

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐẠI HỌC QUỐC GIA HÀ NỘI**

=====\*



**BÁO CÁO MÔN HỌC QUẢN TRỊ MẠNG  
CHỦ ĐỀ**

***API(Application Programming Interface)***

**GIẢNG VIÊN: ThS.ĐỖ HOÀNG KIÊN**

***SINH VIÊN: Nguyễn Đức Hải - 17020715***

***Ngô Trung Kiên - 17020840***

***Lê Thành Vinh - 17021132***

***Trương Hoàng Long - 18020853***

Hà Nội, Ngày 06 Tháng 04 Năm 2021

II<sub>222</sub>

## Mục Lục

<b>1. GIỚI THIỆU</b>	<b>2</b>
<b>2. TỔNG QUAN VỀ API</b>	<b>2</b>
Về request	<b>5</b>
Về response	<b>5</b>
Status code của response	<b>6</b>
<b>3. REST-API</b>	<b>6</b>
REST và RESTful API	<b>6</b>
Ưu điểm của REST	<b>8</b>
Nhược điểm của REST	<b>9</b>
<b>4. SOAP (SIMPLE OBJECT ACCESS PROTOCOL)</b>	<b>9</b>
<b>5.RESTCONF-API</b>	<b>10</b>
<b>6. Tầm quan trọng của API</b>	<b>17</b>
<b>7. Đánh giá ưu, nhược điểm của API</b>	<b>17</b>
<b>8. Mức độ bảo mật của API</b>	<b>185</b>
<b>9. Phân chia công việc trong nhóm</b>	<b>15</b>

## **1. GIỚI THIỆU**

Hiện nay API nói chung và Web API nói riêng đang được ứng dụng ngày càng nhiều. Kiến trúc ứng dụng hiện đại ngày nay ngày càng phân tán, không phụ thuộc ngôn ngữ đã thúc đẩy việc ứng dụng API. API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface. API cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.. API là một cơ chế giúp phần mềm trên một thiết bị này nói chuyện với phần mềm trên một thiết bị khác. API được dùng rất nhiều trên Internet và trong thế giới của các nhà phát triển ứng dụng, trong thế giới của phần mềm. API hiện diện ở khắp nơi, ngoại trừ trong các thiết bị mạng. Tuy nhiên những năm gần đây, các hãng sản xuất thiết bị mạng viễn thông đều đã chú ý hỗ trợ APIs trong các sản phẩm của mình. Một thiết bị có hỗ trợ giao tiếp APIs sẽ giúp các công cụ phần mềm quản lý truy xuất trực tiếp các thông tin cấu hình, các trạng thái hoạt động và các dòng lưu lượng trên thiết bị. API là giao tiếp giữa phần mềm với phần mềm. Các thiết bị mạng hiện đại hầu hết đều hỗ trợ APIs. API sẽ sớm trở thành một công cụ chủ chốt để cấu hình và quản trị các thiết bị mạng trong một tương lai rất gần. CLI sẽ sớm nghỉ hưu sau hơn 25 năm làm việc.

## **2. TỔNG QUAN VỀ API**

- **Về cơ bản API có những đặc điểm sau:**

- API là một phần mềm cho phép truy cập dữ liệu hoặc dịch vụ giữa các ứng dụng khác nhau. Đó là một tập hợp các quy tắc mô tả cách thức một ứng dụng tương tác với một ứng dụng khác và các hướng dẫn để cho phép tương tác này xảy ra.
- Người sử dụng gửi yêu cầu truy cập API request đến máy chủ để yêu cầu thông tin và nhận được trả lời API response từ server cùng với thông tin được yêu cầu.
- API như một cổng kết nối các phần mềm lại với nhau phục vụ nhu cầu trao đổi thông tin hoặc thực thi một hay nhiều nghiệp vụ nào đó.

- **Phân loại API:**

Sự khác biệt giữa API mở (Open APIs), API nội (Internal APIs) và API cho đối tác (Partner APIs):

- API mở (Open APIs/Public APIs): Các API này có sẵn công khai và có thể được sử dụng không bị hạn chế. Vì các API này công khai nên nhiều nhà cung cấp API yêu cầu người sử dụng nhận khóa hoặc mã thông báo trước khi sử dụng nhằm giúp kiểm soát số lượng yêu cầu API mà họ nhận được và xử lý.
- API nội bộ: Đây là các API được sử dụng bởi một tổ chức hoặc công ty để truy cập dữ liệu và dịch vụ nội bộ.
- APIs cho đối tác: Đây là các API được sử dụng giữa một tổ chức hoặc công ty với đối tác để tạo thuận lợi cho việc kinh doanh giữa họ. Đối tác kinh doanh phải có giấy phép hoặc được sự cho phép để sử dụng API này.

