



Native App

- Native App là các ứng dụng viết **cho từng nền tảng** iOS, Android, Windows bằng ngôn ngữ tương ứng trên mỗi nền tảng.
 - Android có thể viết bằng Java hoặc **Kotlin**.
 - iOS có thể viết bằng **Swift**, trước đây là Objective C.



Kotlin



Swift



Native App

- Native App chỉ **vận hành** trên một nền tảng cụ thể giúp **hiệu năng hoạt động** ứng dụng tốt nhất có thể mà không cần engine bên thứ ba.
- Native App có dữ liệu an toàn hơn, ứng dụng dễ dàng **tương tác** với **phần cứng** của thiết bị, **ít tốn bộ nhớ** do bản build có dung lượng nhẹ.
- Native App **tốn nhiều thời gian và chi phí** để phát triển.

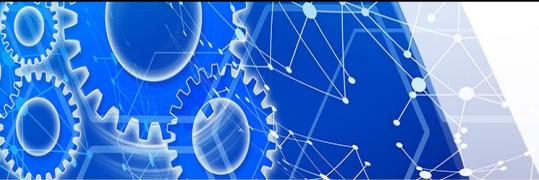


Hybrid App

- Hybrid App là các ứng dụng được phát triển trên các **nền tảng công nghệ phát triển Web** như HTML, CSS, Javascript.
- Hybrid có **build thành ứng dụng chạy trên các nền tảng thiết bị di động** khác nhau như Android, iOS.

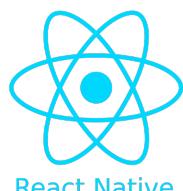
Dương Hữu Thành

5



Hybrid App

- Một số công nghệ phát triển Hybrid App
 - **React Native**: là công cụ mã nguồn mở được phát triển bởi Facebook.
 - **Ionic**: tạo các ứng dụng mobile với các công nghệ Web như HTML, CSS, Javascript.
 - **Xamarin**: được phát triển bởi Microsoft.



Dương Hữu Thành

6



Hybrid App

- Hybrid App hoạt động được trên **đa nền tảng** nên giúp **tiết kiệm nhiều chi phí** phát triển trên từng nền tảng riêng như Native App. Từ đó, sản phẩm phát triển nhanh hơn và dễ bảo trì hơn.
- Hybrid App **yêu cầu phải có kết nối internet** để hoạt động, **tốn dung lượng** nhớ do bản build thành ứng dụng khá lớn.

Dương Hữu Thành

7



Web Services

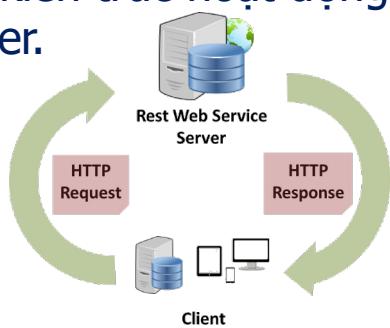
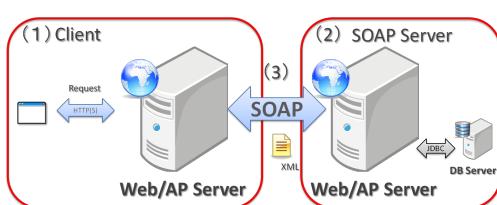
- Web services là tập các giao thức giúp **trao đổi dữ liệu** giữa các ứng dụng hoặc giữa các hệ thống.
- Web services hoạt động độc lập **không phụ thuộc ngôn ngữ hay nền tảng nào**. Các ứng dụng viết bằng các ngôn ngữ lập trình khác nhau, trên các nền tảng khác nhau có thể sử dụng Web services thông qua mạng internet.

Dương Hữu Thành

8

Web Services

- Có hai loại Web Service chính
 - **SOAP** (Simple Object Access Protocol) Web Services là một giao thức để gửi dữ liệu XML (gọi là SOAP Messages) giữa các ứng dụng.
 - **RESTful Web Services** là kiến trúc hoạt động theo kiến trúc client-server.



Dương Hữu Thành

9

REST

- REST (Representational State Transfer) là một dạng một **kiến trúc** phần mềm dựa trên giao thức HTTP cung cấp các API nghiệp vụ cho nhiều loại ứng dụng khác nhau.
- REST hỗ trợ **nhiều định dạng dữ liệu** khác nhau như JSON, HTML, XML.

