Table of Contents

[REVIEW of REST API Exercises 4 & 5 2](#_Toc48631297)

[API REST query parameter table template: 2](#_Toc48631298)

[Documenting HTTP Headers 2](#_Toc48631299)

[HTTP Headers for Content Negotiation 3](#_Toc48631300)

[Accept: What data format this API is expecting to receive (MOST COMMON) 4](#_Toc48631301)

[Content-Type: What format of data the server is expecting; what the client needs to send 4](#_Toc48631302)

[Bearer: A type of token authentication messaging 4](#_Toc48631303)

[Header name table: 4](#_Toc48631304)

[Sample POST request with HTTP headers: 5](#_Toc48631305)

[Anatomy of a POST Request 5](#_Toc48631306)

[APPENDIX 6](#_Toc48631307)

[XML example 6](#_Toc48631308)

[XML documentation 6](#_Toc48631309)

[XML documentation with many Attributes 7](#_Toc48631310)

[JSON + XML file 7](#_Toc48631311)

[JSON example 8](#_Toc48631312)

[JSON documentation for menu UI 8](#_Toc48631313)

[CRUD == POST, GET, PUT, DELETE 8](#_Toc48631314)

[CRUD Methods 9](#_Toc48631315)

[Header name table: 10](#_Toc48631316)

[HTTP headers Content-Type 10](#_Toc48631317)

# REVIEW of REST API Exercises 4 & 5

## API REST query parameter table template:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Type** | **Required** | **Note** |
| startDate | The start date for when the bug was created | Date | Optional | YYYY-MM-DD. Default is the earliest recorded bug |
| endDate | The end date for when the bug was created | Date | Optional | YYYY-MM-DD. Default is today’s date |
| priority | Priority value | integer | Optional | 1-4; 1=highest; The default is all |
| severity | Severity value | integer | Optional | 1-4; 1=most severe; default is all |
| status | Status of bug | string | Optional | Valid values: open, closed, duplicate, notabug;  Default is all |
| start | For pagination Starting index | integer | Optional | 0=first bug;  Default is zero |
| total | For pagination | integer | Optional | Total number of bugs returned;  Default is All |

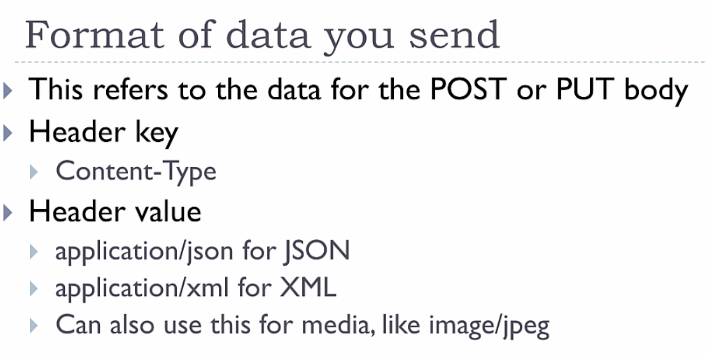
Sample URL where startDate and status are parameters unique to this API URL; limit is a standard pagination parameter.

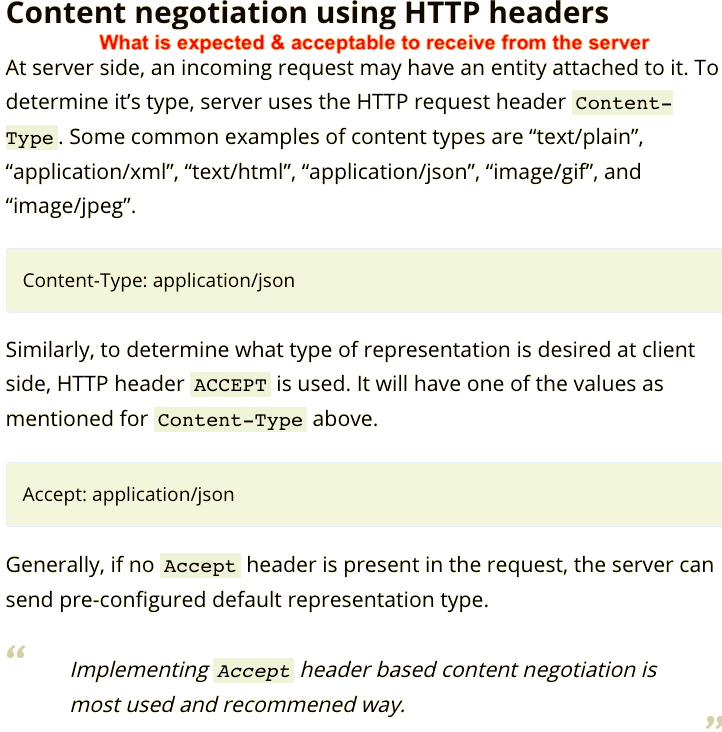
GET <https://api.bugsquisher.com/bug/?startDate=2020-06-01&status=1&limit=20>

