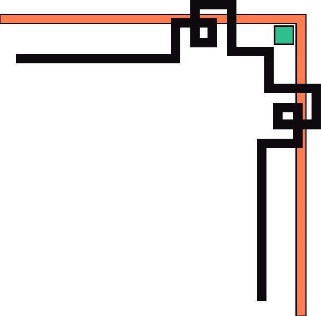
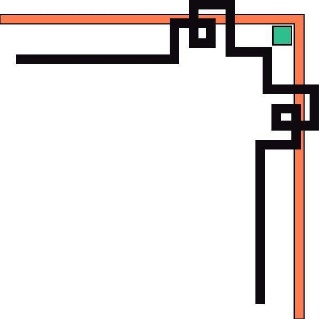
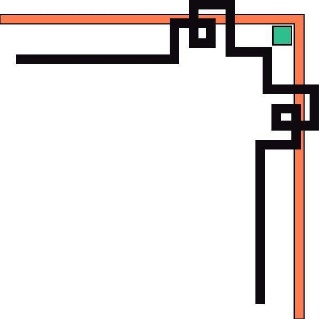
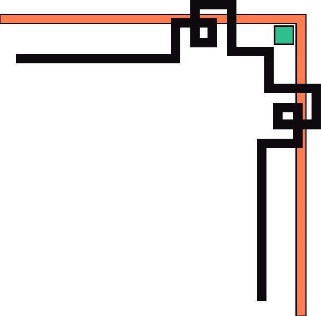
**TRƯỜNG ĐẠI HỌC QUY NHƠN**



**KHOA CÔNG NGHỆ THÔNG TIN**

---🙦🕮🙤---

****

**BÁO CÁO CUỐI KỲ**

**CÔNG NGHỆ JAVA**

**Quản Lý Sinh Viên**

|  |  |
| --- | --- |
| Sinh viên thực hiện : | **Trần Văn Hoài Nhiên** |
|  | **Võ Quang Bình**  **Lê Phi Khánh** |
| Lớp : | **Công nghệ phần mềm K39** |
| Giáo viên hướng dẫn : | **TS. Nguyễn Thanh Bình** |

**Quy Nhơn, tháng 9 năm 2020**

**MỤC LỤC**

[**LỜI CẢM ƠN** 4](#_Toc51696203)

[**Phần I : Giới Thiệu và Phân Tích đề tài** 5](#_Toc51696207)

[**I. TỔNG QUAN** 5](#_Toc51696208)

[**1. Mục tiêu đề tài** 5](#_Toc51696209)

[***1.1. Ý tưởng*** 5](#_Toc51696210)

[***1.2. Phát biểu mục tiêu đề tài*** 5](#_Toc51696211)

[***1.3. Các chức năng chính của hệ thống*** 5](#_Toc51696212)

[**II. PHÂN TÍCH THIẾT KẾ** 6](#_Toc51696213)

[**1. Yêu cầu chức năng** 6](#_Toc51696214)

[***1.1. Sơ đồ use cases*** 6](#_Toc51696215)

[***1.2. Đặc tả use cases*** 8](#_Toc51696216)

[**2. Yêu cầu phi chức năng** 21](#_Toc51696217)

[**3. Lược đồ cơ sở dữ liệu** 21](#_Toc51696218)

[**4. Đặc tả giao diện màn hình** 22](#_Toc51696219)

[**Phần II : Các Framework sử dụng và Triển Khai** 36](#_Toc51696220)

[**I. FRAMEWORK SỬ DỤNG TRONG PROJECT** 36](#_Toc51696221)

[**1. Spring MVC** 36](#_Toc51696222)

[**1.1. Spring MVC (Spring Web MVC)** 36](#_Toc51696223)

[***1.2. Flow trong Spring MVC.*** 36](#_Toc51696224)

[**2. Spring Security** 37](#_Toc51696225)

[**II. CẤU HÌNH VÀ TRIỂN KHAI** 48](#_Toc51696226)

[**1. Cấu hình và triển khai lên host** 48](#_Toc51696227)

[**2. Cấu hình và triển khai lên localhost** 53](#_Toc51696228)

[**Deploy từ Eclipse** 54](#_Toc51696229)

[**III. KẾT QUẢ ĐẠT ĐƯỢC** 59](#_Toc51696230)

[**1. Kết quả đạt được** 59](#_Toc51696231)

[**3.** **Hạn chế của đề tài** 60](#_Toc51696232)

[**3. Hướng phát triển của đề tài** 60](#_Toc51696233)

[**CÔNG CỤ HỖ TRỢ** 60](#_Toc51696234)

[**TÀI LIỆU THAM KHẢO** 60](#_Toc51696235)

**LỜI CẢM ƠN**

**Chúng em xin chân thành cảm ơn Thầy Nguyễn Thanh Bình đã tận tình giúp đỡ trong suốt quá trình học tập và thực hiện để chúng em có thể hoàn thành đượcc môn học và đề tài cuối kì.**

**Mặc dù đã cố gắng hết sức để hoàn thành đề tài cuối kì nhưng do sự thiếu kinh nghiệm và kiến thức nên còn gặp sai sót trong cách trình bày và triển khai . Kính mong thầy thông cảm và nhận được ý kiến đánh giá của thầy .**

**Nhóm chúng em xin cảm ơn !**

**Phần I : Giới Thiệu và Phân Tích đề tài**

**I. TỔNG QUAN**

**1. Mục tiêu đề tài**

***1.1. Ý tưởng***

Xây dựng chương trình Quản lý sinh viên nhằm hỗ trợ cho công tác quản lý họ tên, ngày sinh, điểm. Bài toán đặt ra là phân tích thiết kế hệ thống thông tin vấn đề đặt ra là tại sao  phải quản lý? Và quản lý cái j và quản lý như thế nào để công việc có hiệu quả, tiết kiệm được thời gian cho cán bộ công nhân viên.

***1.2. Phát biểu mục tiêu đề tài***

Quản lý sinh viên là công việc nhằm quản lý tất cả quá trình hoạt động và học tập của sinh viên trong các trường Đại học cũng để nâng cao về công nghệ thông tin.

Quản lý sinh viên trong các trường Đại học chính là quản lý quá trình học tập, trong đó có tất cả hồ sơ của sinh viên và điểm trong quá trình học tập tại trường đều được lưu trong chương trình “ Quản lý sinh viên”

Trong quản lý sinh viên có nhiều đầu điểm, có nhiều môn và có điểm của nhiều lần thi. Chương trình “ Quản lý sinh viên gồm nhiều lĩnh vực như quản lý họ tên, ngày sinh, giới tính, dân tốc, nơi sinh….”

Nghiên cứu và kết hợp framework Spring và Hibernate trên nền tảng Java vào phát triển đề tài.

***1.3. Các chức năng chính của hệ thống***

Đăng Nhập, Đăng Ký

Đổi Mật Khẩu

Thêm Thông Tin Cá Nhân

Sửa Thông Tin Cá Nhân

Xóa Thông Tin Cá Nhân

Xem Thông Tin Cá Nhân

Báo Cáo, Report Sinh Viên

….

***1.4. Các tác nhân***

SINH VIÊN: là người sử dụng, thay đổi các thông tin cá nhân của bản thân, xem tin tức từ trường và thực hiện các chức năng như thêm xóa sửa thông tin cá nhân, đăng ký tín chỉ, xem lịch học, lịch thi, học phí, các thông báo từ nhà trường….

NGƯỜI QUẢN LÝ: là người điều hành, quản lý và đưa ra các hướng giải quyết trên hệ thống

**II. PHÂN TÍCH THIẾT KẾ**

**1. Yêu cầu chức năng**

***1.1. Sơ đồ use cases***

*1.1.1. Use case tổng quan hệ thống*

|  |  |
| --- | --- |
| Mã use case | UC-01 |
| Tên use case | **Sinh viên đăng nhập** |
| Tác nhân (Actor) | Người dùng |
| Level | User goal |
| Mô tả ngắn (Brief) | Người dùng đăng nhập vào hệ thống |
| Tiền điều kiện (Preconditions) | Người dùng phải có tài khoản đăng nhập |
| Kết quả (Postconditions) | Chuyển tới trang quản lý sinh viên |
| Điều kiện kích hoạt use case  (Triggers – specific business event) | Người dùng nhấn nút đăng nhập |
| Luồng sự kiện chính  (Main scenario, basic flow) | 1. Người dùng truy cập vào trang đăng nhập 2. Nhập tài khoản và mật khẩu 3. Nhấn nút đăng nhập 4. Chuyển tới trang quản lý sinh viên |
| Luồng sự kiện phụ  (Extensions) | 1. Bước 3:  * Nếu tài khoản và mật khẩu không đúng hiển thị thông báo “Tài khoản hoặc mật khẩu không đúng mời bạn kiểm tra lại” |

|  |  |
| --- | --- |
| Mã use case | UC-02 |
| Tên use case | **Sinh viên đổi mật khẩu** |
| Tác nhân (Actor) | Người dùng |
| Level | User goal |
| Mô tả ngắn (Brief) | Người dùng muốn đổi mật khẩu tải khoản |
| Tiền điều kiện (Preconditions) | Người dùng phải đăng nhập thành công |
| Kết quả (Postconditions) | Hiển thị thông báo “ Đổi mật khẩu thành công “ |
| Điều kiện kích hoạt use case  (Triggers – specific business event) | Người dùng nhấn nút đổi mật khẩu |
| Luồng sự kiện chính  (Main scenario, basic flow) | 1. Người dùng chọn chức năng đổi mật khẩu 2. Trang đổi mật khẩu hiện thị 3. Người dùng nhập đầy đủ các trường thông tin ( mật khẩu cũ, mật khẩu mới, xác nhận mật khẩu mới ) 4. Nhấn nút đổi mật khẩu 5. Hiển thị thông báo “ Đổi mật khẩu thành công “ |
| Luồng sự kiện phụ  (Extensions) | 1. Tại bước 4:  * Nếu người dùng không nhập đầy đủ các trường thông tin sẽ hiển thị thông báo “Bạn cần nhập đầy đủ thông tin để đổi mật khẩu” * Nếu người dùng nhập không đúng mật khẩu cũ hiển thị thông báo “Mật khẩu cũ của bạn không đúng” * Nếu người dùng nhập không đúng xác nhận mật khẩu mới hiện thị thông báo “Xác nhận mật khẩu mới của bạn không đúng” |

|  |  |
| --- | --- |
| Mã use case | UC-03 |
| Tên use case | **Sinh viên sửa thông tin** |
| Tác nhân (Actor) | Người dùng |
| Level | User goal |
| Mô tả ngắn (Brief) | Người dùng muốn sửa thông tin |
| Tiền điều kiện (Preconditions) | Người dùng phải đăng nhập thành công |
| Kết quả (Postconditions) | Hiển thị thông báo “ Sửa thông tin thành công “ |
| Điều kiện kích hoạt use case  (Triggers – specific business event) | Người dùng nhấn nút lưu thông tin |
| Luồng sự kiện chính  (Main scenario, basic flow) | 1. Người dùng chọn chức năng sửa thông tin 2. Trang thông tin cá nhân hiển thị 3. Người dùng chọn vào thông tin cần sửa và sửa thông tin 4. Người dùng nhấn nút lưu thông tin 5. Hiển thị thông báo “ Sửa thông tin thành công “ |
| Luồng sự kiện phụ  (Extensions) |  |

|  |  |
| --- | --- |
| Mã use case | UC-04 |
| Tên use case | **Admin xem thông tin sinh viên** |
| Tác nhân (Actor) | Admin |
| Level | User goal |
| Mô tả ngắn (Brief) | Admin muốn sửa thông tin sinh viên |
| Tiền điều kiện (Preconditions) | Admin phải đăng nhập thành công vào hệ thống |
| Kết quả (Postconditions) | Trang thông tin sinh viên hiển thị |
| Điều kiện kích hoạt use case  (Triggers – specific business event) | Admin chọn vào sinh viên cần xem |
| Luồng sự kiện chính  (Main scenario, basic flow) | 1. Admin chọn chức năng xem danh sách sinh viên 2. Trang danh sách sinh viên được hiển thị 3. Admin chọn sinh viên cần xem thông tin 4. Trang thông tin sinh viên hiển thị |
| Luồng sự kiện phụ  (Extensions) |  |

**2. Yêu cầu phi chức năng**

***2.1. Môi trường***

Hệ thống hoạt động trên Google Chrome, FireFox và CocCoc.

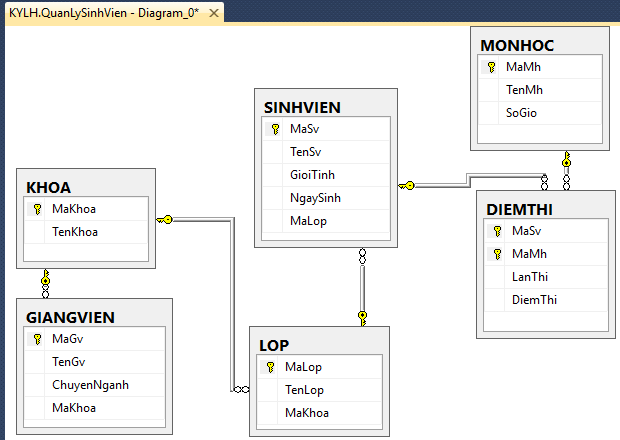
***2.2. Hiệu suất***

Các chức năng thực hiện nhanh gọn, chính xác

Thời gian hiển thị thông tin chi tiết của một chiếc xe trong vòng 2 giây.

Hệ thống có thể xử lý tối đa 1000 người dùng cùng lúc.

**3. Lược đồ cơ sở dữ liệu**

*­*****

**4. Đặc tả giao diện màn hình**

***4.1. Đặc tả màn hình Đăng Nhập***

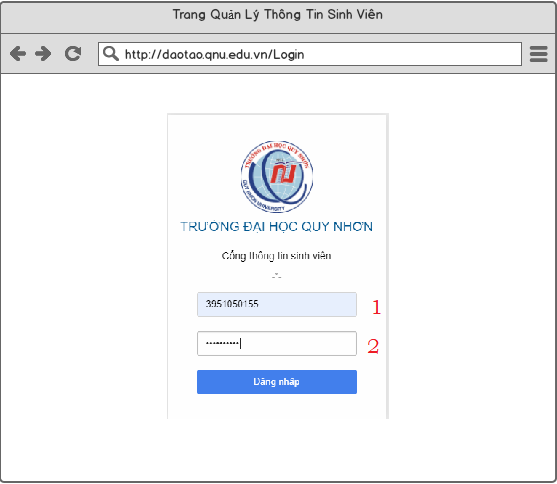
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Mã màn hình* | SCR-01 | *Tên chức năng* | Đăng Nhập | *Ngày tạo* | 20/9/2020 |
| *Người tạo* | Trần Văn Hoài Nhiên |

*1. Tổng quan*

Màn hình hiển thị trang chủ của website cho người dùng.

Màn hình này hiện lên khi người dùng nhấn vào nút “TRANG CHỦ” trên giao diện chính hoặc khi website được truy cập.

*2. Giao diện màn hình*

****

3. *Các thành phần của màn hình*

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Mô tả |
| 1 | Trường đại học Quy Nhơn | Icon, label | Hiển thị thông tin |
| 2 | Mã sinh viên | Textbox | Nhập mã sinh viên |
| 3 | Mật Khẩu | Textbox | Nhập mật khẩu mã sinh viên |
| 4 | Đăng Nhập | Button | Chuyển đến trang chủ |

***4.2. Đặc tả màn hình Trang Chủ***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Mã màn hình* | SCR-02 | *Tên chức năng* | Trang Chủ | *Ngày tạo* | 20/9/2020 |
| *Người tạo* | Trần Văn Hoài Nhiên |

­­­­

*1. Tổng quan*

Màn hình hiển thị các thông tin cơ bản và menu tùy chọn

Màn hình này hiện lên khi người dùng nhấn vào nút bên thanh list bên cạnh sẽ hiện ra thông tin ứng với thông tin được chọn

*2. Giao diện màn hình*



1

*3. Các thành phần của màn hình*

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Mô tả |
| 1 | Trang chủ | href | Quay trở lại trang chủ |
| 2 | Đăng ký học phần | href | Chuyển sang trang đăng ký học phần |
| 3 | Thông báo chung | href | Chuyển sang trang thông báo chung |
| 4 | Các quy định | href | Chuyển sang trang quy định |
| 5 | Thông báo học phí | href | Chuyển sang trang thông báo học phí |

***4.3. Đặc tả màn hình xem thông tin chi tiết sinh viên***

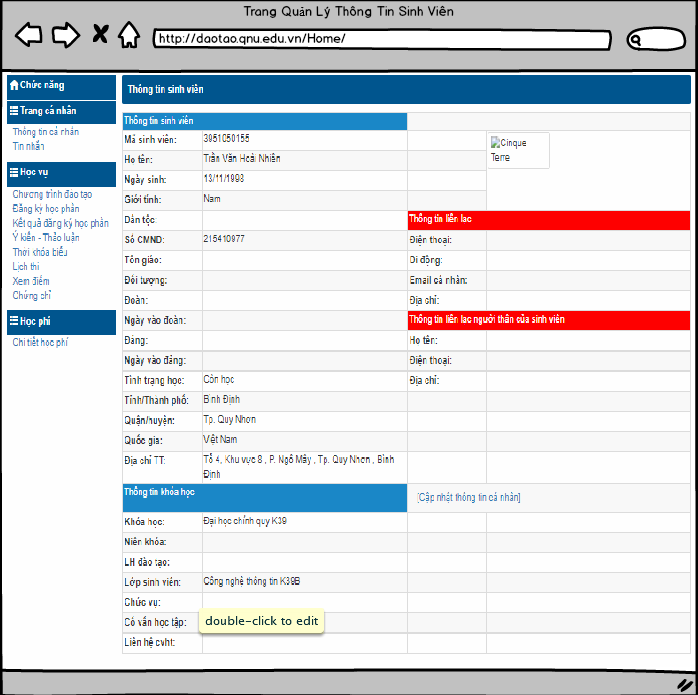
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Mã màn hình* | SCR-03 | *Tên chức năng* | Thông tin chi tiết | *Ngày tạo* | 20/9/2020 |
| *Người tạo* | Trần Văn Hoài Nhiên |

­­­­

*1. Tổng quan*

Màn hình hiển thị các thông tin sinh viên như : họ tên, ngày tháng năm sinh, dân tộc, CMND, các thông tin cá nhân khác, thông tin về khóa học, thông tin liên lạc……

*2. Giao diện màn hình*



*3. Các thành phần của màn hình*

1

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Mô tả |
| 1 | Thông tin sinh viên, thông tin khóa học, thông tin liên lạc.. | Label | Hiển thị những Thông tin sinh viên, thông tin khóa học, thông tin liên lạc.. |
| 2 | Mã sinh viên, họ tên, ngày sinh…. | Label | Hiển thị những thông tin mã sinh viên, họ tên, ngày sinh…. |
| 3 | Thông tin cá nhân, tin nhắn, chương trình đào tạo, đăng kí học phần, lịch thi, thời khóa biểu, học phí… | Href | Chuyển sang những trang như Thông tin cá nhân, tin nhắn, chương trình đào tạo, đăng kí học phần, lịch thi, thời khóa biểu, học phí… |
| 4 | Cập Nhật thông tin cá nhân | Href | Hiển thị lên bảng thay đổi thông tin cá nhân của sinh viên |

