

1. Speisen abfragen

Request

Methode	URL
GET	Foodscout/api/food/:filter

Beispiel

- GET <http://Foodscout/api/food/all HTTP/1.1>
- GET <http://Foodscout/api/food/Lasagne HTTP/1.1>

Parameter

Feld	Beschreibung	Typ
filter	,all' für alle Speisen ansonsten der Name der gesuchten Speise(n)	string

Response **application/json**

Status	Response
200	<pre>[{ "food": "Lasagne", "ingredients": "Nudelaufguss mit Hackfleisch in Tomatensauce mit Käse überbacken", "price": " 8,20", "restaurant": "Restaurant Roma" }, { "food": "Lasagne", "ingredients": "Hackfleisch und Schinken, überbacken ", "price": "5,50 ", "restaurant": "Pizzeria Chili Hot" }]</pre>
500	<pre>{"error":":Fehlerbeschreibung"}</pre>

Rückgabewerte

Feld	Beschreibung	Typ
food	Name der Speise	string
ingredients	Zutatenbeschreibung der Speise	string
price	Preis der Speise	string
restaurant	Name des Restaurants indem die Speise angeboten wird	string

2. Restaurants abfragen

Request

Methode	URL
GET	Foodscout/api/restaurant/:name

Beispiel

- GET <http://Foodscout/api/restaurant/fellini> HTTP/1.1
- GET <http://Foodscout/api/restaurant/roma> HTTP/1.1

Parameter

Feld	Beschreibung	Typ
name	Name des gewünschten Restaurants	string

Response `application/json`

Status	Response
200	<pre>[{ "id": "5b2e2671aa5ad", "name": "Ristorante Fellini", "street": "Groner-Tor-Str. 28", "plz": "37073", "city": "Göttingen", "tel": "(0551) 30 97 963", "lastupdate": "2018-06-24 20:12:20", "fax": "(0551) 49 95 954", "email": "kontakt@fellini-goettingen.de", "opening": "täglich 11:30 - 24:00 Uhr" }]</pre>
400	<code>{"error":":Fehlerbeschreibung"}</code>
500	<code>{"error":":Fehlerbeschreibung"}</code>

Rückgabewerte

Feld	Beschreibung	Typ
id	ID des Datensatzes	<code>string</code>
name	Name des Restaurants	<code>string</code>
street	Straße des Restaurants	<code>string</code>
plz	Postleitzahl des Restaurants	<code>string</code>
tel	Telefonnummer des Restaurants	<code>string</code>
lastupdate	Letztes Aktualisierungsdatum des Datensatzes	<code>timestamp</code>
fax	Faxnummer des Restaurants	<code>string</code>
email	Emailadresse des Restaurants	<code>string</code>
opening	Öffnungszeiten des Restaurants	<code>string</code>

3. Speisen eines Restaurants abfragen

3.1. Speisen des Restaurants Fellini abfragen

Request

Methode	URL
GET	Foodscout/api/fellini/:filter

Beispiel

- GET <http://Foodscout/api/fellini/all HTTP/1.1>
- GET <http://Foodscout/api/fellini/pizza HTTP/1.1>

Parameter

Feld	Beschreibung	Typ
filter	,all' für alle Speisen ansonsten der Name der gesuchten Speise(n)	string

Response **application/json**

Status	Response
200	<pre>[{ "name": "Crostini alla Luciano ", "ingredients": "Verschiedene Variationen auf getoastetem Brot ", "price": "4,80 " }, { "name": "Antipasto Verdure sott olio ", "ingredients": "Verschiedene Gemüsesorten eingelegt in Olivenöl und Knoblauch ", "price": "8,90 " }, // [...]]</pre>
500	<pre>{"error":":Fehlerbeschreibung"}</pre>

Rückgabewerte

Feld	Beschreibung	Typ
name	Name der Speise	string
ingredient	Zutaten der Speise	string
price	Preis der Speise	string

3.2. Speisen des Restaurants Roma abfragen

Request

Methode	URL
GET	Foodscout/api/roma/:filter

Beispiel

- GET <http://Foodscout/api/roma/all HTTP/1.1>
- GET <http://Foodscout/api/roma/lasagne HTTP/1.1>

Parameter

Feld	Beschreibung	Typ
filter	,all' für alle Speisen ansonsten der Name der gesuchten Speise(n)	string

Response `application/json`

Status	Response
200	<pre>[{ "name": "Spaghetti Napoli", "ingredients": "mit frischer Tomatensauce und Basilikum", "price": " 6,90" }, { "name": "Spaghetti Bolognese", "ingredients": "mit Hackfleisch und Tomatensauce ", "price": " 7,30" }, { "name": "Spaghetti Carbonara", "ingredients": "mit Ei und Vorderschinken in Sahnesauce", "price": " 7,80" }, { "name": "Spaghetti Frutti de mare", "ingredients": "mit Meeresfrüchten in Tomatensauce ", "price": " 9,90 " }]</pre>
500	<pre>{"error":":Fehlerbeschreibung"}</pre>

Rückgabewerte

Feld	Beschreibung	Typ
name	Name der Speise	<code>string</code>
ingredient	Zutaten der Speise	<code>string</code>
price	Preis der Speise	<code>string</code>

3.3. Speisen des Restaurants MontRaw abfragen

Request

Methode	URL
GET	Foodscout/api/montraw/:filter

Beispiel

- GET <http://Foodscout/api/montraw/all HTTP/1.1>
- GET <http://Foodscout/api/montraw/toast HTTP/1.1>

Parameter

Feld	Beschreibung	Typ
filter	,all' für alle Speisen ansonsten der Name der gesuchten Speise(n)	string

Response **application/json**

Status	Response
200	<pre>[{ "name": "Gegrillter Käsetoast", "ingredients": " Kastenbrot, Gouda Käse, Basilikum, Tomaten, Spiegelei mit Tomaten, Minze und Chili-Salsa ", "price": "9.50" }, { "name": "French Toast ", "ingredients": " French Toast mit Schlagsahne und frischem Obst", "price": "8.50" }]</pre>
500	<pre>{"error":"","Fehlerbeschreibung"}</pre>

Rückgabewerte

Feld	Beschreibung	Typ
name	Name der Speise	string
ingredient	Zutaten der Speise	string
price	Preis der Speise	string

4. Datenbank auf Update prüfen

Function `checkUpdate`

Überprüft ob ein Update der Datenbank nötig ist

Beschreibung

bool **checkUpdate** (Object \$conn, string \$id)

Gibt True zurück, wenn die Felder *lastupdate* derjenigen Datensätze, die \$id als Fremdschlüssel enthalten, älter als 24 Stunden sind, ansonsten False.

Parameter

Name	Beschreibung	Typ
conn	Enthält die Datenbankverbindung.	Object
id	DB Feld restId der zu überprüfenden Speisen aus der Relation food.	string