## Documenting HTTP Headers

* Format a photo spec’d with Content-Type header (what you are **sending**) as jpeg, gif, or png. Default is jpeg
* What you can receive is spac’d in the Accept header as application/json or application/xml. Default is application/json.  
  The authorization token us spec’d with a query parameter – no header needed.
* The method is POST
* The resource is photo.

## HTTP Headers for Content Negotiation





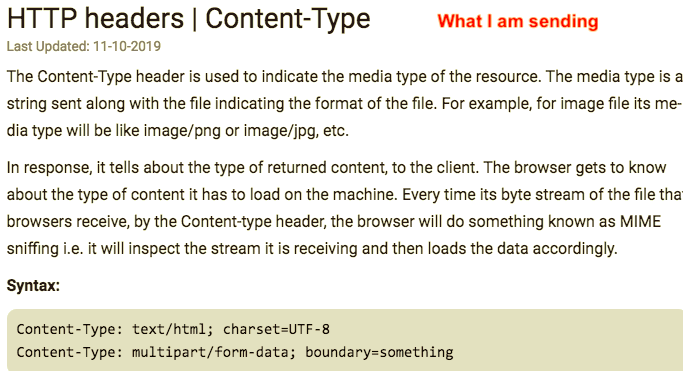
### 

### 

### Accept: What data format this API is expecting to receive (MOST COMMON)

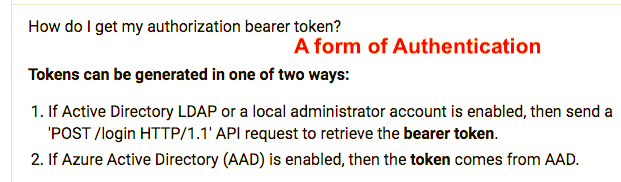
|  |
| --- |
| What the client side is expecting to receive. **What I am receiving**  Accept: application/json |

### Content-Type: What format of data the server is expecting; what the client needs to send

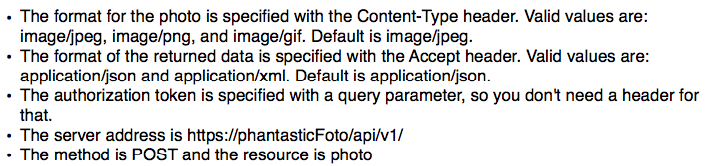


* Content-Type: text/html; charset=UTF-8
* Content-Type: multipart/form-data; boundary=something

### Bearer: A type of token authentication messaging



Exercise 5: Documenting Headers with a table from following dev team:



The format of the photo to upload.