***4.4. Đặc tả màn hình chỉnh sửa thông tin cá nhân sinh viên***

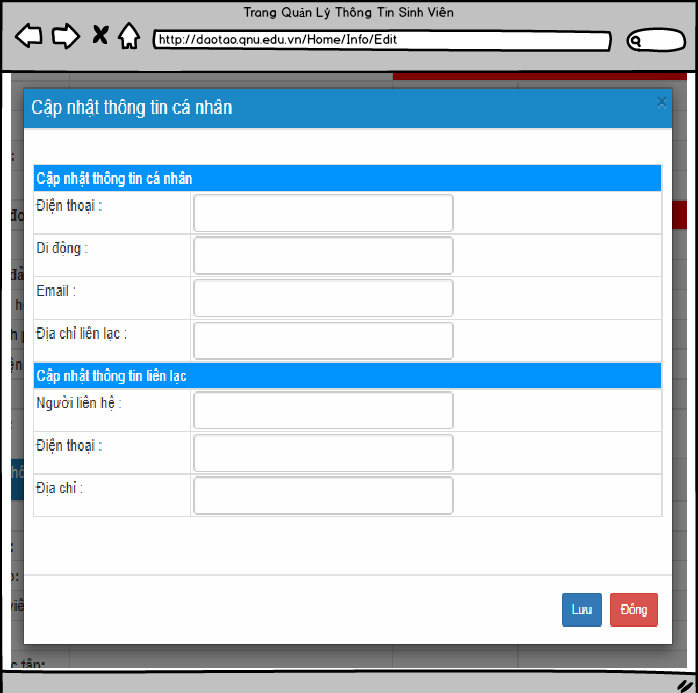
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Mã màn hình* | SCR-05 | *Tên chức năng* | Chỉnh sửa thông tin cá nhân | *Ngày tạo* | 20/9/2020 |
| *Người tạo* | Trần Văn Hoài Nhiên |

­­­­

*1. Tổng quan*

Màn hình hiển thị bảng thông tin để sinh viên có thể chỉnh sửa thông tin cá nhân

*2. Giao diện màn hình*



*3. Các thành phần của màn hình*

1

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Mô tả |
| 1 | Cập nhật thông tin cá nhân, cập nhật thông tin liên lạc | Label | Hiển thị thông tin |
| 2 | Điện thoại, di động, email, địa chỉ liên lạc, người liên hệ, địa chỉ, điện thoại | Label | Hiển thị thông tin |
| 3 | Nhập Điện thoại, di động, email, địa chỉ liên lạc, người liên hệ, địa chỉ, điện thoại | Textbox | Nơi điền thông tin tương ứng với các label ở trên |
| 4 | Nút Lưu | Button | Lưu thông tin đã chỉnh sửa lại và thoát hộp thoại |
| 3 | Nút đóng | Button | Nhấn để thoát hộp thoại |

***4.5. Đặc tả màn hình xem thời khóa biểu***

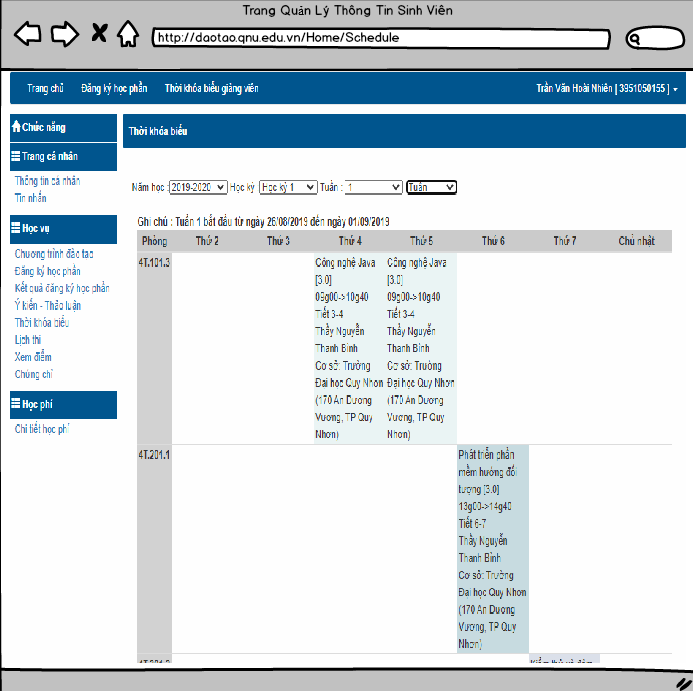
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Mã màn hình* | SCR-06 | *Tên chức năng* | Xem thời khóa biểu | *Ngày tạo* | 20/9/2020 |
| *Người tạo* | Lê Phi Khánh |

­­­­

*1. Tổng quan*

Màn hình hiển thị danh sách các các môn học đã đăng ký và lịch học.

*2. Giao diện màn hình*



*2. Các thành phần của màn hình*

1

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Mô tả |
| 1 | Năm học, Học kỳ, Tuần | Label | Hiển thị các thông tin |
| 2 | Thông tin lịch học | Table | Hiển thị thông tin các môn học đã đăng kí |
| 3 | 2019-2020, học kì I, 1, tuần | Dialog | Lựa chọn thông tin hiển thị theo năm – học kì – tuần |

***4.6. Đặc tả màn hình xem lịch thi***

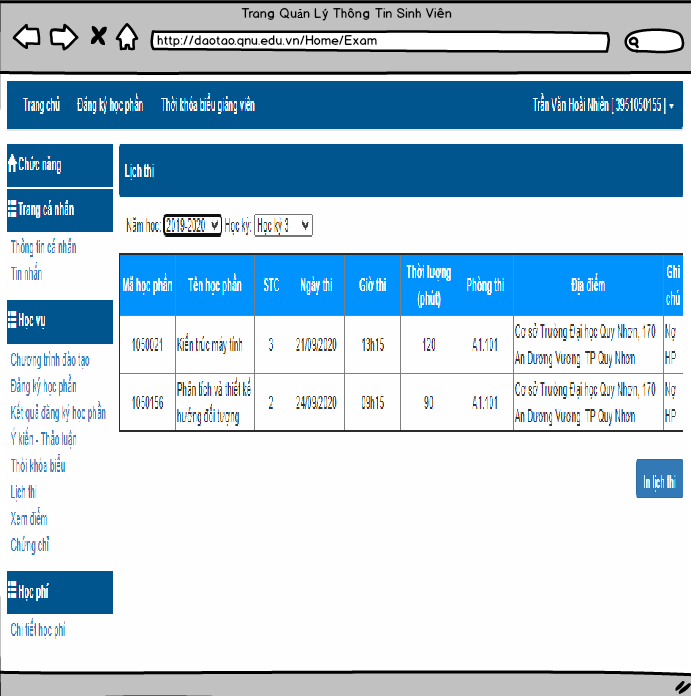
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Mã màn hình* | SCR-08 | *Tên chức năng* | Xem lịch thi | *Ngày tạo* | 20/9/2020 |
| *Người tạo* | Lê Phi Khánh |

­­­­

*1. Tổng quan*

Màn hình cho phép sinh viên xem lịch thi

*2. Giao diện màn hình*



*7.6.3. Các thành phần của màn hình*

1

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Mô tả |
| 1 | Năm học, Học kỳ | Label | Hiển thị các thông tin |
| 2 | Thông tin lịch thi | Table | Hiển thị thông tin các môn thi theo lựa chọn |
| 3 | 2019-2020, học kì I | Dialog | Lựa chọn thông tin hiển thị theo năm – học kì |

**Phần II : Các Framework sử dụng và Triển Khai**

**I. FRAMEWORK SỬ DỤNG TRONG PROJECT**

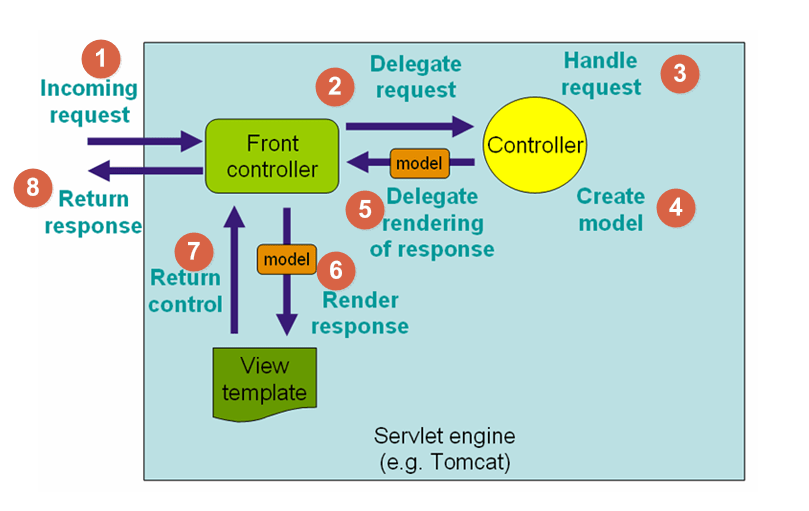
**1. Spring MVC**

# **1.1. Spring MVC (Spring Web MVC)**

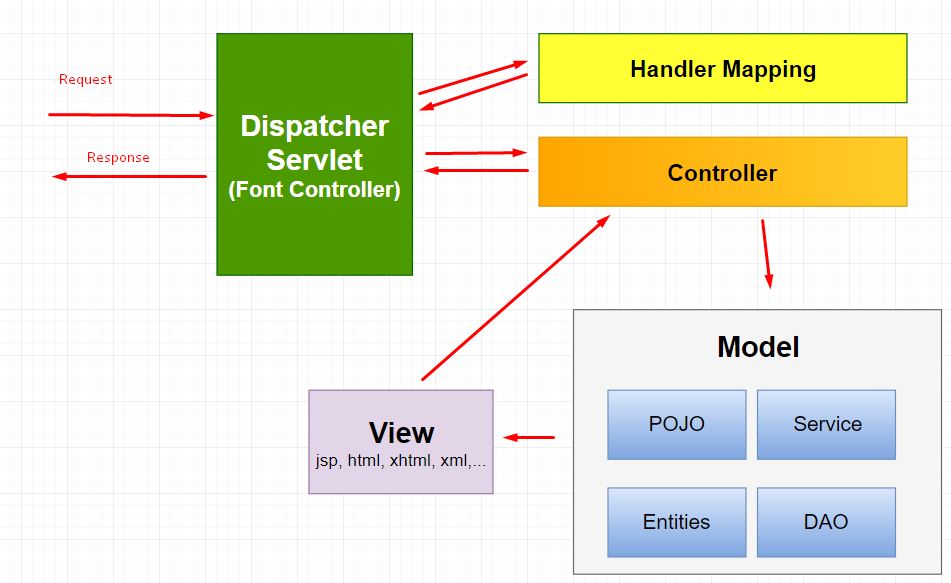
Spring MVC là một Framework / 1 Project mã nguồn mở của Spring.

Spring MVC Framewrok cung cấp kiến truc MVC (Model-View-Controller) và các component được sử dụng để phát triển các ứng dụng web một cách linh hoạt và giảm sư

# ***1.2. Flow trong Spring MVC.***



* Bất kỳ request nào tới ứng dụng web đều sẽ được gửi tới Front Controller (Dispatcher Servlet)
* Front Controller sẽ sử dụng Handler Mapping để biết được controller nào sẽ xử lý request đó
* Controller nhận request, gọi tới các class service thích hợp để xử lý yêu cầu.
* Sau khi xử lý xong, Controller sẽ nhận được model từ tầng Service hoặc tầng DAO.
* Controller gửi model vừa nhận được tới Front Controller (Dispatcher Servlet)
* Dispatcher Servlet sẽ tìm các mẫu view, sử dụng view resolver và truyền model vào nó.
* View template, model, view page được build và gửi trả lại Front Controller
* Front Controller gửi một page view tới trình duyệt để hiển thị nó cho người dùng.



Trong Mô hình MVC thì:

* **Model**: là các file POJO, Service, DAO thực hiện truy cập database, xử lý business
* **View**: là các file JSP, html…
* **Control**: là Dispatcher Controller, Handler Mapping, Controller – thực hiện điều hướn các request.

***1.3. Lợi ích của Spring MVC***

Các tầng trong Spring MVC độc lập nên việc unit test dễ dàng hơn.

Phần view có thể tích hợp với nhiều framework về UI như JSF, Freemarker, Themeleaf…

Spring MVC base trên các POJO class nên các hành động của nó khá đơn giản.

Hỗ trợ cả Annotation và XML config giúp việc phát triển nhanh hơn và sạch hơn.

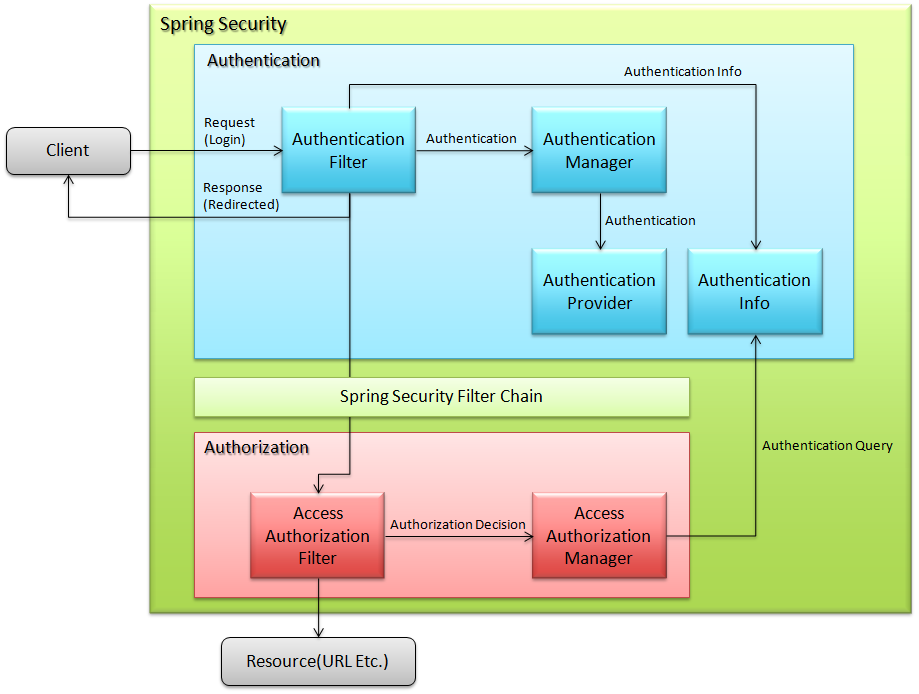
Cung cấp việc phân chia một cách rõ ràng, linh hoạt giữa controller, service, DAO.

**2. Spring Security**

***2.1. Giới thiệu***

**Spring security:**nó là một framework hay đúng hơn là một công cụ cung cấp và xử lý các vấn đề về xác thực và phân quyền cho các ứng dụng web nó là một công cụ luôn song hành và luôn được ưu tiên sử dụng trong các dự án về java trong đó nổi bật nhất là  spring boot.

**Cơ chế :**

[](http://devtayto.com/wp-content/uploads/2018/09/spring_security_overview.png)

Spring security cung cấp cho ta 2 cơ chế cơ bản

* **Authentication:**cơ chế xác thực xem bạn có được quyền vào ứng dụng của bạn không túm váy như kiểu đăng nhập bằng username và password đúng hay sai vậy đây là ải đầu tiên của spring security.
* **Authorization:**cung cấp cho chúng ta cơ chế phân quyền xem user này có được thao tác hành động nào đó không, đơn giản nhất là admin có quyền thêm xóa thành viên có quyền xem bài vậy tất nhiên tới ải này được thì chúng ta phải vượt qua được ải thứ nhất.

***2.2. Các thành phần***

*2.2.1. SecurityContext*

***SecurityContext*** là interface cốt lõi của Spring Security, lưu trữ tất cả các chi tiết liên quan đến bảo mật trong ứng dụng. Khi chúng ta kích hoạt Spring Security trong ứng dụng thì SecurityContext cũng sẽ được kích hoạt theo.

*2.2.2. SecurityContextHolder*

Chúng ta sẽ không truy cập trực tiếp vào SecurityContext, thay vào đó sẽ sử dụng lớp ***SecurityContextHolder***. Lớp này lưu trữ security context hiện tại của ứng dụng, bao gồm chi tiết của principal đang tương tác với ứng dụng. Spring Security sẽ dùng một đối tượng Authentication để biểu diễn thông tin này.

*2.2.3. UserDetails*

***UserDetails*** là một interface cốt lõi của Spring Security. Nó đại diện cho một principal nhưng theo một cách mở rộng và cụ thể hơn.UserDetails bao gồm các method sau:

* getAuthorities(): trả về danh sách các quyền của người dùng
* getPassword(): trả về password đã dùng trong quá trình xác thực
* getUsername(): trả về username đã dùng trong quá trình xác thực
* isAccountNonExpired(): trả về true nếu tài khoản của người dùng chưa hết hạn
* isAccountNonLocked(): trả về true nếu người dùng chưa bị khóa
* isCredentialsNonExpired(): trả về true nếu chứng thực (mật khẩu) của người dùng chưa hết hạn
* isEnabled(): trả về true nếu người dùng đã được kích hoạt

Đoạn code saulà kết hợp giữa ***SecurityContextHolder*** và method ***getUsername()*** trong ***UserDetails***, để lấy tìm người dùng bằng chính username đó(username ở đây ta nên hiểu là username trong cặp username - password mà người dùng nhập vào khi đăng nhập):



*2.2.4. UserDetailsService*

***UserDetailsService*** là một interface có duy nhất một phương thức. Tham số truyền vào chỉ gồm có username của người dùng. Ta sẽ tìm kiếm trong CSDL, phảitồn tại username. Nếu không tìm thấy, ta sẽ ném ra ngoại lệ UsernameNotFoundException.

Phương thức đó là loadUserByUsername(), có thể trả về mộtimplementation của UserDetails. Có thể là org.springframework.security.core.userdetails.User.

Và ta cần phải tạo một lớp để implements UserDetailsService.

