pulsante *Elimina* nella scheda *Utenti*, verrà visualizzato un alert che chiede conferma per l'operazione di eliminazione. Cliccando *Conferma*, verrà avviata la funzione che chiede l'eliminazione del dato all'API; mentre cliccare *Annulla*, significa non mandare tale richiesta all'API.

FONT AWESOME

Font Awesome è un toolkit di icone open-source ampiamente utilizzato che offre immagini vettoriali scalabili che possono essere personalizzate con CSS.

E' stato utilizzato esclusivamente all'interno dei *button*, piuttosto che utilizzare il set di icone messe a disposizione da Bootstrap, poiché offre una gamma più ampia di scelta rispetto a quest'ultimo.