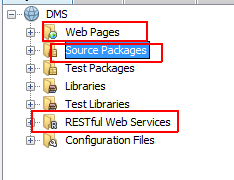
# Cấu trúc Project

Cấu trúc Project gồm



Trong đó Webpage chứa các trang jsp, các tài nguyên sử dụng trong trang jsp, các style, script…

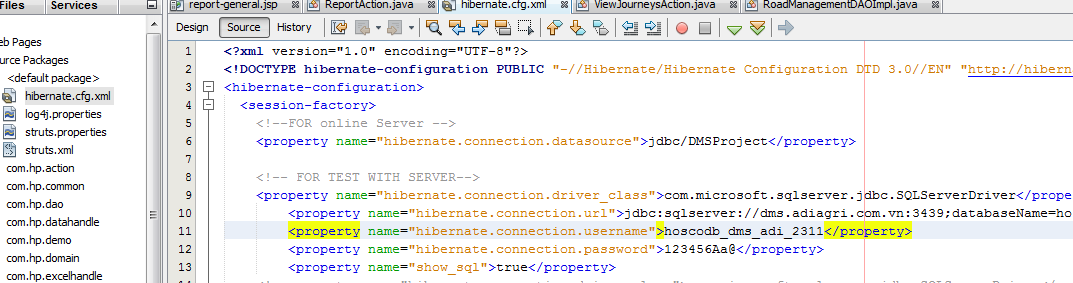
Trong Source Pakages chứa các lớp Java, các hàm xây dựng để xử lí dữ liệu và đẩy cho trang jsp

Trong RESTful Web Services chứa danh sách webservice ta cung cấp cho Client sử dụng

### Project sử dụng thêm thư viện hibernate và struts2

#### Hibernate

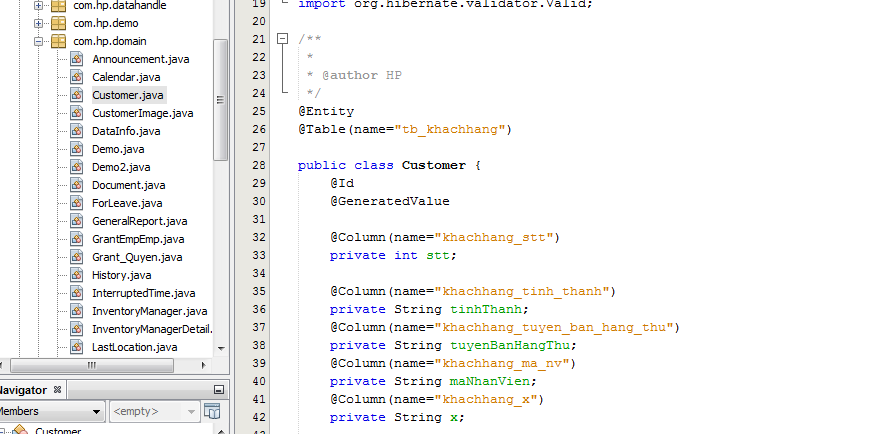
Thông tin cấu hình Database trong file hibernate.cfg



Trong file này còn chứa danh sách các lớp ánh xạ từ CSDL vào class của Java

Ta có thể đổi tên các trường trong CSDL vào class trong java cho phù hợp sử dụng.

