200 = alles gut

300 = keine veränderung

400 = Client fehler

500 = Interner Fehler

in der Funktion ausgabe kalories erstes else if.

## Response

|  |  |  |  |
| --- | --- | --- | --- |
| **HTTP Status code** |  | **Type** | **Description** |
| 200 OK |  |  | Recipe object containing number of servings (yield), total calories for the recipe (calories), nutrient content by nutrient type (totalNutrients, totalDaily), diet and health classification (dietLabels, healthLabels) |
| 404 Not Found |  | HTML | The specified URL was not found or couldn’t be retrieved |
| 422 Uprocessable Entity |  | HTML | Couldn’t parse the recipe or extract the nutritional info |
| 555 |  | HTML | Recipe with insufficient quality to process correctly |

404

Get

Header

Put

Patch

Delete

200 OK OK

Die Anfrage war erfolgreich. Die Antwort enthält je nach Anfragemethode die angeforderten Daten.

201 Created Erzeugt

Die Anfrage hat dazu geführt, dass erfolgreich eine neue Resource erzeugt wurde. Deren URI kann im Antwortheader Location mitgeteilt werden.

303 See Other Siehe anderswo

Die Antwort kann unter einer anderen URI mit der GET-Methode geholt werden. Zweck ist es, nach einer POST-Anfrage nahtlos zu einer ausgewählten Resource weiterleiten zu können.

304 Not Modified Nicht verändert

HTTP 1.0HTTP 1.1

Die Resource hat sich seit der letzten Anfrage nicht geändert, die Antwort enthält keine Daten. Dies ist z.B. der Fall, wenn der Client durch einen If-Modified-Since-Header erklärt, dass er die Resource in einem älteren Stand bereits besitzt.

400 Bad Request Ungültige Anfrage

HTTP 1.0HTTP 1.1

Der Client hat eine fehlerhafte Anfrage geschickt, die der Server aufgrund fehlerhafter Syntax nicht bearbeiten kann.

401 Unauthorized Unautorisiert

HTTP 1.0HTTP 1.1

Die Anfrage kann ohne Autorisierung nicht verarbeitet werden. Zu diesem Zweck muss der Server einen WWW\_Authenticate-Antwortheader mitschicken, der erläutert, wie die Authentifizierung vorzunehmen ist. HTTP definiert die zwei Methoden „Basic“ und „Digest“ vor.

403 Forbidden Verboten

HTTP 1.0HTTP 1.1

Der Zugriff ist dauerhaft verboten. Eine Authorisation wird nicht anerkannt, und der Client soll die Anfrage nicht noch einmal stellen.

404 Not Found Nicht gefunden

HTTP 1.0HTTP 1.1

Der Server konnte die vom Client angeforderte Ressource nicht finden. Häufigste Ursache ist ein sogenannter Toter Link.

Der 404-Status kann ebenfalls zurückgeliefert werden, wenn kein anderer Statuscode zutreffend ist oder der Client bewusst ohne Angabe eines näheren Grundes abgewiesen werden soll.

405 Method Not Allowed Methode nicht erlaubt

HTTP 1.1

Die Anfrage-Methode (wie GET oder POST) ist nicht erlaubt. Diese Antwort kann z.B. zurückkommen, wenn man WebDAV-Methoden verwendet, ohne dass der Server diese beherrscht. In einem Allow-Antwortheader muss der Server erlaubte Methoden auflisten.

408 Request Time-out Anfrage-Zeitüberschreitung

HTTP 1.1

Die Anfrage wurde vom Client nicht in der vom Server vorgegebenen Zeit beendet. Der Fehler kann z.B. beim Hochladen großer Dateien auftreten.

409 Conflict Konflikt

HTTP 1.1

Die Anfrage kann nicht bearbeitet werden, weil sich dadurch ein Konfliktfall bei der angefragten Ressource ergeben würde. Soll z.B. eine Ressource per PUT-Methode verändert werden, die aber auf dem Server bereits eine neuere Version aufweist, würde die Bearbeitung einen Konflikt erzeugen.

410 Gone Verschwunden

HTTP 1.1

Die Ressource existiert nicht mehr, der Server kennt aber keine Weiterleitungsadresse. Dieser Zustand ist als dauerhaft anzunehmen.

