**Hướng dẫn cài đặt websites**

**ECommerceProject và WebProviders trên Netbeans**

**A. Cài đặt Netbeans, MySQL và tạo New Project**

Để cài đặt, chúng ta cần có một số phần mềm được cài đặt sẵn trên máy tính. Những phần mềm này chúng tôi đề nghị tham khảo các đường link dưới đây để thực hiện trước khi tiến hành cài đặt websites: ECommerceProject và WebProviders

Lưu ý: chúng ta phải tự cài đặt thành công tất cả những chương trình sau đây:

* Cấu hình Java: <http://o7planning.org/vi/10377/huong-dan-cai-dat-va-cau-hinh-java>
* NetBeans: <http://netbeans.org/downloads/start.html?platform=windows&lang=en&option=javaee>
* MySQL Community Server: <http://dev.mysql.com/downloads/mysql/>

Các bước cài đặt websites: chúng ta nên copy tất cả thư mục trong đĩa CD vào một nơi trên máy tính của người dùng.

* **Bước 1: Tạo và Import CSDL vào MySQL.**

Mở Workbench được tích hợp sẵn trong lúc cài đặt MySQL Server (chúng ta vẫn có thể sử dụng một số chương trình khác như PHPAdmin hay Navicat), sau đó đăng nhập vào hệ thống MySQL bằng tài khoản khi cài MySQL Server.

Trên thanh công cụ, tạo một CSDL mới, có tên là *ecmrproj.* Sau đó import tập tin *“e-commerce.sql”* được cung cấp sẵn trong đĩa CD vào CSDL ecmrproj vừa mới tạo.

Lưu ý: tên CSDL phải bắt buộc phải là “*ecmrproj”,* nếu đổi tên khác thì quá trình chạy thử chương trình sẽ không thành công.

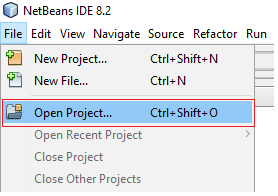
Chúng ta có thể thảm khảo cách import dữ liệu tại đây: <https://goweb.vn/kien-thuc-web-vi/thiet-ke-web/cach-export-va-import-bang-cau-lenh-co-so-du-lieu-mysql-434.html>

* **Bước 2: Cài đặt project website trên NetBeans**

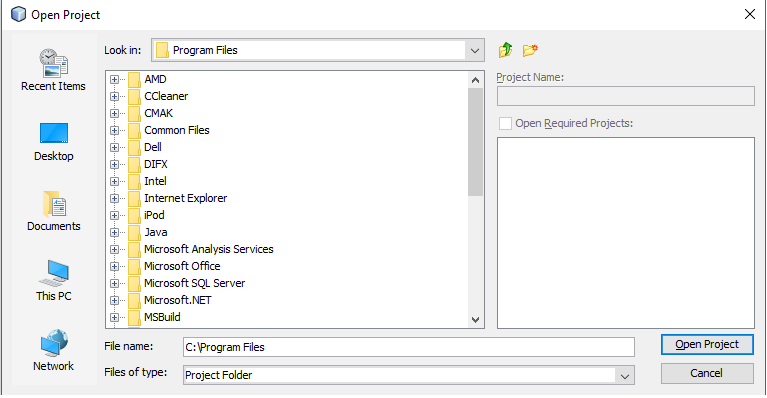
Có 2 cách để cài đặt một dự án web có sẵn trên NetBeans:

* **Cách 1: Mở trực tiếp file thư mục project NetBeans:**

Ở màn hình chính của NetBeans, chúng ta chọn **File → Open Project**…



Sau đó, cửa sổ chọn dự án hiện ra như dưới đây:

****

Chúng ta chọn đến thư mục dự án **ECommerceProject** được cung cấp sẵn và Open Project. Với **WebProviders**, chúng ta làm tương tự.

