



# LAB- Designing API using RAML

## Use Case : Blogger API

```
documentation:  
  - title: Headline  
    content: docs/headline.md
```

### 1) Retrieve User by userId

schema ---user-schema.json

example - user-example.json

### 2) Retrieve all blogs of a user identified by user

schema ---getListByUser-schema.json

example - getListByUser-example.json

### Optional query parameters

```
view:  
  description: |  
    Acceptable values are:  
    "ADMIN": Admin level detail  
    "AUTHOR": Author level detail  
    "READER": Admin level detail  
fetchUserInfo:  
  description: Whether the response is a list of blogs with per-user information  
instead of just blogs.  
  type: boolean  
  example: true
```

### 3) Retrieve a blog by blogId of a user identified by user

schema --- blogUserInfo-schema.json

example - blogUserInfo-example.json

### Optional query parameters

```
maxPosts:  
  description: Maximum number of posts to pull back with the blog.  
  type: integer  
  example: 25
```



4) Retrieve all posts of blog by blogId of a user identified by user

schema [postUserInfosList-schema.json](#)

example - [postUserInfosList-example.json](#)

query parameters

**endDate:**  
description: Latest date of comment to fetch, a date-time with RFC 3339 formatting.

**fetchBodies:**  
description: Whether the body content of the comments is included.

**maxResults:**  
description: Maximum number of comments to include in the result.

**pageToken:**  
description: Continuation token if request is paged.

**startDate:**  
description: Earliest date of comment to fetch, a date-time with RFC 3339 formatting.

**status:**  
description: |  
Acceptable values are:  
"emptied": Comments that have had their content removed  
"live": Comments that are publicly visible  
"pending": Comments that are awaiting administrator approval  
"spam": Comments marked as spam by the administrator

**view:**  
description: |  
Acceptable values are:  
"ADMIN": Admin level detail  
"AUTHOR": Author level detail  
"READER": Admin level detail

**labels:**  
description: Comma-separated list of labels to search for.

**orderBy:**  
description: |  
Sort order applied to search results. Default is published.  
  
Acceptable values are:  
"published": Order by the date the post was published  
"updated": Order by the date the post was last updated

5) Retrieve a posts by postId of blog by blogId of a user identified by user

schema [postUserInfo-schema.json](#)

example - [postUserInfo-example.json](#)

query parameters

**maxComments:**



**description:** Maximum number of comments to retrieve for the post.  
**type:** integer  
**example:** 25

## 6) Retrieve a blog by blogId

schema [blogResource-schema.json](#)

example - [blogResource-example.json](#)

query parameters

**maxPosts:**  
**description:** Maximum number of posts to retrieve along with the blog. When this parameter is not specified, no posts will be returned as part of the blog resource.  
**type:** integer  
**example:** 25

## 6) Retrieve all comments for a blog by blogId, across all posts

schema [commentsList-schema.json](#)

example - [commentsList-example.json](#)

query parameters

**endDate:**  
**description:** Latest date of comment to fetch, a date-time with RFC 3339 formatting.  
**type:** date  
**example:** "2016-10-02T10:00:00"

**fetchBodies:**  
**description:** Whether the body content of the comments is included.  
**type:** boolean  
**example:** true

**maxResults:**  
**description:** Maximum number of comments to include in the result.  
**type:** integer  
**example:** 25

**pageToken:**  
**description:** Continuation token if request is paged.

**startDate:**  
**description:** Earliest date of comment to fetch, a date-time with RFC 3339 formatting.  
**type:** date  
**example:** "2016-09-02T10:00:00"



6) Retrieve all pages of a blog by blogId, across all posts

schema [pages-schema.json](#)

example - [pages-example.json](#)

query parameters

```
fetchBodies:
  description: Whether to retrieve the Page bodies.
  type: boolean
  example: true

status:
  description: |
    Acceptable values are:
    "draft": Draft (unpublished) Pages
    "imported": Pages that have had their content removed
    "live": Pages that are publicly visible
  example: live

view:
  description: |
    Acceptable values are:
    "ADMIN": Admin level detail
    "AUTHOR": Author level detail
    "READER": Admin level detail
```

7) add a new page to a blog identified by blogId

request body schema [page-schema.json](#)

request body example - [page-example.json](#)

response body schema [page-schema.json](#)

response body example - [page-example.json](#)

8) get details of a page identified by pageId belonging to a blog identified by blogId

response body schema [page-schema.json](#)

response body example - [page-example.json](#)

query parameters

```
view:
  description: |
    Acceptable values are:
    "ADMIN": Admin level detail
    "AUTHOR": Author level detail
```



"READER": Admin level detail

9) delete a page identified by pageId belonging to a blog identified by blogId

No Response body

10) update a page identified by pageId belonging to a blog identified by blogId

request body schema [page-schema.json](#)

request body example - [page-example.json](#)

response body schema [page-schema.json](#)

response body example - [page-example.json](#)

11) Retrieve page view stats for a blog identified by blogId

response body schema [pageViews-schema.json](#)

response body example - [pageViews-example.json](#)

Query Parameters :

```
range:
    description: |
        Acceptable values are:
        "30DAYS": Page view counts from the last thirty days.
        "7DAYS": Page view counts from the last seven days.
        "all": Total page view counts from all time.
    example: 30DAYS
```