### Header name table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Header name** | **Description** | **Required** | **Values** |
| Content-Type | The format of the photo to upload. | optional | image/jpeg, image/ png, image/ gif  DEFAULT: jpeg |
| Accept | The format of the photo returned. | optional | application/json, application/xml  DEFAULT: json |

### Sample POST request with HTTP headers:

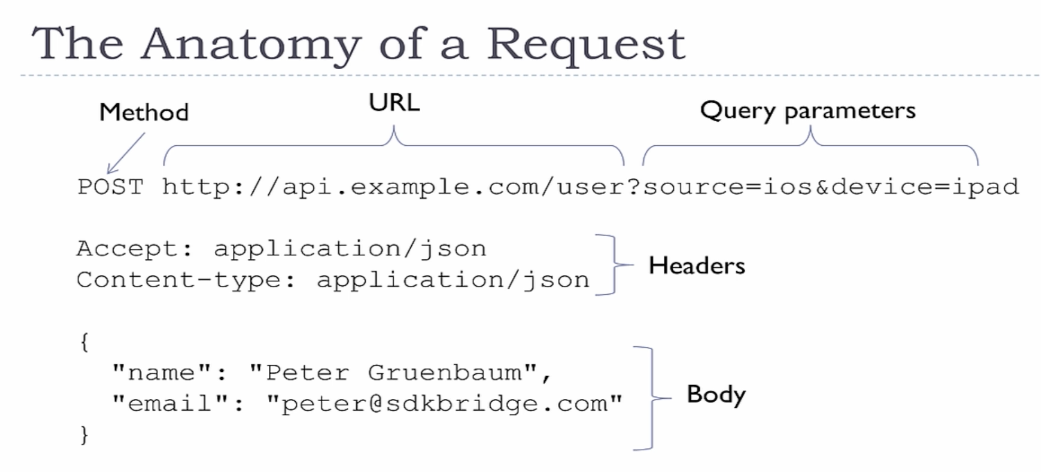
POST [https://phantasticFoto/api/v1/photo](https://phantasticfoto/api/v1/photo)