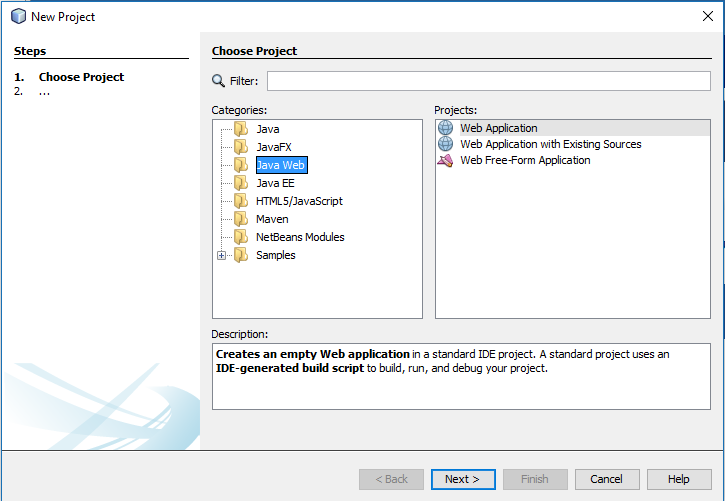
Lưu ý 1: Sau khi mở thành công dự án, một số NetBean phiên bản mới sẽ tự động cài đặt **Application Server** là **Tomcat** hay **Glassfish** cho dự án, nếu như không thì chúng tôi khuyến khích các bạn chọn Tomcat.

Lưu ý 2: Sau bước trên, một số thư mục class có thể bị lỗi, điều đó xảy ra là do chúng ta chưa thêm một số thư viện Jar, chúng tôi có hướng dẫn việc add thêm thư viện ở bước 3.

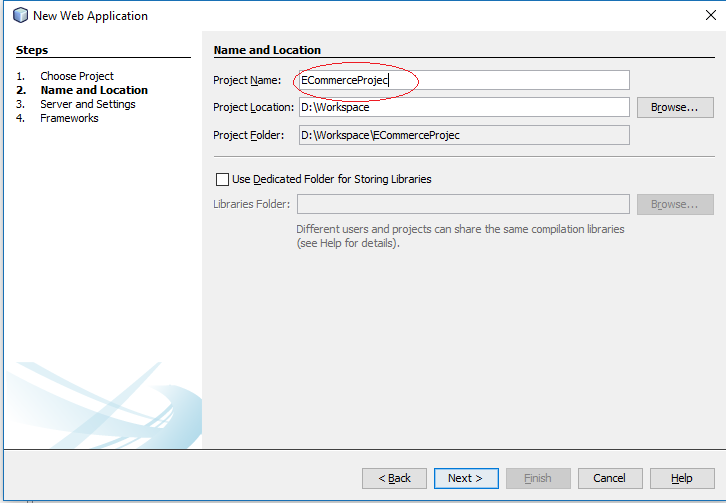
* **Cách 2: Tạo và sao chép các thư mục project.**

Chúng ta có thể tạo một web project theo các bước sau:

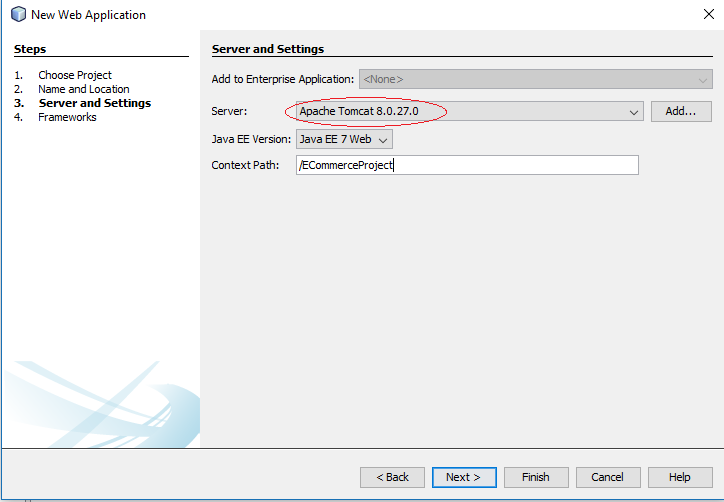
**1.** Học viện chọn **File → New Project** trên thanh công cụ của NetBeans, cửa sổ tạo project sẽ hiện ra như sau:



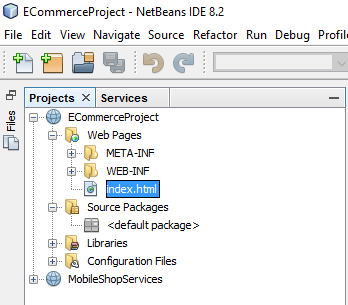
**2.** Trong cửa sổ Categories, chọn Java Web, và trong cửa sổ Projects chọn Web Application, sau đó nhấn **Next.**