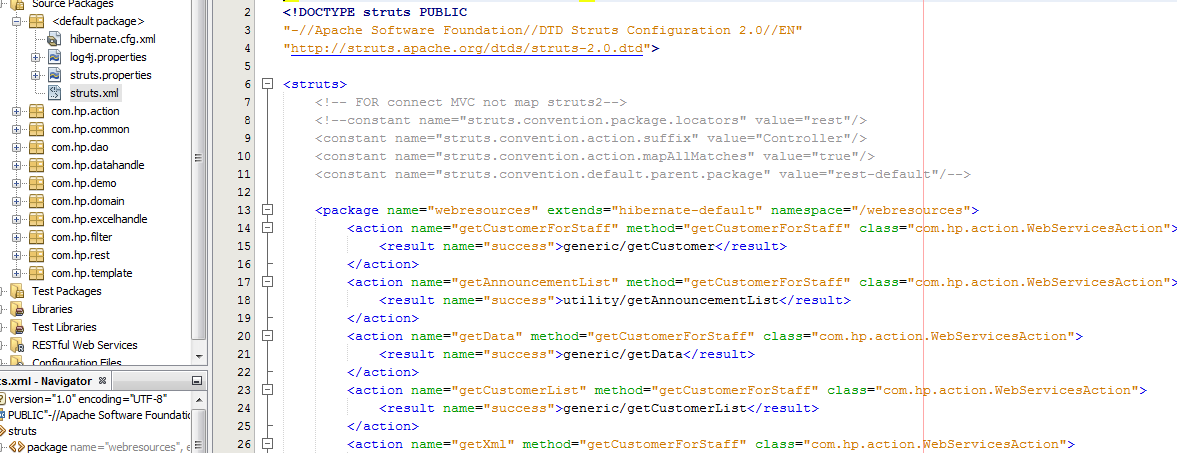
Các Class được ánh xạ được tạo trong group com.hp.domain



## Thư viện struts2

Struts2 ánh xạ các action tới các hàm trong lớp Java ta xây dựng. Trong các trang jsp, mỗi lần gọi action tương ứng tới một lời gọi hàm trong Java, Java xử lí xong lại đẩy ra một kết quả, struts2 xử lí kết quả và đẩy tới trang jsp tương ứng kết quả trả về.

Để truyền dữ liệu từ lớp Java tới jsp ta có thể sử dụng biến session hoặc truyền qua paramerter trên URL 🡪cần mã hóa và giải mã

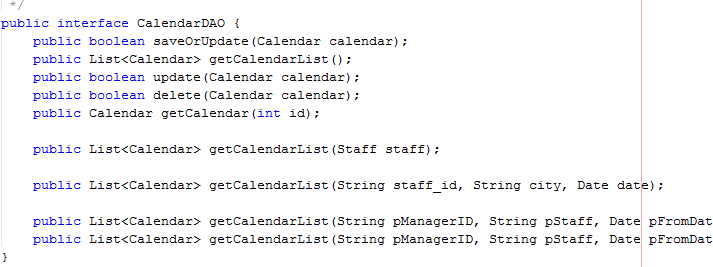


#### Source pakages bao gồm

#### 

+ hp.action chứa các lớp điều khiển, struts2 thường ánh xạ tên mỗi action tới một hàm trong các lớp này, trong các lớp này thường sử dụng các lớp trong hp.dao để hỗ trợ xử lí

+ hp.dao là các lớp hỗ trợ xử lí cho mỗi đối tượng trong lớp domain như thêm, sửa, xóa, lấy danh sách,…



+ hp.rest chứa các service cung cấp cho client

# Quản lí hành trình (chỉ xem được trong một ngày)

Ý tưởng:

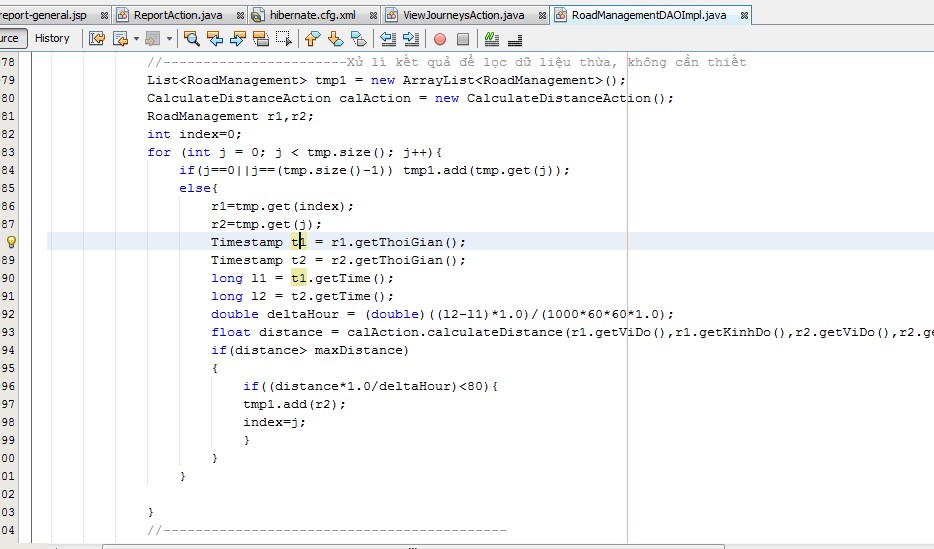
Sau khi lấy danh sách hành trình đi trong ngày, ta sẽ lọc dữ liệu đó theo một số quy tắc trước khi gửi lên jsp để vẽ maps.