Content-type: image/jpeg

Accept: application/json

POST body is the image to upload

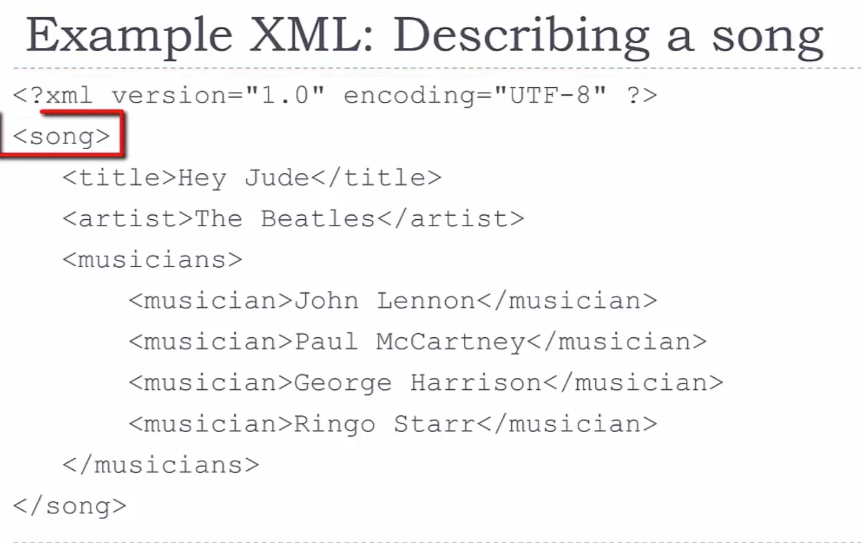
### Anatomy of a POST Request



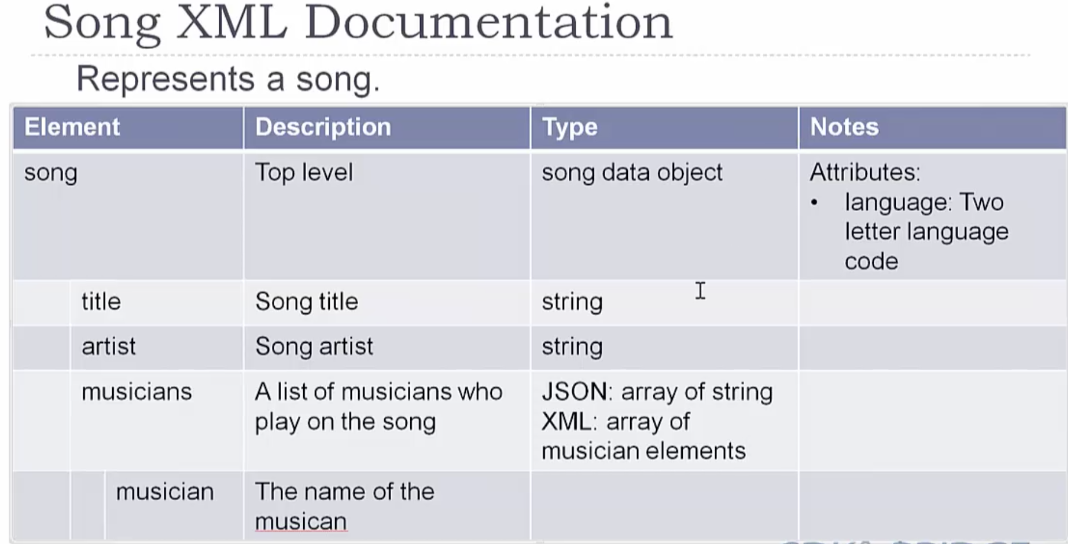
