Recipe - Food - Nutrition 🔖 FREEMIUM Verified ✓

By David | Updated 17 days ago | Food | Featured in Food & Restaurant

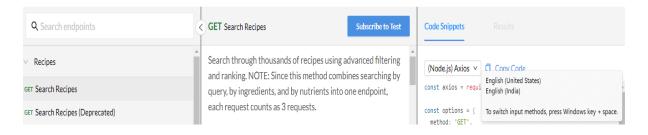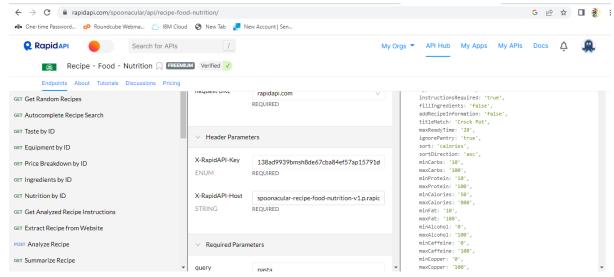| ↙ Popularity	⏱ Latency	✓ Service Level
9.9 / 10	685ms	100%

Endpoints   About   Tutorials   Discussions   Pricing

Recipe - Food - Nutrition API Documentation

The spoonacular Recipe - Food - Nutrition API gives you to access to thousands of recipes, storebought packaged foods, and chain restaurant menu items. Our food ontology and semantic recipe search engine makes it possible to search for recipes using natural language queries, such as "gluten free brownies without sugar" or "low fat vegan cupcakes." You can automatically calculate the nutritional information for any recipe, estimate recipe costs, visualize ingredient lists, find recipes for what's in your fridge, find recipes based on special diets, nutritional requirements, or favorite ingredients, classify recipes into types and cuisines, convert ingredient amounts, or even compute an entire meal plan. With our powerful API, you can create many kinds of food and nutrition apps.

Special diets/dietary requirements currently available include: vegan, vegetarian, pescetarian, gluten free, grain free, dairy free, high protein, low sodium, low carb, Paleo, Primal, ketogenic, and more.

🔍 Search endpoints	‹ **GET** Search Recipes Subscribe to Test	Code Snippets Results
⌄ Recipes	Search through thousands of recipes using advanced filtering and ranking. NOTE: Since this method combines searching by query, by ingredients, and by nutrients into one endpoint, each request counts as 3 requests.	(Node.js) Axios ⌄ ⧉ Copy Code
GET Search Recipes		English (United States)
GET Search Recipes (Deprecated)		English (India)

```
const axios = requi

const options = {
  method: 'GET',
```

To switch input methods, press Windows key + space.

RapidAPI   Search for APIs   My Orgs ▾   API Hub   My Apps   My APIs   Docs   🔔   👤

Recipe - Food - Nutrition 🔖 FREEMIUM Verified ✓

Endpoints   About   Tutorials   Discussions   Pricing

GET Get Random Recipes	Request URL rapidapi.com ⌄	instructionsRequired: 'true',
GET Autocomplete Recipe Search	REQUIRED	fillIngredients: 'false',
GET Taste by ID		addRecipeInformation: 'false',
GET Equipment by ID	⌄ Header Parameters	titleMatch: 'Crock Pot', maxReadyTime: '20',
GET Price Breakdown by ID		ignorePantry: 'true',
GET Ingredients by ID	X-RapidAPI-Key 138ad9939bmsh8de67cba84ef57ap15791d	sort: 'calories', sortDirection: 'asc',
GET Nutrition by ID	ENUM REQUIRED	minCarbs: '10', maxCarbs: '100',
GET Get Analyzed Recipe Instructions		minProtein: '10', maxProtein: '100',
GET Extract Recipe from Website	X-RapidAPI-Host spoonacular-recipe-food-nutrition-v1.p.rapic	minCalories: '50', maxCalories: '800',
POST Analyze Recipe	STRING REQUIRED	minFat: '10', maxFat: '100',
GET Summarize Recipe		minAlcohol: '0', maxAlcohol: '100',
	⌄ Required Parameters	minCaffeine: '0', maxCaffeine: '100',
	query pasta	minCopper: '0', maxCopper: '100',

Recipe - Food - Nutrition 🔖 FREEMIUM Verified ✓

Endpoints   About   Tutorials   Discussions   Pricing

GET Autocomplete Recipe Search	
GET Taste by ID	
GET Equipment by ID	
GET Price Breakdown by ID	
GET Ingredients by ID	
GET Nutrition by ID	
GET Get Analyzed Recipe Instructions	
GET Extract Recipe from Website	
POST Analyze Recipe	
GET Summarize Recipe	

the recipe must have.

maxZinc
```
100
```
NUMBER   OPTIONAL   The maximum number of milligrams of zinc
                     the recipe can have.

offset
```
0
```
NUMBER   OPTIONAL   The number of results to skip (between 0
                     and 900).

number
```
10
```
NUMBER   OPTIONAL   The number of results to return (between 1
                     and 100).

```javascript
    minZinc: '0',
    maxZinc: '100',
    offset: '0',
    number: '10',
    limitLicense: 'false',
    ranking: '2'
  },
  headers: {
    'X-RapidAPI-Key': '138ad9939bmsh8de67cba84ef57ap15791djsn479
21b7da04e',
    'X-RapidAPI-Host': 'spoonacular-recipe-food-nutrition-v1.p.r
apidapi.com'
  }
};

axios.request(options).then(function (response) {
    console.log(response.data);
}).catch(function (error) {
    console.error(error);
});
```