```
HTTP/1.1 200 OK
Expires: Sat, 25 Jul 2015 21:20:52 GMT
Date: Sat, 25 Jul 2015 21:15:52 GMT
Vary: X-Origin
Content-Type: application/json; charset=UTF-8
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
Server: GSE
Cache-Control: public, max-age=300, must-revalidate, no-transform
Age: 0
Alternate-Protocol: 443:quic,p=1
Accept-Ranges: none
Vary: Origin,Accept-Encoding
Connection: close

{
 "kind": "discovery#restDescription",
 "etag": "\"ye6orv2F-1npMW3u9suM3a7C5Bo/ZRgj7Ah7gwQETAdvqLpx-xGEmV8\"",
 "discoveryVersion": "v1",
 "id": "blogger:v3",
 "name": "blogger",
 "version": "v3",
 "revision": "20150422",
 "title": "Blogger API",
 "description": "API for access to the data within Blogger.",
 "ownerDomain": "google.com",
 "ownerName": "Google",
 "icons": {
  "x16": "https://www.google.com/images/icons/product/blogger-16.png",
  "x32": "https://www.google.com/images/icons/product/blogger-32.png"
 },
 "documentationLink": "https://developers.google.com/blogger/docs/3.0/
     getting_started",
 "labels": [
  "limited_availability"
 ],
 "protocol": "rest",
 "baseUrl": "https://www.googleapis.com/blogger/v3/",
 "basePath": "/blogger/v3/",
 "rootUrl": "https://www.googleapis.com/",
 "servicePath": "blogger/v3/",
 "batchPath": "batch",
 "parameters": {
  "alt": {
   "type": "string",
   "description": "Data format for the response.",
   "default": "json",
   "enum": [
    "json"
   ],
   "enumDescriptions": [
    "Responses with Content-Type of application/json"
   ],
   "location": "query"
  },
        "fields": {
         "type": "string",
         "description": "Selector specifying which fields to include in a partial
             response.",
         "location": "query"
        },
        "key": {
         "type": "string",
         "description": "API key. Your API key identifies your project and provides you
             with API access, quota, and reports. Required unless you provide an OAuth
             2.0 token.",
         "location": "query"
        },
        "oauth_token": {
         "type": "string",
         "description": "OAuth 2.0 token for the current user.",
         "location": "query"
        },
        "prettyPrint": {
         "type": "boolean",
         "description": "Returns response with indentations and line breaks.",
         "default": "true",
         "location": "query"
        },
        "quotaUser": {
         "type": "string",
         "description": "Available to use for quota purposes for server-side
             applications. Can be any arbitrary string assigned to a user, but should
             not exceed 40 characters. Overrides userIp if both are provided.",
         "location": "query"
        },
        "userIp": {
         "type": "string",
         "description": "IP address of the site where the request originates. Use this
             if you want to enforce per-user limits.",
         "location": "query"
        }
       },
       "auth": {
        "oauth2": {
         "scopes": {
          "https://www.googleapis.com/auth/blogger": {
           "description": "Manage your Blogger account"
          },
          "https://www.googleapis.com/auth/blogger.readonly": {
           "description": "View your Blogger account"
          }
         }
        }
       },
       "schemas": {
        "Blog": {
         "id": "Blog",
         "type": "object",
         "properties": {
          "customMetaData": {
           "type": "string",
           "description": "The JSON custom meta-data for the Blog"
          },
          "description": {
           "type": "string",
           "description": "The description of this blog. This is displayed underneath
               the title."
          },
          "id": {
           "type": "string",
           "description": "The identifier for this resource."
          },
          "kind": {
           "type": "string",
           "description": "The kind of this entry. Always blogger#blog",
           "default": "blogger#blog"
          },
          "locale": {
           "type": "object",
           "description": "The locale this Blog is set to.",
           "properties": {
            "country": {
             "type": "string",
             "description": "The country this blog's locale is set to."
            },
            "language": {
             "type": "string",
             "description": "The language this blog is authored in."
            },
            "variant": {
             "type": "string",
             "description": "The language variant this blog is authored in."
            }
           }
          },
          "name": {
           "type": "string",
           "description": "The name of this blog. This is displayed as the title."
          },
          "pages": {
           "type": "object",
           "description": "The container of pages in this blog.",
           "properties": {
            "selfLink": {
             "type": "string",
             "description": "The URL of the container for pages in this blog."
            },
            "totalItems": {
             "type": "integer",
             "description": "The count of pages in this blog.",
             "format": "int32"
            }
           }
          },
          "posts": {
           "type": "object",
          "description": "The container of posts in this blog.",
          "properties": {
           "items": {
            "type": "array",
            "description": "The List of Posts for this Blog.",
            "items": {
             "$ref": "Post"
            }
           },
           "selfLink": {
            "type": "string",
            "description": "The URL of the container for posts in this blog."
           },
           "totalItems": {
            "type": "integer",
            "description": "The count of posts in this blog.",
            "format": "int32"
           }
          }
         },
         "published": {
          "type": "string",
          "description": "RFC 3339 date-time when this blog was published.",
          "format": "date-time"
         },
         "selfLink": {
          "type": "string",
          "description": "The API REST URL to fetch this resource from."
         },
         "status": {
          "type": "string",
          "description": "The status of the blog."
         },
         "updated": {
          "type": "string",
          "description": "RFC 3339 date-time when this blog was last updated.",
          "format": "date-time"
         },
         "url": {
          "type": "string",
          "description": "The URL where this blog is published."
         }
        }
       },
       "BlogList": {
        "id": "BlogList",
        "type": "object",
        "properties": {
         "blogUserInfos": {
          "type": "array",
          "description": "Admin level list of blog per-user information",
          "items": {
           "$ref": "BlogUserInfo"
          }
         },
         "items": {
```

```
   "type": "array",
   "description": "The list of Blogs this user has Authorship or Admin rights
       over.",
   "items": {
    "$ref": "Blog"
   }
  },
  "kind": {
   "type": "string",
   "description": "The kind of this entity. Always blogger#blogList",
   "default": "blogger#blogList"
  }
 }
},
"BlogPerUserInfo": {
 "id": "BlogPerUserInfo",
 "type": "object",
 "properties": {
  "blogId": {
   "type": "string",
   "description": "ID of the Blog resource"
  },
  "hasAdminAccess": {
   "type": "boolean",
   "description": "True if the user has Admin level access to the blog."
  },
  "kind": {
   "type": "string",
   "description": "The kind of this entity. Always blogger#blogPerUserInfo",
   "default": "blogger#blogPerUserInfo"
  },
  "photosAlbumKey": {
   "type": "string",
   "description": "The Photo Album Key for the user when adding photos to the
       blog"
  },
  "role": {
   "type": "string",
   "description": "Access permissions that the user has for the blog (ADMIN,
       AUTHOR, or READER)."
  },
  "userId": {
   "type": "string",
   "description": "ID of the User"
  }
 }
},
"BlogUserInfo": {
 "id": "BlogUserInfo",
 "type": "object",
 "properties": {
  "blog": {
   "$ref": "Blog",
   "description": "The Blog resource."
  },
  "blog_user_info": {
     "$ref": "BlogPerUserInfo",
     "description": "Information about a User for the Blog."
