Xây dựng ứng dụng CRUD với Spring 5

**Bài 4: Thực hiện việc update dữ liệu và bắt ràng buộc về dữ liệu khi thực hiện chức năng update**

Mã Hoàng Nhật Phi

* **Mục đích:** Tiếp nối series bài viết về xây dựng ứng dụng web dùng Spring 5 Framework, bài viết trước chúng ta đã trải nghiệm việc truy vấn dữ liệu từ DB sử dụng Spring Framework cộng thêm việc trình bày dữ liệu trên giao diện client sử dụng XMLHTTPRequest để gọi RESTful Webservices nhằm chuyển đổi dữ liệu từ object thành dạng JSON để dùng JavaScript trình bày ở client. Trong bài viết này chúng ta sẽ thực hiện chức năng update và thực hiện kiểm tra ràng buộc để thông báo lỗi cho người dùng đồng thời cập nhật lại giao diện lưới trình bày sau khi thực hiện chức năng update xong (refresh hay gọi lại chức năng khác một lần nữa).
* **Yêu cầu kiến thức cơ bản:**
* Nắm vững và sử dụng các kỹ năng để về việc xây dựng hoàn tất ứng dụng với mô hình MVC2
* Nắm vững các khái niệm về ngôn ngữ lập trình Java, lập trình thao tác hướng đối tượng.
* Nắm vững các khái niệm về lập trình web sử dụng J2EE hay JavaEE với các kiến thức về Servlet, JSP. <http://www.kieutrongkhanh.net/search/label/Servlet%26JSP>
* Tìm hiểu và nắm vững các khái niệm và ứng dụng XMLHttpRequest trong việc gọi ứng dụng hạn chế postback ở client (behind the scene) tại địa chỉ <https://www.w3schools.com/xml/xml_http.asp> và <http://www.kieutrongkhanh.net/2016/12/du-lieu-uoc-request-tu-server-va-chuyen.html>
* Tìm hiểu và nắm được các khái niệm cùng với các kỹ năng sử dụng RESTful Web Services, tham khảo tại địa chỉ <http://www.kieutrongkhanh.net/2016/08/xay-dung-ung-dung-ap-dung-restful-web.html> hay <http://www.kieutrongkhanh.net/2016/08/gioi-thieu-ve-restful-web-services-cong.html>
* Đã hoàn tất bài xây dựng chức năng Search sử dụng Spring MVC5 tại địa chỉ <http://www.kieutrongkhanh.net/2020/08/bai-3-thuc-hien-viec-truy-van-va-trinh.html>
* **Tool sử dụng:**
  + Netbeans 8.x
  + JDK 8 update 66
  + DBMS: SQL Server từ 2005 đến 2019, PostgreSQL 10 (Đón xem nội dung bổ sung trong loạt bài viết tiếp theo)

# Giới thiệu

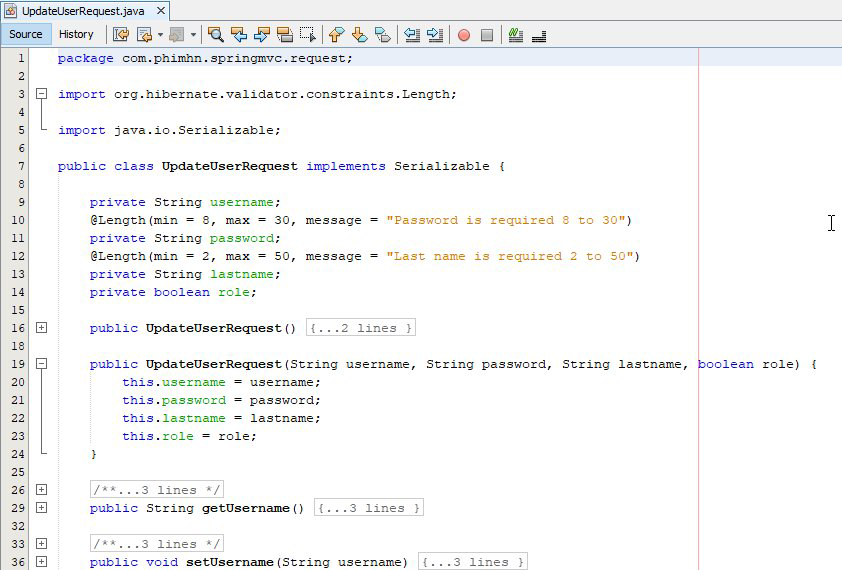
* Sau bài viết số 3, chúng ta đã nắm cơ bản về cách truy vấn dữ liệu và cung cấp thành API dưới dạng web services để từ đó có thể consume một cách độc lập ở server sử dụng format trung gian (JSON) mà không cần render dữ liệu ở phía server (giảm performance và không phù hợp với việc client có thể sử dụng bất cứ giao diện gì từ browser đến mobile hay bất kỳ thiết bị trình bày khác có khả năng trình bày). Tiếp theo chúng ta thực hiện việc bắt ràng buộc dữ liệu trước khi đồng bộ vào DB thông qua chức năng update