*2.2.5. GrantedAuthority*

Ở phần trên, đã đề cập đến phương thức getAuthorities(). Phưong thức này sẽ trả về một tập hợp các đối tượng ***GrantedAuthority***. Một GrantedAuthority là một quyền được cấp cho principal. Các quyền đều có tiền tố là ROLE\_, ví dụ như ROLE\_ADMIN, ROLE\_MEMBER…

***2.3. Vidu về Spring Security***

Đầu tiên mình sẽ  thêm những  **dependency**cần thiết dưới đây vào vào file **pom.xml**

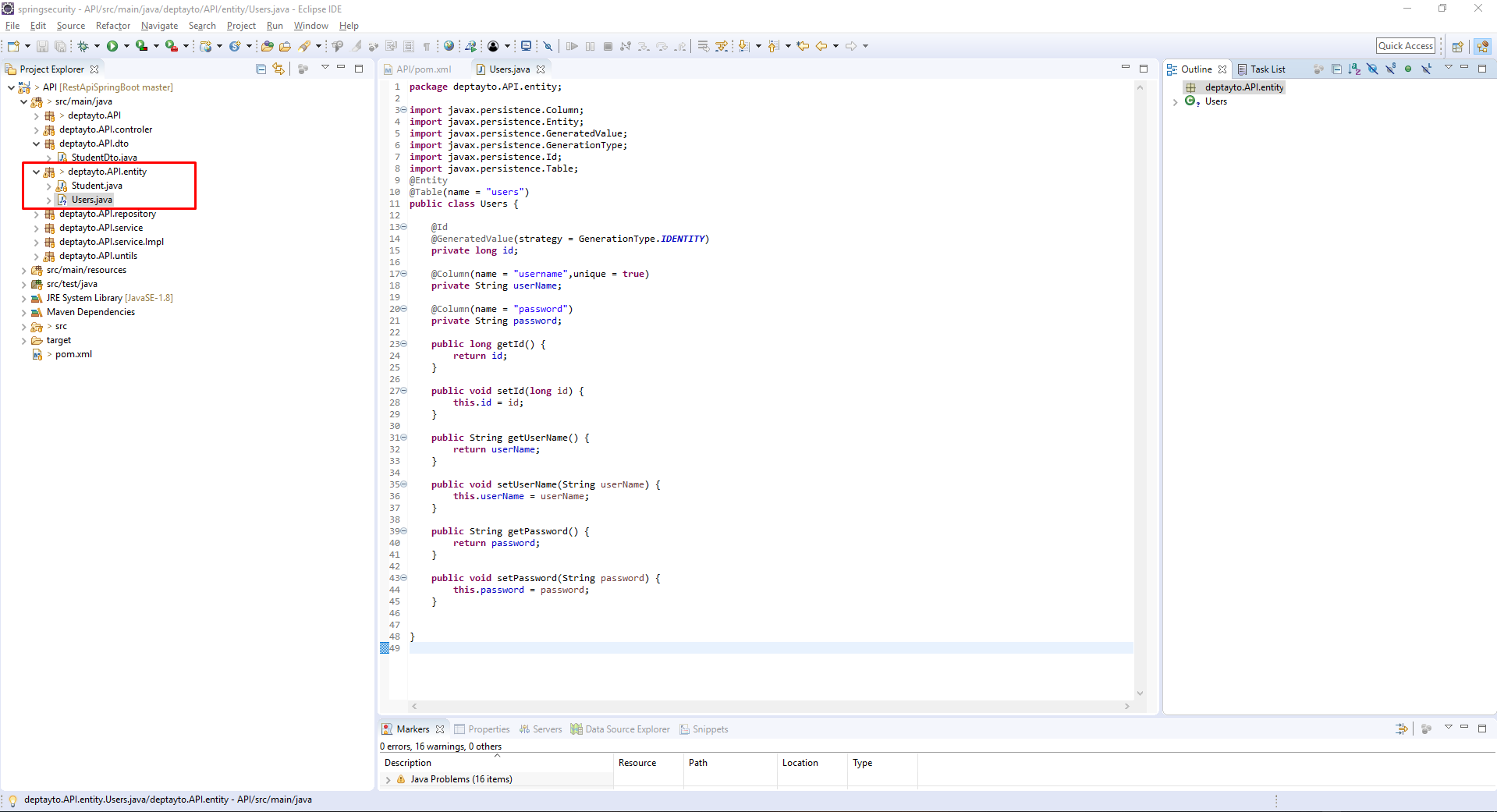
**<dependency>  
<groupId>io.jsonwebtoken</groupId>  
<artifactId>jjwt</artifactId>  
<version>0.9.0</version>  
</dependency>**

**<!– https://mvnrepository.com/artifact/com.auth0/auth0 –>  
<dependency>  
<groupId>com.auth0</groupId>  
<artifactId>auth0</artifactId>  
<version>1.5.0</version>  
</dependency>**

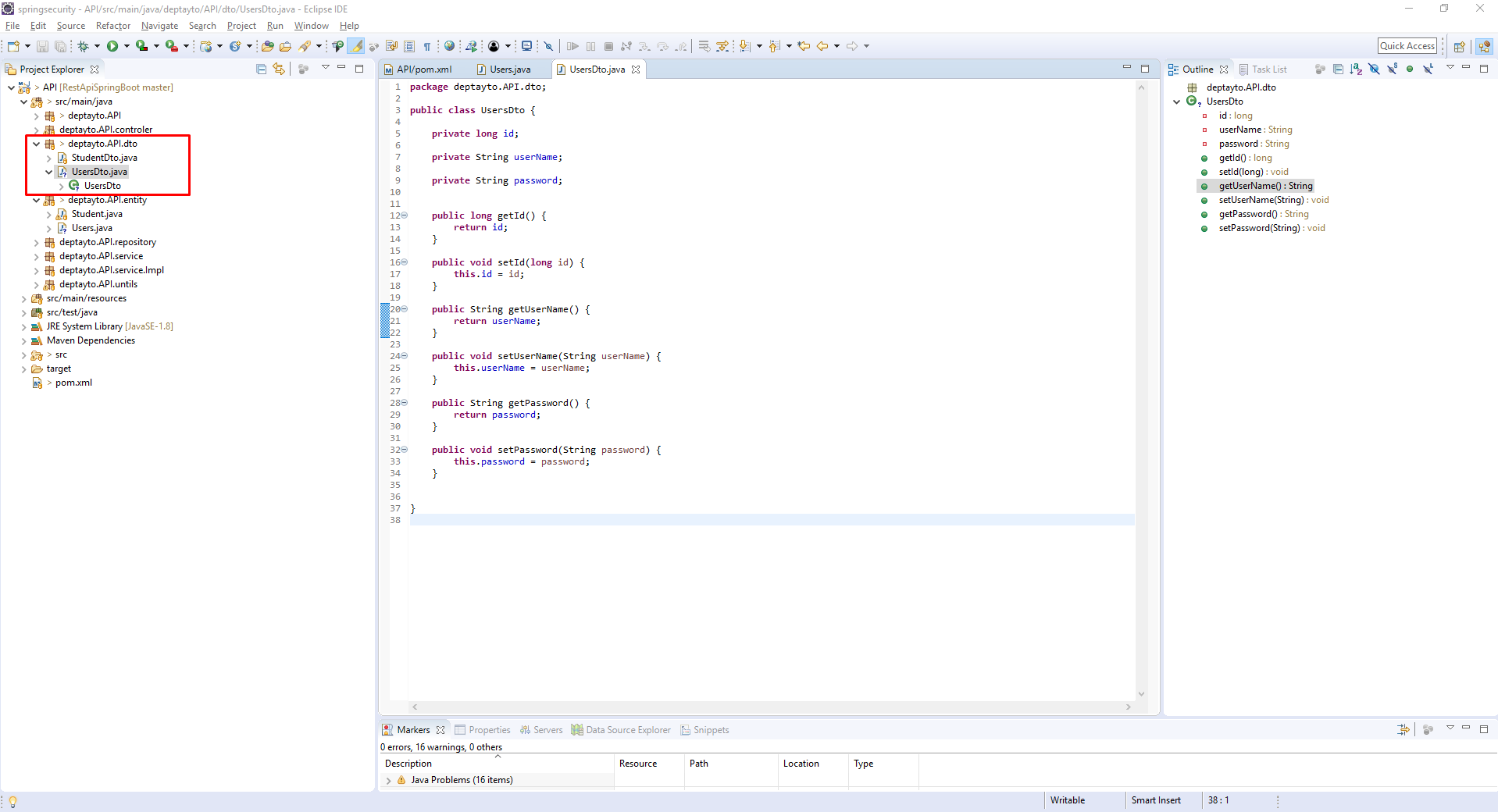
**<!– https://mvnrepository.com/artifact/com.auth0/auth0-spring-security-api –>  
<dependency>  
<groupId>com.auth0</groupId>  
<artifactId>auth0-spring-security-api</artifactId>  
<version>1.0.0</version>  
</dependency>**

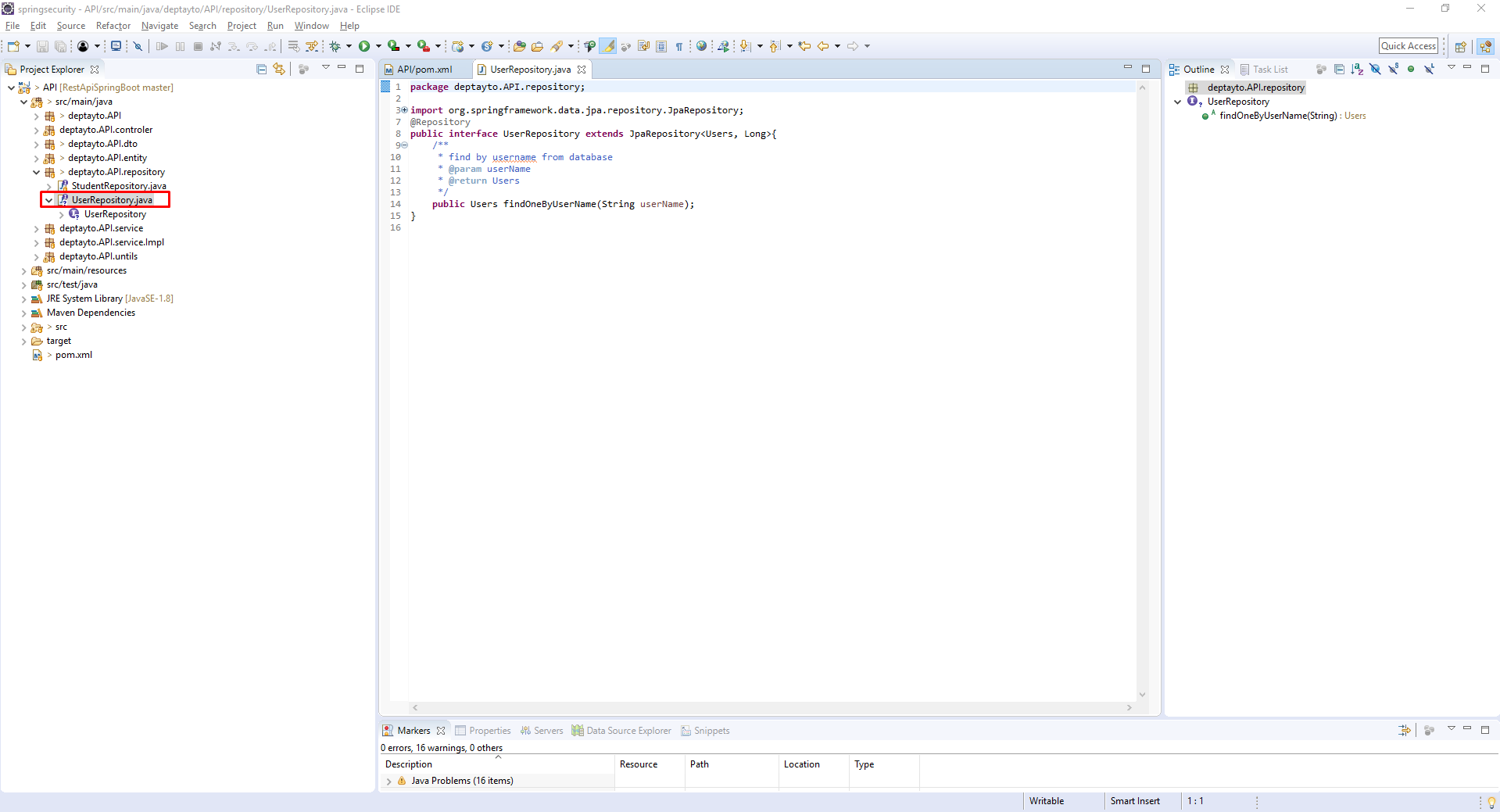
[](http://devtayto.com/wp-content/uploads/2018/09/springscurity1-1.png)

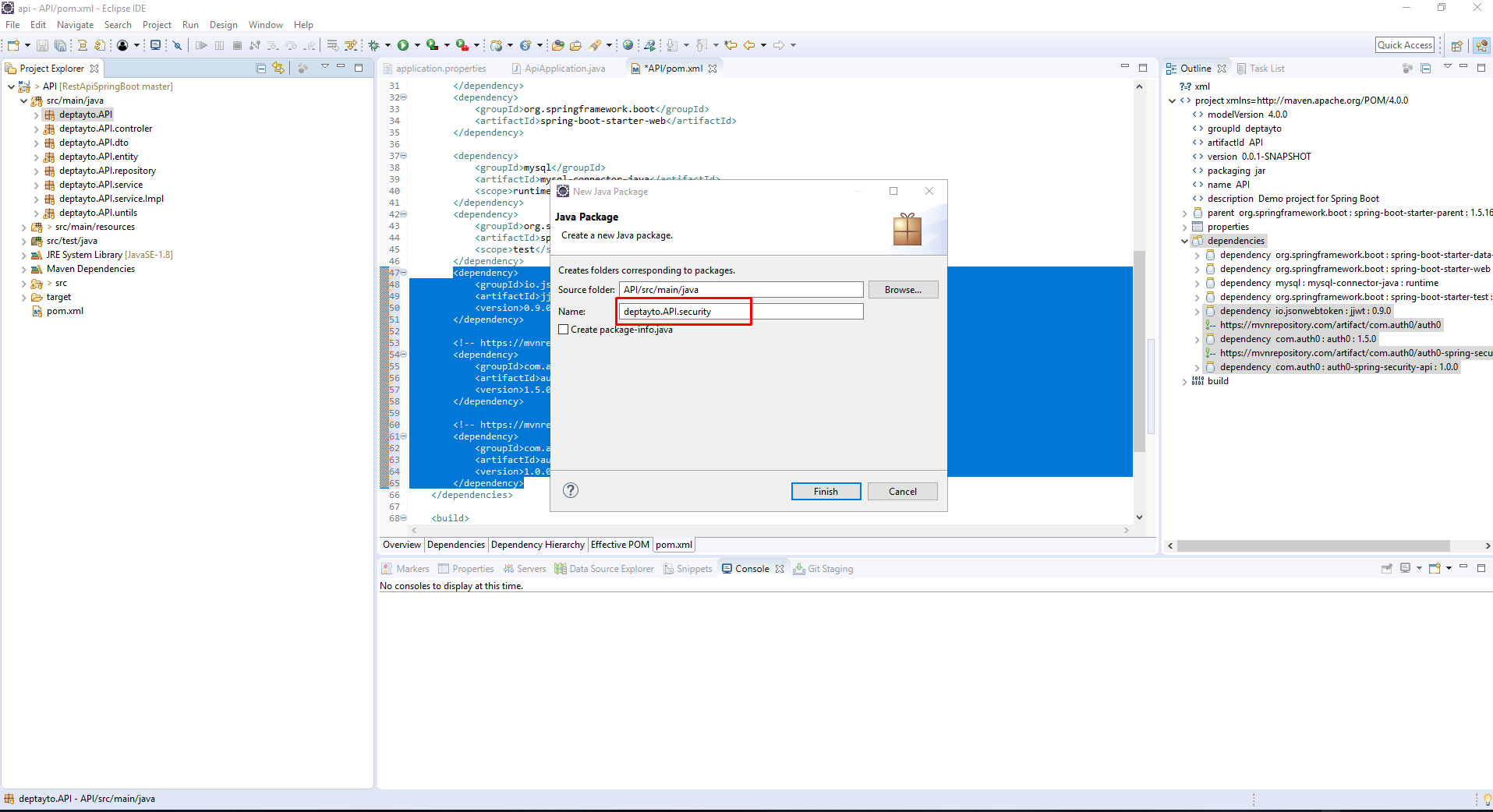
* Đầu tiên để bắt add authentication vào source ta cần tạo một đối tượng user để chứa thông tin của các user login ở đây mình tạo 1 cái đơn giản chỉ gồm username và password

[](http://devtayto.com/wp-content/uploads/2018/09/springscurity6.png)

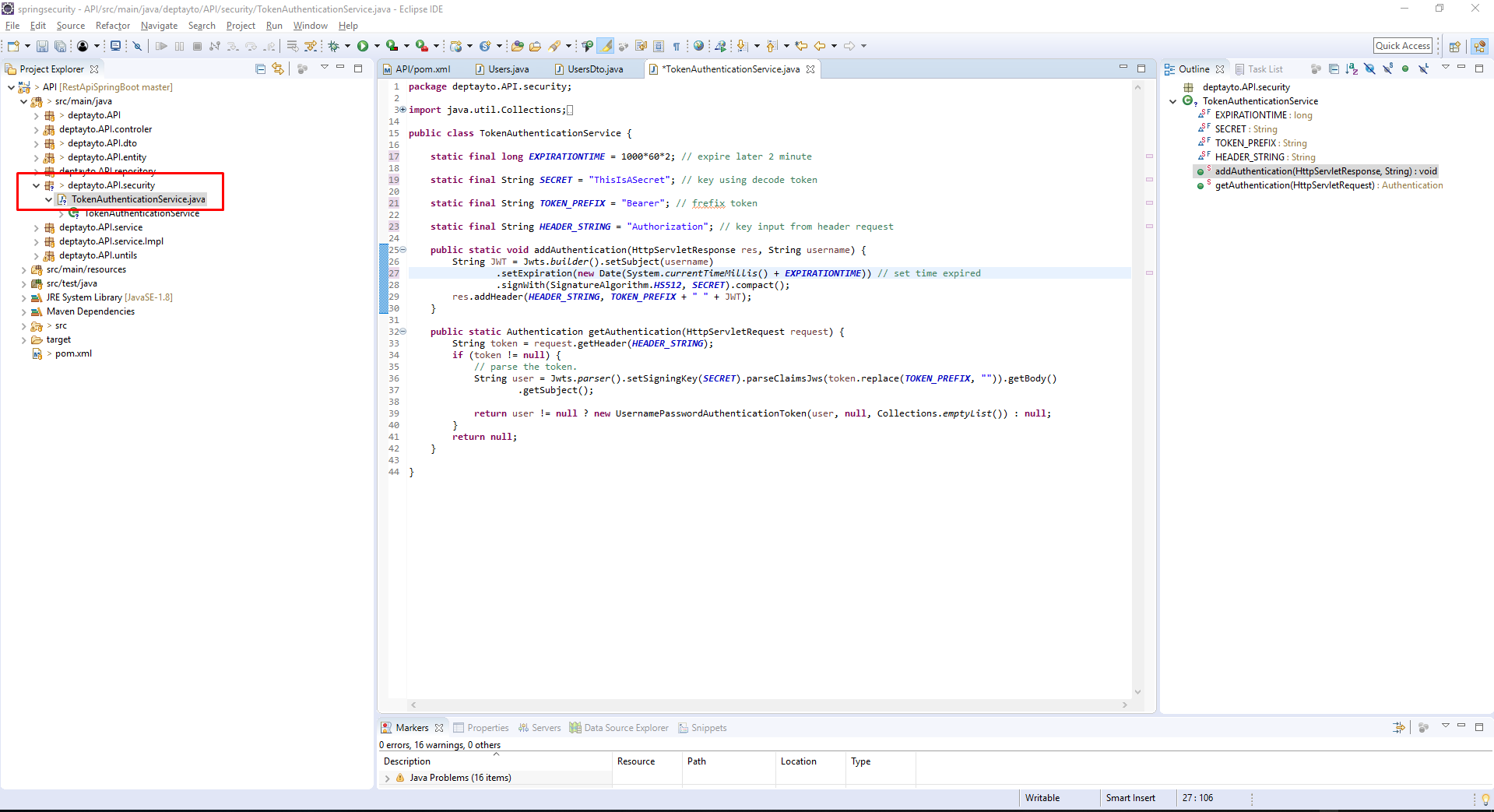
* Tiếp tới ta tạo 1 class **UsersDto.java**