# 

# APPENDIX

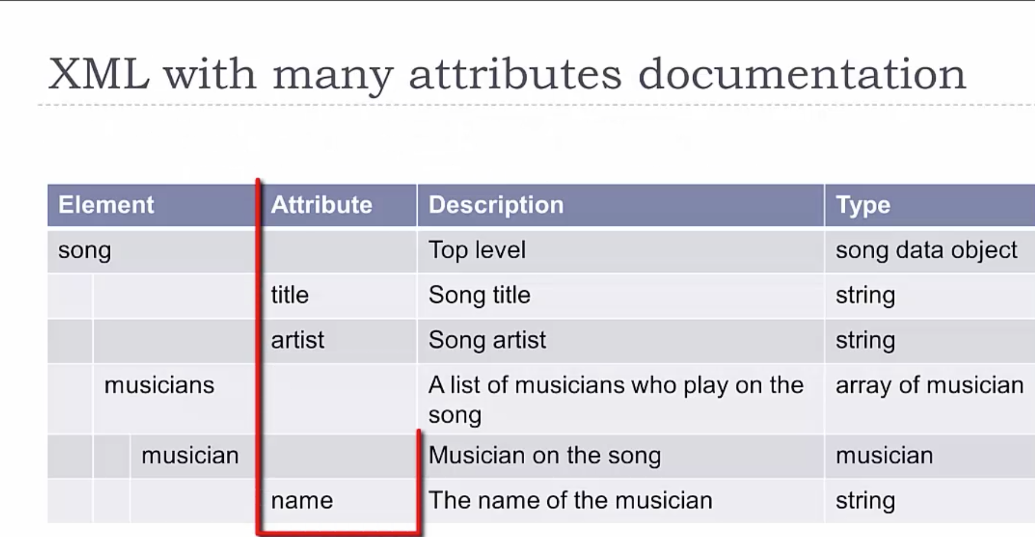
## XML example



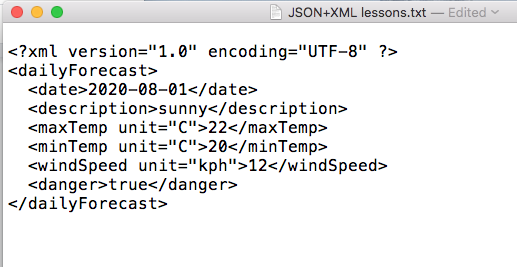
## XML documentation



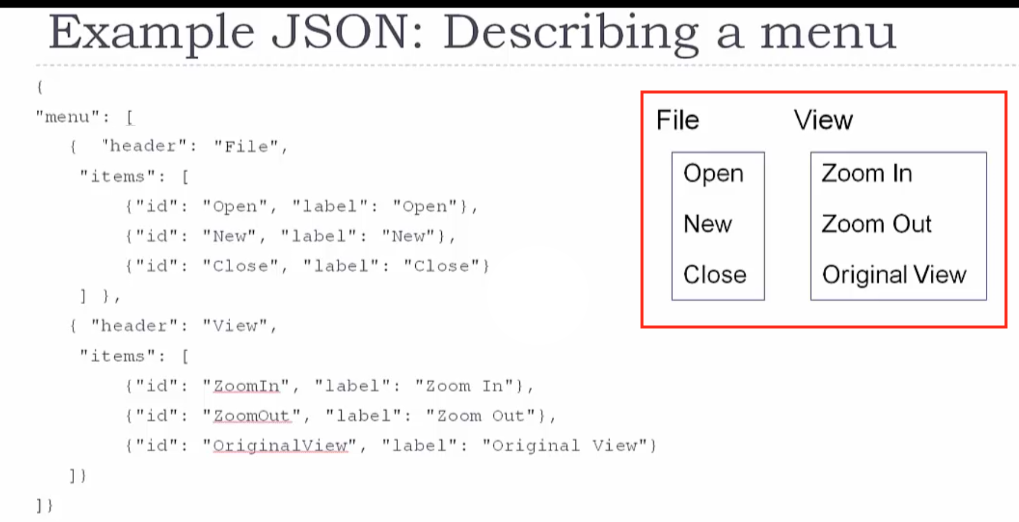
## XML documentation with many Attributes