# Nội dung

## Thực hiện tính năng update

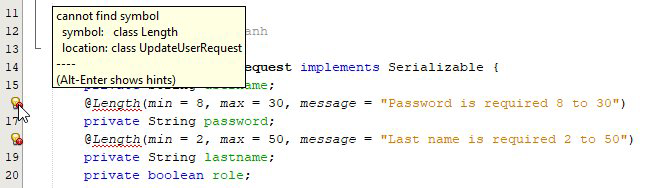
Tương tự cách suy nghĩ như tính năng Search, chúng ta sẽ phải xây dựng Response Model để mapping dữ liệu từ request vào DTO để xử lý với tên là **UpdateUserModelMapper được đặt tại package modelmapper**

* Đầu tiên công việc cần làm với chức năng update là xác định ràng buộc khi dữ liệu được đưa vào để cập nhật phải tương thích với dữ liệu và ràng buộc dưới DB. Chúng ta cần định nghĩa các ràng buộc về dữ liệu cần bắt trong quá trình nhập liệu của người dùng. Việc bắt ràng buộc này được diễn ra trong khi request đang xảy ra.
  + Chúng ta xây dựng class UpdateUserRequest trong package request để xử lý các ràng buộc về dữ liệu, cụ thể là password từ 8 đến 30 ký tự; lastname từ 2 đến 50 ký tự

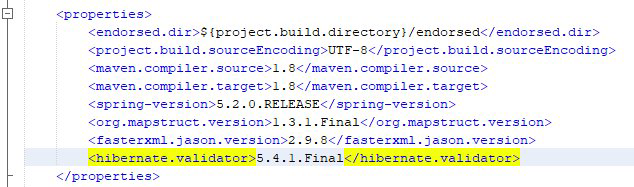


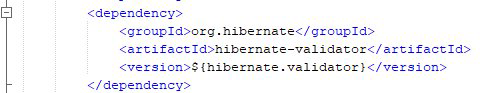


* + **Lưu ý,** các tên **property** **tương ứng** phải được mapping tên tương ứng với **DTO** và **entity** **Registration** (**xem tham khảo lại** các **case study** trong bài số 3 để có thể phát hiện lỗi cần chỉnh sửa)
  + Tuy nhiên, khi gõ code chúng ta sẽ phát sinh lỗi tại dòng @Length

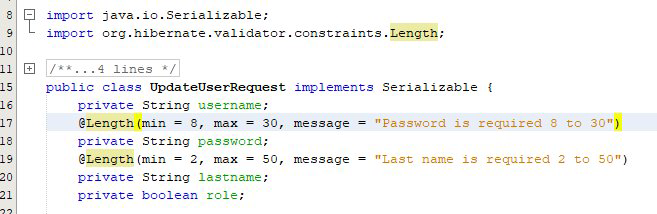


* + Lỗi này phát sinh là chưa do import thư viện hỗ trợ ràng buộc khi check dữ liệu. Chúng ta cần bổ sung thư viện validation với maven vào tập tin pom như sau
  + Bổ sung property hibernate.validator version và dependency cho hibernate validator