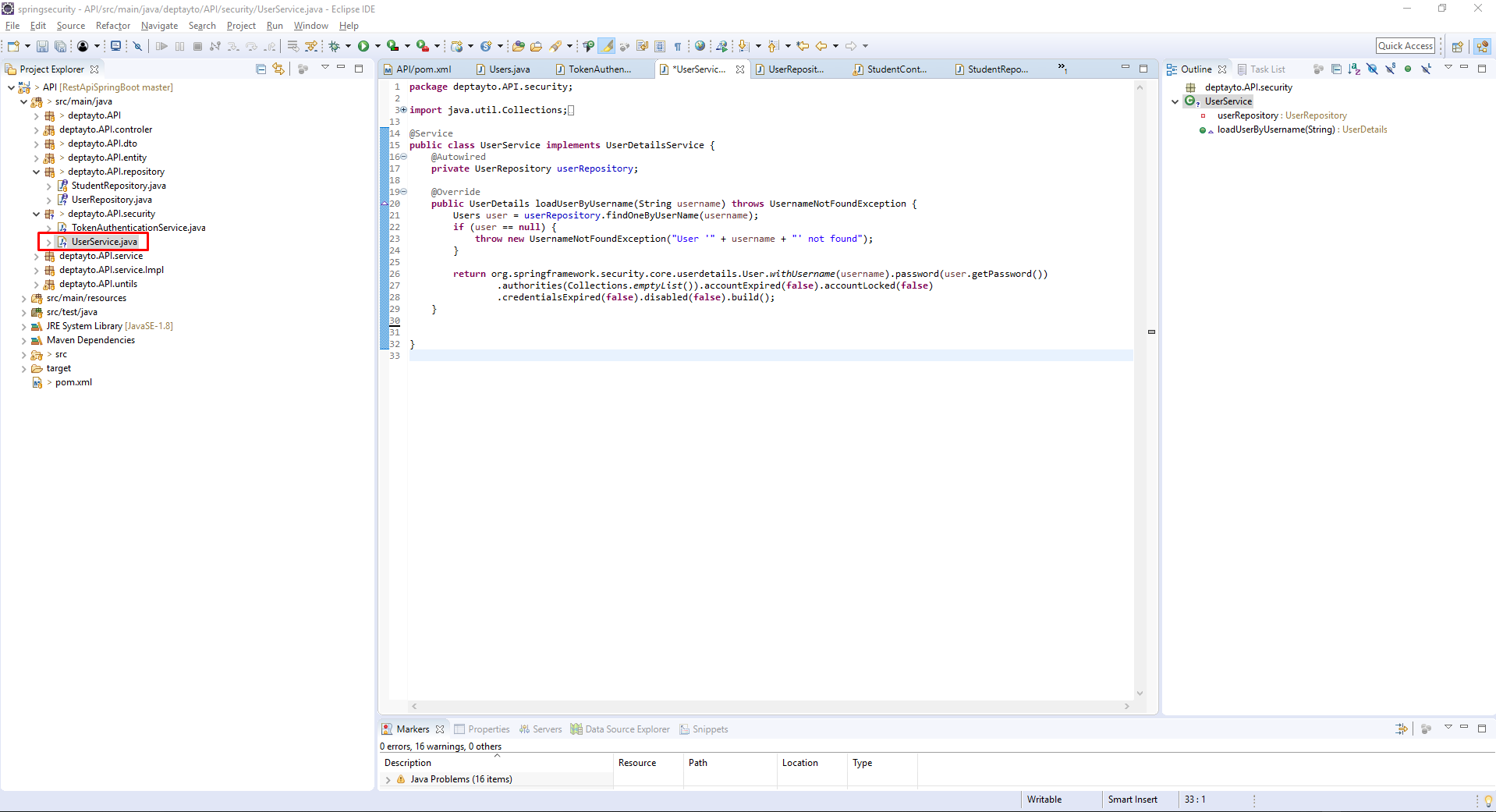
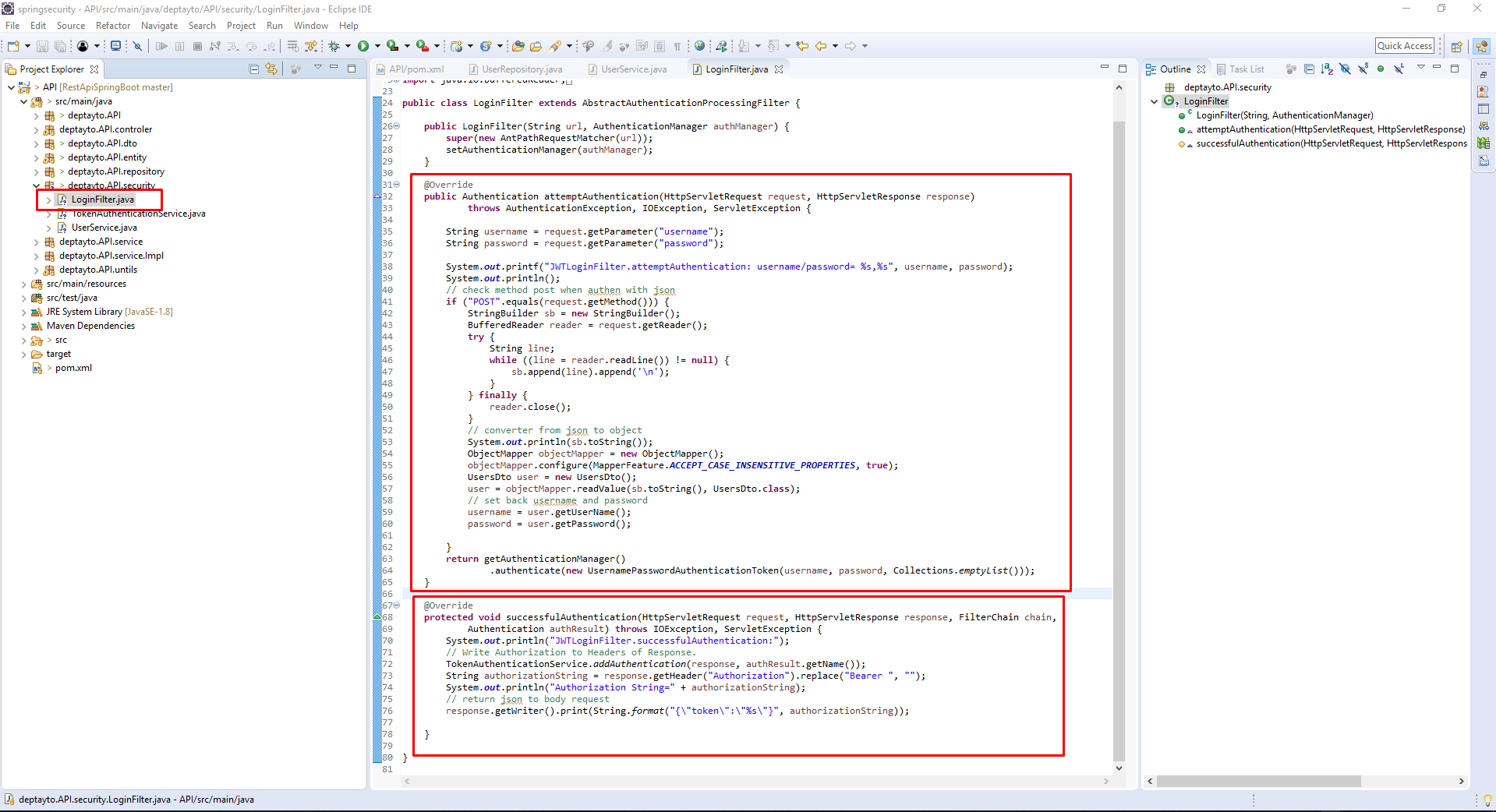
[](http://devtayto.com/wp-content/uploads/2018/09/springscurity7.png)

* Tiếp tới mình tạo thêm 1 class **UserRepository.java** để lấy thông tin User từ database phục vụ cho việc authentication.
* [](http://devtayto.com/wp-content/uploads/2018/09/springsecurity3.png)
* Tiếp theo chúng ta tạo thêm 1 package **security** như hình vì để cho các bạn dễ theo dõi cũng như dễ hiểu mình sẽ để tất cả các class liên quan tới authentication trong package khi code được rồi thì các bạn tự tách ra cho đúng với cấu trúc source nhé.

[](http://devtayto.com/wp-content/uploads/2018/09/springscurity2.png)

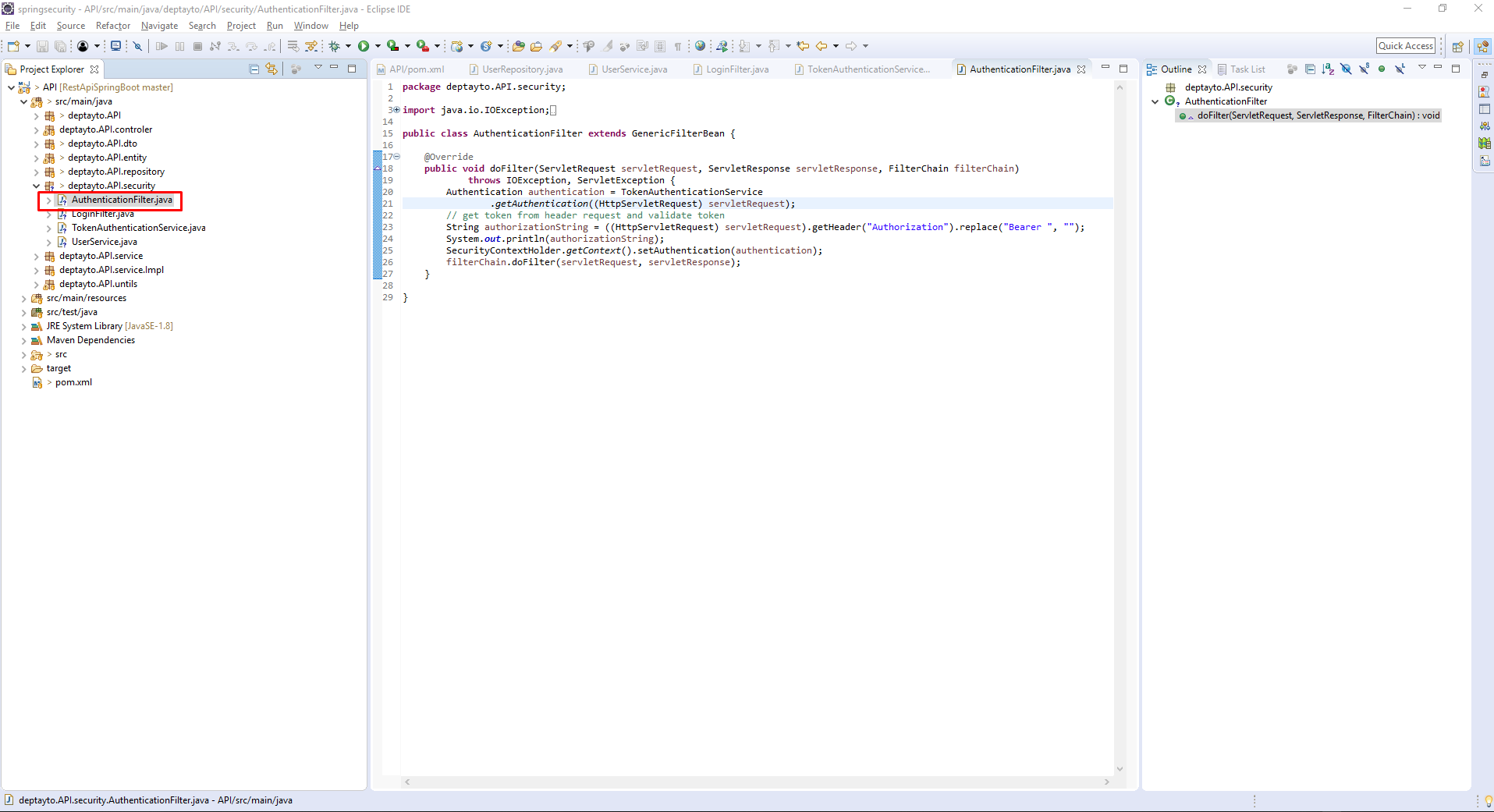
* Đầu tiên ta sẽ tạo 1 file **TokenAuthenticationService.java**để xử lý nhiệm vụ generate ra jwt (json web token)

[](http://devtayto.com/wp-content/uploads/2018/09/springscurity8.png)

* Ở đây vì mình sẽ lấy thông tin user từ database để authentication nên mình phải có 1 service để tìm user được nhập vào khi authentication xem nó có trong database không vì thế ở đây mình sẽ tạo tiếp 1 class **UserService.java** và implements lại **UserDetailsService.java**mà spring cung cấp để có thể so sánh được thông tin của user trong database và thông tin user được nhập vào[](http://devtayto.com/wp-content/uploads/2018/09/springsecurity2.png)
* Khi authentication chúng ta cần nhập vào username và password để authentication chính vì vậy chúng ta cần tạo 1 class để hứng thông tin nhập vào và tiến hành trả lại token ra cho người dùng nếu thành công vì thế mình sẽ tạo thêm 1 class **LoginFilter.java**kế thừa lại **AbstractAuthenticationProcessingFilter.java**của spring nhằm mục đích custom lại 2 hàm **attemptAuthentication**có nhiệm vụ **authen và successfulAuthentication**có nhiệm vụ xử lý trả ra token cho người dùng nếu authen thành công.
* [](http://devtayto.com/wp-content/uploads/2018/09/springsecurity4.png)

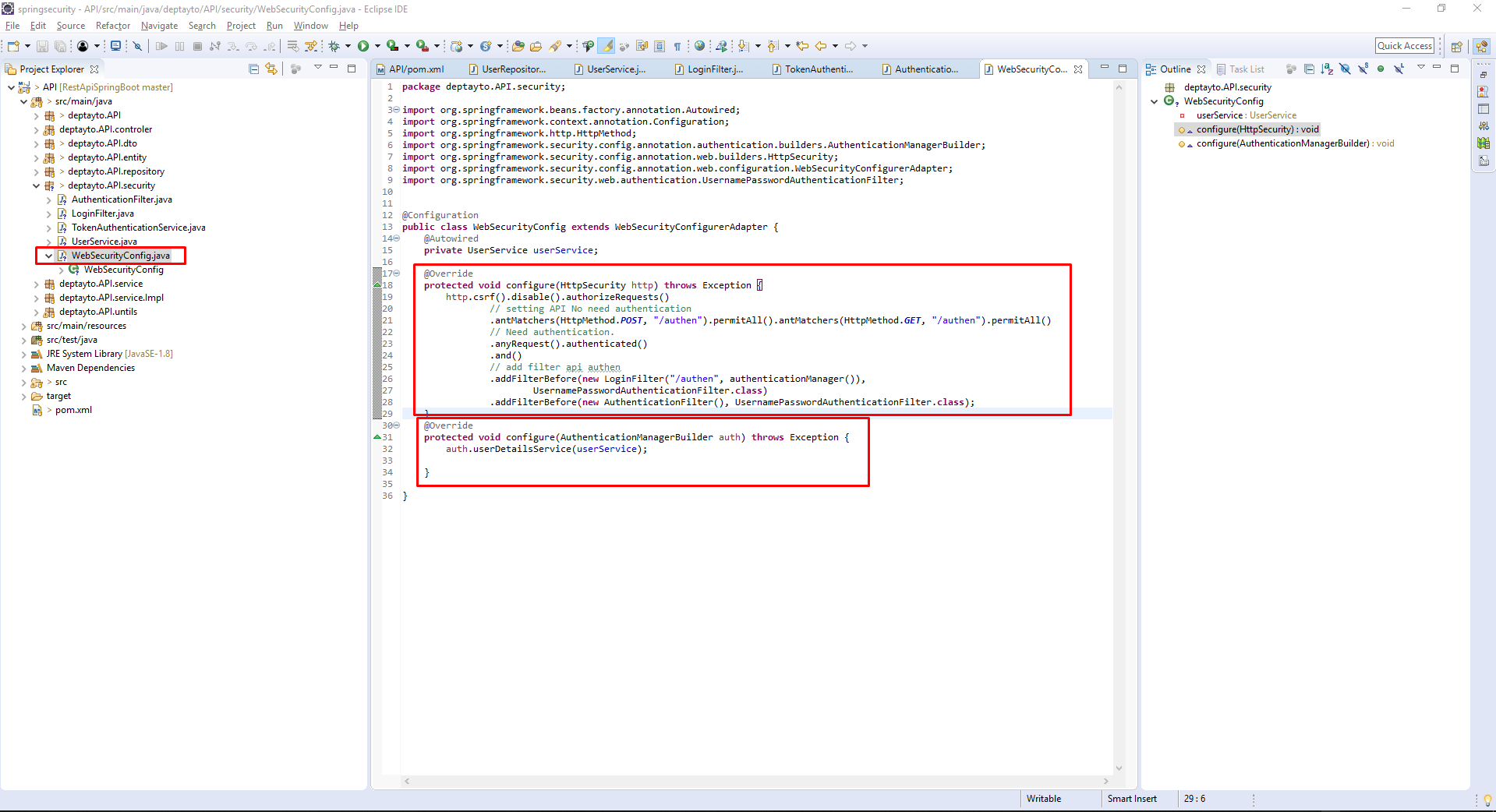
Như chúng ta đã biết khi authen thành công thì chúng ta sẽ dùng token để có thể truy cập các API khác, vậy thì class xử lý token đó xem có đúng hay là sai ?vì vậy chúng ta cần phải tạo ra 1 file filter để hứng tất cả các request trước khi cho nó có quyền vào bất kì 1 API nào được chỉ định là phải có token mới được vào.