- **Các dịch vụ truy cập API qua web:**

Web service là dịch vụ có sẵn qua Internet, sử dụng World Wide Web. Có 4 loại Web service APIs:

- Simple Object Access Protocol (SOAP)
- Representational State Transfer (REST)
- eXtensible Markup Language-Remote Procedure Call (XML-RPC)
- JavaScript Object Notation-Remote Procedure Call (JSON-RPC)

- **SOAP (Simple Object Access Protocol):**

Đây là giao thức sử dụng XML làm định dạng để truyền dữ liệu. Chức năng chính của SOAP là xác định cấu trúc của các thông điệp và phương thức giao tiếp.

- **REST (Representational State Transfer):**

REST không phải là một giao thức như các Web service khác mà nó là một tập hợp các nguyên tắc kiến trúc có một số đặc điểm nhất định như là giao diện đơn giản, các tài nguyên được yêu cầu một cách dễ dàng và các thao tác trên tài nguyên thông qua giao diện.

- **So sánh SOAP and REST:**

SOAP

- Tuân theo các quy tắc nghiêm ngặt và bảo mật tiên tiến
- Được điều khiển bởi các hàm (function)
- Đòi hỏi nhiều băng thông

REST

- Tuân theo những hướng dẫn mở cho phép các nhà phát triển đưa ra khuyến nghị dễ dàng
- Được điều khiển bởi dữ liệu
- Đòi hỏi băng thông tối thiểu

- **XML-RPC (eXtensible Markup Language-Remote Procedure Call):**

Đây là giao thức sử dụng định dạng chung XML để truyền dữ liệu, khác với SOAP sử dụng định dạng riêng biệt XML. XML-RPC sử dụng băng thông tối thiểu và đơn giản hơn nhiều so với SOAP.

- **JSON-RPC (JavaScript Object Notation-Remote Procedure Call):**

Giao thức này tương tự như XML-RPC nhưng thay vì sử dụng định dạng XML để truyền dữ liệu thì JSON-RPC sử dụng định dạng JSON.

- **So sánh JSON-RPC and XML-RPC :**

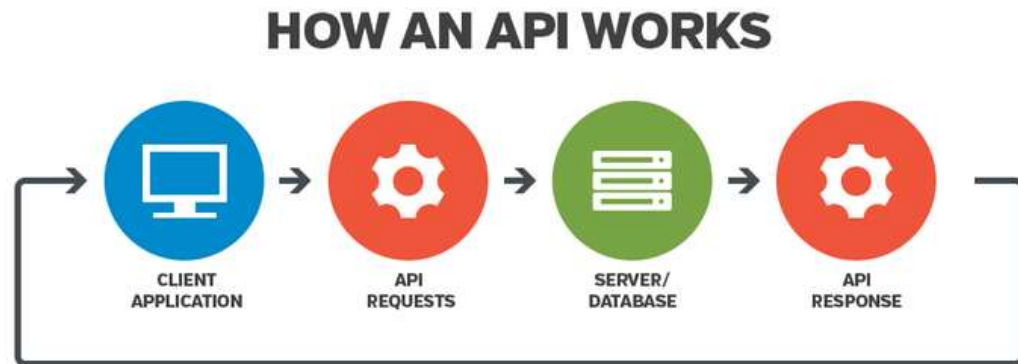
JSON

- Chỉ hỗ trợ text và numbers
- Tập trung chủ yếu vào Data
- Bảo mật thấp

XML

- Hỗ trợ các loại dữ liệu khác nhau như là text, numbers, images, charts, ...
- Tập trung chủ yếu vào Document
- Bảo mật tốt

- API được xây dựng trên chính 2 thành phần: Request và Response



- Về request

Một cái request đúng chuẩn cần có 4 thứ:

1. URL

URL là địa chỉ duy nhất cho 1 request, thường là đường dẫn tới một hàm xử lý logic.