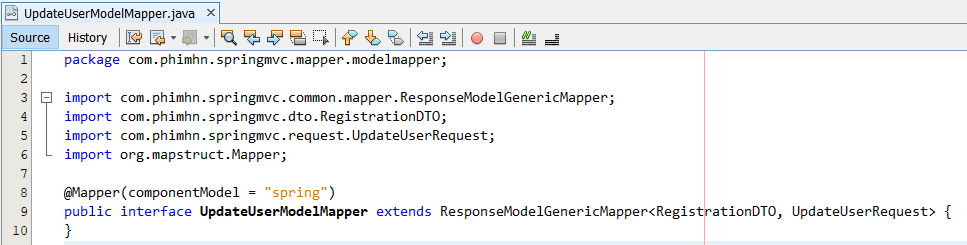




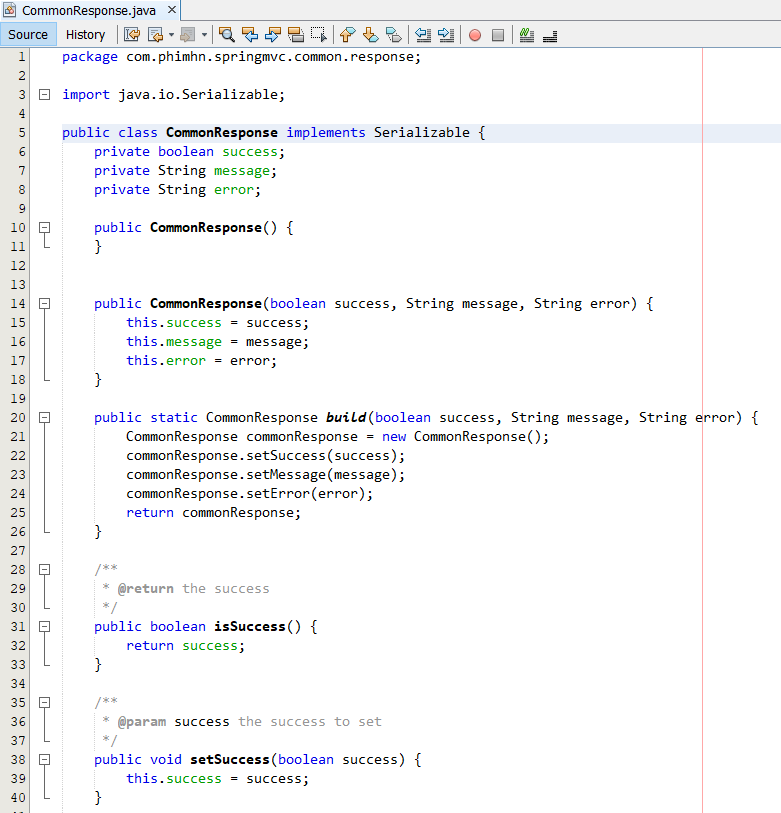
* + Thực hiện lệnh clean – install thì code sẽ hỗ trợ visual để chúng ta import để khắc phục lỗi

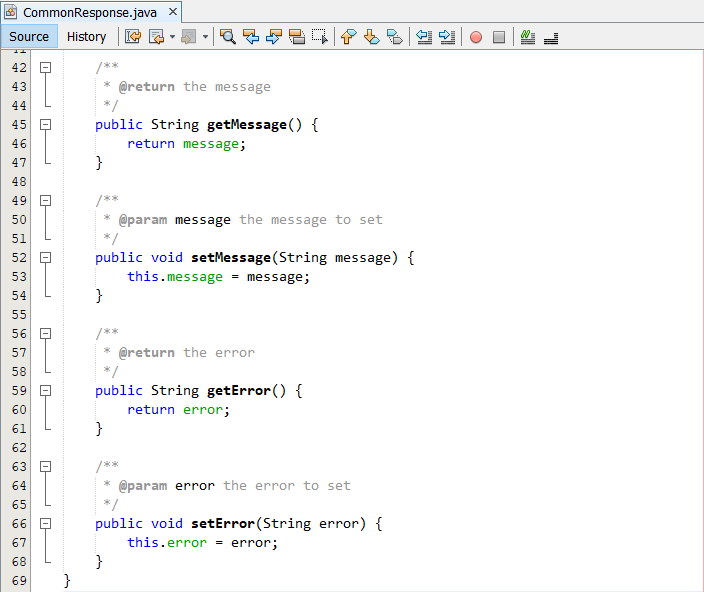


* + Tiếp theo, chúng ta implement interface UpdateUserModelMapper với thành phần dữ liệu mapping DTO với dữ liệu được đưa vào UpdateUserRequest sau khi kiểm tra tính toàn vẹn



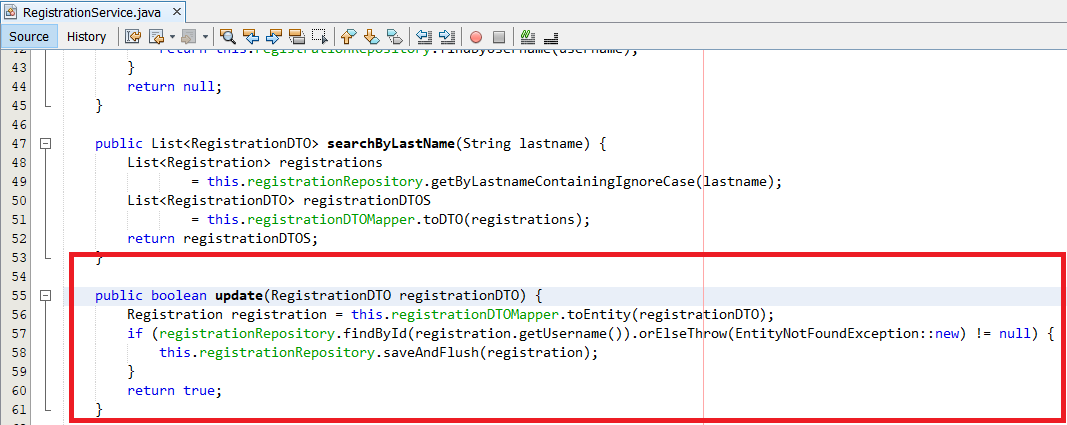
Bởi vì việc xử lý update là có xử lý liên quan đến việc bắt lỗi nhập liệu của người dùng cũng như lỗi phát sinh trong quá trình cập nhật dữ liệu, cho nên chúng ta cần xây dựng instance chứa các property cần thiết để có thể trả về cấu trúc message chung cho toàn ứng dụng. Common Response là đối tượng được implement thành class để thực hiện điều này. Chúng ta cài đặt class tại CommonResponse tại package response và chứa các property xác định kết quả việc xử lý (success), thông báo (message) và lỗi phát sinh (error)