Ở đây mình sẽ tạo thêm 1 class để hứng các request đó  và đặt tên là **AuthenticationFilter.java**

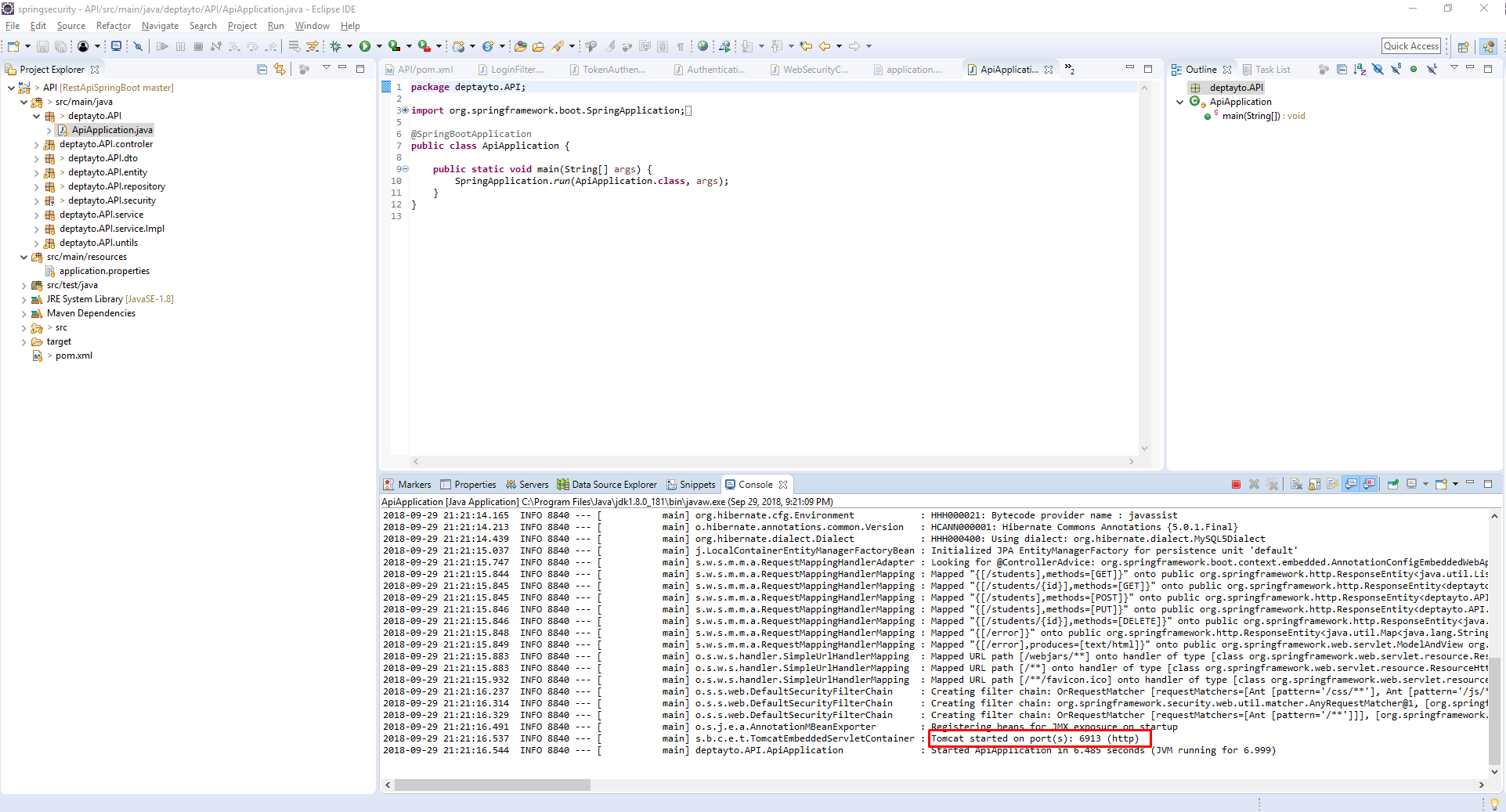
[](http://devtayto.com/wp-content/uploads/2018/09/springsecurity6.png)

Vậy là tạm xong giờ chúng ta chỉ cần tạo thêm 1 class để config xem API nào sẽ bị chặn và API nào sẽ được truy cập mà không cần qua class authen ở trên nên  mình sẽ tạo thêm 1 class **WebSecurityConfig.java .**

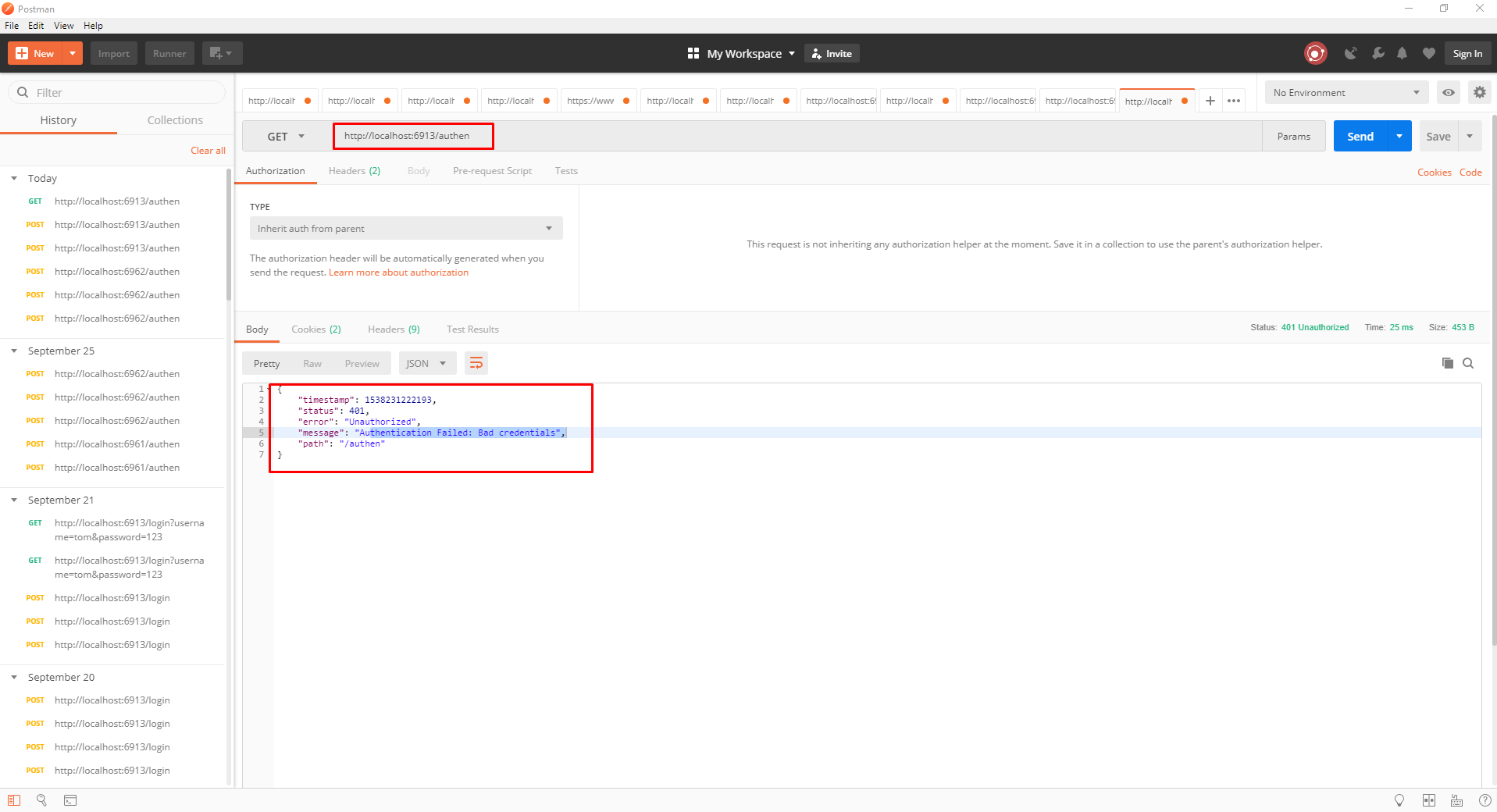
Ở đây mình sẽ Override 2 method configure với configure đầu dùng để lọc API và configure 2 dùng để chỉ định sẽ lấy user trong DB để authen

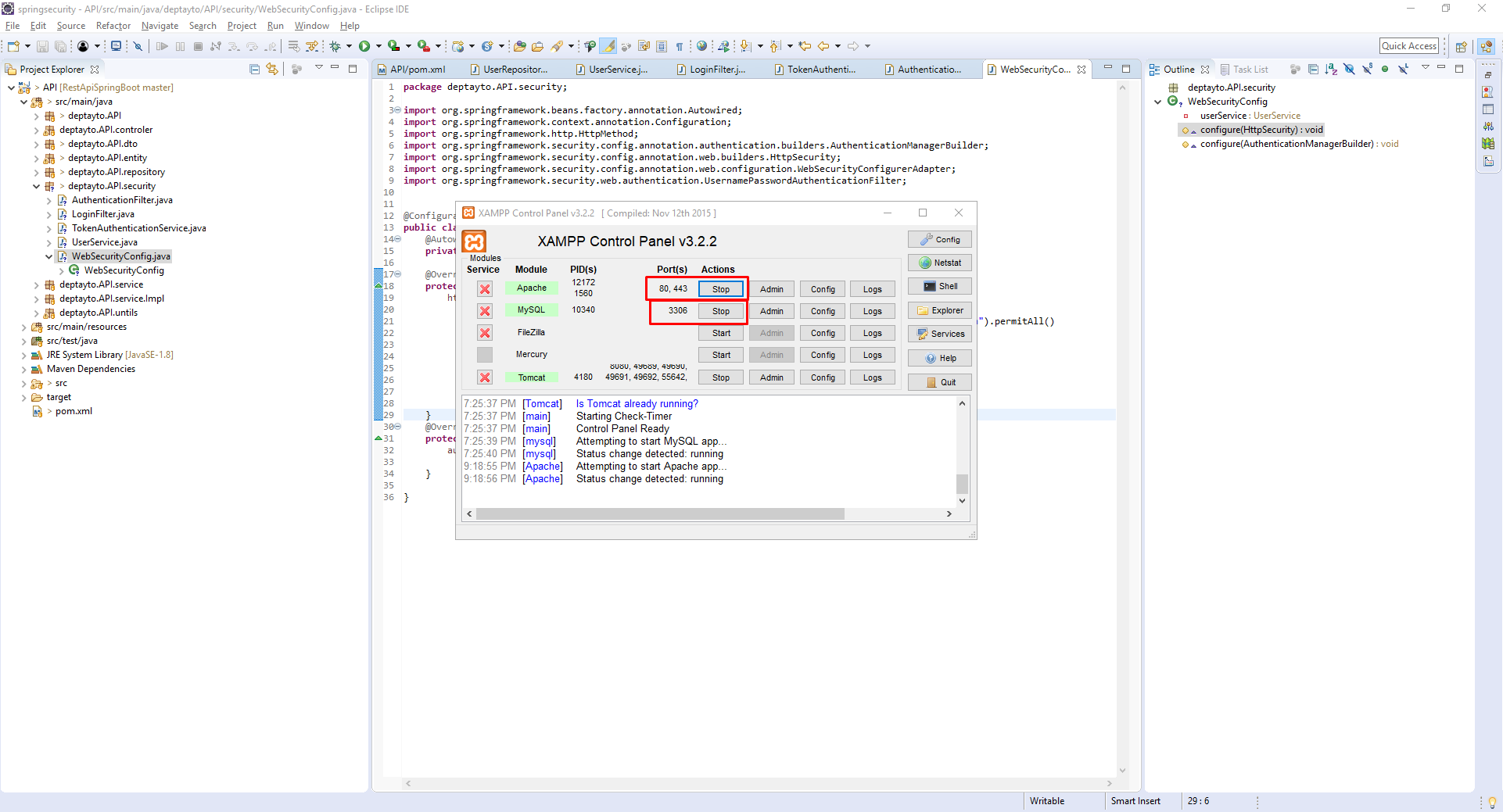
[](http://devtayto.com/wp-content/uploads/2018/09/springsecurity7.png)

Ok vậy là xong chạy thử chương trình lên ta được như hình

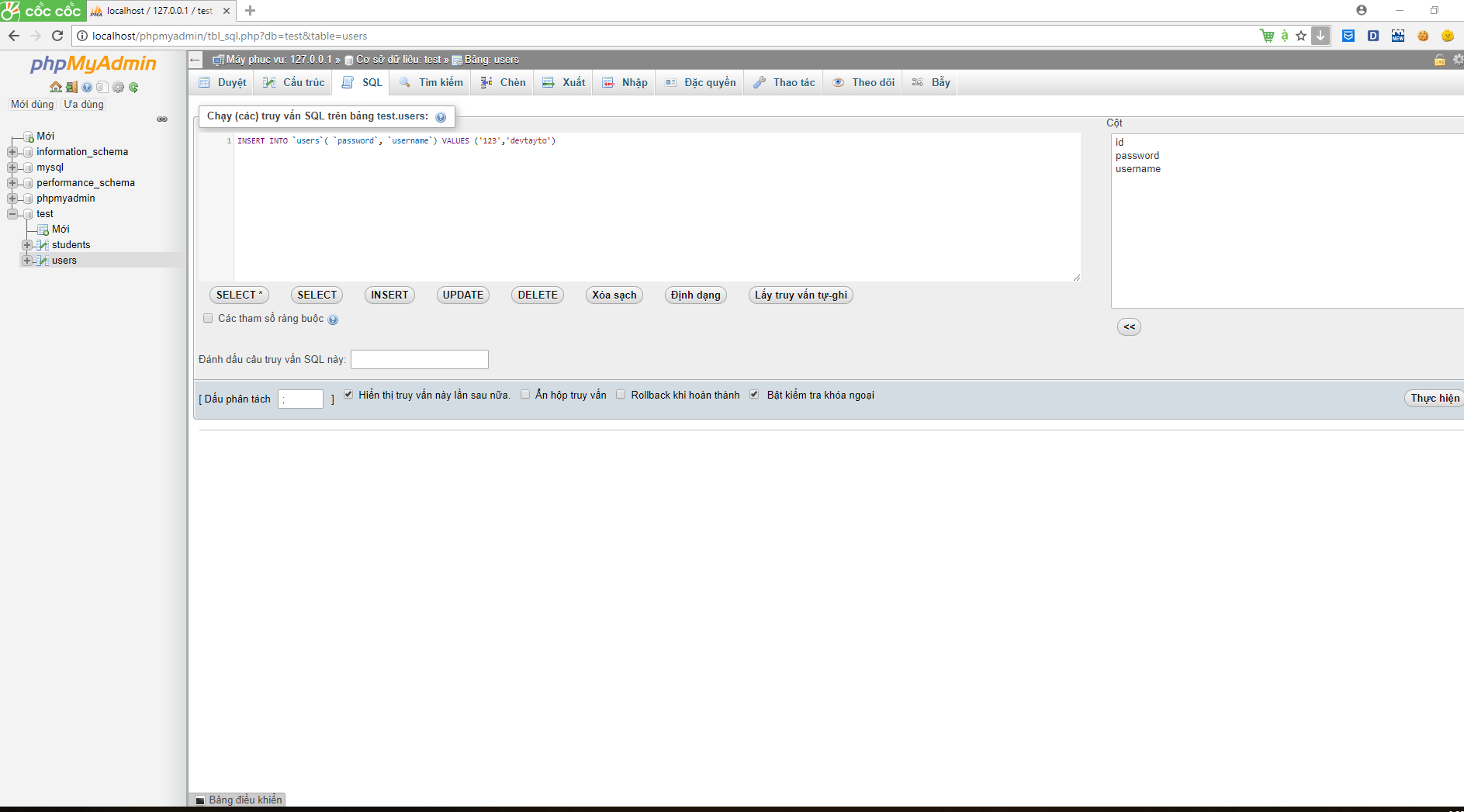
[](http://devtayto.com/wp-content/uploads/2018/09/springsecurity10.png)

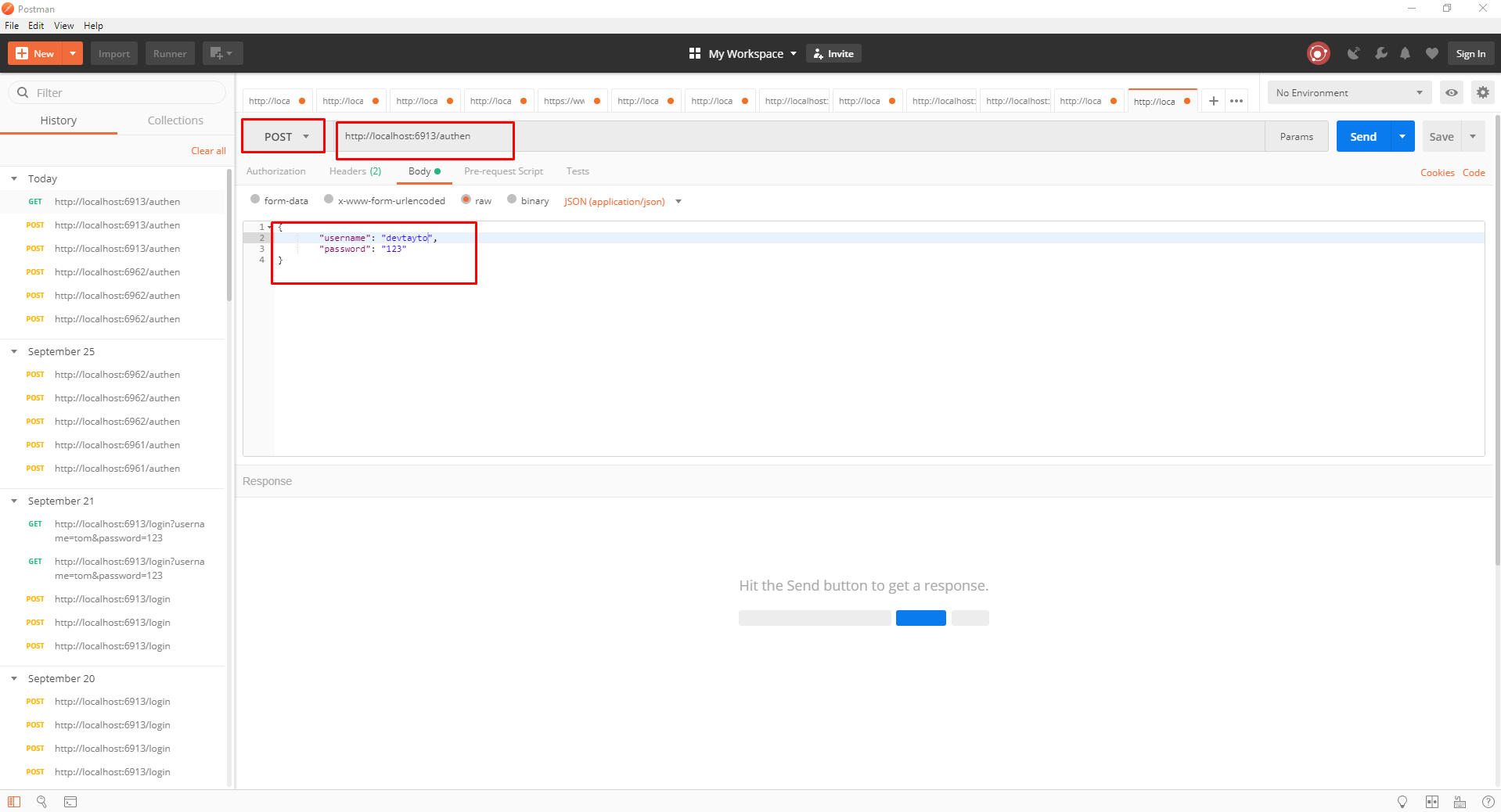
ở đây mình đang để port là 6913 lên ta sẽ mở postman test thử API vừa tạo

