Image search

Introduction

We have already discussed basic web search on a number of occasions. In this task you will need to extend it to image search with local storage in a database. The general algorithm of the program will be the following:

1. User enters a query string
2. The program sends a request to the Google Custom Search API to retrieve information about relevant images. Each item in the search result contains a URL pointing to an image thumbnail (small version of the image)
3. The program downloads all thumbnails and saves them on the local hard drive. A function for downloading a single image is already given to you. For each file a database entry should also be created.
4. Images are displayed in the UI (the XAML part is already implemented)

Part 1. 3 points

Refresh the Google Custom Search API. You are given a slightly modified version of a basic web search project (from one of the seminars). You need to restore your API key and Search Engine ID.

The XAML markup has been changed to display images, so you won't see any results in the UI. Just set a breakpoint to make sure your keys work correctly and no exception is thrown.

To change the API call from web page search to image search you need to modify two things:

- Add an additional parameter to the query URL: searchType=image
- Turn on image search on the search engine setup page (see figure 1)

Take a look at the structure of the incoming json data. You need to extract the thumbnailLink property. It is a URL pointing to a small version of the image, which you will then download.

Using the DownloadFile function implement the logic of downloading of all thumbnails to a local hard drive. Files are downloaded to the **images** folder

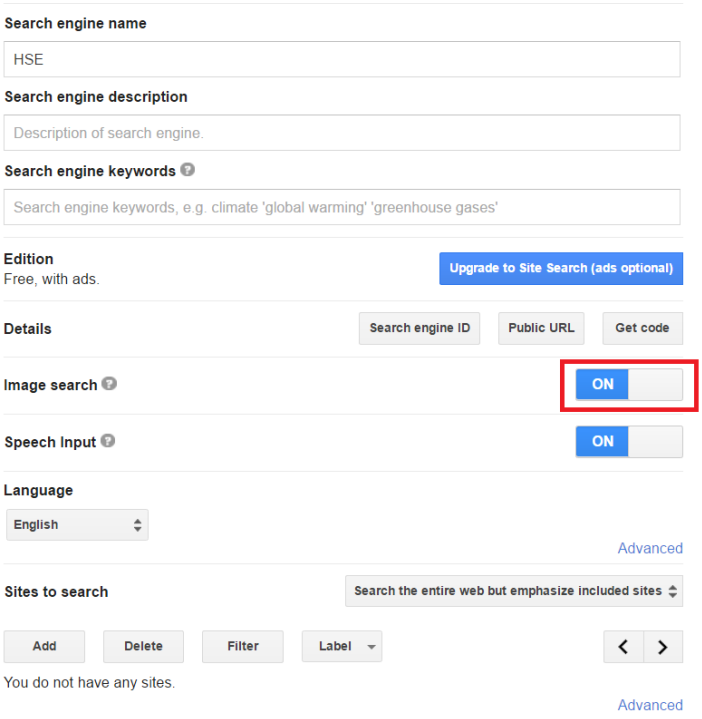

Figure 1: Custom search engine setup

inside bin/debug. The function returns the absolute path to a saved file on the local drive. When using the function make sure you follow the asynchronous pattern.

Use breakpoints and step-by-step debugging to localize errors if you come across any.

To display thumbnails in the UI you need to fill the Path property of each SearchResult so that it points to the corresponding image file on disk (see figure 2)

Part 2. 2-4 points

Implement the Context class to store the following information in a local database:

- Query
- Title
- Description
- Url of the full image

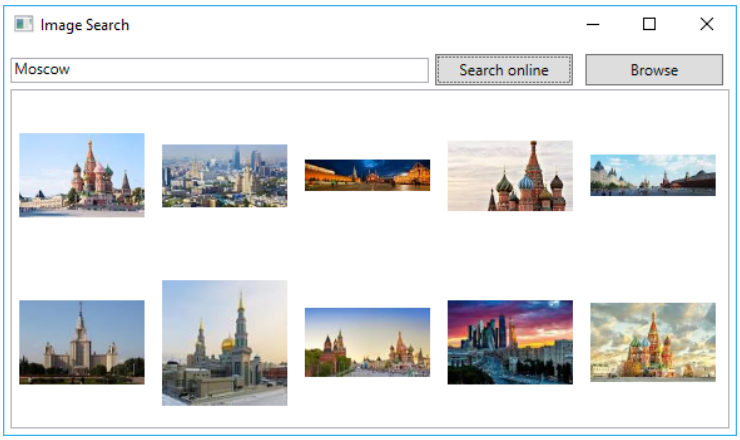

Figure 2: Image Search UI

- Path to a thumbnail on the local drive

To get the maximal 4 points divide this information into two entities - request (query) and response item (other 4 properties - title, description, url, path) with a many-to-many relationship (a query has multiple associated images and a single image can be related to multiple queries). A simpler option would be to store everything in a single entity.

Add the necessary logic to the GoogleService class to save information about downloaded images to the database after each request.

You don't need to save images in the DB, just absolute paths leading to them (strings)

Part 3. 3 points

Decouple dependencies between program components. Make an interface for a search service and two classes implementing it - google service and database service. The database service should retrieve all related items from the database.