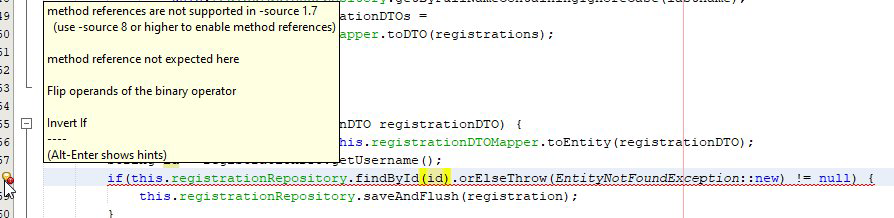


* Cần giải thích tại sao trong class trên phải xây dựng hàm build static

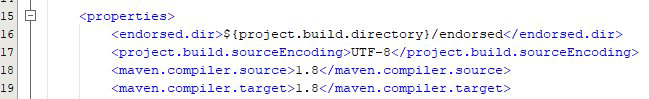
Chúng ta tiếp tục bổ sung phương thức update vào bên trong services

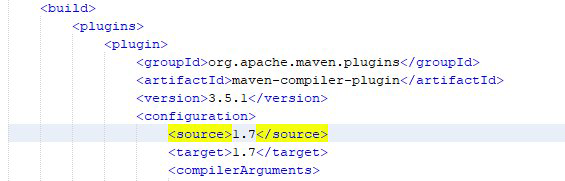


* Trong phương thức update, chúng ta đã thực hiện kiểm tra xem ở dưới database đã có dữ liệu tương ứng hay chưa nếu có thì thực hiện update, ngược lại thì throw EntityNotFoundException
* Chúng ta cần lưu ý đến phương thức **findById** đã được có sẵn trong **JpaRepository** với tham số là kiểu dữ liệu của khóa chính (ở **Spring 4 thì** hàm trên sẽ trả về chính entity) nhưng từ **Spring 5** entity sẽ được wrap lại trong một **Optional** (kiểu dữ liệu có từ java 8). Do vậy, chúng ta cũng có thể dùng hàm **findByUsername** đã định nghĩa ở phần login. Tuy nhiên, để chúng ta cùng nghiên cứu các tình huống khác nhau trong sử dụng chúng ta sử dụng kiểu dữ liệu của Java 8 sẳn có để minh họa cho việc bắt exception tại **ExceptionHandlerController.**
* Sau khi thực hiện kiểm tra được hành động update đã được thực hiện, chúng ta thực hiện gọi phương thức **saveAndFlush** để đưa dữ liệu xuống database ngay mà không cần phải chờ commit của transaction do spring quản lý như phương thức **save.**
* **Lưu ý**: ở đây chúng ta có khả năng phát sinh lỗi khi viết code tại dòng 57 như sau

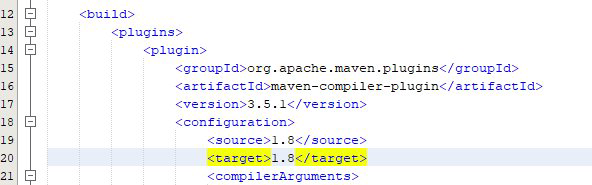


* + Lỗi này phát sinh là cách sử dụng Java 8 phải được build với maven 1.8, do vậy cho dù chúng ta cấu hình property là 1.8 nhưng nếu plugin build của chúng ta điền 1.7 (trong file pom.xml) thì lỗi phát sinh như trên