[](http://devtayto.com/wp-content/uploads/2018/09/springsecurity11.png)

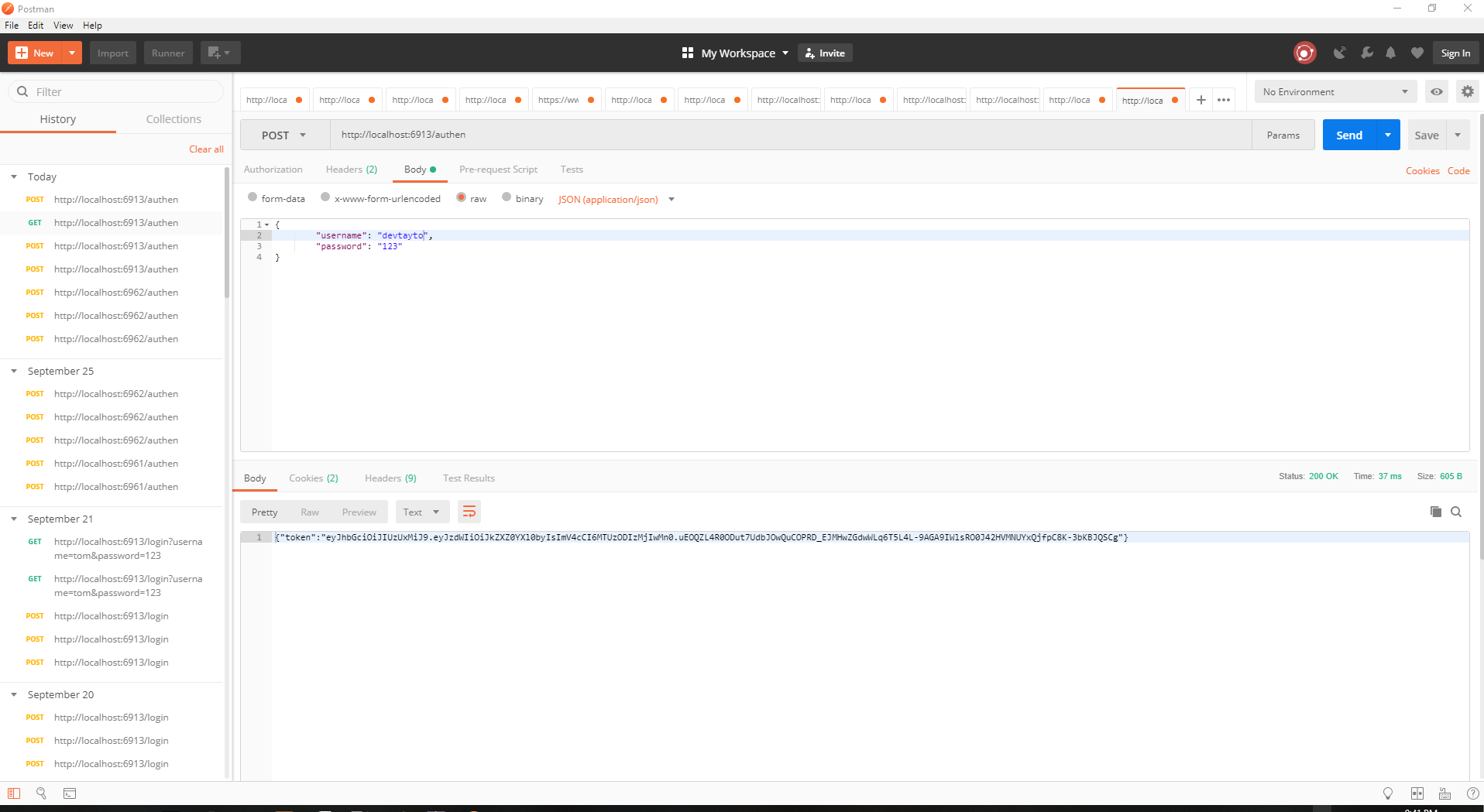
Mở xampp bật localhost và MySQL nên để add thử 1 user vào sau đó tiến hành authen [](http://devtayto.com/wp-content/uploads/2018/09/springsecurity8.png)

truy cập vào http://localhost/phpmyadmin/ ở đây trong application.xml mình đang sử dụng DB test ok vào sửa thôi. ở đây mình sẽ add username = hoainhien và pass: 123

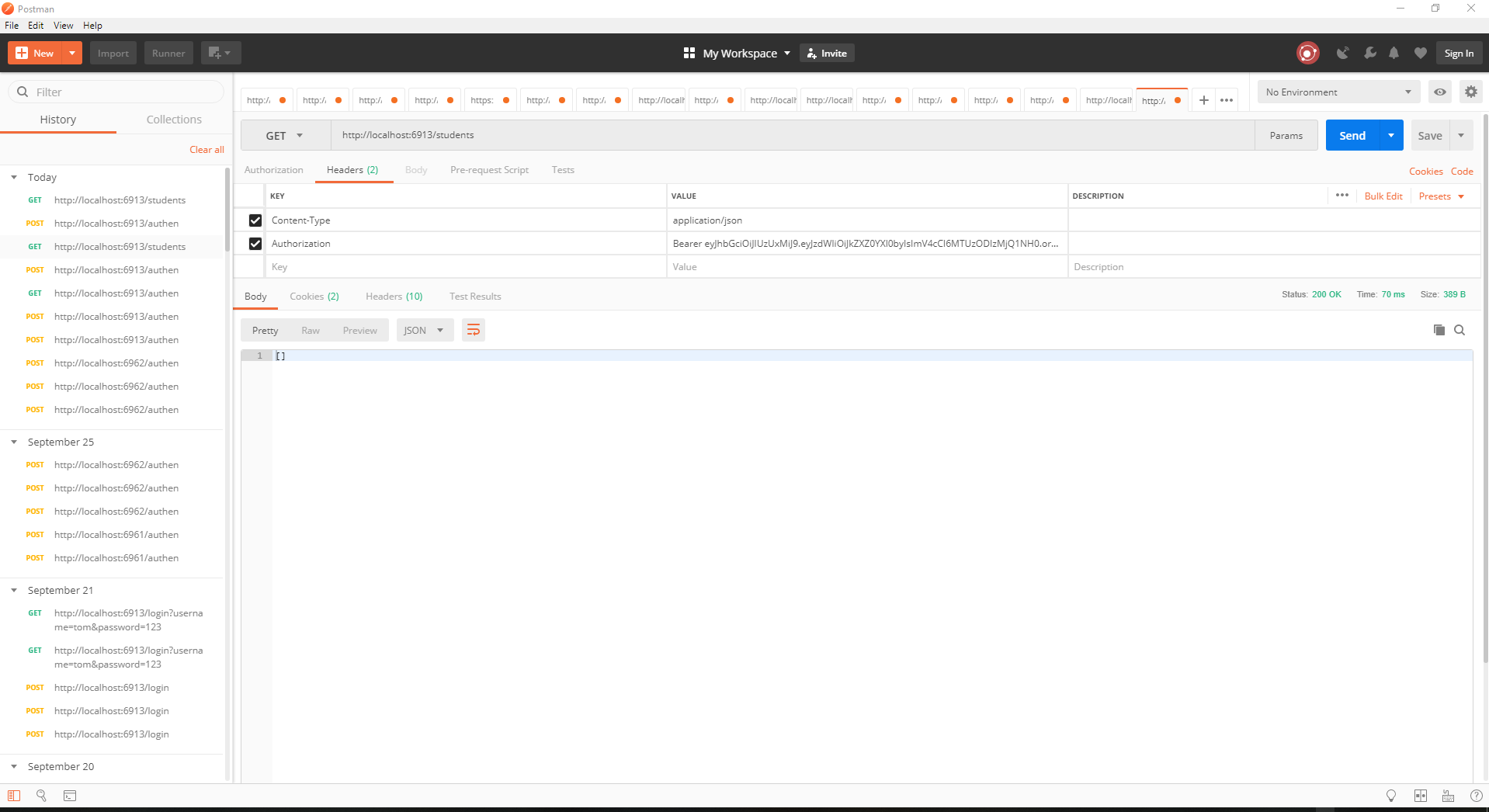
[](http://devtayto.com/wp-content/uploads/2018/09/springsecurity12.png)

[](http://devtayto.com/wp-content/uploads/2018/09/springsecurity13.png)

Kết  quả ta được

[](http://devtayto.com/wp-content/uploads/2018/09/springsecurity14.png)

Đã lấy được token giờ ta thử dùng để truy cập các API khác, thêm Bearer ở đầu cách ra xong điền token như hình để authen

[](http://devtayto.com/wp-content/uploads/2018/09/springsecurity15.png)

**II. CẤU HÌNH VÀ TRIỂN KHAI**

**1. Cấu hình và triển khai lên host**

***1.1. Chuẩn bị***

Project java đã chạy được trên localhost, đã được push lên github.

Hosting để triển khai ứng dụng web java. Nhóm dùng host free có tên Heroku.

* Link: <https://www.heroku.com/>

Vì hostHeroku không hỗ trợ MySQL nên phải cần database online, ở đây nhóm dùng Freemysqlhosting.

* Link: <https://www.freemysqlhosting.net/>

Nếu có thẻ VISA hoặc MasterCard thì có thể dùng host AWS của Amazon vì cái này nó sẽ chạy nhanh hơn và nó hỗ trợ cả MySQL

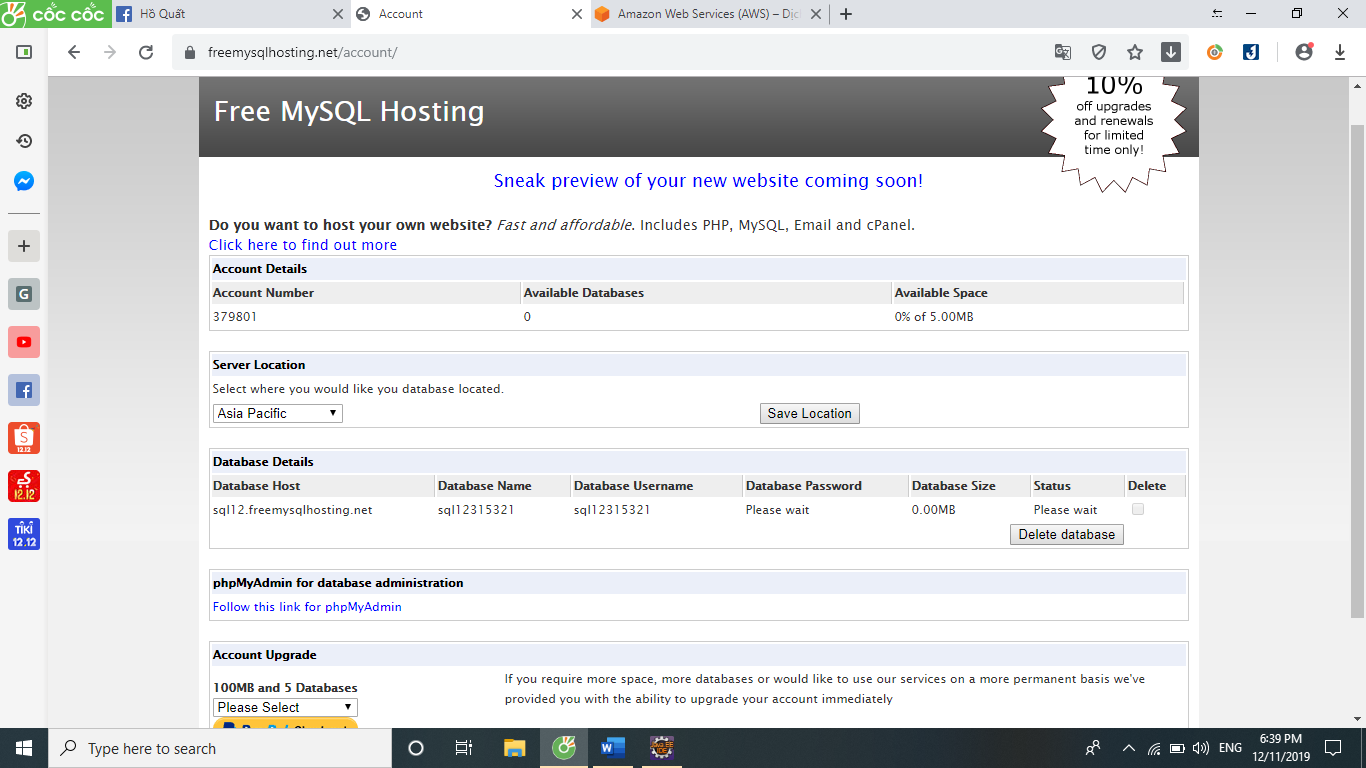
* Link: <https://aws.amazon.com/vi/>

***1.2. Triển khai***

*1.2.1. Triển khai database lên database online*

**Bước 1:** Truy cập vào link ở trên để vô trang chủ Freemysqlhosting.

**Bước 2:**Đăng ký tài khoản để sử dụng để đăng nhập vào Freemysqlhosting. Dưới đây là giao diện khi đăng nhập thành công.



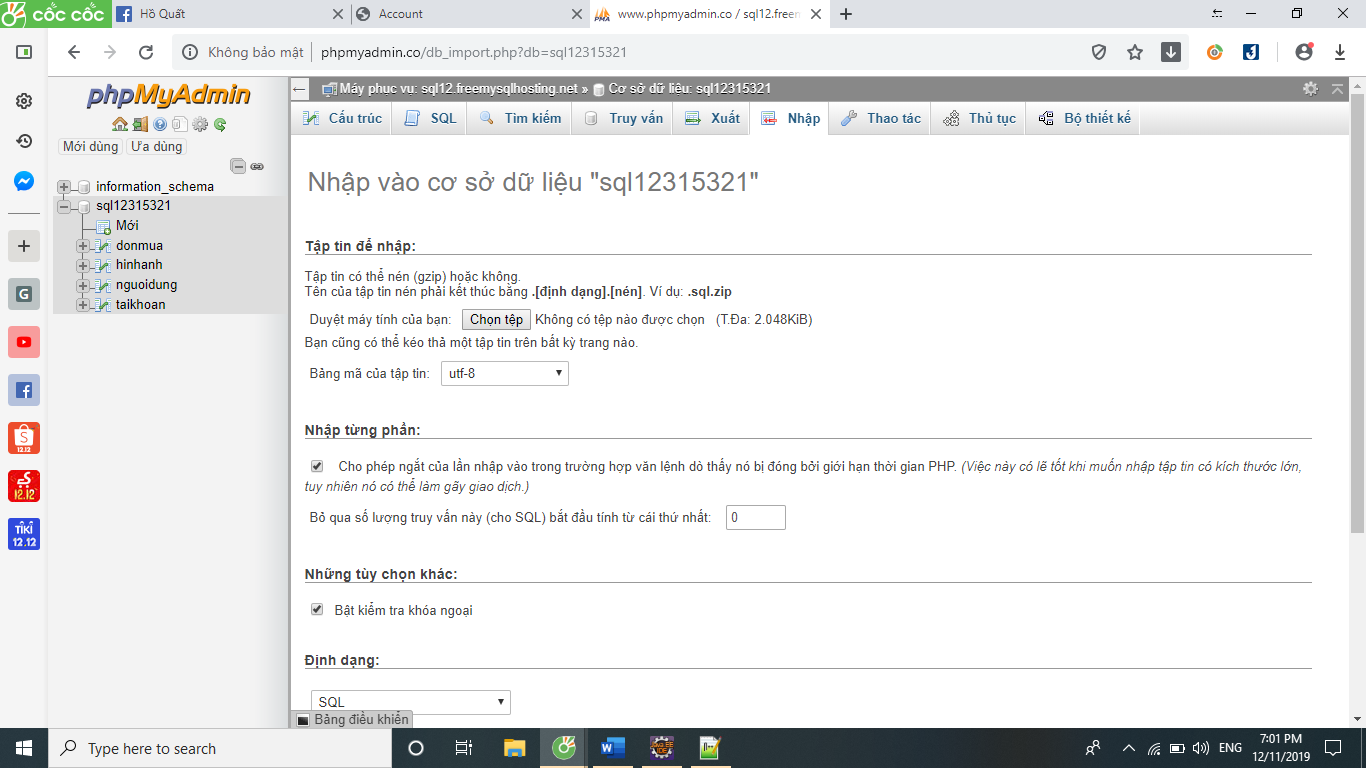
**Bước 3:** Nhấn vào [Follow this link for phpMyAdmin](http://www.phpmyadmin.co/) để truy cập vào phpmyadmin.



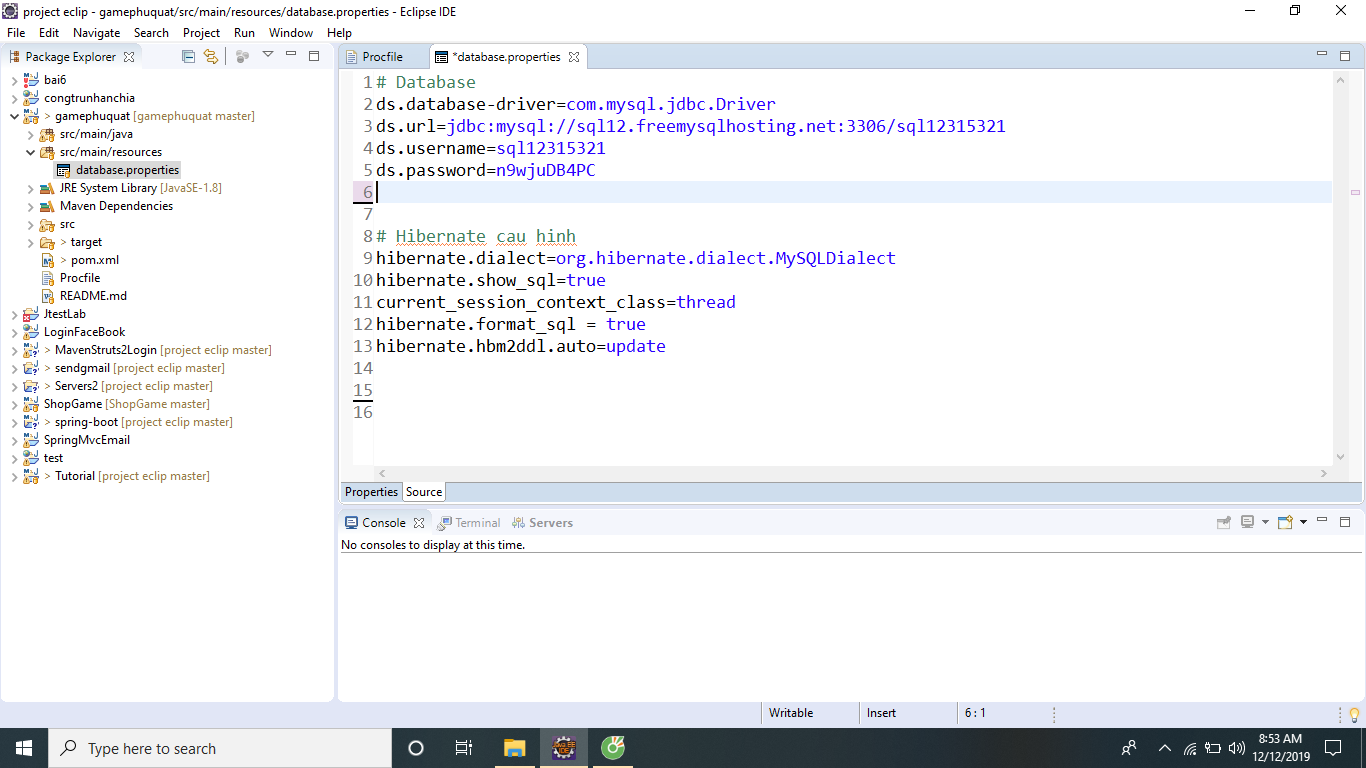
**Bước 4:** Thông tin cần điền:

* Máy chủ và tài khoản: có trong giao diện khi đăng nhập thành công.
* Mật khẩu: sẽ gửi vào email (cái này nó gửi về hơi lâu).