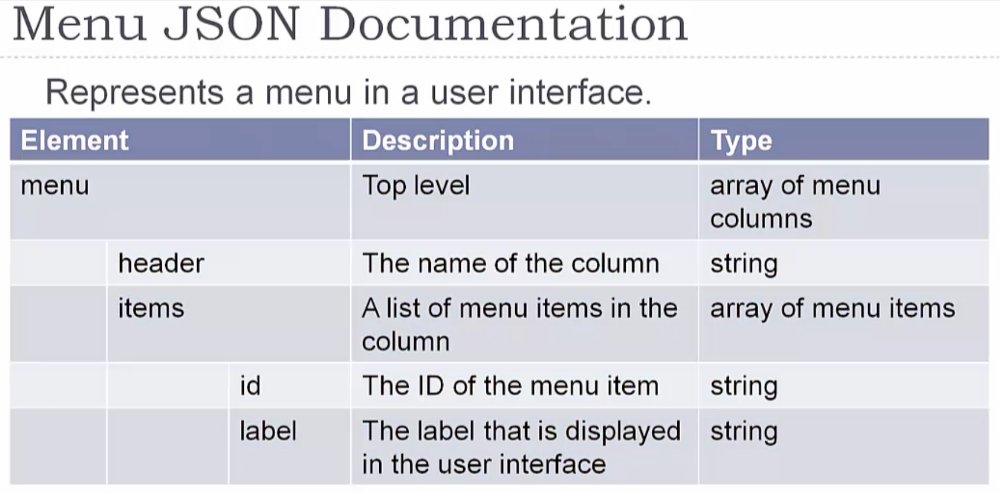
## JSON + XML file



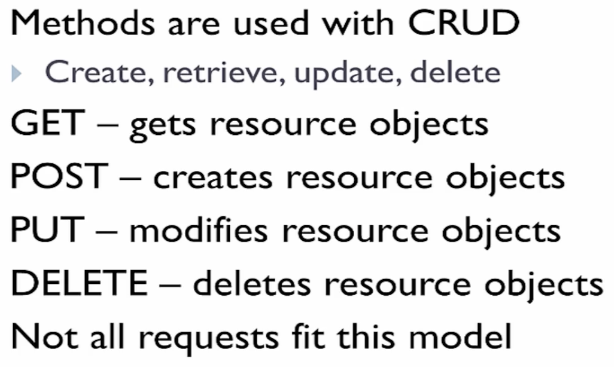
## JSON example



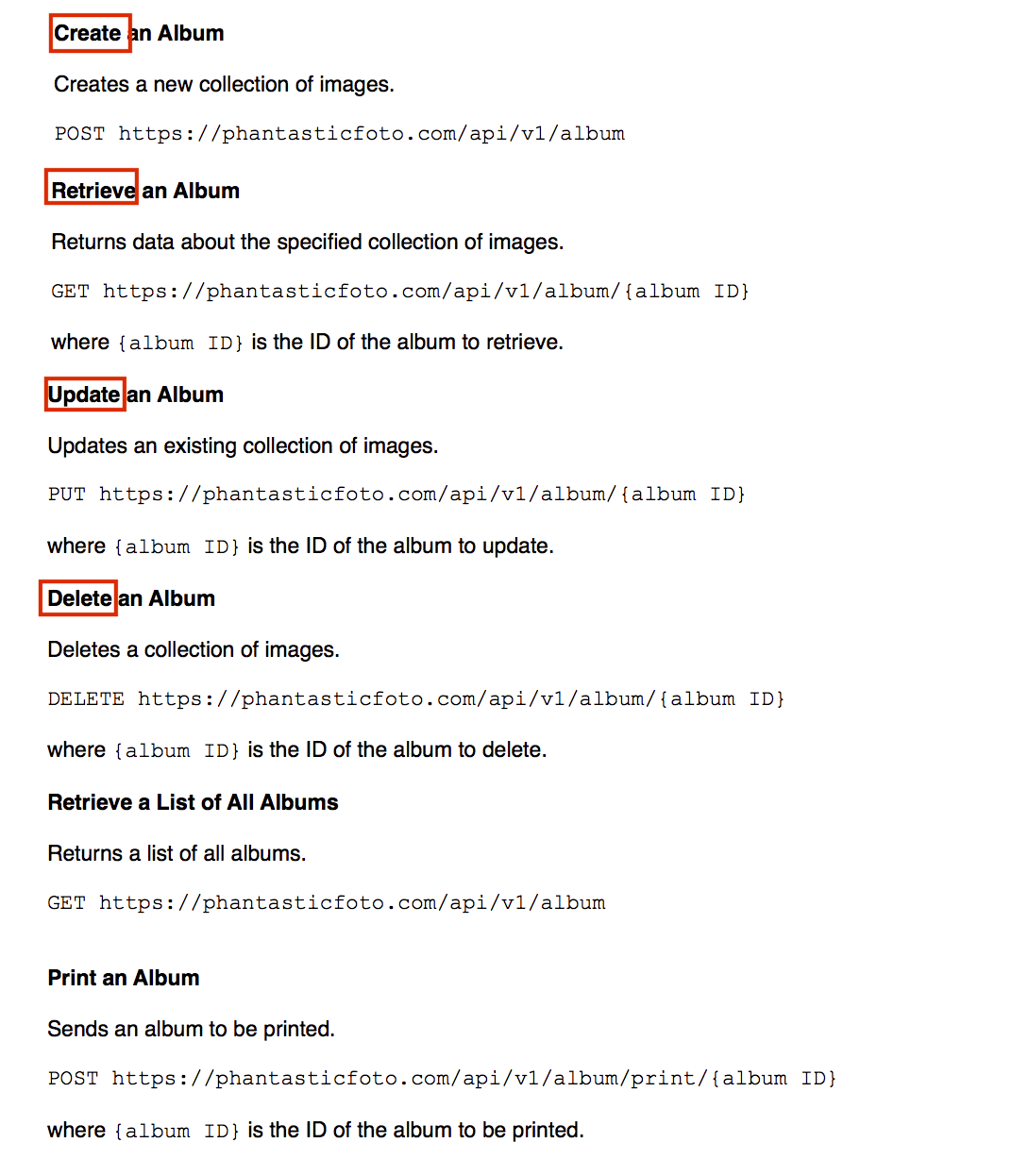
## JSON documentation for menu UI



## CRUD == POST, GET, PUT, DELETE



## CRUD Methods



## Header name table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Header name** | **Description** | **Required** | **Values** |
| Content-Type | The format of the photo to upload. | optional | image/jpeg, image/ png, image/ gif  DEFAULT: jpeg |
| Accept | The format of the photo returned. | optional | application/json, application/xml  DEFAULT: json |

### 

## HTTP headers Content-Type