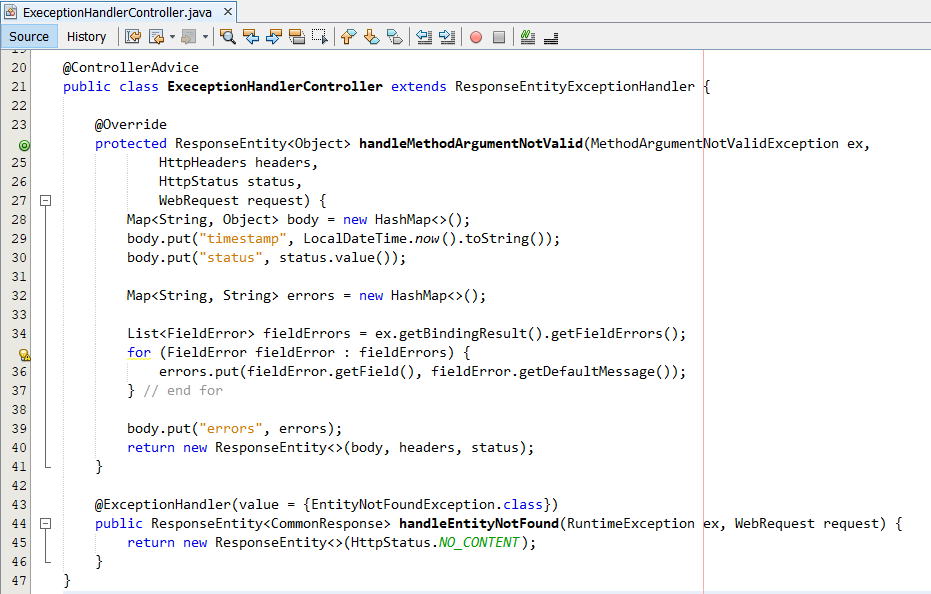




* + Chúng ta thực hiện đổi source và target trong plugin thành 1.8 thì code sẽ hết lỗi



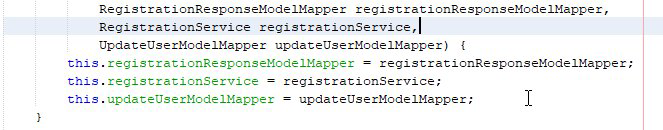
* Chúng ta tiếp tục việc xử lý lỗi nếu có bằng **ExceptionHandlerController. Đây cũng** là một controller nhưng được gọi xử lý khi có exception throw ra. ControllerAdvice có liên quan đến một khái niệm lập trình là AOP. (Viết cái này là cái gì, nó có ăn nhậu gì với ExceptionHandlerController – giải thích. Ngoài ra, AOP là cái gì – mục tiêu và chúng ta ứng dụng gì)
  + Khi cài đặt class này chúng ta phải sử dụng annotation ControllerAdvice ….
* Chúng ta thực hiện tạo ExceptionHandlerController class tại package controller
  + Sau khi tạo class và extends, chúng ta gõ chữ handleMethod và nhấn tổ hợp phím control – space bar để phát sinh phương thức override cho chúng ta. Cần giải thích handleMethodArgumentNotValid dùng để làm gì
  + Giải thích handleEntityNotFound làm gì, trigger khi nào, việc return của nó nghĩa là gì !??



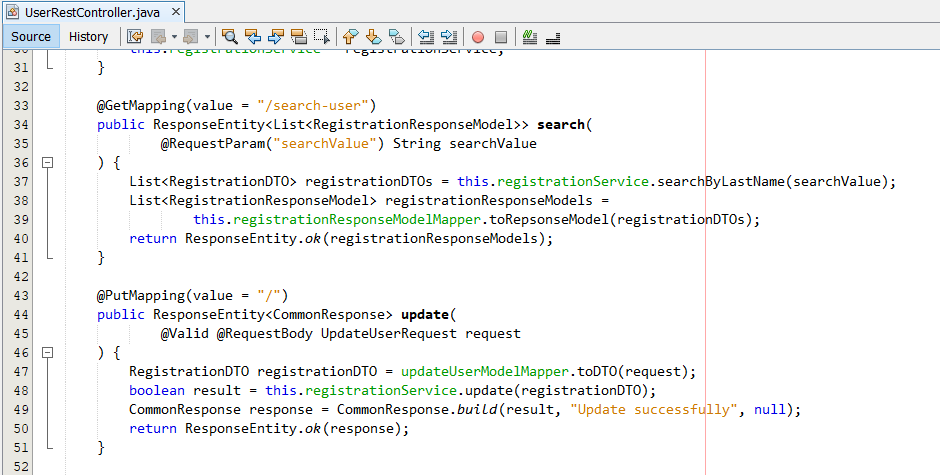
* Mọi thứ đã sẵn sàng, chúng ta bổ sung chức năng vào trong Rest Controller
  + Đầu tiên chúng ta khai báo bổ sung property cho Rest Controller để sử dụng injection với IoC cho UpdateUserModelMapper

**private final UpdateUserModelMapper updateUserModelMapper;**

* + Cập nhật thêm tham số cho contructstor để đảm bảo injection được Spring hỗ trợ thực hiện