- Bắt đầu từ điểm xuất phát đầu tiên, nếu khoảng cách điểm tiếp theo nhỏ hơn 300m ( hoặc cho phép người xem cấu hình trên trang jsp với tham số maxDistance thì mới vẽ điểm tiếp theo

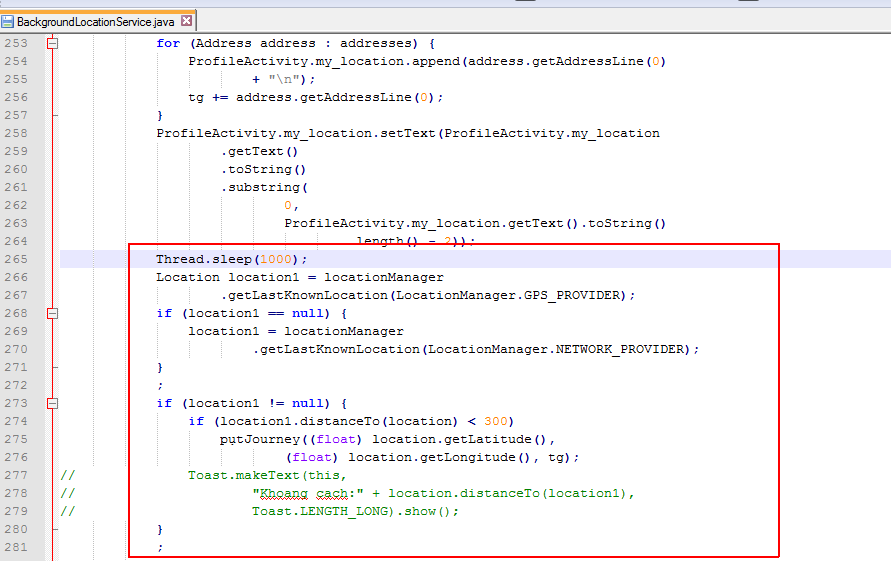
- Vì mỗi tọa độ gửi lên có tính cả thời gian nên có thể tính được khoảng thời gian di chuyển giữa 2 điểm. Để loại bỏ trường hợp điểm kì dị do client lấy sai vị trí, ta lọc thêm điều kiện “*nếu vận tốc lớn hơn x=80km/h thì điểm tiếp theo đó coi như là kì dị, không được vẽ*”.

Các xử lí này được xử lí trước ở trong lớp Java RoadmanagementDAO trước khi đẩy ra ngoài trang jsp để vẽ. Tất cả các hàm liên quan tới việc lấy danh sách hành trình đều phải xử lí lọc dữ liệu như vậy trước khi tính toán, ví dụ như hàm tính đường đi trong ngày, …

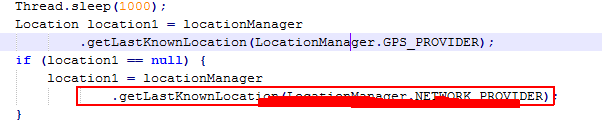
Đoạn code lọc dữ liệu như dưới:



- Trên Client : có hiện tượng lấy sai, thường sai tập trung vào một số điểm (gọi là điểm kì dị), có thể xử lí bằng cách lấy một lần 2 vị trí cách nhau 1s, nếu hai điểm không quá xa nhau thì gửi một vị trí lên, nếu cách nhau quá xa thì không gửi vị trí lên.



