



# HTTP

## Application Integration



# HTTP Query Strings

http://john:password@www.example.com:123/forum?tag=networking&order=newest#top



# HTTP Headers

GET /johnberrydev/JusdaTeamData/projects/1/columns/7360105/cards

HTTP/1.1

Host: github.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0)

Gecko/20100101 Firefox/82.0

Accept: text/html

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate, br

Referer: <https://github.com/johnberrydev/JusdaTeamData/projects/1>

X-Requested-With: XMLHttpRequest

Connection: keep-alive

X-API-Key: 289f19d18c1877a28

# HTTP Methods

## GET

Retrieves the resource specified by the URI.  
This only retrieves data. .



## POST

Submits a new entity to the resource specified by the URI. This adds data to the server.



## PUT

Replaces an entity at the resource specified by the URI with the request body of the PUT.



## DELETE

Removes an entity at the resource specified by the URI



## OPTIONS

describes the communications options that are available for the target resource. The resource will reply with a list of HTTP methods that are allowed.

# Request Body

Content Type	Description
application/x-www-form-urlencoded	HTML form data
multipart/form-data	HTML form data, can handle binary data
text/plain	Raw text
application/json	JSON payload, commonly used in APIs
application/xml	XML payload
Binary	Binary data like image, sound, or video files

# Status Codes

Category	Description
1XX	Informational
2XX	Successful
3XX	Redirection
4XX	Client Error
5XX	Server Error

# Most Common Status Codes

Code	Explanation
200	Everything worked as expected
201	Record created. Sometimes used to indicate a successful addition of a new record in the receiving system.
400	Bad request. Something is wrong with the sender's payload or URL.
403	Forbidden. The sender has not authenticated properly, and the server is refusing access.
404	Record not found. Returned when the sender attempts to get a resource that doesn't exist on the server.
429	Too many requests. Returned when the sender has exceeded rate limiting thresholds. See Rate Limiting learning unit for more info.
500	Something went wrong on the server. The sender should either try the request later or report the problem to the receiver.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

# Authentication



## Basic Authentication

Client provides User ID and password in the HTTP header of the request. The key *Authorization* is used in the header. User ID and password are separated by a colon and encoded with base 64.

Example: *Authorization: Basic c3RldmU6cGFzcEYMW==*

## Bearer Token

Client provides an access key like a JSON Web Token (JWT) in the HTTP header of the request. The header key "Authorization" should be used with the value "Bearer" followed by the key or token. If your key is 123456 the header would look like *Authorization: Bearer 123456*.

## API Key

The header key name is specified by the API provider. Examples of key names are *api-key* or *x-api-key*. The actual API key is passed as the header value. For example, *x-api-key : 123456*.

<https://www.bing.com/search?q=base64+decode>