1. 閱讀英文 Google Photos API 文件，並回答問題
https://vivipic.notion.site/Google-Photos-API-208ff38d22ec473ca0a49d729d8400b4

a. 某位使用者在 Google Photos 儲存的所有照片

列出程式庫內容
https://developers.google.com/photos/library/guides/list#listing-library-contents

```
try {
  // Make a request to list all media items in the user's library
  // Iterate over all the retrieved media items
  // Pagination is handled automatically
  ListMediaItemsPagedResponse response = photosLibraryClient.listMediaItems();
  for (MediaItem item : response.iterateAll()) {
    // Get some properties of a media item
    String id = item.getId();
    String description = item.getDescription();
    String mimeType = item.getMimeType();
    String productUrl = item.getProductUrl();
    String filename = item.getFilename();
  }
} catch (ApiException e) {
  // Handle error
}
```

- 使用 photosLibraryClient 呼叫 listMediaItems 方法，它返回一個 ListMediaItemsPagedResponse 物件。
- 使用 for 迴圈遍歷 response 中的所有媒體項目，item.getId() 取得媒體項目的唯一識別碼

取得媒體項目
https://developers.google.com/photos/library/guides/access-media-items#get-media-item

The photo property contains metadata for photo items.

```
try {
  // Get a media item using its ID
  String mediaItemId = "...";
  MediaItem item = photosLibraryClient.getMediaItem(mediaItemId);
  // Get some properties from the retrieved media item
  String id = item.getId();
  String description = item.getDescription();
  String baseUrl = item.getBaseUrl();
  String productUrl = item.getProductUrl();
  // ...
  if (item.hasMediaMetadata()) {
    // The media item contains additional metadata, such as the height and width
    MediaMetadata metadata = item.getMediaMetadata();
    long height = metadata.getHeight();
    long width = metadata.getWidth();
    Timestamp creationTime = metadata.getCreationTime();
    // ...
    if (metadata.hasPhoto()) {
      // This media item is a photo and has additional photo metadata
      Photo photoMetadata = metadata.getPhoto();
      String cameraMake = photoMetadata.getCameraMake();
      String cameraModel = photoMetadata.getCameraModel();
      float aperture = photoMetadata.getApertureFNumber();
      int isoEquivalent = photoMetadata.getIsoEquivalent();
      // ...
    }
  }
  if (item.hasContributorInfo()) {
    // A user has contributed this media item  to a shared album
    ContributorInfo contributorInfo = item.getContributorInfo();
    String profilePictureBaseUrl = contributorInfo.getProfilePictureBaseUrl();
    String displayName = contributorInfo.getDisplayName();
  }
} catch (ApiException e) {
  // Handle error
}
```

- 使用 photosLibraryClient 呼叫 getMediaItem 方法，傳入 mediaItemId 作為參數。
- 使用 item.hasMediaMetadata() 檢查是否存在媒體項目的附加資料 metadata，使用 metadata.hasPhoto() 檢查媒體項目是否是一張照片，如果是照片，則可以從 metadata 中取得照片相關資訊。

b. 某位使用者在 Google Photos 儲存的相簿名稱，以及各相簿裏面的照片

列出專輯

https://developers.google.com/photos/library/guides/list#listing-albums

```
try {
  // Make a request to list all albums in the user's library
  // Iterate over all the albums in this list
  // Pagination is handled automatically
  ListAlbumsPagedResponse response = photosLibraryClient.listAlbums();

  for (Album album : response.iterateAll()) {
    // Get some properties of an album
    String id = album.getId();
    String title = album.getTitle();
    String productUrl = album.getProductUrl();
    String coverPhotoBaseUrl = album.getCoverPhotoBaseUrl();
    // The cover photo media item id field may be empty
    String coverPhotoMediaItemId = album.getCoverPhotoMediaItemId();
    boolean isWritable = album.getIsWriteable();
    long mediaItemsCount = album.getMediaItemsCount();
  }

} catch (ApiException e) {
  // Handle error
}
```

- 使用 photosLibraryClient 呼叫 listAlbums 方法，它返回一個 ListAlbumsPagedResponse 物件。
- 使用 for 迴圈遍歷 response 中的所有專輯項目，album.getId() 取得專輯項目的唯一識別碼。

列出專輯內容
https://developers.google.com/photos/library/guides/list#listing-album-contents

```
try {
  // Make a request to list all media items in an album
  // Provide the ID of the album as a parameter in the searchMediaItems call
  // Iterate over all the retrieved media items
  SearchMediaItemsPagedResponse response = photosLibraryClient.searchMediaItems(albumId);

  for (MediaItem item : response.iterateAll()) {
    // ...
  }

} catch (ApiException e) {
  // Handle error
}
```