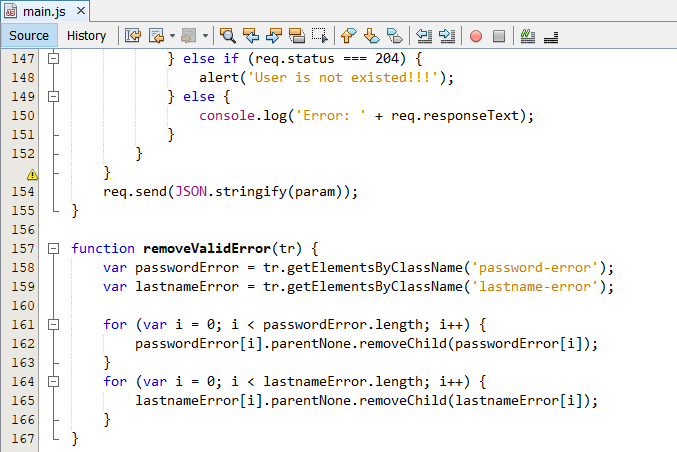
* + Cài đặt phương thức update. Giải thích các tham số truyền và các annotation trong code và giải thích một số code xử lý trong thân phương thức



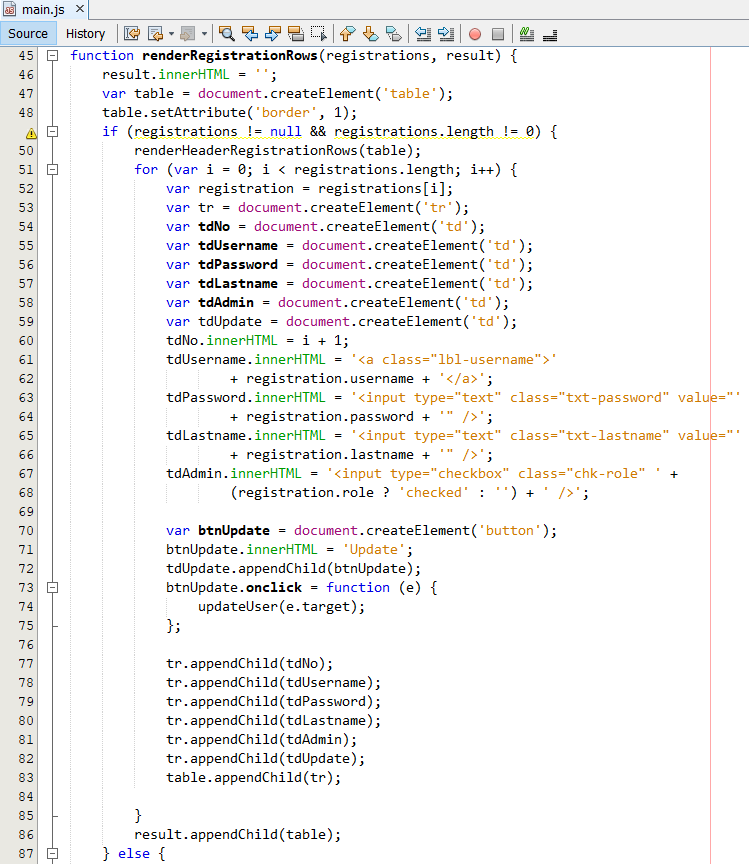
Sau khi thực hiện thao tác, chúng ta thực hiện cập nhật giao diện trình bày ở client

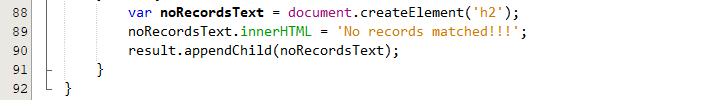
* Chúng ta thực hiện bổ sung chức năng Update user vào file main.js. function updateUser thực hiện cơ bản nội dung như sau
  + Lấy các giá trị của các control trong table trình bày kết quả search được thể hiện qua các class name được chúng ta định nghĩa
  + Bổ sung các phần xử lý trong chức năng của update tổng quát từ xử lý lỗi đến gọi services sử dụng XMLHTTPRequest, đến checking kq, …
  + …



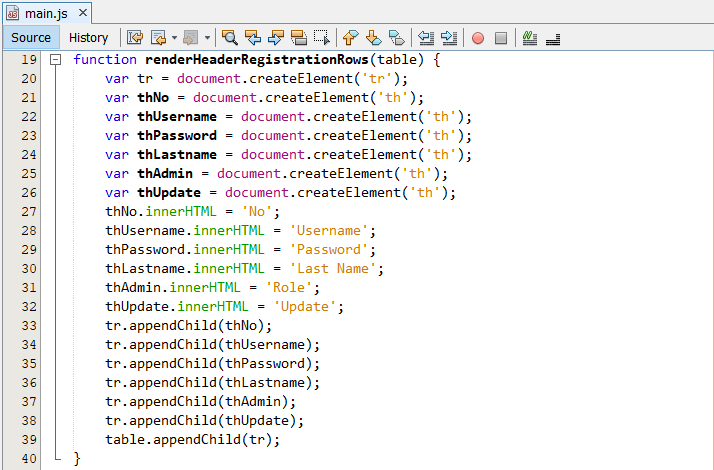


* Tiếp theo, chúng ta thực hiện rendering lại dữ liệu thông qua hàm renderRegistrationRows để cập nhật thêm thành phần liên quan đến Update