Hoặc xử lí bằng cách chỉ xử dụng GPS của máy để lấy vị trí, trường hợp này một số máy lấy chậm, không dung được.

Một số còn có hiện tượng không lấy được vị trí sau một thời gian không đăng nhập phần mềm

# Báo cáo tổng hợp

- Lấy dữ liệu từ các báo cáo về đường đi, báo cáo chấm công, báo cáo kế hoạch công tác

- Tổng hợp dữ liệu lấy được ghi vào lớp GeneralReport ứng với mỗi (nhân viên + ngày)

- Từ chuỗi json trong trường *province* xử lí xem danh sách khách hàng dự kiến, danh sách đã đi dựa vào trường thời gian trong đó, tìm ra danh sách khách hàng sai tuyến …

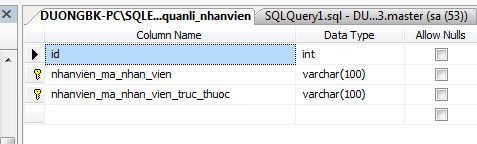
- Cần bổ sung chức năng xem trên bản đồ đường đi thực tế và đường đi dự kiến

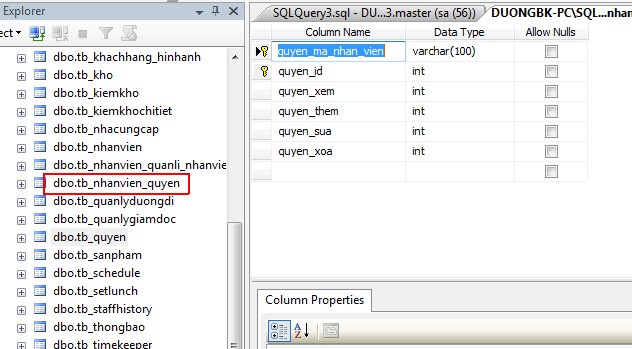
(🡪 open new window, truyền tham số hành trình dự kiến và thực tế vào và vẽ lên

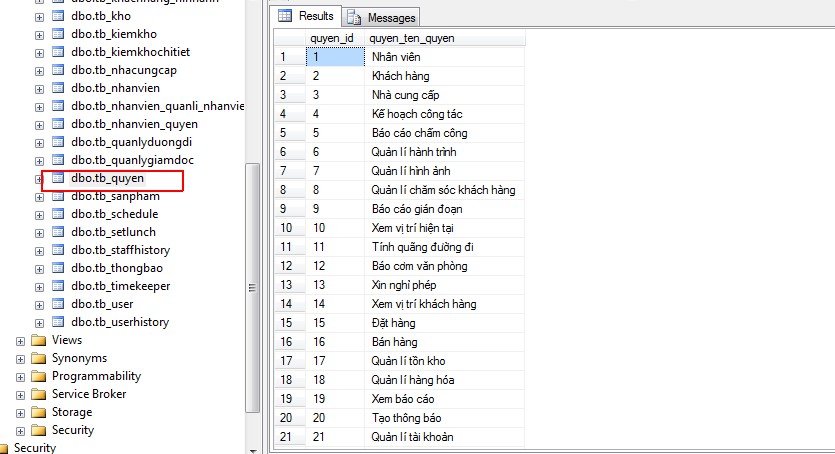
🡪 cần giải quyết vấn đề vị trí khách hàng trước)

# Phân quyền

Phân quyền cho tài khoản dựa trên 2 bảng như sau:







Mỗi nhân viên sẽ có quyền quản lý một list nhân viên nào đó, mỗi nhân viên cũng được cấp một list quyền, mỗi quyền lại có các cấp khác nhau như xem, thêm, sửa, xóa

Nếu muốn thêm quyền khác, có thể thêm vào bảng tb\_quyen, các quyền xem, thêm, sửa, xóa sẽ được gán mặc định cho nhân viên là 0

Khi đăng nhập bởi admin, admin sẽ được gán cho full quyền quản lí nhân viên, full quyền chức năng