**Bước 5:** Sau khi đăng nhập thành công, vào phần import => chọn file database (.sql)



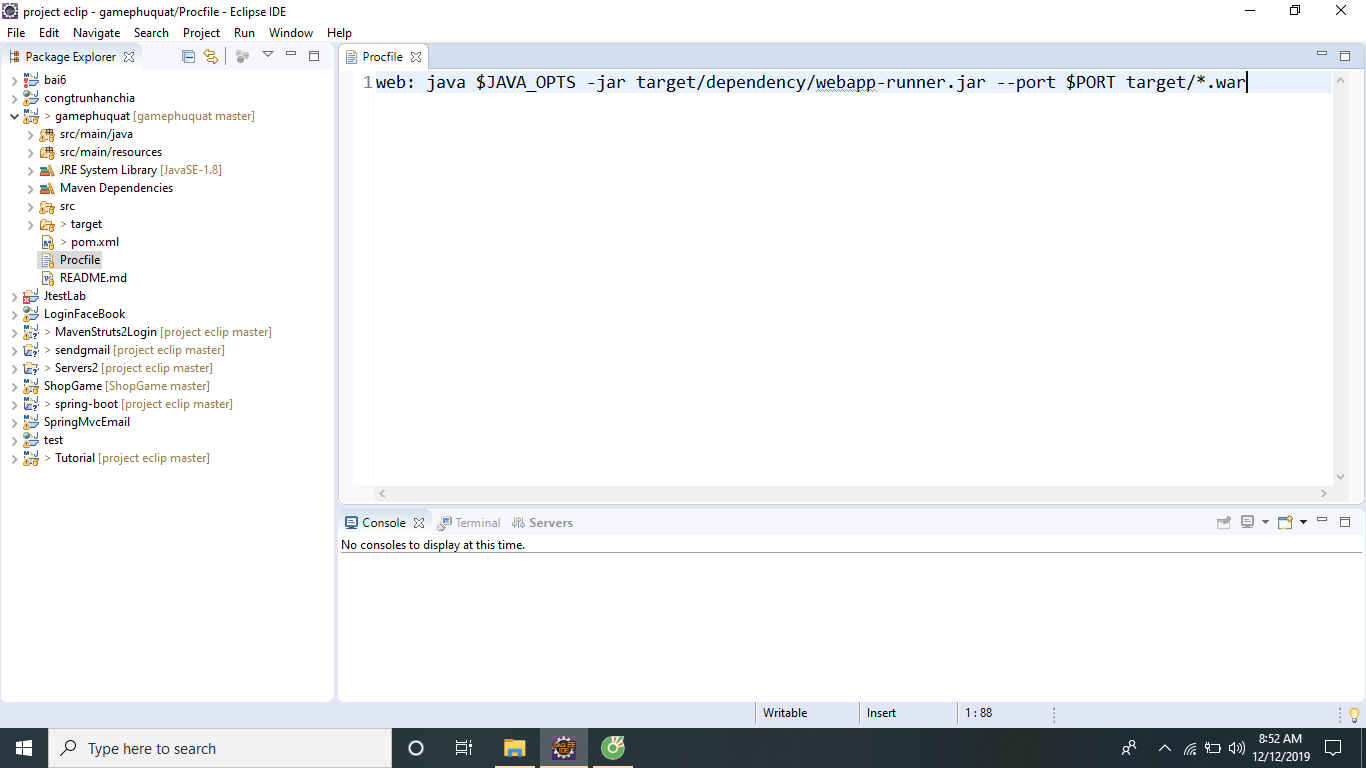
**Bước 6:** Sau khi hoàn tất tất cả thì việc còn lại là vào project cấu hình lại đường dẫn trong database (nhớ push lên github).

****

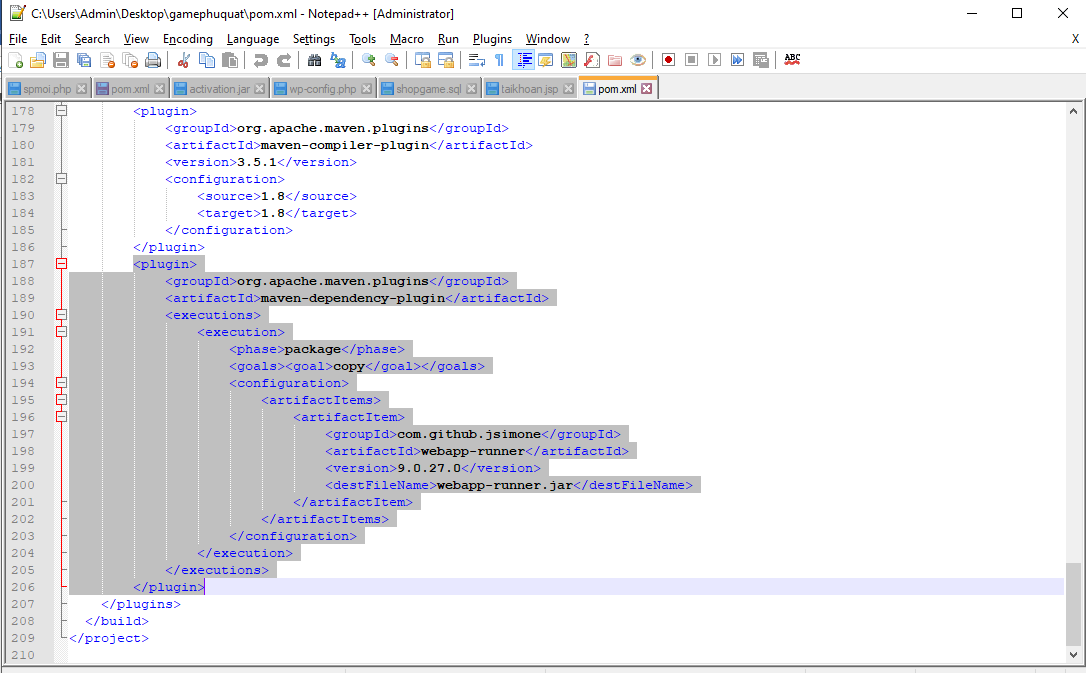
*1.2.2. Triển khai ứng dụng lên host*

**Bước 1:**Cần cấu hình một chút trong project:

* Thêm 1 procfile vào trong project với nội dung như sau:

****

* Thêm đoạn plugin này vào file pom.xml

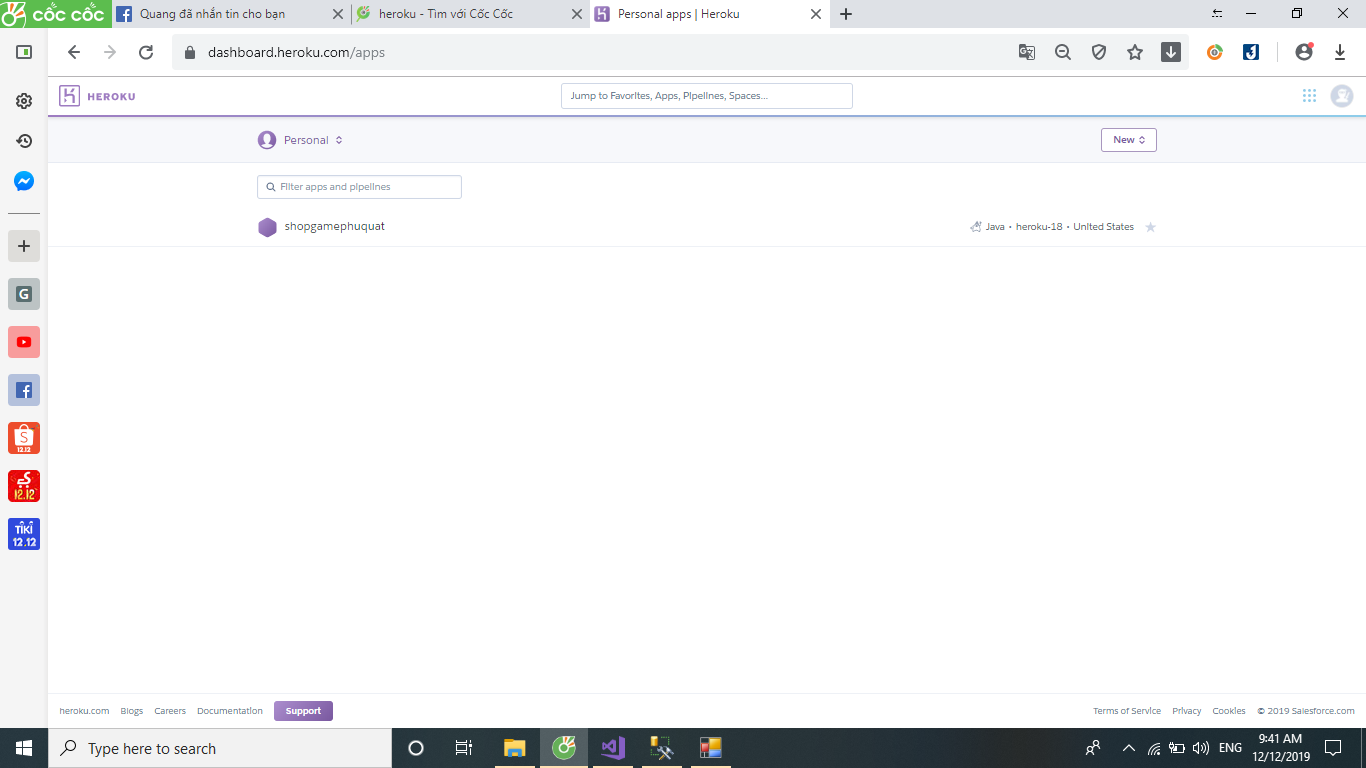


**Bước 2:**Truy cập vào đường link Heroku ở trên.

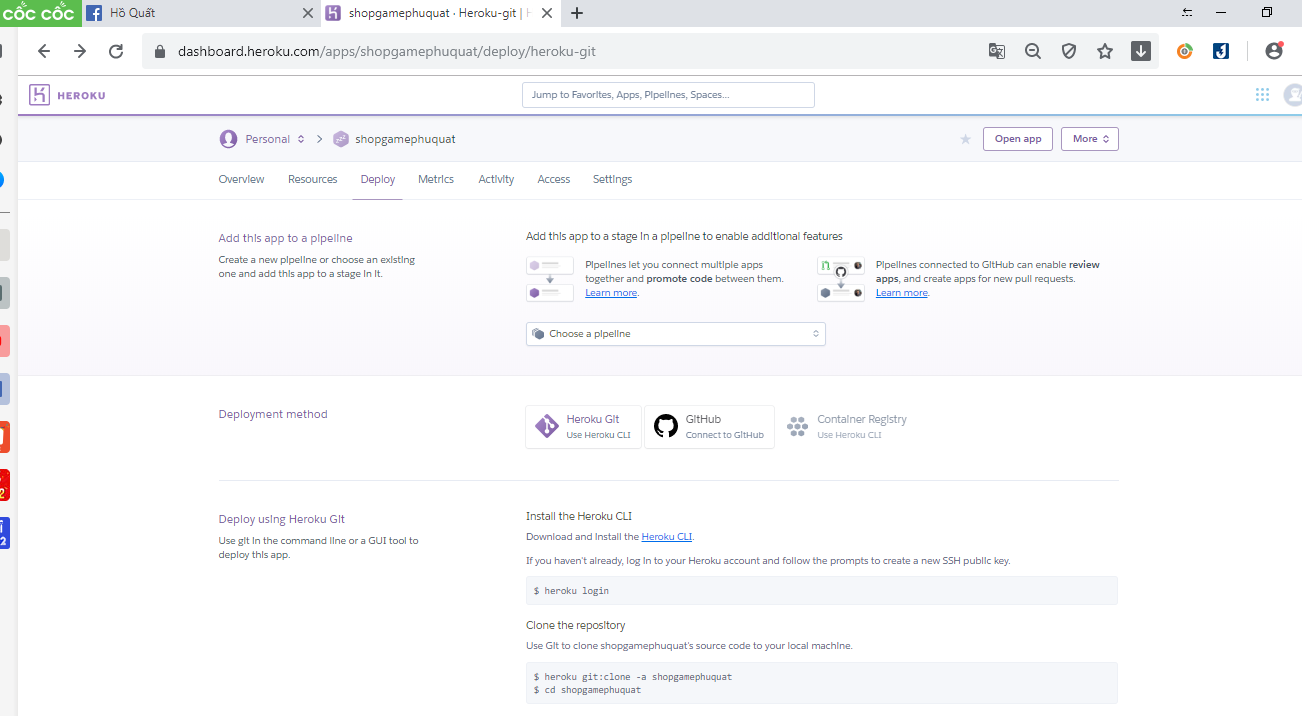
**Bước 3:**Đăng ký tài khoản và sau đó đăng nhập.

**Bước 4:**Khi đăng nhập thành công, giao diện trang chủ sẽ hiện ra.

**Bước 5:** Nhấn **New** và đặt tên cho website.

****

**Bước 6:** Nhấn vào tên website mới tạo ra để tiến hành deploy



**Bước 7:**Nhấn vào deloy phần Deployment methodta chọn **biểu tượng github** => chọn **connect to github**.

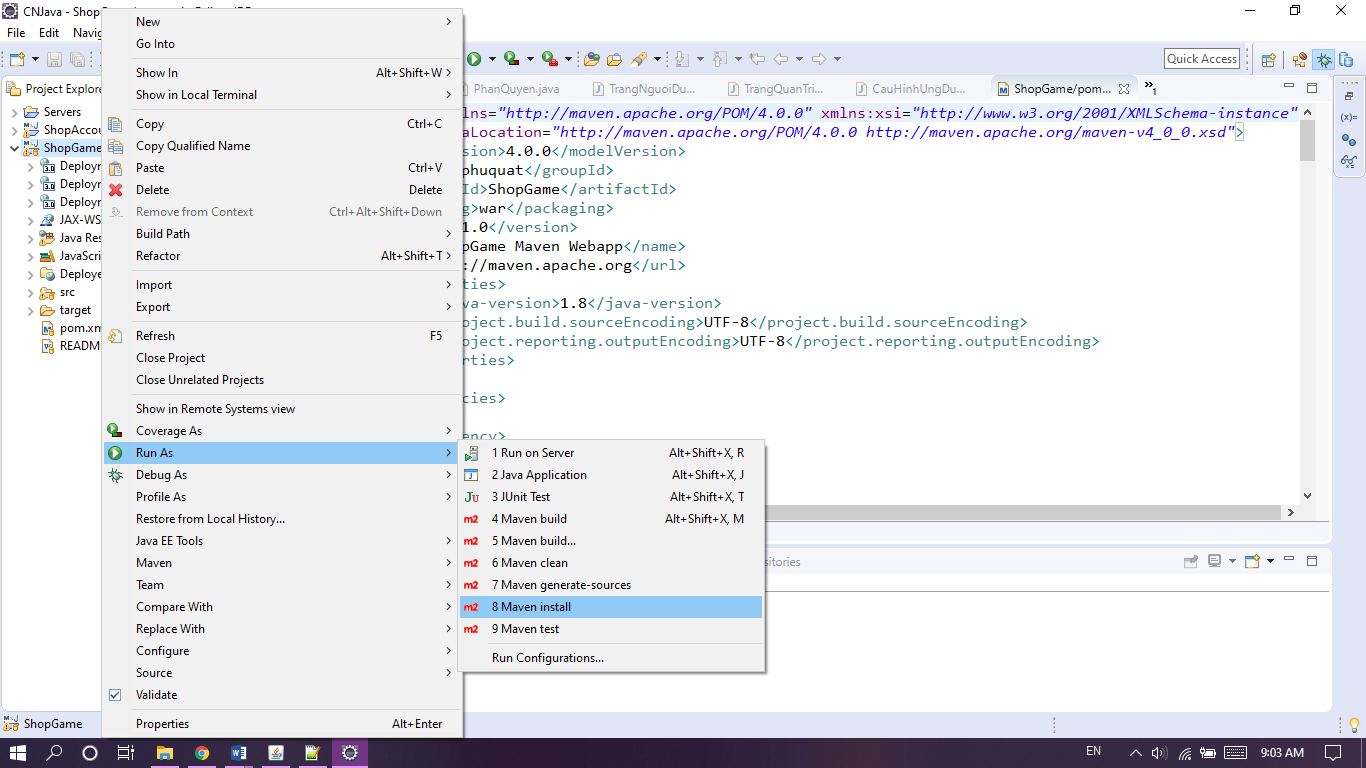
**Bước 8:**Sau khi đăng nhập github xong chọn project cần đưa lên host và nhấn **connect** => nhấn tiếp **Deloy Branch**

**Bước 9:**Sau khi hệ thống build xong nhấn vào **View** nếu hiện lên giao diện trang chủ thì đã thành công.

**2. Cấu hình và triển khai lên localhost**

***2.1. Chuẩn bị***

**Bước 1:**File .war của project: Chuột phải vào project => Run As => Maven install



**Bước 2:** Nếu build file .war thành công thì sẽ có thông báo ở màn hình console như sau:



Để triển khai được file .war lên tomcat thì phải có tài khoản đăng nhập vào Tomcat Web Application Manager.

**Bước 3:** Vào thư mục tomcat => thư mục conf => mở file “tomcat-users.xml” =>thêm đoạn code sau vào trong thẻ <tomcat-users>, với password và username tùy ý.