2. Method

HTTP request có tất cả 9 loại method , 2 loại được sử dụng phổ biến nhất là GET và POST

- GET: Sử dụng để lấy thông tin từ server theo URI đã cung cấp.
- HEAD: Giống với GET nhưng response trả về không có body, chỉ có header.
- POST: Gửi thông tin tới sever thông qua các parameters HTTP.
- PUT: Ghi đè tất cả thông tin của đối tượng với những gì được gửi lên.
- PATCH: Ghi đè các thông tin được thay đổi của đối tượng.
- DELETE: Xóa Resource trên server.
- CONNECT: Thiết lập một kết nối tới server theo URI.
- OPTIONS: Mô tả các tùy chọn giao tiếp cho resource.
- TRACE: Thực hiện một bài test loop-back theo đường dẫn đến resource.

3. Headers

Là nơi chứa các thông tin cần thiết của 1 request nhưng end-users không biết có sự tồn tại của nó. Ví dụ: độ dài của request body, thời gian gửi request, loại thiết bị đang sử dụng, loại định dạng cái response mà client có đọc được...

4. Body

Là nơi chứa thông tin mà client sẽ điền.

- Về response

Sau khi nhận được request từ phía client, server sẽ xử lý cái request đó và gửi ngược lại cho client 1 cái response. Cấu trúc của 1 response tương đối giống phần request nhưng Status code sẽ thay thế cho URL và Method. Tóm lại, nó có cấu trúc 3 phần:

- Status code
- Headers
- Body

Phần Header và body tương đối giống với request.

#### - **Status code của response**

Status code (Mã hóa trạng thái thường được gọi là mã trạng thái) là một số nguyên 3 ký tự, trong đó ký tự đầu tiên của Status-Code định nghĩa loại Response và hai ký tự cuối không có bất cứ vai trò phân loại nào. Có 5 giá trị của ký tự đầu tiên:

- 1xx: Information (Thông tin): Khi nhận được những mã như vậy tức là request đã được server tiếp nhận và quá trình xử lý request đang được tiếp tục.
- 2xx: Success (Thành công): Khi nhận được những mã như vậy tức là request đã được server tiếp nhận, hiểu và xử lý thành công
- 3xx: Redirection (Chuyển hướng): Mã trạng thái này cho biết client cần có thêm action để hoàn thành request
- 4xx: Client Error (Lỗi Client): Nó nghĩa là request chứa cú pháp không chính xác hoặc không được thực hiện.
- 5xx: Server Error (Lỗi Server): Nó nghĩa là Server thất bại với việc thực hiện một request nhìn như có vẻ khả thi.

### **3. REST-API**

Website API là kiểu thiết kế mới nhất giúp tuân thủ các tiêu chuẩn **API** phổ biến nhất với nhiều ưu điểm vượt trội hơn so với các công nghệ cũ, nhưng trước khi nói rõ hơn về web API thì chúng ta cần biết sơ qua các tiêu chuẩn thiết kế **API** phổ biến và quan trọng nhất đó chính là chuẩn REST.

#### **REST và RESTful API**

REST là tên viết tắt của Representational State Transfer đây là một kiến trúc hình mẫu quy định về các cách ứng dụng giao tiếp tương tác với nhau. Một kiến trúc REST cơ bản gồm có 3 phần chính: một máy chủ ngoài còn gọi là external server dùng để chứa dữ liệu, một máy chủ API và một máy khách còn gọi là Client. Riêng về bộ phận máy khách là có thể là bất cứ thứ gì từ ứng dụng web cho đến các thư viện hoặc thậm chí là các phần khác nhau của một phần mềm máy tính được phân ra. Nhiệm vụ chính của REST đó



chính là cho phép client truy cập vào máy chủ API và thực hiện các lệnh ví dụ như lệnh lấy về, chỉnh sửa và xóa dữ liệu từ external server, ngoài ra REST còn cho phép các lập trình viên có thể thoải mái về việc truy xuất chỉnh sửa dữ liệu từ máy chủ mà không cần biết rõ hệ thống hoạt động như thế nào, chỉ cần sử dụng **API** là họ có thể thực hiện được các kết quả mong muốn bất kể máy chủ dữ liệu là một máy tính Linux hay Windows hay một phần mềm nào khác. Giao thức chính được REST sử dụng đó chính là HTTP, hầu hết các ứng dụng web hiện nay đều sử dụng giao thức này, đây là một trong những giao thức nhanh chóng và hiệu quả trong bối cảnh tốc độ đường truyền mạng ngày càng mạnh mẽ và kéo theo REST cũng là một kiến trúc có tốc độ nhanh bậc nhất.

