

Tổng quan ứng dụng từ điển Anh - Anh.

Tính năng :

- Tra từ
- AutoComplete
- Hình ảnh kèm theo từ
- Dành cho admin :
 - + Chỉnh sửa, xóa từ, thêm từ
 - + Top các từ truy cập nhiều nhất

Các tính năng trừ Cache được gọi thông qua Thrift. Thrift Client sử dụng TClientPool, borrow client mỗi khi hàm được gọi.

```
service DictionaryService
{
    string getTranslation(1:string word),
    set<string> getAutoCompletes(1:string word),
    set<string> getImage(1: string word,),
    bool updateTranslation (1:string word, 2:string translation),
    bool deleteTranslation (1:string word),
    i32 getAuthenLevel(1:string username, 2:string password),
    void addUser(1:string username, 2:string password, 3:i32 role),
    i32 getReadTimes(1:string word),
    bool addReadTimes(1:string word, 2:i32 times),
    string getTopRead(),
}
```

Cấu trúc dữ liệu :

- DataBase :
 - Nghĩa từ : **Key** < Prefix + word > -- **Value** < Translation >
 - Hình ảnh : **Key** < Prefix + word > -- **Value** < URLS >
 - User : **Key** < Prefix + UserName > -- **Value** < Object User được serialized >
 - Biến đếm số lần truy cập từ : **Key** < Prefix + word > -- **Value** < count >
 - List từ truy cập nhiều nhất : **Key** <"TopLis"> -- **Value** < JSON Object các từ + count >
- Elastic Search (dùng cho autoComplete) :
Index (int) - **Value** (String : word)
- ZiCache :
 - Nghĩa từ : tương tự db
 - Hình ảnh : tương tự db

- Biến đếm số lần truy cập từ : tương tự db

API

- /api : api cho translate, autocomplete, images, các tính năng admin
- /view : giao diện chính
- /authen : đăng nhập, đăng xuất

Cấu trúc source code

Packages :

- **ZiCache** : Quản lý cache
- **dictionaryService.Thrift** : Các file do Thrift sinh ra.
- **elasticSearch** : Connect tới server elastic search, các tác vụ tìm kiếm theo prefix, thêm xóa từ trên ES
- **eventBus** : thông qua ZEvbusSubsys, dành riêng tác vụ crawl hình ảnh.
- **testapp.handlers.Thrift** : Handler cho thrift Server.
- **testapp.handlers.webapp** : Handler cho web.
- **testapp.model.Thrift** : Model cho các tác vụ truy xuất tới DB và ES để host trên thriftServer.
- **testapp.model.webapp** : Model cho webapp, truy xuất trực tiếp package Cache, các tác vụ khác gọi qua thriftClient.
- **Một số package mặc định khác*

Cách thức cập nhật biến đếm số lần search 1 từ

Mỗi một từ khi được tìm sẽ cập nhật trên cache để tránh hit db quá nhiều.

→ Khi vượt quá một số threshold sẽ cập nhật với db.

DB lưu trữ biến đếm (count) của tất cả các từ và một topList các từ được truy cập nhiều nhất.

→ Khi nhận update từ cache sẽ cập nhật lại biến đếm của từ và so sánh lại topList

Hàm update count trong Model :

```
private void updateCount(String word) {

    // synchronized to prevent many concurrent request

    synchronized (this) {
        int currentCount = CACHE.getCount(word);

        //if count not found on Cache, check and sync with db
        //happen when reset cache
    }
}
```

```

    if (currentCount == -1) {
        currentCount = CLIENT.getCount(word);
    }

    // store count on cache, write to db when reach a threshold
    currentCount++;
    if (currentCount % countThreshold == 0) {
        CLIENT.addCount(word, countThreshold);
    }
    CACHE.setCount(word, currentCount);
}
}

```

- TopList lưu trong db chứa nhiều từ hơn topList hiển thị cho admin, dùng để warmup cache

Cache được warmup bằng topList