- 使用 photosLibraryClient 呼叫 searchMediaItems 方法，它返回一個 SearchMediaItemsPagedResponse 物件。
- 使用 for 迴圈遍歷 response 中的所有媒體項目，item.getId() 取得媒體項目的唯一識別碼

取得媒體項目
https://developers.google.com/photos/library/guides/access-media-items#get-media-item

The photo property contains metadata for photo items.

```
try {
  // Get a media item using its ID
  String mediaItemId = "...";
  MediaItem item = photosLibraryClient.getMediaItem(mediaItemId);
  // Get some properties from the retrieved media item
  String id = item.getId();
  String description = item.getDescription();
  String baseUrl = item.getBaseUrl();
  String productUrl = item.getProductUrl();
  // ...
  if (item.hasMediaMetadata()) {
    // The media item contains additional metadata, such as the height and width
    MediaMetadata metadata = item.getMediaMetadata();
    long height = metadata.getHeight();
    long width = metadata.getWidth();
    Timestamp creationTime = metadata.getCreationTime();
    // ...
    if (metadata.hasPhoto()) {
      // This media item is a photo and has additional photo metadata
      Photo photoMetadata = metadata.getPhoto();
      String cameraMake = photoMetadata.getCameraMake();
      String cameraModel = photoMetadata.getCameraModel();
      float aperture = photoMetadata.getApertureFNumber();
      int isoEquivalent = photoMetadata.getIsoEquivalent();
      // ...
    }
  }
  if (item.hasContributorInfo()) {
    // A user has contributed this media item  to a shared album
    ContributorInfo contributorInfo = item.getContributorInfo();
    String profilePictureBaseUrl = contributorInfo.getProfilePictureBaseUrl();
    String displayName = contributorInfo.getDisplayName();
  }
} catch (ApiException e) {
  // Handle error
}
```

- 使用 photosLibraryClient 呼叫 getMediaItem 方法，傳入 mediaItemId 作為參數。
- 使用 item.hasMediaMetadata() 檢查是否存在媒體項目的附加資料 metadata，使用 metadata.hasPhoto() 檢查媒體項目是否是一張照片，如果是照片，則可以從 metadata 中取得照片相關資訊。

上述兩題補充：

a. 如無需取得檔案中的中繼資料，或處理影片類型的媒體項目，則可以使用媒體類型先進行過濾，獲取照片類型的媒體項目，再進行遍歷取得所有照片項目的相關資訊。

媒體類型
https://developers.google.com/photos/library/guides/apply-filters#media-types

```
// Create a new MediaTypeFilter for Photo media items
MediaTypeFilter mediaType = MediaTypeFilter.newBuilder()
    .addMediaTypes(MediaType.PHOTO)
    .build();
// Create a new Filters object
Filters filters = Filters.newBuilder()
    .setMediaTypeFilter(mediaType)
    .build();
// Specify the Filters object in the searchMediaItems call
SearchMediaItemsPagedResponse response = photosLibraryClient.searchMediaItems(filters);
```

- 創建一個 MediaTypeFilter，用於篩選媒體類型為照片的媒體項目。
- 創建一個 Filters 物件，並將此過濾器設定為上述建立的媒體類型過濾器。
- 調用 searchMediaItems 方法，以搜尋符合過濾條件為照片的媒體項目。

b. 官方建議如果 albumId 已設定，就不應在 mediaItems:search 要求中使用篩選器。如果在設定相簿 ID 時使用篩選器，則會傳回 INVALID_ARGUMENT 錯誤 (400)。

可用的篩選器
https://developers.google.com/photos/library/guides/apply-filters#available-filters

Available filters

You can search a user's library for specific kinds of media. For example, you might only want items that are of animals, from a certain day, or you may wish to exclude photos of receipts. You can exclude or include specific items by applying filters to an album or library listing. There are five available filters based on media item properties:

- Content categories (`includedContentCategories`, `excludedContentCategories`)
- Dates and date ranges (`dates`, `ranges`)
- Features (`featureFilter`)
- Media types (`mediaTypeFilter`)
- Archived state (`includeArchivedMedia`)

Filters shouldn't be used in a `mediaItems:search` request if the `albumId` is set. If a filter is used when the albumId is set, an `INVALID_ARGUMENT` error (400) is returned.

Results are sorted according to the media item creation time. The sort order can be modified for queries using date filters.

Allow some time for newly uploaded media to appear in your searches. The media appears in unfiltered searches immediately.

Media items that have a date in the future don't appear in filtered searches. They appear in unfiltered searches and album searches.