412 Precondition Failed Vorbedingung missglückt

HTTP 1.1

Eine vom Client vorgegebene Bedingung ist nicht erfüllbar.

413 Request Entity Too Large Anfrage-Entität zu groß

HTTP 1.1

Der Inhalt der Anfrage (z.B. eine hochgeladene Datei) ist zu groß, die Anfrage wird abgelehnt.

414 Request-URI Too Large Anfrage-URI zu lang

HTTP 1.1

Die Anfrage-URI ist zu lang. Der HTTP-Standard definiert keine feste Obergrenze für URI-Längen, d.h., das Erscheinen dieses Fehlers hängt vom Server-Programm und seiner internen Einstellung ab.

416 Requested range not satisfiable Anfrage-Bereich nicht erfüllbar

HTTP 1.1

Der angefragte Teilbereich der Ressource existiert nicht oder ist ungültig. Dies kann z.B. im Zusammenhang mit partiellen Downloads auftreten.

423 Locked gesperrt

HTTP 1.1

Die angeforderte Ressource ist zurzeit gesperrt ( RFC 4918)

429 Too Many Requests zu viele Anfragen

HTTP 1.1

Der Client hat zu viele Anfragen in einem bestimmten Zeitraum gesendet. ( RFC 6585)

500 Internal Server Error Interner Server-Fehler

HTTP 1.0HTTP 1.1

Der Server entdeckt einen internen Fehler und kann deshalb die Anfrage nicht bearbeiten. Fehler in Programmen auf dem Server, z.B. PHP-Skripten, können diese Antwort provozieren.

(Wird von uns genutzt, wenn kein anderer, genauerer Statuscodes übernommen werden kann.)

501 Not Implemented Nicht implementiert

HTTP 1.0HTTP 1.1

Der Server verfügt nicht über die nötige Funktionalität, um die Anfrage zu verarbeiten. Diese Antwort ist angemessen, wenn die in der Anfrage verwendete HTTP-Methode nicht erkannt oder unterstützt wird.

503 Service Unavailable Dienst nicht verfügbar

HTTP 1.0HTTP 1.1

Der Dienst ist derzeit nicht verfügbar. Dies kann aufgrund hohen Datenaufkommens oder wegen Wartungsarbeiten passieren.

505 HTTP Version Not Supported HTTP-Version nicht unterstützt

HTTP 1.1

Die angeforderte HTTP-Version wird nicht unterstützt.

507 Insufficient Storage Speicher des Servers reicht nicht aus Die Anfrage konnte nicht bearbeitet werden, weil der Speicherplatz des Servers dazu zurzeit nicht mehr ausreicht.

508 Loop Detected Endlosschleife Die Operation wurde nicht ausgeführt, weil die Ausführung in eine Endlosschleife gelaufen wäre. Definiert in der Binding-Erweiterung für WebDAV gemäß RFC 5842, weil durch Bindings zyklische Pfade zu WebDAV-Ressourcen entstehen können.

511 Network Authentication Required Identizifierung benötigt Der Client muss sich zuerst authentifizieren um Zugang zum Netzwerk zu erhalten ( RFC 6585)

HTML Statuscodes:

An dieser Stelle standen wir vor der Entscheidung, ob wir die für uns in Frage kommenden Statuscodes einfach auf paar wenige, allumfassende reduzieren wollten, oder eben eine größere, umfassendere Menge auflisten wollten.

Wir entschieden uns für die letztere Wahl, aus einer Reihe aus Gründen:

Zum einen, weil wir glauben dass das Nutzen von spezifischeren Fehlercodes grundsätzlich eine bessere Art zu arbeiten darstellt, da sie Nutzern genauere, und damit potentiell hilfreichere Informationen zur verfügung stellt.

Zum anderen, weil es im Rahmen dieses Projektes darum geht, die Arbeits und Funktonsweise von verteilten Systemen im Netz kennenzulernen, und es kaum schaden kann, sich dafür weiter mit den verschiedenen HTML Statuscodes auseinanderzusetzen.

Deshalb haben wir eine vielleicht etwas zu ausufernde Liste an Statuscodes zusammengestellt, die wir aber trotzdem grundsätzlich für sinnvoll halten.