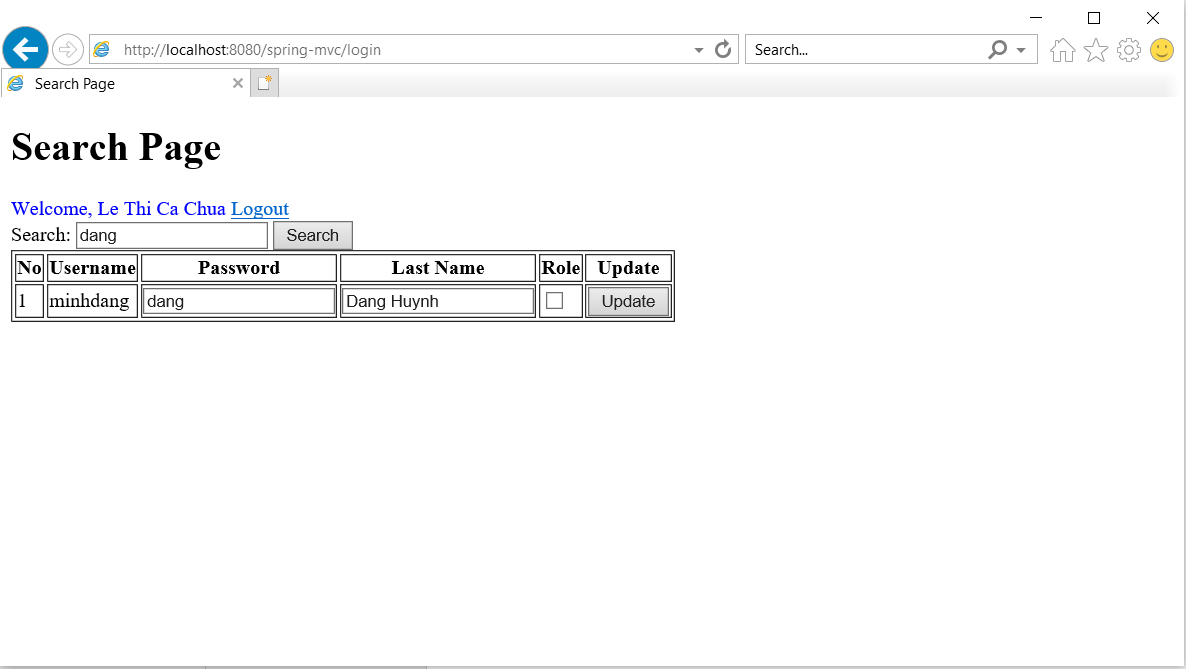




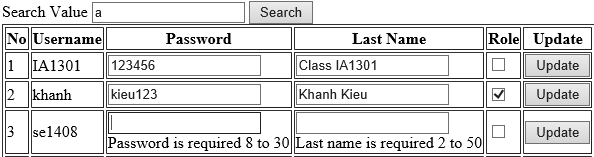
* Tương tự, chúng ta thực hiện update lại hàm renderRegistrationRows để cập nhật thêm thành phần liên quan đến Update



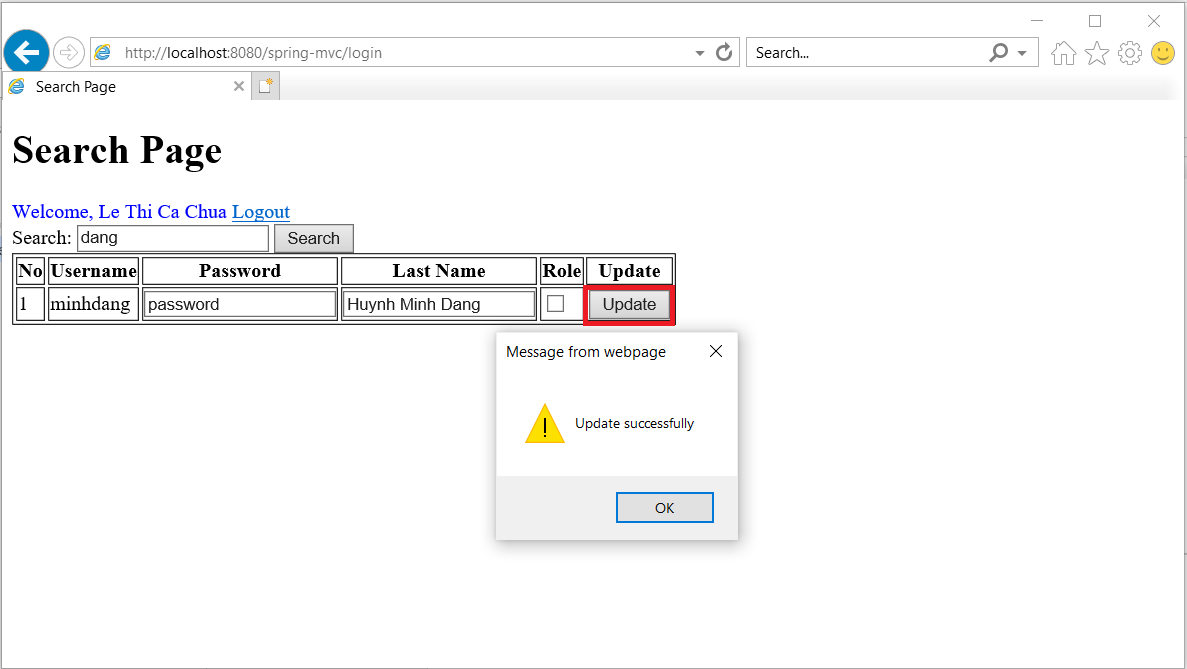
* Chúng ta thực hiện lệnh clean – install. Sau đó, deploy để test chức năng update
  + Chúng ta thực hiện đến chức năng search