***2.2. Triển khai***

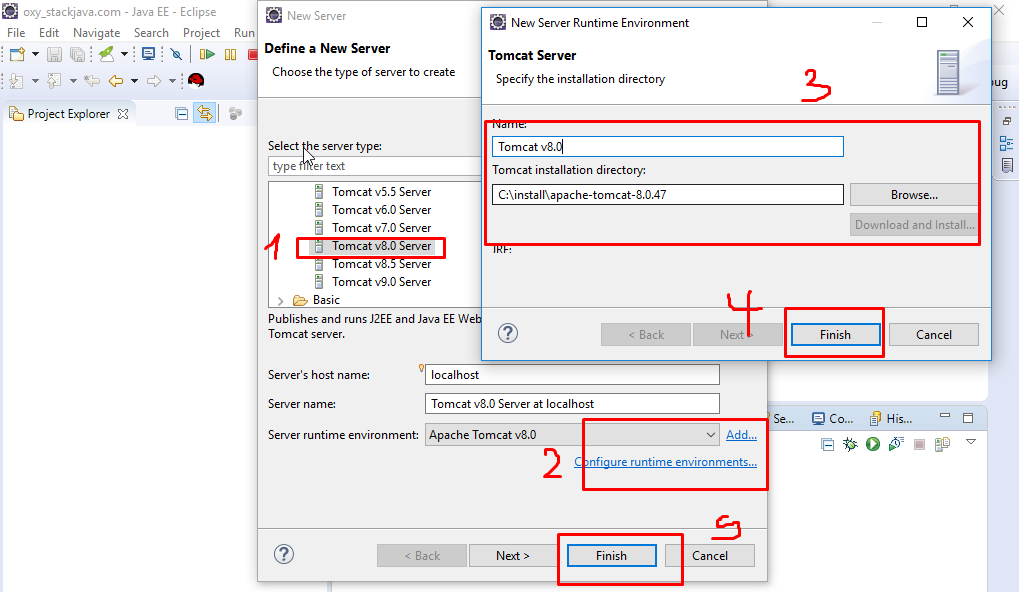
# **Deploy từ Eclipse**

Trước tiên bạn cài tomcat vào eclipse:

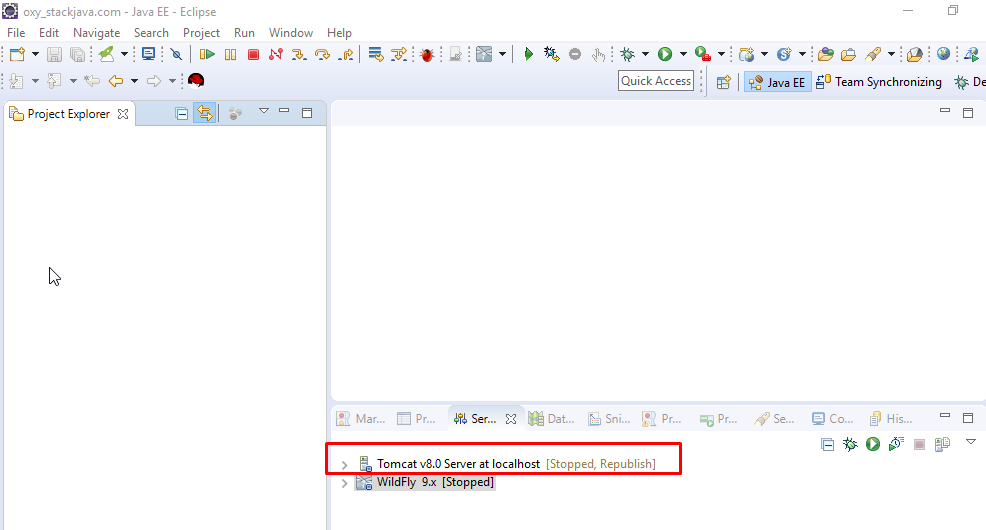
* Chọn tab Servers > click chuột phải vào vùng trống của tab Servers và chọn new > Server



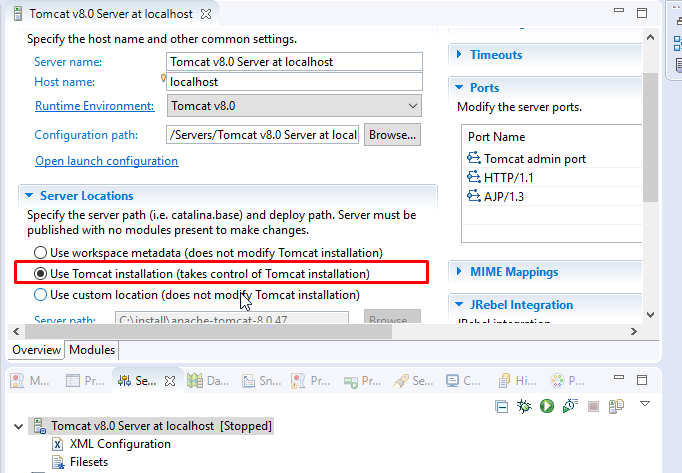
Chọn Apache > Tomcat v8.0 Server > click vào Add Configure runtime enviroments để trỏ tới folder tomcat.



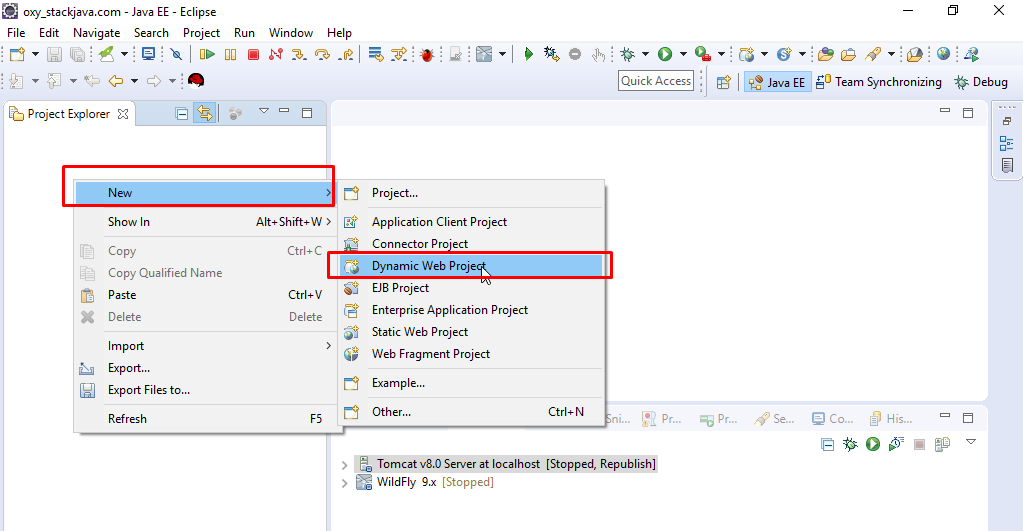
Kết quả sau khi cài thành công:

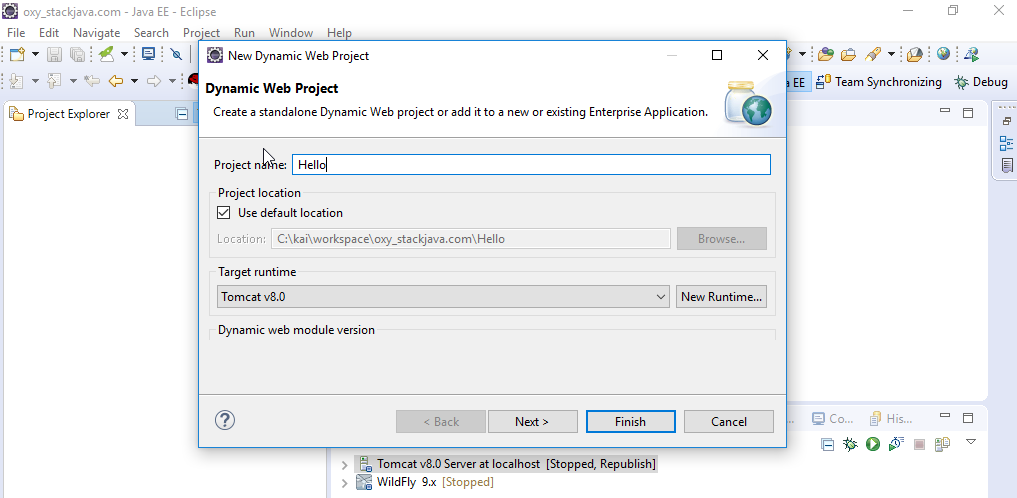


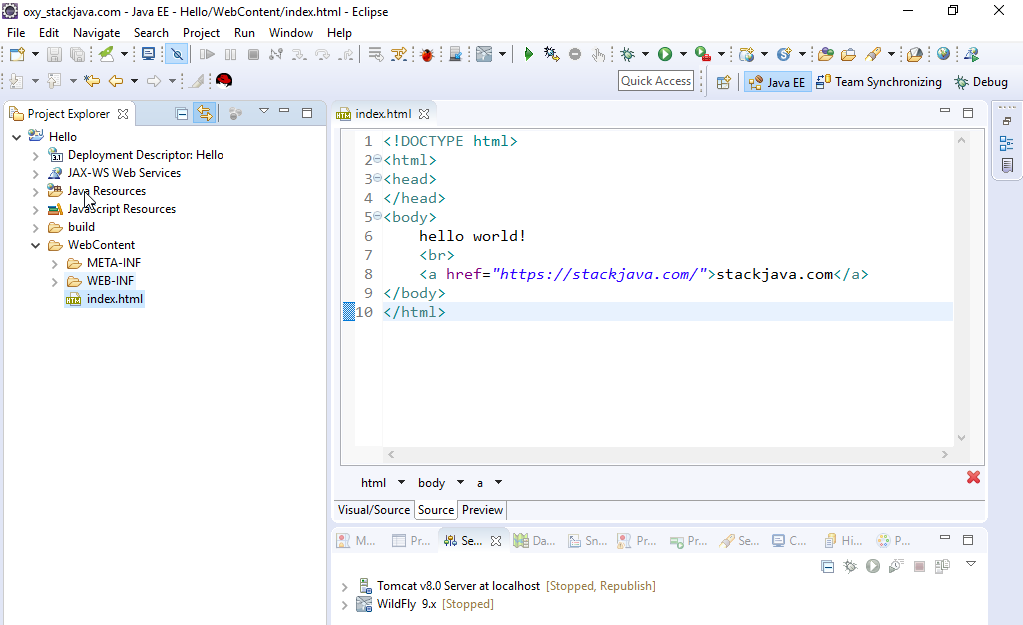
Trỏ cài đặt tomcat tới folder tomcat để có thể thực hiện truy cập trang quản lý của tomcat.



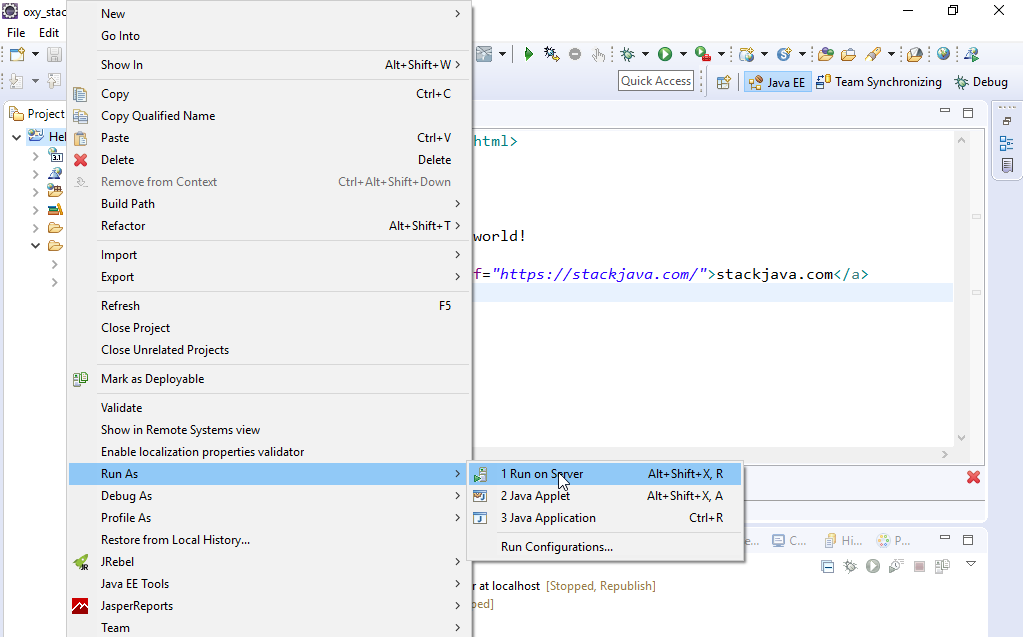
* Tạo 1 ứng dụng web đơn giản trên eclipse: click chuột phải vào vùng trống của Project Explorer > new > Dynamic Web Project



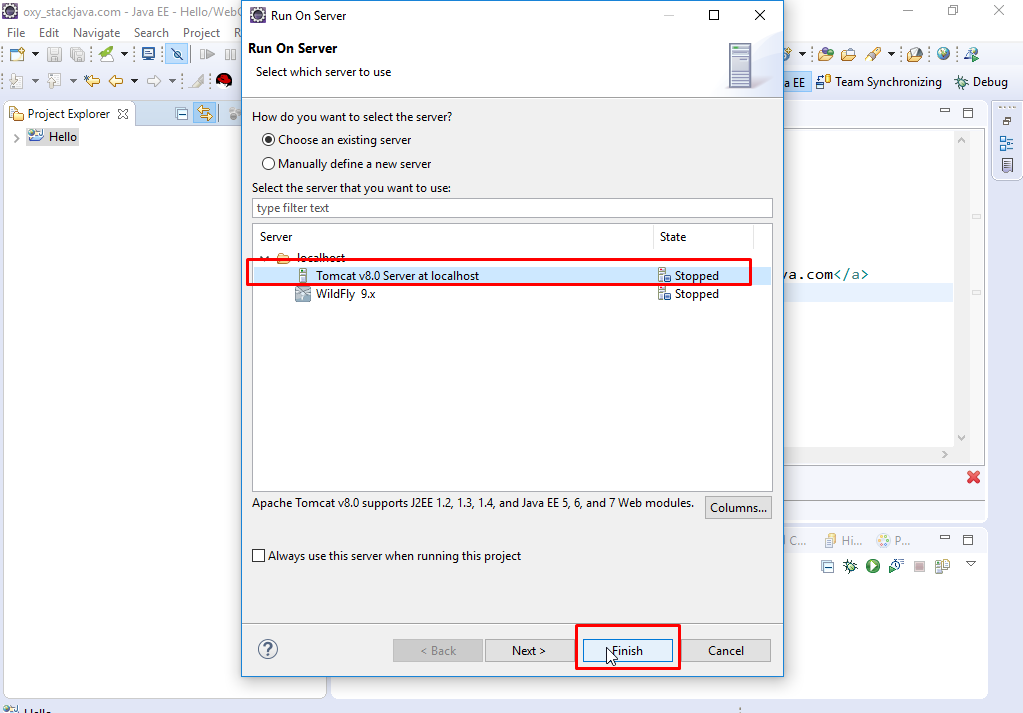
Nhập tên cho ứng dụng web

Tạo một file index.html (mặc định ban đầu chạy ứng dụng web nó sẽ chạy file index.html này, bạn có thể sửa file mặc định chạy ban đầu ở trong file WEB-INF/web.xml)

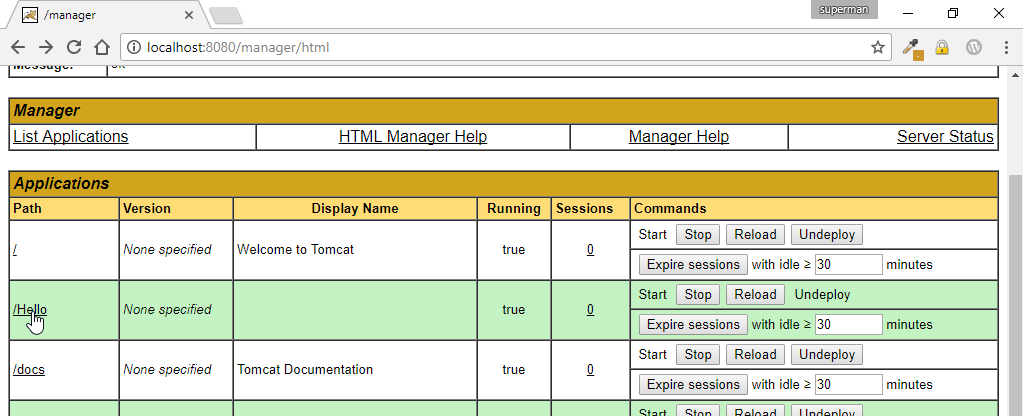
Chạy ứng dụng web vừa tạo trên tomcat: chuột phải vào project > Run As > Run on Server



Chọn server Tomcat và click Finish



Sau khi deploy bạn sẽ thấy ứng dụng Hello hiện lên trên trang manager của tomcat.



Kết quả:



**III. KẾT QUẢ ĐẠT ĐƯỢC**

**1. Kết quả đạt được**

Sau thời gian thực hiện đề tài, hệ thống cũng đã hoàn thành và đạt được một số kết quả sau:

* Hiểu rõ quy trình mua – bán xe thông qua hệ thống
* Xây dựng được hệ thống mua – bán xe đáp ứng 1 số nhu cầu nhất định của mọi người
* Tìm hiểu và nắm rõ về các công cụ để xây dựng hệ thống như: Eclipse, MySQL, Git.
* Học được cách thức xây dựng một website sử dụng framework Spring MVC và Hibernate kết hợp với một số công nghệ khác.
* Giao diện của website phù hợp với mọi người, thu hút và dễ sử dụng
* Hệ thống đã giúp khách hàng tiết kiệm thời gian, công sức để có được những chiếc xe ưng ý
* Thực hiện kiểm thử phần mềm ở mức độ lập trình.

1. **Hạn chế của đề tài**

Chưa thể cài đặt chức năng nạp thẻ trực tuyến cho khách hàng.

Chưa thể cập nhật nhiều loại thể thao điện tử.

Có thể hình ảnh đăng lên không chính xác 100% với đời thực.

**3. Hướng phát triển của đề tài**

Vì thời gian có hạn nên đề tài còn sai sót cũng như thiếu sót về nhiều mặc chức năng và phi chức năng, giao diện….

Trong khoản thời gian tiếp theo, nhóm chúng em sẽ cố gắng hoàn thiện hệ thống và được mọi người sử dụng rộng rãi. Đồng thời cung cấp các chuyên ngành khác để đảm bảo nhu cầu của khách hàng. Và trong tương lai gần nhóm sẽ hoàn thiện các chức năng :

* Thêm chức năng gửi SMS báo thông tin tài khoản đã mua cho khách hàng.
* Cài đặt chức năng nạp thẻ trực tuyến cho khách hàng.
* Thêm chức năng chat cho khách hàng và người quản trị.
* Tăng độ tin cậy cho khách hàng mua bằng cách xác minh rõ các đối tượng bán.

**CÔNG CỤ HỖ TRỢ**

* Eclipse IDE for Enterprise Java Developers – version: 2019-06 (4.12.0)
* MySQL Workbench – version: 8.0.18
* Tomcat Server – version: 8.5.45

**TÀI LIỆU THAM KHẢO**

* <https://o7planning.org/vi/11281/huong-dan-spring-mvc-security-voi-hibernate>
* <https://www.javatpoint.com/spring-mvc-crud-example>
* <https://giasutinhoc.vn/website/spring-framework/huong-dan-xu-ly-database-trong-spring-mvc-5-voi-hibernate/>
* <https://stackjava.com/spring/code-vi-du-spring-mvc-spring-jdbc.html>
* <https://stackoverflow.com/>
* <https://www.w3schools.com/>
* [https://getbootstrap.com/](https://getbootstrap.com/docs/4.0/getting-started/introduction/)