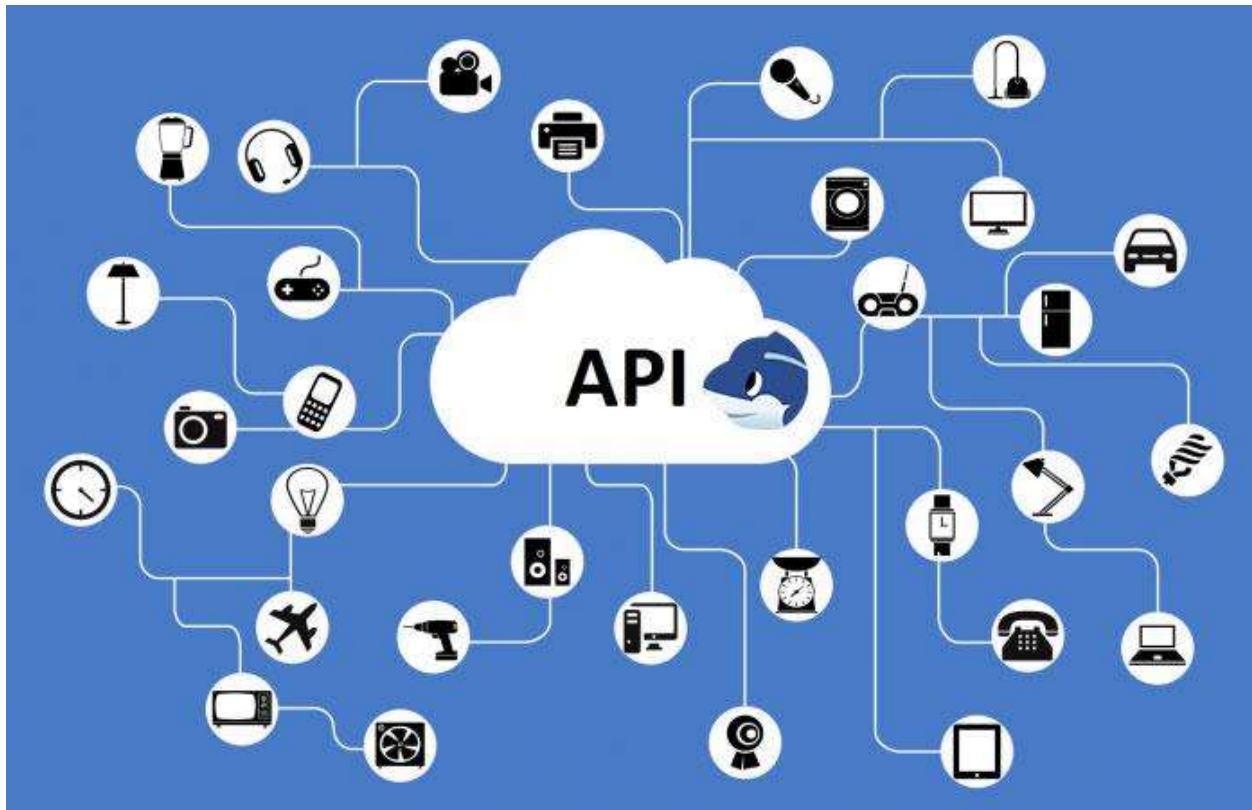
Ngoài ra REST còn có một điểm cộng đó chính là sử dụng **giao thức stateless** nghĩa là trạng thái. Nói rõ hơn đó chính là hệ thống REST không dùng session và cookie không cần biết thông tin trong mỗi lần yêu cầu đến máy chủ ngoài, vì vậy REST giảm tải được cho các máy chủ và tăng được tốc độ và hiệu suất làm việc. REST bản chất được thiết kế để sử dụng trong các dịch vụ web vì web app phát triển ngày càng phổ biến nhưng hiện nay bất kỳ một phần mềm nào cũng có thể sử dụng ứng dụng REST để giao tiếp với các ứng dụng khác một cách nhanh chóng và hiệu quả. Nhờ vậy mà tính đến hiện nay REST đã trở thành tiêu chuẩn mặc định cho bất kỳ giao thức nào và những API được thiết kế theo cấu trúc của REST được gọi là RESTFUL API.

