# LAB 3: HTTP REQUEST VÀ REST API (OPENWEATHERMAP)

THÔNG TIN SINH VIÊN  
Họ và tên: Trần Trung Đức  
MSSV: SE183038

MỤC TIÊU HỌC TẬP  
Sau khi hoàn thành bài lab, các bạn có khả năng:  
  
- Hiểu và sử dụng HTTP Request để gọi API  
- Làm việc với REST API của OpenWeatherMap  
- Xử lý dữ liệu JSON trong R  
- Tạo visualizations từ dữ liệu weather API  
- Áp dụng kiến thức vào việc phân tích dữ liệu thời tiết

CHUẨN BỊ  
- Cài đặt gói và load Thư viện:  
install.packages(c("httr", "httr2", "jsonlite", "dplyr", "lubridate", "ggplot2"))  
- Tài khoản API:  
 + Đăng ký tài khoản miễn phí tại OpenWeatherMap  
 + Lấy API key từ tài khoản của bạn

## PHẦN I: THIẾT LẬP VÀ GỌI API CƠ BẢN (25 điểm)

Yêu cầu thực hiện:

Cài đặt thông tin API (5 điểm)  
- Khai báo API key của bạn  
- Đặt tên thành phố cần tra cứu (ví dụ: "Ho Chi Minh City")  
- Thiết lập URL endpoint cho Current Weather Data

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Tạo HTTP GET Request (10 điểm)  
- Sử dụng hàm GET() từ package httr  
- Thiết lập các query parameters:  
 q: tên thành phố  
 appid: API key  
 units: "metric" (để có đơn vị °C)  
 lang: "vi" (ngôn ngữ tiếng Việt)  
- Kiểm tra status code của response  
Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

Xử lý response (10 điểm)  
- Parse JSON response thành R object  
- Tạo data frame chứa thông tin thời tiết với các cột:  
 city, country, description, temp\_c, feels\_like\_c, humidity, wind\_speed, dt

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Câu 1: (3 điểm)

- Status code 200: Request thành công  
- 404: Không tìm thấy tài nguyên  
- 401: Không có quyền truy cập (Unauthorized)  
- 500: Lỗi server nội bộ

Câu 2: (4 điểm)

- REST API: Giao diện lập trình cho phép giao tiếp client-server qua HTTP  
- GET: Lấy dữ liệu  
- POST: Gửi dữ liệu mới  
- PUT: Cập nhật dữ liệu  
- DELETE: Xóa dữ liệu

Câu 3: (3 điểm)

- URLencode() dùng để mã hóa ký tự đặc biệt (dấu, khoảng trắng)  
- Nếu không encode, API có thể không hiểu đúng tên thành phố  
- Đảm bảo request chính xác

## PHẦN II: DỮ LIỆU DỰ BÁO 5 NGÀY (30 điểm)

Yêu cầu thực hiện:

Tạo function lấy dữ liệu forecast (15 điểm)  
  
- Viết function get\_forecast\_5day(city, api\_key)  
- Sử dụng endpoint forecast API của OpenWeatherMap  
- Function trả về data frame với các cột:  
 datetime, temp, temp\_min, temp\_max, humidity, wind\_kph, rain\_mm, weather

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Nội dung do AI tạo ra có thể không chính xác.  
  
Xử lý dữ liệu JSON (15 điểm)  
- Parse dữ liệu forecast (có cấu trúc nested)  
- Chuyển đổi timestamp thành datetime object  
- Xử lý trường hợp dữ liệu mưa có thể null  
- Chuyển đổi đơn vị gió từ m/s sang km/h

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

Câu 4: (4 điểm)

- JSON: Định dạng trao đổi dữ liệu dạng text (key-value)  
- XML: Dùng thẻ mở/đóng, phức tạp hơn  
- JSON ngắn gọn, dễ đọc, phổ biến trong REST API

Câu 5: (4 điểm)

- Nested JSON: Dữ liệu lồng nhau nhiều cấp  
- Truy cập trong R bằng $ hoặc [[]]  
- Có thể dùng jsonlite::fromJSON() để parse

Câu 6: (2 điểm)

- Cần chuyển Unix timestamp sang datetime để dễ đọc  
- Epoch time: số giây từ 01/01/1970 đến hiện tại

## PHẦN III: THU THẬP DỮ LIỆU NHIỀU THÀNH PHỐ (20 điểm)

Yêu cầu thực hiện:

- Lấy dữ liệu cho danh sách thành phố (15 điểm)  
  
- Sử dụng danh sách 10 thành phố Việt Nam theo danh sách trong có (tham khảo cách tìm địa danh trên OpenWeather.)  
- Sử dụng function đã tạo cho tất cả các thành phố  
- Gộp tất cả dữ liệu thành một data frame duy nhất  
- Xử lý lỗi khi một số thành phố không trả về dữ liệu

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu

Nội dung do AI tạo ra có thể không chính xác.

Kiểm tra và làm sạch dữ liệu (5 điểm)  
  
- Kiểm tra missing values  
- Xem tổng quan về dataset (số dòng, cột, kiểu dữ liệu)  
- Hiển thị 10 dòng đầu tiên

Câu 7: (3 điểm)

- Rate limiting: Giới hạn số lần gọi API trong một khoảng thời gian  
- Tránh server quá tải  
- Giải pháp: chờ đợi, retry sau, hoặc dùng gói trả phí

Câu 8: (4 điểm)

- API key: Chuỗi đơn giản để xác thực  
- OAuth: Cơ chế bảo mật, cấp quyền qua token  
- Bearer token: Token trong header, an toàn hơn API key

Câu 9: (3 điểm)

- lapply() chạy hàm cho từng phần tử, ngắn gọn hơn for  
- Functional programming: ít lỗi, code gọn, dễ bảo trì

## PHẦN IV: TRỰC QUAN HÓA DỮ LIỆU (25 điểm)

Yêu cầu thực hiện:

Biểu đồ Boxplot (8 điểm)  
  
- Tạo boxplot so sánh phân bố lượng mưa giữa các thành phố  
- Sử dụng màu sắc phù hợp và title rõ ràng  
Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.  
Biểu đồ Scatter Plot (8 điểm)  
  
- Tạo scatter plot (jitter) hiển thị lượng mưa theo từng thành phố  
- Sử dụng alpha và size phù hợp để tránh overlap  
- Thêm theme và formatting

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.  
  
Biểu đồ Time Series (9 điểm)  
  
- Lọc dữ liệu cho một thành phố cụ thể (TP.HCM)  
- Tạo time series plot cho lượng mưa theo thời gian  
- Kết hợp point và line plot  
- Formatting trục thời gian

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Câu 10: (3 điểm)

- Boxplot: dùng khi so sánh phân bố, phát hiện outlier  
- Thành phần: median, quartiles, whiskers, outliers

Câu 11: (4 điểm)

- Overplotting: quá nhiều điểm chồng lên nhau  
- Giải pháp:  
 + Dùng jitter  
 + Giảm alpha (độ trong suốt)  
 + Dùng hexbin hoặc binning

Câu 12: (3 điểm)

- Standardize giúp so sánh dữ liệu công bằng  
- Nếu đơn vị khác nhau thì kết quả sai lệch  
- Đảm bảo tính chính xác khi phân tích