    },
    "kind": {
     "type": "string",
     "description": "The kind of this entity. Always blogger#blogUserInfo",
     "default": "blogger#blogUserInfo"
    }
   }
  },
  "Comment": {
   "id": "Comment",
   "type": "object",
   "properties": {
    "author": {
     "type": "object",
     "description": "The author of this Comment.",
     "properties": {
      "displayName": {
       "type": "string",
       "description": "The display name."
      },
      "id": {
       "type": "string",
       "description": "The identifier of the Comment creator."
      },
      "image": {
       "type": "object",
       "description": "The comment creator's avatar.",
       "properties": {
        "url": {
         "type": "string",
         "description": "The comment creator's avatar URL."
        }
       }
      },
      "url": {
       "type": "string",
       "description": "The URL of the Comment creator's Profile page."
      }
     }
    },
    "blog": {
     "type": "object",
     "description": "Data about the blog containing this comment.",
     "properties": {
      "id": {
       "type": "string",
       "description": "The identifier of the blog containing this comment."
      }
     }
    },
    "content": {
     "type": "string",
     "description": "The actual content of the comment. May include HTML markup."
    },
          "id": {
           "type": "string",
           "description": "The identifier for this resource."
          },
          "inReplyTo": {
           "type": "object",
           "description": "Data about the comment this is in reply to.",
           "properties": {
            "id": {
             "type": "string",
             "description": "The identified of the parent of this comment."
            }
           }
          },
          "kind": {
           "type": "string",
           "description": "The kind of this entry. Always blogger#comment",
           "default": "blogger#comment"
          },
          "post": {
           "type": "object",
           "description": "Data about the post containing this comment.",
           "properties": {
            "id": {
             "type": "string",
             "description": "The identifier of the post containing this comment."
            }
           }
          },
          "published": {
           "type": "string",
           "description": "RFC 3339 date-time when this comment was published.",
           "format": "date-time"
          },
          "selfLink": {
           "type": "string",
           "description": "The API REST URL to fetch this resource from."
          },
          "status": {
           "type": "string",
           "description": "The status of the comment (only populated for admin users)"
          },
          "updated": {
           "type": "string",
           "description": "RFC 3339 date-time when this comment was last updated.",
           "format": "date-time"
          }
         }
        },
        "CommentList": {
         "id": "CommentList",
         "type": "object",
         "properties": {
          "etag": {
           "type": "string",
           "description": "Etag of the response."
```

```
    },
    "items": {
     "type": "array",
     "description": "The List of Comments for a Post.",
     "items": {
      "$ref": "Comment"
     }
    },
    "kind": {
     "type": "string",
     "description": "The kind of this entry. Always blogger#commentList",
     "default": "blogger#commentList"
    },
    "nextPageToken": {
     "type": "string",
     "description": "Pagination token to fetch the next page, if one exists."
    },
    "prevPageToken": {
     "type": "string",
     "description": "Pagination token to fetch the previous page, if one exists."
    }
   }
  },
  "Page": {
   "id": "Page",
   "type": "object",
   "properties": {
    "author": {
     "type": "object",
     "description": "The author of this Page.",
     "properties": {
      "displayName": {
       "type": "string",
       "description": "The display name."
      },
      "id": {
       "type": "string",
       "description": "The identifier of the Page creator."
      },
      "image": {
       "type": "object",
       "description": "The page author's avatar.",
       "properties": {
        "url": {
         "type": "string",
         "description": "The page author's avatar URL."
        }
       }
      },
      "url": {
       "type": "string",
       "description": "The URL of the Page creator's Profile page."
      }
     }
    },
    "blog": {
           "type": "object",
           "description": "Data about the blog containing this Page.",
           "properties": {
            "id": {
             "type": "string",
             "description": "The identifier of the blog containing this page."
            }
           }
          },
          "content": {
           "type": "string",
           "description": "The body content of this Page, in HTML."
          },
          "etag": {
           "type": "string",
           "description": "Etag of the resource."
          },
          "id": {
           "type": "string",
           "description": "The identifier for this resource."
          },
          "kind": {
           "type": "string",
           "description": "The kind of this entity. Always blogger#page",
           "default": "blogger#page"
          },
          "published": {
           "type": "string",
           "description": "RFC 3339 date-time when this Page was published.",
           "format": "date-time"
          },
          "selfLink": {
           "type": "string",
           "description": "The API REST URL to fetch this resource from."
          },
          "status": {
           "type": "string",
           "description": "The status of the page for admin resources (either LIVE or
               DRAFT)."
          },
          "title": {
           "type": "string",
           "description": "The title of this entity. This is the name displayed in the
               Admin user interface."
          },
          "updated": {
           "type": "string",
           "description": "RFC 3339 date-time when this Page was last updated.",
           "format": "date-time"
          },
          "url": {
           "type": "string",
           "description": "The URL that this Page is displayed at."
          }
         }
        },
"PageList": {
 "id": "PageList",
 "type": "object",
 "properties": {
  "etag": {
   "type": "string",
   "description": "Etag of the response."
  },
  "items": {
   "type": "array",
   "description": "The list of Pages for a Blog.",
   "items": {
    "$ref": "Page"
   }
  },
  "kind": {
   "type": "string",
   "description": "The kind of this entity. Always blogger#pageList",
   "default": "blogger#pageList"
  },
  "nextPageToken": {
   "type": "string",
   "description": "Pagination token to fetch the next page, if one exists."
  }
 }
},
"Pageviews": {
 "id": "Pageviews",
 "type": "object",
 "properties": {
  "blogId": {
   "type": "string",
   "description": "Blog Id"
  },
  "counts": {
   "type": "array",
   "description": "The container of posts in this blog.",
   "items": {
    "type": "object",
    "properties": {
     "count": {
      "type": "string",
      "description": "Count of page views for the given time range",
      "format": "int64"
     },
     "timeRange": {
      "type": "string",
      "description": "Time range the given count applies to"
     }
    }
   }
  },
  "kind": {
   "type": "string",
   "description": "The kind of this entry. Always blogger#page_views",
   "default": "blogger#page_views"
```

```
    }
   }
  },
  "Post": {
   "id": "Post",
   "type": "object",
   "properties": {
    "author": {
     "type": "object",
     "description": "The author of this Post.",
     "properties": {
      "displayName": {
       "type": "string",
       "description": "The display name."
      },
      "id": {
       "type": "string",
       "description": "The identifier of the Post creator."
      },
      "image": {
       "type": "object",
       "description": "The Post author's avatar.",
       "properties": {
        "url": {
         "type": "string",
         "description": "The Post author's avatar URL."
        }
       }
      },
      "url": {
       "type": "string",
       "description": "The URL of the Post creator's Profile page."
      }
     }
    },
    "blog": {
     "type": "object",
     "description": "Data about the blog containing this Post.",
     "properties": {
      "id": {
       "type": "string",
       "description": "The identifier of the Blog that contains this Post."
      }
     }
    },
    "content": {
     "type": "string",
     "description": "The content of the Post. May contain HTML markup."
    },
    "customMetaData": {
     "type": "string",
     "description": "The JSON meta-data for the Post."
    },
    "etag": {
     "type": "string",
     "description": "Etag of the resource."
```

```
},
"id": {
 "type": "string",
 "description": "The identifier of this Post."