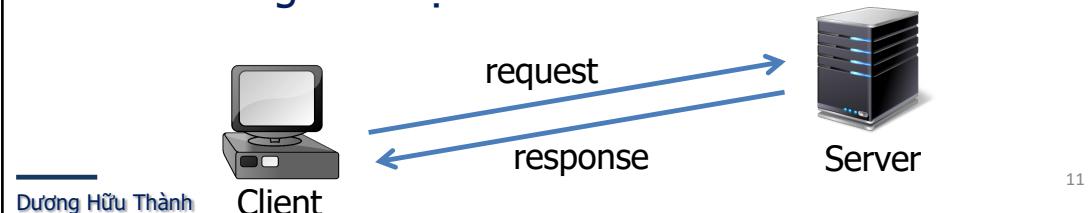
Dương Hữu Thành

10

10

Các đặc tính của REST

- **Client-server:** REST làm việc theo kiến trúc client-server.
 - Client gửi yêu cầu (request) lên server để yêu cầu lấy thông tin hoặc xử lý nào đó.
 - Server xử lý và phản hồi (response) từ yêu cầu của phía Client.
 - Tương tác một chiều từ client đến server



11

Các đặc tính của REST

- **Stateless:** server không lưu trữ thông tin mỗi request từ client, client phải gửi đầy đủ thông tin được yêu cầu bởi server trong mỗi request.
- **Cacheable:** đối với các request dùng thường xuyên, ít thay đổi, client có thể lưu response trong bộ nhớ đệm (cache), điều này có thể giúp cải thiện hiệu năng của hệ thống.

Dương Hữu Thành

12

12



Các đặc tính của REST

- **Layered System:** một vài layer phụ có thể được thêm vào giữa client và server, chẳng hạn middleware layer ở giữa thực hiện một số xử lý từ phía client trước khi đưa request đến server thật sự, chú ý là những layer như thế này phải trong suốt (transparent) giữa client và server.

Dương Hữu Thành

13

13



Các đặc tính của REST

- **Uniformed Interface:** REST cung cấp giao diện thông nhất cho phát triển các ứng dụng độc lập phía client.
- **Code on demand (tùy chọn):** cho phép code hoặc applet được truyền qua API để sử dụng trong ứng dụng.