**3.** Đặt tên project tương ứng là **ECommerceProject**, sau đó nhấn **Next**, cửa sổ thiếp lập **Server** sẽ hiện ra:



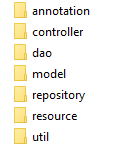
**4.** Chúng ta có thể chọn **Server** là **Tomcat** hoặc **Glassfish**, ở đây chúng tôi chọn Tomcat server. Sau đó chọn **Finish**. Dự án **ECommerceProjec**t sẽ xuất hiện ở cửa sổ Projects như dưới đây:



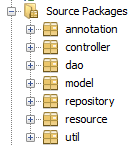
Lưu ý: File index.html là không cần thiết, chúng ta có thể xóa đi.

**5.** Sao chép thư mục vào dự án mới tạo. Chúng ta vào thư mục dự án được cung cấp sẵn, sau đó thực hiện như sau:

Vào thư mục **ECommerceProject** → ***src – java***, chúng ta sao chép toàn bộ như thư mục sau:

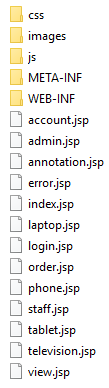


Sau đó dán vào thư mục Source Packages trong dự án ECommerceProject vừa tạo trên NetBeans:

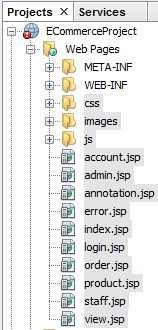


Lưu ý: Sau bước trên, một số thư mục class có thể bị lỗi, điều đó xảy ra là do chúng ta chưa thêm một số thư viện Jar (chúng tôi có hướng dẫn việc thêm thư viện ở bước 3).

Chúng ta tiếp tục vào ECommerceProject, vào thư mục web, sao chép toàn bộ như thư mục sau:



Sau đó dán vào thư mục Webpages trong dự án ECommerceProject vừa tạo trên NetBeans:

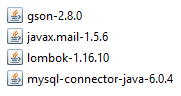


Như vậy là đã có thể hoàn thành cài đặt một dự án web trên NetBeans bằng cách sao chép. Chúng ta làm tương tự cho website nhà cung cấp **WebProviders** (trong cùng thư mục trên đĩa CD đã được copy vào máy tính của người dùng).

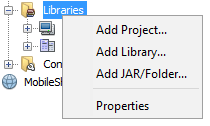
*Lưu ý: chúng ta cũng có thể cài đặt project bằng cách import một file zip trực tiếp vào NetBeans.*

* **Bước 3: Thêm một số thư viện Jar được cung cấp sẵn và điều chỉnh CSDL phù hợp:**

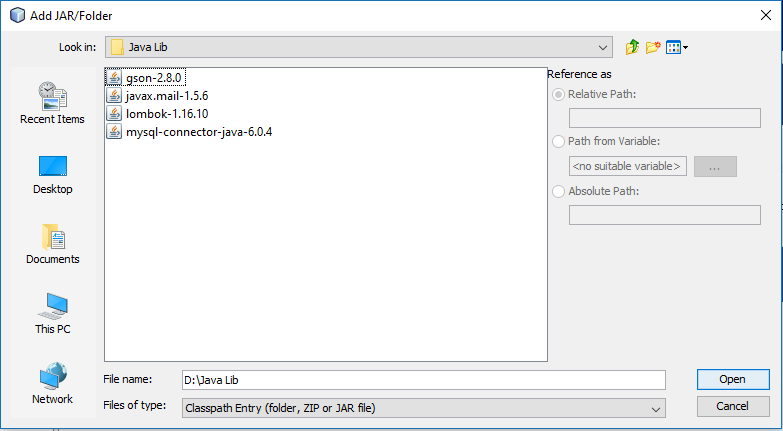
Để dự án được cài đặt thành công, chúng ta phải thêm số thư viện được chúng tôi cung cấp sẵn trong thư mục JavaLib như sau:



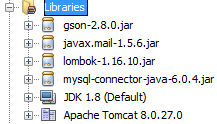
Trong dự án **ECommerceProject** trên NetBeans, chúng ta nhấn chuột phải vào **Libraries** và chọn **Add Jar/Folder**:



Browse đến thư mục JavaLib chứa thư viện xuất hiện như sau:



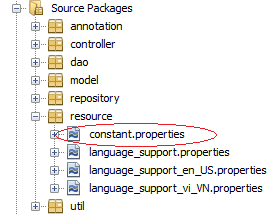
Chúng ta chọn tất cả thư viện Jar trên. Sau đó chọn **Open**. Nếu thành công thì phần **Libraries** sẽ xuất hiện như sau:



Chúng ta làm tương tự cho dự án **WebProviders.**

Do đặc điểm từng cở sở dữ liệu và server khác nhau nên chúng ta phải điều chỉnh một số phần như sau trên thư mục dự án **ECommerceProject:**

* Vào package *resource* và mở file *constant.properties*



* Điều chỉnh USERNAME, PASSWORD, và SERVER.PORT tương ứng với database và application server trên máy người dùng.



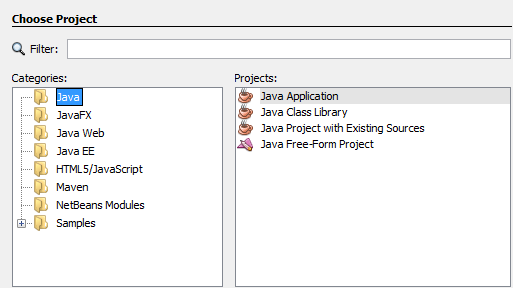
Như vậy là chúng ta đã hoàn thành cài đặt website trên NetBeans IDE. Xin chúc mừng!

**B. Hướng dẫn tạo trang web đa ngôn ngữ trên ứng dụng web bằng ResourceBundle.**

Đây cũng là cách mà chúng tôi áp dụng cho thư mục dự án ECommerceProject. Sau khi học xong bài hướng dẫn này, chúng ta có thể tham khảo cách chúng tôi tạo trang web đa ngôn ngữ bằng ResourceBundle ở file Language.java nằm trong package *controller*. Bây giờ chúng tôi sẽ hướng dẫn các bạn tạo một ứng dụng đa ngôn ngữ trên NetBeans.

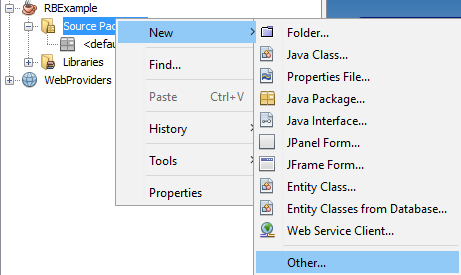
* **Bước 1:**

Trước hết, chúng ta sẽ tạo một dự án trên NetBeans, các bạn chọn **File → New Project..,** cửa sổ sau sẽ xuất hiện….

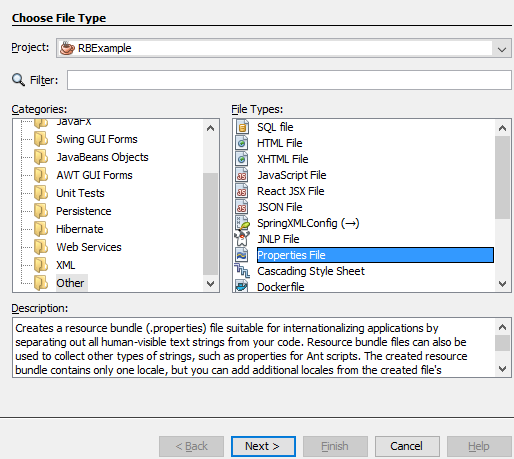


Chọn Java Application và đặt tên dự án này là **RBExample.**

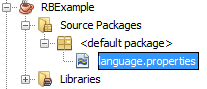
Tiếp theo chúng ta tạo một file properties trong source packages bằng cách nhấn chuột phải vào **Source Packages → New → Other:**



Cửa sổ chọn file sẽ xuất hiện, trong phần **Categories**, chọn **Other**, phần **File Type** chúng ta chọn **Properties File**, sau đó chọn **Next**:

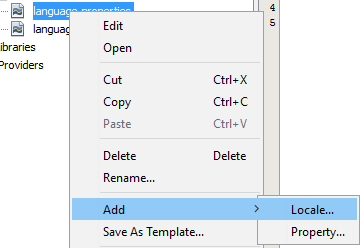


Tiếp theo chúng ta đặt tên file là *language*, sau đó chọn **Finish**, file vừa tạo sẽ nằm trong default package:

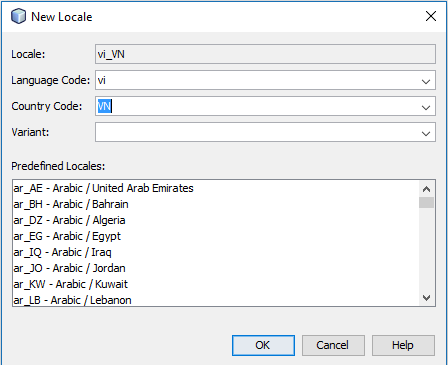


* **Bước 2:**

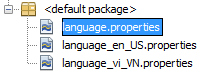
Chúng ta tiếp tục tạo 2 files property thành viên so với file property vừa khởi tạo ở bước một, file đầu tiên sử dụng cho cho tiếng việt, file còn lại sử dụng cho tiếng Anh. Lần này chúng ta nhấn chuột phải vào file language.properties và chọn **Add → Locale:**



Cửa sổ chọn Locale xuất hiện:



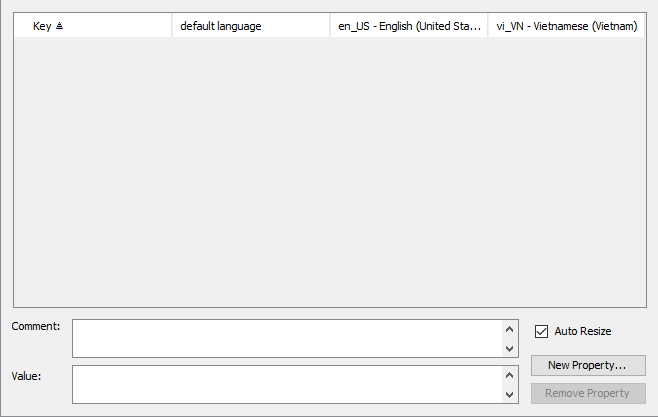
Đối với file properties sử dụng cho tiếng việt, phần *Language Code* chúng ta chọn *vi* và phần *Country Code* chúng ta chọn *VN*, sau đó nhấn **Finish.** Chúng ta tiếp tục tạo thêm 1 file properties giống như trên để sử dụng cho cho tiếng anh với *Language Code* là *en* và *Country Code* là *US.* Sau khi tạo xong, chúng ta sẽ có *3 file properties* như sau:



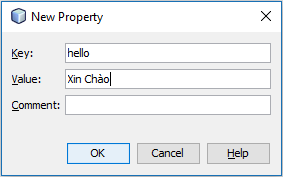
* **Bước 3:**

Hoàn thành xong bước 2, chúng ta đã có 3 file properties, trong đó file *language.properties* tạm gọi là file chủ, 2 file *language\_en\_US.properties* và *language\_vi\_VN.properties* là 2 files thành viên với file đầu tiên sử dụng cho cho tiếng Anh và file thứ 2 sử dụng cho cho tiếng Việt.

Chúng ta sẽ nhập vào một số properties bằng cách nhấn chuột phải vào file *language.properties* và chọn Open, cửa số sau hiện ra:



Chúng ta chọn New Property ở góc phải cửa sổ, sau đó phần Key chúng ta nhập hello; phần Value nhập Xin chào, sau đó chọn OK:



Sau khi nhấn OK, 1 property đầu tiên đã xuất hiện trên cửa sổ file *language.properties:*