},
"images": {
 "type": "array",
 "description": "Display image for the Post.",
 "items": {
  "type": "object",
  "properties": {
   "url": {
    "type": "string"
   }
  }
 }
},
"kind": {
 "type": "string",
 "description": "The kind of this entity. Always blogger#post",
 "default": "blogger#post"
},
"labels": {
 "type": "array",
 "description": "The list of labels this Post was tagged with.",
 "items": {
  "type": "string"
 }
},
"location": {
 "type": "object",
 "description": "The location for geotagged posts.",
 "properties": {
  "lat": {
   "type": "number",
   "description": "Location's latitude.",
   "format": "double"
  },
  "lng": {
   "type": "number",
   "description": "Location's longitude.",
   "format": "double"
  },
  "name": {
   "type": "string",
   "description": "Location name."
  },
  "span": {
   "type": "string",
   "description": "Location's viewport span. Can be used when rendering a map preview."
  }
 }
},
"published": {
 "type": "string",
   "description": "RFC 3339 date-time when this Post was published.",
   "format": "date-time"
  },
  "readerComments": {
   "type": "string",
   "description": "Comment control and display setting for readers of this
       post."
  },
  "replies": {
   "type": "object",
   "description": "The container of comments on this Post.",
   "properties": {
    "items": {
     "type": "array",
     "description": "The List of Comments for this Post.",
     "items": {
      "$ref": "Comment"
     }
    },
    "selfLink": {
     "type": "string",
     "description": "The URL of the comments on this post."
    },
    "totalItems": {
     "type": "string",
     "description": "The count of comments on this post.",
     "format": "int64"
    }
   }
  },
  "selfLink": {
   "type": "string",
   "description": "The API REST URL to fetch this resource from."
  },
  "status": {
   "type": "string",
   "description": "Status of the post. Only set for admin-level requests"
  },
  "title": {
   "type": "string",
   "description": "The title of the Post."
  },
  "titleLink": {
   "type": "string",
   "description": "The title link URL, similar to atom's related link."
  },
  "updated": {
   "type": "string",
   "description": "RFC 3339 date-time when this Post was last updated.",
   "format": "date-time"
  },
  "url": {
   "type": "string",
   "description": "The URL where this Post is displayed."
  }
 }
```

```
  },
  "PostList": {
   "id": "PostList",
   "type": "object",
   "properties": {
    "etag": {
     "type": "string",
     "description": "Etag of the response."
    },
    "items": {
     "type": "array",
     "description": "The list of Posts for this Blog.",
     "items": {
      "$ref": "Post"
     }
    },
    "kind": {
     "type": "string",
     "description": "The kind of this entity. Always blogger#postList",
     "default": "blogger#postList"
    },
    "nextPageToken": {
     "type": "string",
     "description": "Pagination token to fetch the next page, if one exists."
    }
   }
  },
  "PostPerUserInfo": {
   "id": "PostPerUserInfo",
   "type": "object",
   "properties": {
    "blogId": {
     "type": "string",
     "description": "ID of the Blog that the post resource belongs to."
    },
    "hasEditAccess": {
     "type": "boolean",
     "description": "True if the user has Author level access to the post."
    },
    "kind": {
     "type": "string",
     "description": "The kind of this entity. Always blogger#postPerUserInfo",
     "default": "blogger#postPerUserInfo"
    },
    "postId": {
     "type": "string",
     "description": "ID of the Post resource."
    },
    "userId": {
     "type": "string",
     "description": "ID of the User."
    }
   }
  },
  "PostUserInfo": {
   "id": "PostUserInfo",
   "type": "object",
   "properties": {
    "kind": {
     "type": "string",
     "description": "The kind of this entity. Always blogger#postUserInfo",
     "default": "blogger#postUserInfo"
    },
    "post": {
     "$ref": "Post",
     "description": "The Post resource."
    },
    "post_user_info": {
     "$ref": "PostPerUserInfo",
     "description": "Information about a User for the Post."
    }
   }
  },
  "PostUserInfosList": {
   "id": "PostUserInfosList",
   "type": "object",
   "properties": {
    "items": {
     "type": "array",
     "description": "The list of Posts with User information for the post, for
         this Blog.",
     "items": {
      "$ref": "PostUserInfo"
     }
    },
    "kind": {
     "type": "string",
     "description": "The kind of this entity. Always blogger#postList",
     "default": "blogger#postUserInfosList"
    },
    "nextPageToken": {
     "type": "string",
     "description": "Pagination token to fetch the next page, if one exists."
    }
   }
  },
  "User": {
   "id": "User",
   "type": "object",
   "properties": {
    "about": {
     "type": "string",
     "description": "Profile summary information."
    },
    "blogs": {
     "type": "object",
     "description": "The container of blogs for this user.",
     "properties": {
      "selfLink": {
       "type": "string",
       "description": "The URL of the Blogs for this user."
      }
     }
    },
    "created": {
     "type": "string",
     "description": "The timestamp of when this profile was created, in seconds since epoch.",
     "format": "date-time"
    },
    "displayName": {
     "type": "string",
     "description": "The display name."
    },
    "id": {
     "type": "string",
     "description": "The identifier for this User."
    },
    "kind": {
     "type": "string",
     "description": "The kind of this entity. Always blogger#user",
     "default": "blogger#user"
    },
    "locale": {
     "type": "object",
     "description": "This user's locale",
     "properties": {
      "country": {
       "type": "string",
       "description": "The user's country setting."
      },
      "language": {
       "type": "string",
       "description": "The user's language setting."
      },
      "variant": {
       "type": "string",
       "description": "The user's language variant setting."
      }
     }
    },
    "selfLink": {
     "type": "string",
     "description": "The API REST URL to fetch this resource from."
    },
    "url": {
     "type": "string",
     "description": "The user's profile page."
    }
   }
  }
 },
 "resources": {
  "blogUserInfos": {
   "methods": {
    "get": {
     "id": "blogger.blogUserInfos.get",
     "path": "users/{userId}/blogs/{blogId}",
     "httpMethod": "GET",
     "description": "Gets one blog and user info pair by blogId and userId.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "The ID of the blog to get.",
       "required": true,
       "location": "path"
      },
      "maxPosts": {
       "type": "integer",
       "description": "Maximum number of posts to pull back with the blog.",
       "format": "uint32",
       "location": "query"
      },
      "userId": {
       "type": "string",
       "description": "ID of the user whose blogs are to be fetched. Either the
           word 'self' (sans quote marks) or the user's profile identifier.",
       "required": true,
       "location": "path"
      }
     },
     "parameterOrder": [
      "userId",
      "blogId"
     ],
     "response": {
      "$ref": "BlogUserInfo"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    }
   }
  },
  "blogs": {
   "methods": {
    "get": {
     "id": "blogger.blogs.get",
     "path": "blogs/{blogId}",
     "httpMethod": "GET",
     "description": "Gets one blog by ID.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "The ID of the blog to get.",
       "required": true,
       "location": "path"
      },
      "maxPosts": {
       "type": "integer",
       "description": "Maximum number of posts to pull back with the blog.",
       "format": "uint32",
       "location": "query"
      },
      "view": {
       "type": "string",
       "description": "Access level with which to view the blog. Note that some
           fields require elevated access.",
       "enum": [
        "ADMIN",
        "AUTHOR",
        "READER"
       ],
       "enumDescriptions": [
        "Admin level detail.",
        "Author level detail.",
        "Reader level detail."