Dương Hữu Thành

14

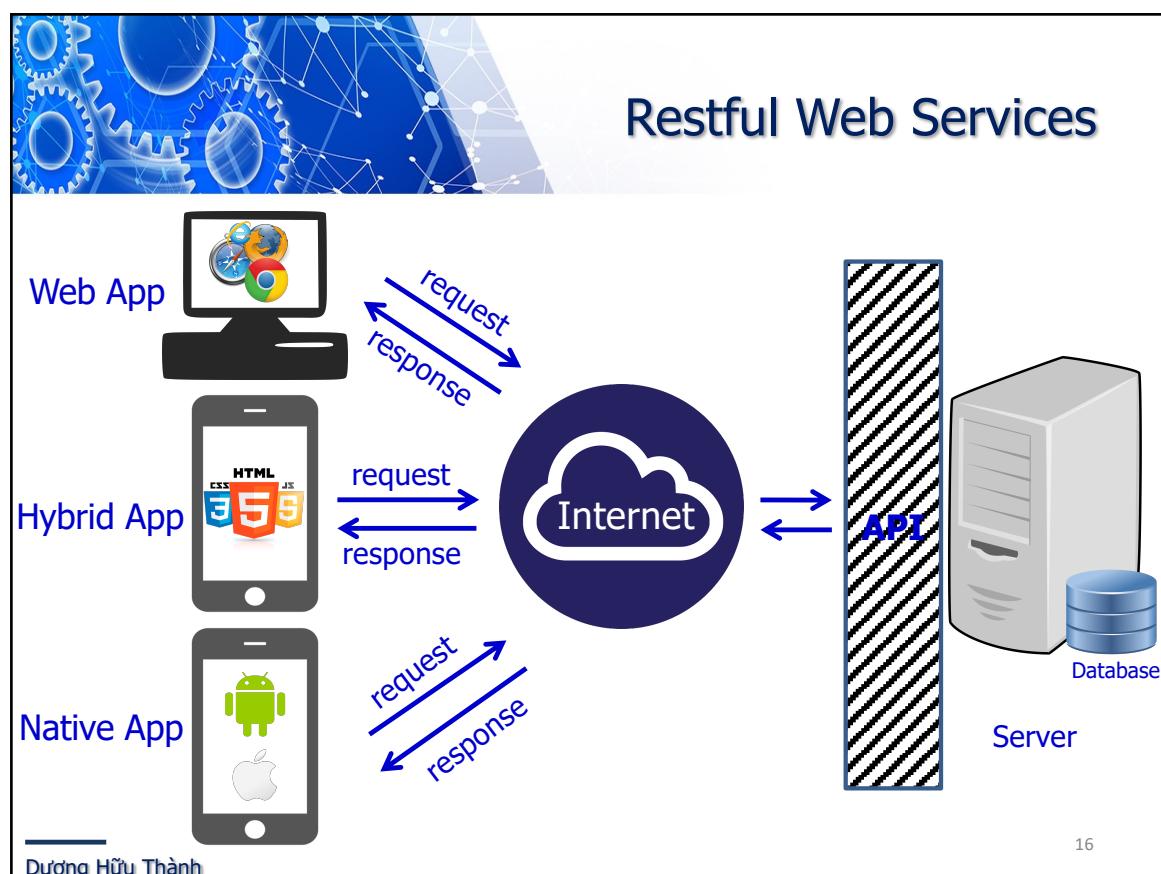
14

Restful Web Services

- Các Web service tuân thủ theo dạng kiến trúc của REST gọi là Restful Web services, nó cung cấp các API cho các ứng dụng client.
- Để hiện thực Restful API cần thiết kế các thông tin sau:
 - Resource
 - Request (Verb, Header, Body)
 - Response (Body, Status code)

Dương Hữu Thành

15



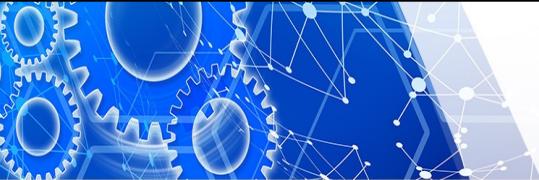
Dương Hữu Thành

16



Restful Web Services

- API (Application Programming Interface) cho phép **giao tiếp** giữa hai hệ thống phần mềm tách biệt.
- Trong phát triển phần mềm API có vai trò **trung gian** giữa tầng giao diện (Presentation Layer) và tầng tương tác cơ sở dữ liệu (Database Layer), nó cho phép giao tiếp và trao đổi dữ liệu từ hệ thống này đến hệ thống khác.



Hiện thực Restful API

- **Resources**: danh sách các endpoint để yêu cầu server tương tác với một tài nguyên nào đó.
- **Ví dụ**
 - <http://localhost/courses>: cung cấp danh sách khoá học
 - <http://localhost/courses/1>: cung cấp chi tiết thông tin một khoá học.



Hiện thực Restful API

- **Request Verbs:** mô tả các thức muốn làm việc với các tài nguyên
 - **GET:** dùng lấy tài nguyên hoặc dữ liệu gửi lên không tính chất nhạy cảm (sensitive).
 - **POST:** dùng tạo mới tài nguyên hoặc dữ liệu quan trọng, có tính nhạy cảm cần gửi lên server.
 - **PUT:** dùng cập nhật tài nguyên.
 - **DELETE:** dùng xoá tài nguyên.
 - **PATCH:** dùng cập nhật một phần tài nguyên.



Hiện thực Restful API

- **Request Headers:** chứa những chỉ thị gửi kèm với request như loại dữ liệu response hoặc các thông tin chứng thực.
- **Request Body:** chứa dữ liệu request gửi lên, thường nó được gửi cho các request sử dụng phương thức POST.
- **Response Body:** dữ liệu phản hồi về cho client.



Hiện thực Restful API

- Response status code
 - 200 OK: request thực hiện hành công
 - 201 Created: tài nguyên tạo thành công
 - 204 No Content: tài nguyên xoá thành công
 - 400 Bad Request: request không hợp lệ
 - 401 Unauthorized: request chưa được chứng thực.
 - 403 Forbidden: request không được quyền truy cập.
 - 404 Not Found: request không được tìm thấy.
 - 500 Internal Server Error: Lỗi server

Dương Hữu Thành

21

21



Security

- Cross site scripting (XSS) là các thức tấn công một người dùng **gắn script phía client** vào các trình duyệt của các người dùng khác.
- Tấn công XSS xảy ra thường do
 - Lưu các mã độc (malicious script) trong CSDL được truy vấn bởi người dùng khác.
 - Người dùng click trên một liên kết làm thực thi một đoạn javascript trên trình duyệt của người dùng.
 - Bắt nguồn từ một số nguồn dữ liệu không đáng tin cậy của cookies hay web services.