* + Nếu chúng ta thực hiện cập nhật sai như bỏ trống cột password và lastname, kết quả chúng ta sẽ nhận được

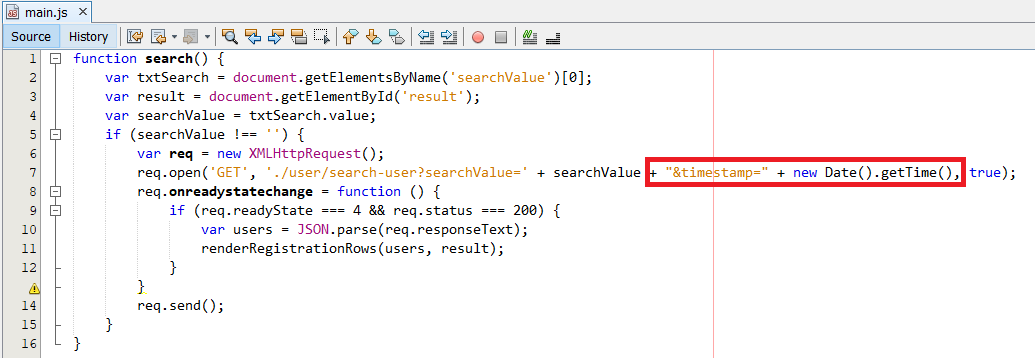


* + Nếu chúng ta thực hiện cập nhật kết quả đúng, chúng ta nhận được kết quả như sau



***Lưu ý: Với trình duyệt IE 11, nếu data trên giao diện quay lại giá trị trước khi update nhưng data ở database đã thay đổi thì chúng ta thực hiện việc xử lý cache khi sử dụng XHR trên IE 11 như sau***

* + Ở một số máy, trình duyệt IE 11 cache lại và trả về kết quả khi cùng url và cùng param giống nhau, điều này không xảy ra trên các trình duyệt khác.
  + Khi chức năng update thực hiện thành công, chúng ta sẽ gọi lại hàm search nhưng IE nhận ra url và param truyền đi tương tự đã truyền nên trả về kết quả đã được cache trên browser. Cho dù, chúng ta dễ dàng nhận thấy tính năng không hoạt động sai dù database đã được cập nhật. Bên cạnh đó, chức năng vẫn chạy đúng trên các trình duyệt khác ngoài IE
  + Để khắc phục vấn đề cache của IE như đã nêu trên, chúng ta thêm một tham số là timestamp là giờ hiện tại với mục tiêu để các param được hình thành trong các request sẽ khác nhau.
  + Chúng ta điều chỉnh sửa main.js như sau:



Bổ sung trường hợp là update có lỗi nhưng khi update lại thành công thì lỗi vẫn như cũ không refresh bỏ lỗi được, điều chỉnh lại code java script nhé Phi

Bổ sung lại việc chữ báo lỗi đang là màu đen ko phải màu đỏ

Chúc mừng quí vị, chúng ta vừa hoàn tất việc bắt ràng buộc về dữ liệu và thông báo lỗi cho người sử dụng thông qua chức năng update. Hy vọng nội dung này sẽ hỗ trợ quí vị trong việc sử dụng Spring Framework để xây dựng ứng dụng

Rất mong quí vị góp ý về nội dung loạt bài viết này. Hẹn gặp lại quí vị ở các bài viết sau về các nội dung liên quan đến việc thêm xóa (CD) sử dụng Spring 5 Framework.