       ],
       "location": "query"
      }
     },
     "parameterOrder": [
      "blogId"
     ],
     "response": {
      "$ref": "Blog"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    },
    "getByUrl": {
     "id": "blogger.blogs.getByUrl",
     "path": "blogs/byurl",
     "httpMethod": "GET",
     "description": "Retrieve a Blog by URL.",
     "parameters": {
      "url": {
       "type": "string",
       "description": "The URL of the blog to retrieve.",
       "required": true,
       "location": "query"
      },
      "view": {
       "type": "string",
       "description": "Access level with which to view the blog. Note that some
           fields require elevated access.",
       "enum": [
        "ADMIN",
        "AUTHOR",
        "READER"
       ],
       "enumDescriptions": [
        "Admin level detail.",
        "Author level detail.",
        "Reader level detail."
       ],
       "location": "query"
```

```
  }
 },
 "parameterOrder": [
  "url"
 ],
 "response": {
  "$ref": "Blog"
 },
 "scopes": [
  "https://www.googleapis.com/auth/blogger",
  "https://www.googleapis.com/auth/blogger.readonly"
 ]
},
"listByUser": {
 "id": "blogger.blogs.listByUser",
 "path": "users/{userId}/blogs",
 "httpMethod": "GET",
 "description": "Retrieves a list of blogs, possibly filtered.",
 "parameters": {
  "fetchUserInfo": {
   "type": "boolean",
   "description": "Whether the response is a list of blogs with per-user
       information instead of just blogs.",
   "location": "query"
  },
  "role": {
   "type": "string",
   "description": "User access types for blogs to include in the results,
       e.g. AUTHOR will return blogs where the user has author level access.
       If no roles are specified, defaults to ADMIN and AUTHOR roles.",
   "enum": [
    "ADMIN",
    "AUTHOR",
    "READER"
   ],
   "enumDescriptions": [
    "Admin role - Blogs where the user has Admin level access.",
    "Author role - Blogs where the user has Author level access.",
    "Reader role - Blogs where the user has Reader level access (to a private
        blog)."
   ],
   "repeated": true,
   "location": "query"
  },
  "status": {
   "type": "string",
   "description": "Blog statuses to include in the result (default: Live
       blogs only). Note that ADMIN access is required to view deleted
       blogs.",
   "default": "LIVE",
   "enum": [
    "DELETED",
    "LIVE"
   ],
   "enumDescriptions": [
    "Blog has been deleted by an administrator.",
        "Blog is currently live."
       ],
       "repeated": true,
       "location": "query"
      },
      "userId": {
       "type": "string",
       "description": "ID of the user whose blogs are to be fetched. Either the
           word 'self' (sans quote marks) or the user's profile identifier.",
       "required": true,
       "location": "path"
      },
      "view": {
       "type": "string",
       "description": "Access level with which to view the blogs. Note that some
           fields require elevated access.",
       "enum": [
        "ADMIN",
        "AUTHOR",
        "READER"
       ],
       "enumDescriptions": [
        "Admin level detail.",
        "Author level detail.",
        "Reader level detail."
       ],
       "location": "query"
      }
     },
     "parameterOrder": [
      "userId"
     ],
     "response": {
      "$ref": "BlogList"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    }
   }
  },
  "comments": {
   "methods": {
    "approve": {
     "id": "blogger.comments.approve",
     "path": "blogs/{blogId}/posts/{postId}/comments/{commentId}/approve",
     "httpMethod": "POST",
     "description": "Marks a comment as not spam.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "The ID of the Blog.",
       "required": true,
       "location": "path"
      },
      "commentId": {
       "type": "string",
       "description": "The ID of the comment to mark as not spam.",
       "required": true,
       "location": "path"
      },
      "postId": {
       "type": "string",
       "description": "The ID of the Post.",
       "required": true,
       "location": "path"
      }
     },
     "parameterOrder": [
      "blogId",
      "postId",
      "commentId"
     ],
     "response": {
      "$ref": "Comment"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger"
     ]
    },
    "delete": {
     "id": "blogger.comments.delete",
     "path": "blogs/{blogId}/posts/{postId}/comments/{commentId}",
     "httpMethod": "DELETE",
     "description": "Delete a comment by ID.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "The ID of the Blog.",
       "required": true,
       "location": "path"
      },
      "commentId": {
       "type": "string",
       "description": "The ID of the comment to delete.",
       "required": true,
       "location": "path"
      },
      "postId": {
       "type": "string",
       "description": "The ID of the Post.",
       "required": true,
       "location": "path"
      }
     },
     "parameterOrder": [
      "blogId",
      "postId",
      "commentId"
     ],
     "scopes": [
            "https://www.googleapis.com/auth/blogger"
           ]
          },
          "get": {
           "id": "blogger.comments.get",
           "path": "blogs/{blogId}/posts/{postId}/comments/{commentId}",
           "httpMethod": "GET",
           "description": "Gets one comment by ID.",
           "parameters": {
            "blogId": {
             "type": "string",
             "description": "ID of the blog to containing the comment.",
             "required": true,
             "location": "path"
            },
            "commentId": {
             "type": "string",
             "description": "The ID of the comment to get.",
             "required": true,
             "location": "path"
            },
            "postId": {
             "type": "string",
             "description": "ID of the post to fetch posts from.",
             "required": true,
             "location": "path"
            },
            "view": {
             "type": "string",
             "description": "Access level for the requested comment (default: READER).