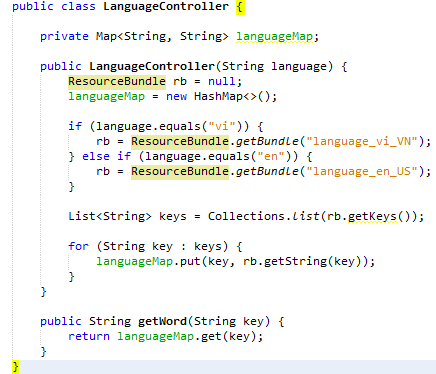


Cột đầu tiên là biểu thị cho từ khóa, cột thứ 2 biểu thị cho ngôn ngữ mặc định, ở đây chúng tôi chọn tiếng Việt, cột thứ 3 biểu thị cho ngôn ngữ tiếng Anh và cột cuối cùng biểu thị cho ngôn ngữ tiếng Việt. Các bạn tiến hành thay đổi lại cột thứ 3 thành Hello để phù hợp với tiếng Anh. Các bạn có thể tạo thêm vài property nữa.



* **Bước 4:**

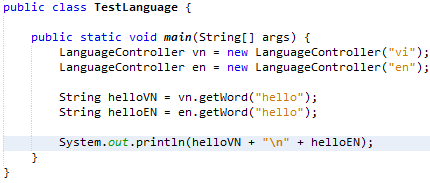
Tiếp theo chúng ta sẽ khởi tạo 1 class controller để xử lý cách tạo trang web đa ngôn ngữ. Các bạn tạo một class tên là LanguageController trong default package:



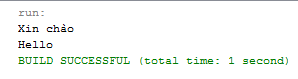
Class LanguageController có một biến là *languageMap* với kiểu dữ liệu là Map, biến này để lưu trữ tất cả dữ liệu mà chúng ta đọc từ file properties tương ứng. *Constructor* của class nhận một biến là language, biến này biểu thị cho kiểu ngôn ngữ mong muốn. Trong Constructor chúng ta khai báo một ResourceBundle, nếu language là *“vi”* thì chúng ta đọc getBundle từ file *language\_vi\_VN.properties* tương ứng với tiếng Việt; còn nếu là *“en”* thì chúng ta đọc getBundle từ file *language\_en\_US.properties* tương ứng với tiếng Anh, sau khi đọc từ file properties, chúng ta bỏ tất cả dữ liệu lại vào biến languageMap. Method *getWord()* dùng để lấy dữ liệu từ biến *languageMap* với từ khóa tương ứng.

* **Bước 5:**

Tiếp theo, chúng ta sẽ khởi tạo 1 class test để kiểm tra. Các bạn tạo một class tên là TestLanguage trong default package:



Kết quả:



Như vậy chúng ta đã hoàn thành việc tạo một ứng dụng đa ngôn ngữ trên NetBeans bằng ResourceBundle. Như chúng tôi đã nói ở phía trên, ở ứng dụng ECommerceProject chúng tôi sử dụng cách như trên để làm đa ngôn ngữ cho sản phẩm của chúng tôi. Cách thức như sau:

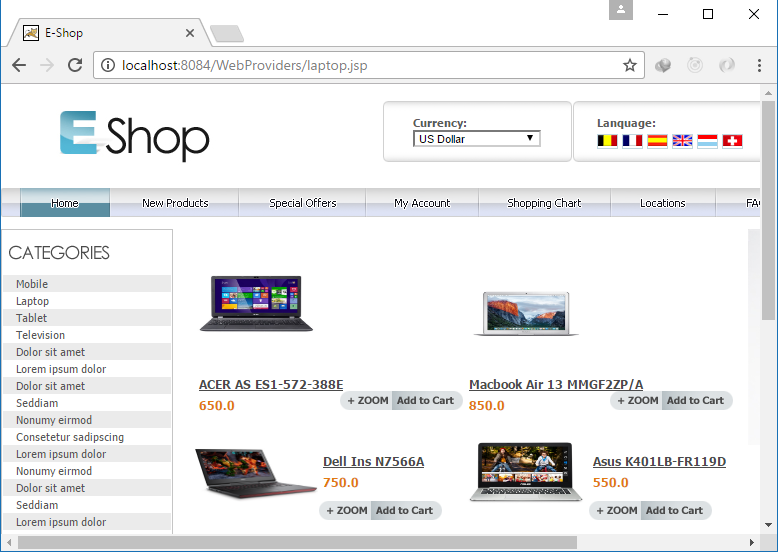
Ở class *Language.java* trong packages *controller*, chúng tôi cũng đọc từ 2 file properties là *language\_support\_en\_US.properties* và *language.support\_vi\_VN.properties*, với file chủ là *language\_support.properties*. Sau khi đọc xong, chúng tôi cũng thêm tất cả từ khóa và giá trị vào kiểu dữ liệu Map. Cách này chúng tôi thực hiện giống như bước 4 ở phần trên. Sau đó chúng tôi lưu trữ Map này vào Session để tiện sử dụng. Vì class Language.java là một Servlet nên nó sẽ nhận request từ người dùng bất cứ lúc nào để chuyển đổi từ tiếng Anh sang tiếng Việt và ngược lại. Sau đó ở mỗi *file .jsp* thành phần, chúng tôi chỉ cần gọi cái Map lưu trữ dữ liệu đọc từ file properties được lưu vào Session và thực hiện các phương thức *get(String key)* của Map để lấy dữ liệu tương ứng. Sau đây là đoạn code lấy Map ra từ Session của chúng tôi, các bạn có thể tìm thấy nó ở bất cứ *file .jsp* mà chúng tôi cung cấp, và chúng tôi thường đặt những dòng code này ở phía trên cùng.