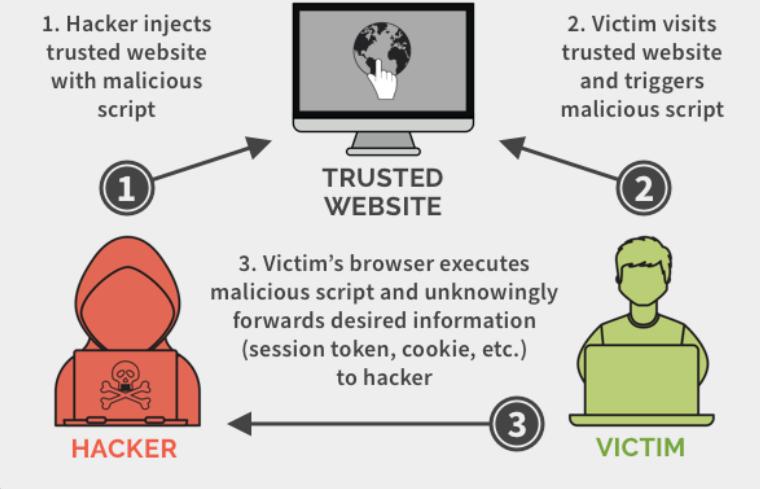
Dương Hữu Thành

22

22

Security

Cross-Site Scripting (XSS)



Dương Hữu Thành

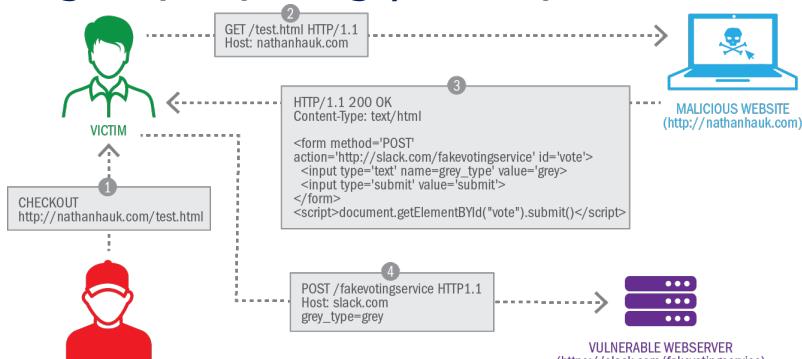
Nguồn ảnh: <https://spanning.com/>

23

23

Security

- Cross site request forgery (CSRF) là cách thức tấn công khi một người dùng xấu thực thi các hành động sử dụng quyền của người dùng khác không được sự đồng ý của họ.



Dương Hữu Thành

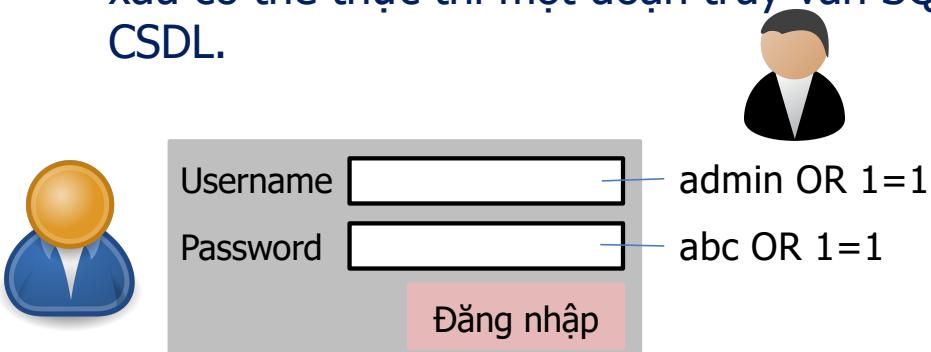
Nguồn ảnh: <https://medium.com/>

24

24

Security

- SQL injection là cách thức tấn công người dùng xấu có thể thực thi một đoạn truy vấn SQL trên CSDL.



"SELECT * FROM user WHERE username=" +
username + " AND password=" + password

25

Dương Hữu Thành

25

Security

- Clickjacking là cách thức tấn công bằng cách dùng một trang web **bọc** (wrap) lấy trang web khác, người dùng có thể thực hiện những hành động không mong muốn trên trang web mục tiêu này.



Dương Hữu Thành

Nguồn ảnh: <https://www.netsparker.com/>

26

26



Q&A

ThS. Dương Hữu Thành,
Khoa CNTT, Đại học Mở TP.HCM,
[thanh.dh@ou.edu.vn.](mailto:thanh.dh@ou.edu.vn)

27