                 Note that some comments will require elevated permissions, for example
                 comments where the parent posts which is in a draft state, or comments
                 that are pending moderation.",
             "enum": [
              "ADMIN",
              "AUTHOR",
              "READER"
             ],
             "enumDescriptions": [
              "Admin level detail",
              "Author level detail",
              "Admin level detail"
             ],
             "location": "query"
            }
           },
           "parameterOrder": [
            "blogId",
            "postId",
            "commentId"
           ],
           "response": {
            "$ref": "Comment"
           },
           "scopes": [
            "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    },
    "list": {
     "id": "blogger.comments.list",
     "path": "blogs/{blogId}/posts/{postId}/comments",
     "httpMethod": "GET",
     "description": "Retrieves the comments for a post, possibly filtered.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "ID of the blog to fetch comments from.",
       "required": true,
       "location": "path"
      },
      "endDate": {
       "type": "string",
       "description": "Latest date of comment to fetch, a date-time with RFC 3339
           formatting.",
       "format": "date-time",
       "location": "query"
      },
      "fetchBodies": {
       "type": "boolean",
       "description": "Whether the body content of the comments is included.",
       "location": "query"
      },
      "maxResults": {
       "type": "integer",
       "description": "Maximum number of comments to include in the result.",
       "format": "uint32",
       "location": "query"
      },
      "pageToken": {
       "type": "string",
       "description": "Continuation token if request is paged.",
       "location": "query"
      },
      "postId": {
       "type": "string",
       "description": "ID of the post to fetch posts from.",
       "required": true,
       "location": "path"
      },
      "startDate": {
       "type": "string",
       "description": "Earliest date of comment to fetch, a date-time with RFC
           3339 formatting.",
       "format": "date-time",
       "location": "query"
      },
      "status": {
       "type": "string",
       "enum": [
        "emptied",
        "live",
        "pending",
        "spam"
       ],
       "enumDescriptions": [
        "Comments that have had their content removed",
        "Comments that are publicly visible",
        "Comments that are awaiting administrator approval",
        "Comments marked as spam by the administrator"
       ],
       "repeated": true,
       "location": "query"
      },
      "view": {
       "type": "string",
       "description": "Access level with which to view the returned result. Note
           that some fields require elevated access.",
       "enum": [
        "ADMIN",
        "AUTHOR",
        "READER"
       ],
       "enumDescriptions": [
        "Admin level detail",
        "Author level detail",
        "Reader level detail"
       ],
       "location": "query"
      }
     },
     "parameterOrder": [
      "blogId",
      "postId"
     ],
     "response": {
      "$ref": "CommentList"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    },
    "listByBlog": {
     "id": "blogger.comments.listByBlog",
     "path": "blogs/{blogId}/comments",
     "httpMethod": "GET",
     "description": "Retrieves the comments for a blog, across all posts,
         possibly filtered.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "ID of the blog to fetch comments from.",
       "required": true,
       "location": "path"
      },
      "endDate": {
       "type": "string",
    "description": "Latest date of comment to fetch, a date-time with RFC 3339
        formatting.",
    "format": "date-time",
    "location": "query"
   },
   "fetchBodies": {
    "type": "boolean",
    "description": "Whether the body content of the comments is included.",
    "location": "query"
   },
   "maxResults": {
    "type": "integer",
    "description": "Maximum number of comments to include in the result.",
    "format": "uint32",
    "location": "query"
   },
   "pageToken": {
    "type": "string",
    "description": "Continuation token if request is paged.",
    "location": "query"
   },
   "startDate": {
    "type": "string",
    "description": "Earliest date of comment to fetch, a date-time with RFC
        3339 formatting.",
    "format": "date-time",
    "location": "query"
   },
   "status": {
    "type": "string",
    "enum": [
     "emptied",
     "live",
     "pending",
     "spam"
    ],
    "enumDescriptions": [
     "Comments that have had their content removed",
     "Comments that are publicly visible",
     "Comments that are awaiting administrator approval",
     "Comments marked as spam by the administrator"
    ],
    "repeated": true,
    "location": "query"
   }
  },
  "parameterOrder": [
   "blogId"
  ],
  "response": {
   "$ref": "CommentList"
  },
  "scopes": [
   "https://www.googleapis.com/auth/blogger",
   "https://www.googleapis.com/auth/blogger.readonly"
  ]
```

```
    },
    "markAsSpam": {
     "id": "blogger.comments.markAsSpam",
     "path": "blogs/{blogId}/posts/{postId}/comments/{commentId}/spam",
     "httpMethod": "POST",
     "description": "Marks a comment as spam.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "The ID of the Blog.",
       "required": true,
       "location": "path"
      },
      "commentId": {
       "type": "string",
       "description": "The ID of the comment to mark as spam.",
       "required": true,
       "location": "path"
      },
      "postId": {
       "type": "string",
       "description": "The ID of the Post.",
       "required": true,
       "location": "path"
      }
     },
     "parameterOrder": [
      "blogId",
      "postId",
      "commentId"
     ],
     "response": {
      "$ref": "Comment"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger"
     ]
    },
    "removeContent": {
     "id": "blogger.comments.removeContent",
     "path": "blogs/{blogId}/posts/{postId}/comments/{commentId}/removecontent",
     "httpMethod": "POST",
     "description": "Removes the content of a comment.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "The ID of the Blog.",
       "required": true,
       "location": "path"
      },
      "commentId": {
       "type": "string",
       "description": "The ID of the comment to delete content from.",
       "required": true,
       "location": "path"
      },
           "postId": {
            "type": "string",
            "description": "The ID of the Post.",
            "required": true,
            "location": "path"
           }
          },
          "parameterOrder": [
           "blogId",
           "postId",
           "commentId"
          ],
          "response": {
           "$ref": "Comment"
          },
          "scopes": [
           "https://www.googleapis.com/auth/blogger"
          ]
         }
        }
       },
       "pageViews": {
        "methods": {
         "get": {
          "id": "blogger.pageViews.get",
          "path": "blogs/{blogId}/pageviews",
          "httpMethod": "GET",
          "description": "Retrieve pageview stats for a Blog.",
          "parameters": {
           "blogId": {
            "type": "string",
            "description": "The ID of the blog to get.",
            "required": true,
            "location": "path"
           },
           "range": {
            "type": "string",
            "enum": [
             "30DAYS",
             "7DAYS",
             "all"
            ],
            "enumDescriptions": [
             "Page view counts from the last thirty days.",
             "Page view counts from the last seven days.",
             "Total page view counts from all time."
            ],
            "repeated": true,
            "location": "query"
           }
          },
          "parameterOrder": [
           "blogId"
          ],
          "response": {
           "$ref": "Pageviews"
          },
          "scopes": [
           "https://www.googleapis.com/auth/blogger"
          ]
         }
        }
       },
       "pages": {
        "methods": {
         "delete": {
          "id": "blogger.pages.delete",
          "path": "blogs/{blogId}/pages/{pageId}",
          "httpMethod": "DELETE",
          "description": "Delete a page by ID.",
          "parameters": {
           "blogId": {
            "type": "string",
            "description": "The ID of the Blog.",
            "required": true,
            "location": "path"
           },
           "pageId": {
            "type": "string",
            "description": "The ID of the Page.",
            "required": true,
            "location": "path"
           }
          },
          "parameterOrder": [
           "blogId",
           "pageId"
          ],
          "scopes": [
           "https://www.googleapis.com/auth/blogger"
          ]
         },
         "get": {
          "id": "blogger.pages.get",
          "path": "blogs/{blogId}/pages/{pageId}",
          "httpMethod": "GET",
          "description": "Gets one blog page by ID.",
          "parameters": {
           "blogId": {
            "type": "string",
            "description": "ID of the blog containing the page.",
            "required": true,
            "location": "path"
           },
           "pageId": {
            "type": "string",
            "description": "The ID of the page to get.",
            "required": true,
            "location": "path"
           },
           "view": {
            "type": "string",
       "enum": [
        "ADMIN",
        "AUTHOR",
        "READER"
       ],