Chúng ta có thể thực hành tiếp bằng cách nhập thêm một số loại ngôn ngữ khác cho website ECommerceProject, ví dụ như tiếng Pháp, tiếng Nhật, v.v nếu chúng ta muốn thử nghiệm.

**C. Hướng dẫn cơ chế thực hiện của một WebServices.**

Tất cả dữ liệu của các trang *phone.jsp, laptop.jsp, tablet.jsp và television.jsp* đều được lấy từ dữ liệu của nhà cung cấp, chúng ta có thể chạy thử thư mục dự án **WebProviders** để tham khảo, tất cả dữ liệu này nằm bên navigation bar, bên trái của giao diện nhà cung cấp:



Để lấy dữ liệu từ nhà cung cấp, thì nhà cung cấp phải cung cấp cho chúng ta một *“api”,* hay gọi cách khác là giao diện người dùng. Website của chúng ta là Ecommerce sẽ *“fetch”* tất cả những dữ liệu từ API mà website nhà cung cấp cho chúng ta, một dạng API cho dữ liệu về mặt hàng Laptop của nhà cung cấp:

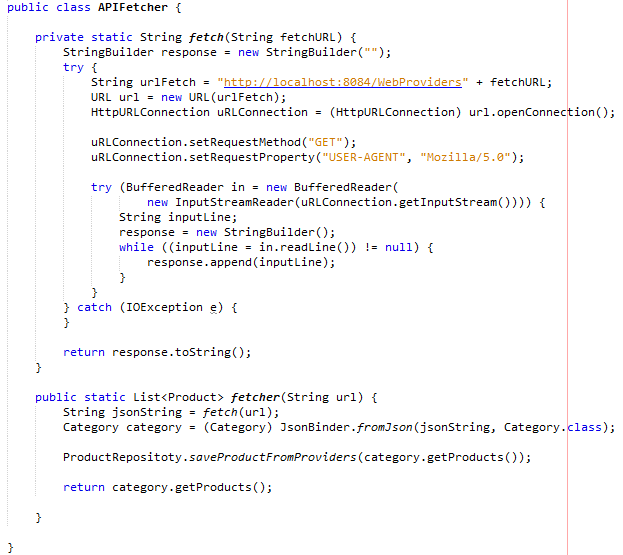


Địa chỉ của *“/laptopApi”* này chính là class *LaptopAPI.java* trong package *core* của thư mục dự án website nhà cung cấp:

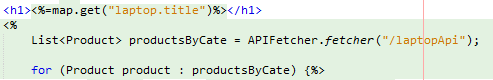


Ở class này chúng tôi khai báo một kiểu dữ liệu dạng **JSON** để lưu trữ dữ liệu các mặt hàng sản phẩm và ở phương thức *doGet()* chúng tôi trả về chính chuỗi **JSON** đó, các bạn có thể tham khảo thêm về JSON Object tại đây: [*http://www.w3schools.com/js/js\_json\_intro.asp*](http://www.w3schools.com/js/js_json_intro.asp)*.*

Về phía website ECommerceProject, có nhiệm vụ là fetch dữ liệu từ API mà nhà cung cấp đưa ra. Các bạn có thể tham khảo class *APIFetcher.java* trong packages *repository:*



