DashLane interview assignment for Chen Dai: Icon Crawler

========

Icon Crawler is a SOAP WSDL web services which is implemented in C#.net environment which return an icon link by giving a domain name.

How to use:

1) Import the web services references by WSDL or the direct link
2) Call the web method : `string GetIcon(string domain)`

Features

- Return an icon image link from a domain name.

Installation

Use the install package, requires the framework 3.5 on IIS.

Presentation

I decided not to return any image content to the final user, just a link to the icon image on the domain web site. For 2 major reasons:

- Legal reason: the distribution of images is always sensible because of copyright issue, especially the icon of a corporation.
- Technical reason: to save the image content exchange from IconCrawler to user and not to save image content locally. That does not change a lot for user, he/she just need to fetch the link returned by web services instead of fetching directly an image content.

The major steps are:

0) Normalized the domain: add http/https if not present.
1) Get a response from domain root. If not found, return empty response.
2) Find out if we already have an image in cache. If yes return it.
3) Try to fetch a potential image list. If fetching succeed, return it.
4) Try to scan the root page and fetch a <Link> with attribute relationship that contains “icon”. If fetching succeed, return it.

- Cache strategy:

It uses a cache in memory, if the domain exists in cache and domain server still return 304 not modified with the last request date, return the image link in cache otherwise update the last request date to now or look for the new icon image.

Ideal cache system for me is a database, the synchronized system is to request the server domain by specifying the IfModifiedSince in http request as my implementation with memory cache. If the server respond with 304 not modified, the image is still available and has not been change. If the server respond with 200 ok, the image is available but should change the last request date in cache. All other responses, we have to re-fetch a new icon.

- Same requests:

If there are 2 requests for same domain which arrived within a short time interval, there is no need to process the second request, just to wait the first request finish the treatment and return the same response for 2 requests. I did not find an existing implementation on IIS and .net.

- Best image:

The best strategy to choose an image is by analyzing the user agent of http request, if that comes from a phablet, it’s better to prior to fetch a specific size of image which better adapted to its resolution.

If the pattern of usage is that many different domains requests. I don’t think that to parallelize the search is necessary, there will be only one user agent in every request, and the worst case is we don’t find the best image, the second and final choice is to look for any icon image. If the pattern of usage is that a few of domains are requested, the best way to manage is to cache all the icon images for every domain and parallelize the fetches of every image.

- Rate limit:

If there is no authentication system to the service, the rate limit strategy can be a requests/ip/day rate limit.

Users from same IP can only make N requests per day.

If there is an authentication system, the rate limit strategy can be a requests/ip/day rate limit with a credit system if one user wants to make more than N requests per day.

An authenticated user consume its credit if he/she made more than N requests per day.

- Architecture for large scale of users

1) Use load balancing for incoming request which distribute the charge to N web servers.
2) Data bases should be partitioned: for instance, Master/Slave - single master server for all write operations, and one or many additional Slave servers that provide read-only operations.
3) Memcached the application layer: cached database query results, objects that can be shared.

- Other thoughts

Potential issues:

1) All issues that because we don't have a copy of image and transfer directly to user. For instance: the internet connection on IconCrawler application's server is down, the domain server can have a wrong mime type description for image, etc.

Thoughts:

2) For those domains that no icon can be found, is there any way to get a logo file and create an icon file?
3) Does this kind of service exist already? Check out what and how they do? (Google S2 seems to do the exactly same thing)

Contribute

- Time spent: on implementation and documentation: 3h for Chen Dai, on thought: 1h for serious and some unaccountable time at night for Chen Dai☺.

- Source Code: https://github.com/ChenDai/DashlaneAssignment

Reference

Html agility pack: http://htmlagilitypack.codeplex.com/

Support

If you are having issues, please let us know.

Mailing address located at: chen.dai.thierry@gmail.com

License