       "enumDescriptions": [
        "Admin level detail",
        "Author level detail",
        "Reader level detail"
       ],
       "location": "query"
      }
     },
     "parameterOrder": [
      "blogId",
      "pageId"
     ],
     "response": {
      "$ref": "Page"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    },
    "insert": {
     "id": "blogger.pages.insert",
     "path": "blogs/{blogId}/pages",
     "httpMethod": "POST",
     "description": "Add a page.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "ID of the blog to add the page to.",
       "required": true,
       "location": "path"
      },
      "isDraft": {
       "type": "boolean",
       "description": "Whether to create the page as a draft (default: false).",
       "location": "query"
      }
     },
     "parameterOrder": [
      "blogId"
     ],
     "request": {
      "$ref": "Page"
     },
     "response": {
      "$ref": "Page"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger"
     ]
    },
    "list": {
     "id": "blogger.pages.list",
     "path": "blogs/{blogId}/pages",
     "httpMethod": "GET",
     "description": "Retrieves the pages for a blog, optionally including non-
        LIVE statuses.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "ID of the blog to fetch Pages from.",
       "required": true,
       "location": "path"
      },
      "fetchBodies": {
       "type": "boolean",
       "description": "Whether to retrieve the Page bodies.",
       "location": "query"
      },
      "maxResults": {
       "type": "integer",
       "description": "Maximum number of Pages to fetch.",
       "format": "uint32",
       "location": "query"
      },
      "pageToken": {
       "type": "string",
       "description": "Continuation token if the request is paged.",
       "location": "query"
      },
      "status": {
       "type": "string",
       "enum": [
        "draft",
        "live"
       ],
       "enumDescriptions": [
        "Draft (unpublished) Pages",
        "Pages that are publicly visible"
       ],
       "repeated": true,
       "location": "query"
      },
      "view": {
       "type": "string",
       "description": "Access level with which to view the returned result. Note
           that some fields require elevated access.",
       "enum": [
        "ADMIN",
        "AUTHOR",
        "READER"
       ],
       "enumDescriptions": [
        "Admin level detail",
        "Author level detail",
        "Reader level detail"
       ],
       "location": "query"
      }
     },
     "parameterOrder": [
      "blogId"
     ],
     "response": {
      "$ref": "PageList"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    },
    "patch": {
     "id": "blogger.pages.patch",
     "path": "blogs/{blogId}/pages/{pageId}",
     "httpMethod": "PATCH",
     "description": "Update a page. This method supports patch semantics.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "The ID of the Blog.",
       "required": true,
       "location": "path"
      },
      "pageId": {
       "type": "string",
       "description": "The ID of the Page.",
       "required": true,
       "location": "path"
      },
      "publish": {
       "type": "boolean",
       "description": "Whether a publish action should be performed when the page
           is updated (default: false).",
       "location": "query"
      },
      "revert": {
       "type": "boolean",
       "description": "Whether a revert action should be performed when the page
           is updated (default: false).",
       "location": "query"
      }
     },
     "parameterOrder": [
      "blogId",
      "pageId"
     ],
     "request": {
      "$ref": "Page"
     },
     "response": {
      "$ref": "Page"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger"
     ]
    },
    "publish": {
     "id": "blogger.pages.publish",
     "path": "blogs/{blogId}/pages/{pageId}/publish",
     "httpMethod": "POST",
     "description": "Publishes a draft page.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "The ID of the blog.",
       "required": true,
       "location": "path"
      },
      "pageId": {
       "type": "string",
       "description": "The ID of the page.",
       "required": true,
       "location": "path"
      }
     },
     "parameterOrder": [
      "blogId",
      "pageId"
     ],
     "response": {
      "$ref": "Page"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger"
     ]
    },
    "revert": {
     "id": "blogger.pages.revert",
     "path": "blogs/{blogId}/pages/{pageId}/revert",
     "httpMethod": "POST",
     "description": "Revert a published or scheduled page to draft state.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "The ID of the blog.",
       "required": true,
       "location": "path"
      },
      "pageId": {
       "type": "string",
       "description": "The ID of the page.",
       "required": true,
       "location": "path"
      }
     },
     "parameterOrder": [
      "blogId",
      "pageId"
     ],
   "response": {
    "$ref": "Page"
   },
   "scopes": [
    "https://www.googleapis.com/auth/blogger"
   ]
  },
  "update": {
   "id": "blogger.pages.update",
   "path": "blogs/{blogId}/pages/{pageId}",
   "httpMethod": "PUT",
   "description": "Update a page.",
   "parameters": {
    "blogId": {
     "type": "string",
     "description": "The ID of the Blog.",
     "required": true,
     "location": "path"
    },
    "pageId": {
     "type": "string",
     "description": "The ID of the Page.",
     "required": true,
     "location": "path"
    },
    "publish": {
     "type": "boolean",
     "description": "Whether a publish action should be performed when the page
         is updated (default: false).",
     "location": "query"
    },
    "revert": {
     "type": "boolean",
     "description": "Whether a revert action should be performed when the page
         is updated (default: false).",
     "location": "query"
    }
   },
   "parameterOrder": [
    "blogId",
    "pageId"
   ],
   "request": {
    "$ref": "Page"
   },
   "response": {
    "$ref": "Page"
   },
   "scopes": [
    "https://www.googleapis.com/auth/blogger"
   ]
  }
 }
},
"postUserInfos": {
 "methods": {
        "get": {
         "id": "blogger.postUserInfos.get",
         "path": "users/{userId}/blogs/{blogId}/posts/{postId}",
         "httpMethod": "GET",
         "description": "Gets one post and user info pair, by post ID and user ID.
             The post user info contains per-user information about the post, such as
             access rights, specific to the user.",
         "parameters": {
          "blogId": {
           "type": "string",
           "description": "The ID of the blog.",
           "required": true,
           "location": "path"
          },
          "maxComments": {
           "type": "integer",
           "description": "Maximum number of comments to pull back on a post.",
           "format": "uint32",
           "location": "query"
          },
          "postId": {
           "type": "string",
           "description": "The ID of the post to get.",
           "required": true,
           "location": "path"
          },
          "userId": {
           "type": "string",
           "description": "ID of the user for the per-user information to be fetched.
               Either the word 'self' (sans quote marks) or the user's profile
               identifier.",
           "required": true,
           "location": "path"
          }
         },
         "parameterOrder": [
          "userId",
          "blogId",
          "postId"
         ],
         "response": {
          "$ref": "PostUserInfo"
         },
         "scopes": [
          "https://www.googleapis.com/auth/blogger",
          "https://www.googleapis.com/auth/blogger.readonly"
         ]
        },
        "list": {
         "id": "blogger.postUserInfos.list",
         "path": "users/{userId}/blogs/{blogId}/posts",
         "httpMethod": "GET",
         "description": "Retrieves a list of post and post user info pairs, possibly
             filtered. The post user info contains per-user information about the
             post, such as access rights, specific to the user.",
         "parameters": {
```

```
"blogId": {
 "type": "string",
 "description": "ID of the blog to fetch posts from.",
 "required": true,
 "location": "path"
},
"endDate": {
 "type": "string",
 "description": "Latest post date to fetch, a date-time with RFC 3339
     formatting.",
 "format": "date-time",
 "location": "query"
},
"fetchBodies": {
 "type": "boolean",
 "description": "Whether the body content of posts is included. Default is
     false.",
 "default": "false",
 "location": "query"
},
"labels": {
 "type": "string",
 "description": "Comma-separated list of labels to search for.",
 "location": "query"
},
"maxResults": {
 "type": "integer",
 "description": "Maximum number of posts to fetch.",
 "format": "uint32",
 "location": "query"
},
"orderBy": {
 "type": "string",
 "description": "Sort order applied to search results. Default is
     published.",
 "default": "PUBLISHED",
 "enum": [
  "published",
  "updated"
 ],
 "enumDescriptions": [
  "Order by the date the post was published",
  "Order by the date the post was last updated"
 ],
 "location": "query"
},
"pageToken": {
 "type": "string",
 "description": "Continuation token if the request is paged.",
 "location": "query"
},
"startDate": {
 "type": "string",
 "description": "Earliest post date to fetch, a date-time with RFC 3339
     formatting.",
 "format": "date-time",
     "location": "query"
    },
    "status": {
     "type": "string",
     "enum": [
      "draft",
      "live",
      "scheduled"
     ],
     "enumDescriptions": [
      "Draft posts",
      "Published posts",
      "Posts that are scheduled to publish in future."