Phương thức *fetcher()* nhận một biến là chuỗi url, tức là địa chỉ API của nhà cung cấp, sau đó gọi phương thức *fetch()* để thực hiên việc lấy chuỗi JSON từ nhà cung cấp với địa chỉ tương ứng, sau đó chúng ta sử dụng thư viện GSON từ Google để thực hiện việc chuyển đổi từ chuỗi JSON thành đối tượng tương ứng và cuối cùng phương thức *fetcher()* trả về một danh sách các mặt hàng, như vậy là chúng ta đã thành công trong việc lấy dữ liệu từ nhà cung cấp. Việc tiếp theo đó chính là gọi phương thức Fetcher ra và hiện lên giao diện của website ECommerceProject. Một ví dụ lấy dữ liệu Laptop từ API của nhà cung cấp WebProviders trong file *laptop.jsp.*



Chúng ta có thể tiếp tục tham khảo cách lấy dữ liệu trong file phone.jsp, tablet.jsp và television.jsp, sau đó thực hiện việc khởi tạo và lấy dữ liệu cho mặt hàng Máy ảnh từ phía nhà cung cấp.

**Chi tiết mô tả chức năng API Fetcher.**

Ở đây chúng ta cần xây dựng một website của nhà cung cấp (WebProviders). Dữ liệu trên website này sẽ đồng bộ và tự động cập nhật vào trang thông tin của từng mặt hàng tương ứng trong website TMĐT (ECommerceProject). Dữ liệu này, sau đó sẽ được lưu vào CSDL của website ECommerceProject.

Trong thư mục website WebProviders, có 1 lớp tên là Api.java, thực chất là một Servlet có địa chỉ truy cập là *“/api”*. Lớp này cung cấp địa chỉ API để website TMĐT có thể truy cập và lấy dữ liệu về. Trong website WebProviders, chúng tôi không sử dụng CSDL, mà sử dụng môt chuỗi JSON (Javascript Object Notation) để lưu dữ liệu cho mặt hàng điện thoại. Người dùng có thể tham khảo lớp Api.java trong Source Packages/Core.

Trong website ECommerceProject, chúng tôi tạo một lớp với tên gọi là APIFetcher.java. Mục đích của lớp này là truy cập tới địa chỉ của website WebProviders để lấy dữ liệu về và hiển thị lên website của mình. Trong trường hợp này là địa chỉ *“/api”* của website WebProviders. Người dùng có thể tham khảo lớp APIFetcher.java trong package repository của website ECommerceProject.

Sau khi lấy dữ liệu thành công về, lúc này dữ liệu vẫn còn hiển thị với định dạng là đối tượng JSON nên chúng tôi đã sử dụng một thư viện được cung cấp bởi Google là GSON để phân tích nó và chuyển nó về một dạng đối tượng của website ECommerceProject. Sau đó hiển thị dữ liệu lên website này.

Dữ liệu của 2 website kể trên được đồng bộ và cập nhật liên tuc. Nghĩa là bất cứ khi nào dữ liệu của website WebProviders thay đổi, thì nó sẽ tự động cập nhật qua website ECommerceProject tương ứng. Người dùng có thể tham khảo trang *“Product.jsp”* trong website ECommerceProject; đây là trang được lấy dữ liệu từ WebProviders

Sau khi hoàn tất các phần A, B, C cua 3 hướng dẫn này, chúng ta có thể sử dụng một số tài khoản chúng tôi cung cấp sẵn trong CSDL cho người dùng kiểm tra website sau khi hoàn tất cài đặt:

* ADMIN: tài khoản: ***admin***, mật khẩu: ***admin123***
* Quản lý website: tài khoản: ***staff***, mật khẩu: ***staff123***
* Người dùng: tài khoản: ***user***, mật khẩu: u***ser123***

Chúc các bạn thành công!

***Mọi thắc mắc xin liên hệ địa chỉ email:***

1. ***Nguyễn Văn Sinh:*** [***nvsinh@hcmiu.edu.vn***](mailto:nvsinh@hcmiu.edu.vn)
2. ***Nguyễn Tấn Đạt: ‎*ntdatjoker@gmail.com**