     ],
     "repeated": true,
     "location": "query"
    },
    "userId": {
     "type": "string",
     "description": "ID of the user for the per-user information to be fetched.
         Either the word 'self' (sans quote marks) or the user's profile
         identifier.",
     "required": true,
     "location": "path"
    },
    "view": {
     "type": "string",
     "description": "Access level with which to view the returned result. Note
         that some fields require elevated access.",
     "enum": [
      "ADMIN",
      "AUTHOR",
      "READER"
     ],
     "enumDescriptions": [
      "Admin level detail",
      "Author level detail",
      "Reader level detail"
     ],
     "location": "query"
    }
   },
   "parameterOrder": [
    "userId",
    "blogId"
   ],
   "response": {
    "$ref": "PostUserInfosList"
   },
   "scopes": [
    "https://www.googleapis.com/auth/blogger",
    "https://www.googleapis.com/auth/blogger.readonly"
   ]
  }
 }
},
    "posts": {
     "methods": {
      "delete": {
       "id": "blogger.posts.delete",
       "path": "blogs/{blogId}/posts/{postId}",
       "httpMethod": "DELETE",
       "description": "Delete a post by ID.",
       "parameters": {
        "blogId": {
         "type": "string",
         "description": "The ID of the Blog.",
         "required": true,
         "location": "path"
        },
        "postId": {
         "type": "string",
         "description": "The ID of the Post.",
         "required": true,
         "location": "path"
        }
       },
       "parameterOrder": [
        "blogId",
        "postId"
       ],
       "scopes": [
        "https://www.googleapis.com/auth/blogger"
       ]
      },
      "get": {
       "id": "blogger.posts.get",
       "path": "blogs/{blogId}/posts/{postId}",
       "httpMethod": "GET",
       "description": "Get a post by ID.",
       "parameters": {
        "blogId": {
         "type": "string",
         "description": "ID of the blog to fetch the post from.",
         "required": true,
         "location": "path"
        },
        "fetchBody": {
         "type": "boolean",
         "description": "Whether the body content of the post is included (default:
             true). This should be set to false when the post bodies are not
             required, to help minimize traffic.",
         "default": "true",
         "location": "query"
        },
        "fetchImages": {
         "type": "boolean",
         "description": "Whether image URL metadata for each post is included
             (default: false).",
         "location": "query"
        },
        "maxComments": {
```

```
       "type": "integer",
       "description": "Maximum number of comments to pull back on a post.",
       "format": "uint32",
       "location": "query"
      },
      "postId": {
       "type": "string",
       "description": "The ID of the post",
       "required": true,
       "location": "path"
      },
      "view": {
       "type": "string",
       "description": "Access level with which to view the returned result. Note that some fields require elevated access.",
       "enum": [
        "ADMIN",
        "AUTHOR",
        "READER"
       ],
       "enumDescriptions": [
        "Admin level detail",
        "Author level detail",
        "Reader level detail"
       ],
       "location": "query"
      }
     },
     "parameterOrder": [
      "blogId",
      "postId"
     ],
     "response": {
      "$ref": "Post"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    },
    "getByPath": {
     "id": "blogger.posts.getByPath",
     "path": "blogs/{blogId}/posts/bypath",
     "httpMethod": "GET",
     "description": "Retrieve a Post by Path.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "ID of the blog to fetch the post from.",
       "required": true,
       "location": "path"
      },
      "maxComments": {
       "type": "integer",
       "description": "Maximum number of comments to pull back on a post.",
       "format": "uint32",
       "location": "query"
      },
      "path": {
       "type": "string",
       "description": "Path of the Post to retrieve.",
       "required": true,
       "location": "query"
      },
      "view": {
       "type": "string",
       "description": "Access level with which to view the returned result. Note
           that some fields require elevated access.",
       "enum": [
        "ADMIN",
        "AUTHOR",
        "READER"
       ],
       "enumDescriptions": [
        "Admin level detail",
        "Author level detail",
        "Reader level detail"
       ],
       "location": "query"
      }
     },
     "parameterOrder": [
      "blogId",
      "path"
     ],
     "response": {
      "$ref": "Post"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    },
    "insert": {
     "id": "blogger.posts.insert",
     "path": "blogs/{blogId}/posts",
     "httpMethod": "POST",
     "description": "Add a post.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "ID of the blog to add the post to.",
       "required": true,
       "location": "path"
      },
      "fetchBody": {
       "type": "boolean",
       "description": "Whether the body content of the post is included with the
           result (default: true).",
       "default": "true",
       "location": "query"
      },
      "fetchImages": {
       "type": "boolean",
       "description": "Whether image URL metadata for each post is included in
           the returned result (default: false).",
       "location": "query"
      },
      "isDraft": {
       "type": "boolean",
       "description": "Whether to create the post as a draft (default: false).",
       "location": "query"
      }
     },
     "parameterOrder": [
      "blogId"
     ],
     "request": {
      "$ref": "Post"
     },
     "response": {
      "$ref": "Post"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger"
     ]
    },
    "list": {
     "id": "blogger.posts.list",
     "path": "blogs/{blogId}/posts",
     "httpMethod": "GET",
     "description": "Retrieves a list of posts, possibly filtered.",
     "parameters": {
      "blogId": {
       "type": "string",
       "description": "ID of the blog to fetch posts from.",
       "required": true,
       "location": "path"
      },
      "endDate": {
       "type": "string",
       "description": "Latest post date to fetch, a date-time with RFC 3339
           formatting.",
       "format": "date-time",
       "location": "query"
      },
      "fetchBodies": {
       "type": "boolean",
       "description": "Whether the body content of posts is included (default:
           true). This should be set to false when the post bodies are not
           required, to help minimize traffic.",
       "default": "true",
       "location": "query"
      },
      "fetchImages": {
       "type": "boolean",
       "description": "Whether image URL metadata for each post is included.",
       "location": "query"
```

```
},
"labels": {
 "type": "string",
 "description": "Comma-separated list of labels to search for.",
 "location": "query"
},
"maxResults": {
 "type": "integer",
 "description": "Maximum number of posts to fetch.",
 "format": "uint32",
 "location": "query"
},
"orderBy": {
 "type": "string",
 "description": "Sort search results",
 "default": "PUBLISHED",
 "enum": [
  "published",
  "updated"
 ],
 "enumDescriptions": [
  "Order by the date the post was published",
  "Order by the date the post was last updated"
 ],
 "location": "query"
},
"pageToken": {
 "type": "string",
 "description": "Continuation token if the request is paged.",
 "location": "query"
},
"startDate": {
 "type": "string",
 "description": "Earliest post date to fetch, a date-time with RFC 3339
     formatting.",
 "format": "date-time",
 "location": "query"
},
"status": {
 "type": "string",
 "description": "Statuses to include in the results.",
 "enum": [
  "draft",
  "live",
  "scheduled"
 ],
 "enumDescriptions": [
  "Draft (non-published) posts.",
  "Published posts",
  "Posts that are scheduled to publish in the future."
 ],
 "repeated": true,
 "location": "query"
},
"view": {
 "type": "string",
   "description": "Access level with which to view the returned result. Note
       that some fields require escalated access.",
   "enum": [
    "ADMIN",
    "AUTHOR",
    "READER"
   ],
   "enumDescriptions": [
    "Admin level detail",
    "Author level detail",
    "Reader level detail"
   ],
   "location": "query"
  }
 },
 "parameterOrder": [
  "blogId"
 ],
 "response": {
  "$ref": "PostList"
 },
 "scopes": [
  "https://www.googleapis.com/auth/blogger",
  "https://www.googleapis.com/auth/blogger.readonly"
 ]
},
"patch": {
 "id": "blogger.posts.patch",
 "path": "blogs/{blogId}/posts/{postId}",
 "httpMethod": "PATCH",
 "description": "Update a post. This method supports patch semantics.",
 "parameters": {
  "blogId": {
   "type": "string",
   "description": "The ID of the Blog.",
   "required": true,
   "location": "path"
  },
  "fetchBody": {
   "type": "boolean",
   "description": "Whether the body content of the post is included with the
       result (default: true).",
   "default": "true",
   "location": "query"
  },
  "fetchImages": {
   "type": "boolean",
   "description": "Whether image URL metadata for each post is included in
       the returned result (default: false).",
   "location": "query"
  },
  "maxComments": {
   "type": "integer",
   "description": "Maximum number of comments to retrieve with the returned
       post.",
   "format": "uint32",
            "location": "query"
           },
           "postId": {
            "type": "string",
            "description": "The ID of the Post.",
            "required": true,
            "location": "path"
           },
           "publish": {
            "type": "boolean",
            "description": "Whether a publish action should be performed when the post
                is updated (default: false).",
            "location": "query"
           },
           "revert": {
            "type": "boolean",
            "description": "Whether a revert action should be performed when the post
                is updated (default: false).",
            "location": "query"
           }
          },
          "parameterOrder": [
           "blogId",
           "postId"
          ],
          "request": {
           "$ref": "Post"
          },
          "response": {
           "$ref": "Post"
          },
          "scopes": [
           "https://www.googleapis.com/auth/blogger"
          ]
         },
         "publish": {
          "id": "blogger.posts.publish",
          "path": "blogs/{blogId}/posts/{postId}/publish",
          "httpMethod": "POST",
          "description": "Publishes a draft post, optionally at the specific time of
              the given publishDate parameter.",
          "parameters": {
           "blogId": {
            "type": "string",
            "description": "The ID of the Blog.",
            "required": true,
            "location": "path"
           },
           "postId": {
            "type": "string",
            "description": "The ID of the Post.",
            "required": true,
            "location": "path"
           },
           "publishDate": {
            "type": "string",
         "description": "Optional date and time to schedule the publishing of the
             Blog. If no publishDate parameter is given, the post is either
             published at the a previously saved schedule date (if present), or the
             current time. If a future date is given, the post will be scheduled to
             be published.",
         "format": "date-time",
         "location": "query"
        }
       },
       "parameterOrder": [
        "blogId",
        "postId"
       ],
       "response": {
        "$ref": "Post"
       },
       "scopes": [
        "https://www.googleapis.com/auth/blogger"
       ]
      },
      "revert": {
       "id": "blogger.posts.revert",
       "path": "blogs/{blogId}/posts/{postId}/revert",
       "httpMethod": "POST",
       "description": "Revert a published or scheduled post to draft state.",
       "parameters": {
        "blogId": {
         "type": "string",
         "description": "The ID of the Blog.",
         "required": true,
         "location": "path"
        },
        "postId": {
         "type": "string",
         "description": "The ID of the Post.",
         "required": true,
         "location": "path"
        }
       },
       "parameterOrder": [
        "blogId",
        "postId"
       ],
       "response": {
        "$ref": "Post"
       },
       "scopes": [
        "https://www.googleapis.com/auth/blogger"
       ]
      },
      "search": {
       "id": "blogger.posts.search",
       "path": "blogs/{blogId}/posts/search",
       "httpMethod": "GET",
       "description": "Search for a post.",
       "parameters": {
            "blogId": {
             "type": "string",
             "description": "ID of the blog to fetch the post from.",
             "required": true,
             "location": "path"
            },
            "fetchBodies": {
             "type": "boolean",
             "description": "Whether the body content of posts is included (default:
                 true). This should be set to false when the post bodies are not
                 required, to help minimize traffic.",
             "default": "true",
             "location": "query"
            },
            "orderBy": {
             "type": "string",
             "description": "Sort search results",
             "default": "PUBLISHED",
             "enum": [
              "published",
              "updated"
             ],
             "enumDescriptions": [
              "Order by the date the post was published",
              "Order by the date the post was last updated"
             ],
             "location": "query"
            },
            "q": {
             "type": "string",
             "description": "Query terms to search this blog for matching posts.",
             "required": true,
             "location": "query"
            }
           },
           "parameterOrder": [
            "blogId",
            "q"
           ],
           "response": {
            "$ref": "PostList"
           },
           "scopes": [
            "https://www.googleapis.com/auth/blogger",
            "https://www.googleapis.com/auth/blogger.readonly"
           ]
          },
          "update": {
           "id": "blogger.posts.update",
           "path": "blogs/{blogId}/posts/{postId}",
           "httpMethod": "PUT",
           "description": "Update a post.",
           "parameters": {
            "blogId": {
             "type": "string",
             "description": "The ID of the Blog.",
   "required": true,
   "location": "path"
  },
  "fetchBody": {
   "type": "boolean",
   "description": "Whether the body content of the post is included with the
       result (default: true).",
   "default": "true",
   "location": "query"
  },
  "fetchImages": {
   "type": "boolean",
   "description": "Whether image URL metadata for each post is included in
       the returned result (default: false).",
   "location": "query"
  },
  "maxComments": {
   "type": "integer",
   "description": "Maximum number of comments to retrieve with the returned
       post.",
   "format": "uint32",
   "location": "query"
  },
  "postId": {
   "type": "string",
   "description": "The ID of the Post.",
   "required": true,
   "location": "path"
  },
  "publish": {
   "type": "boolean",
   "description": "Whether a publish action should be performed when the post
       is updated (default: false).",
   "location": "query"
  },
  "revert": {
   "type": "boolean",
   "description": "Whether a revert action should be performed when the post
       is updated (default: false).",
   "location": "query"
  }
 },
 "parameterOrder": [
  "blogId",
  "postId"
 ],
 "request": {
  "$ref": "Post"
 },
 "response": {
  "$ref": "Post"
 },
 "scopes": [
  "https://www.googleapis.com/auth/blogger"
 ]
}
```

```
   }
  },
  "users": {
   "methods": {
    "get": {
     "id": "blogger.users.get",
     "path": "users/{userId}",
     "httpMethod": "GET",
     "description": "Gets one user by ID.",
     "parameters": {
      "userId": {
       "type": "string",
       "description": "The ID of the user to get.",
       "required": true,
       "location": "path"
      }
     },
     "parameterOrder": [
      "userId"
     ],
     "response": {
      "$ref": "User"
     },
     "scopes": [
      "https://www.googleapis.com/auth/blogger",
      "https://www.googleapis.com/auth/blogger.readonly"
     ]
    }
   }
  }
 }
}
```