CRUD là tên viết tắt của 4 từ sau (Create, Read, Update và Delete) đây là bốn chức năng cơ bản nhất của bất kỳ một hệ thống dữ liệu nào. Kiến trúc REST và các **RESTFUL API** đều hỗ trợ đầy đủ bốn lệnh CRUD này để cho phép bạn thoải mái thao tác với các dữ liệu lấy từ máy chủ theo ý muốn của người dùng.

Trong kiến trúc REST, lập trình viên cần biết đến khi sử dụng REST **API**:

- + POST: tạo một dữ liệu, thông tin mới
- + GET: lấy/đọc một dữ liệu, thông tin
- + PUT: cập nhật dữ liệu, thông tin đã có
- + DELETE: xóa dữ liệu, thông tin trên máy chủ

Trong đó, lệnh GET là một trong những lệnh an toàn và cũng là lệnh phổ biến nhất được cho phép bởi hầu hết các RESTFUL API hiện nay. Còn đối với 3 lệnh còn lại đều có ít nhiều sự ảnh hưởng làm thay đổi dữ liệu gốc trên máy chủ, làm cho máy chủ có thể dẫn đến các tai nạn ngoài ý muốn, nhưng các bạn nên lưu ý chỉ có những API được thiết kế riêng và cấp thẩm quyền mới được thực hiện các lệnh đó.



RESTFul API và ứng dụng trong thiết kế website

### Ưu điểm của REST

- Hiệu năng: các thành phần đảm bảo được việc giao tiếp theo đúng một quy ước giúp hệ thống có thể vận hành tốt hơn.
- Tính khả biến: với các hệ thống cần thay đổi các tài nguyên liên tục, sử dụng **REST** với việc tạo request đơn giản sẽ giúp mọi chuyện trở nên đơn giản hơn.
- Tính mở rộng: các hệ thống **REST** có khả năng mở rộng rất cao nhờ sự tách biệt giữa các thành phần và các quy ước giao tiếp được quy định sẵn.
- Tính linh hoạt: việc chuẩn hoá interface giúp hệ thống trở nên linh hoạt hơn, có thể sử dụng cho cho nhiều nền tảng khác nhau, mobile, web,...
- Trong sáng: trong giao tiếp giữa các thành phần, các request trở nên rất rõ ràng, dễ hiểu.
- Đơn giản: xây dựng rất đơn giản,
- Tính tin cậy: khó để xảy ra lỗi trong giao tiếp giữa các thành phần gây sụp đổ hệ thống.

## Nhược điểm của REST

- Chỉ hoạt động trên các giao thức HTTP.
- Việc bảo mật và xác thực có thể không đảm bảo bằng SOAP (stateless).

## 4. SOAP (SIMPLE OBJECT ACCESS PROTOCOL)

SOAP là một giao thức truy cập Web-Service được sử dụng trong một thời gian và hưởng tất cả các lợi ích của việc sử dụng lâu dài. Được phát triển bởi Microsoft, SOAP không thực sự đơn giản như từ viết tắt của nó. Trong khi đó, REST là một chiến binh mới, tìm kiếm để sửa các vấn đề mà SOAP gặp phải và cung cấp các phương thức thực sự đơn giản để truy cập Web-Service. Tuy nhiên, đôi khi SOAP thực sự dễ sử dụng hơn, đôi khi REST có vấn đề của riêng nó. Cả hai kỹ thuật đều có vấn đề cần xem xét khi quyết định sử dụng phương thức nào.

Cả SOAP và REST đều có những điểm tương đồng trên giao thức HTTP, SOAP là một tổ hợp các pattern truyền tin nghiêm ngặt hơn REST. Quy định trong SOAP là rất quan trọng bởi vì nếu thiếu chúng, bạn không thể đạt được bất kỳ level tiêu chuẩn hóa nào. REST mang một kiến trúc không yêu cầu quy trình và linh hoạt hơn. Cả SOAP và REST đều dựa trên những quy định đã tồn tại trong một thời gian dài mà mọi người đều đồng ý tuân theo do lợi ích mà nó mang lại.

SOAP dựa hoàn toàn vào XML để cung cấp các services truyền tin. Microsoft ban đầu phát triển SOAP để thay thế cho các công nghệ cũ hơn không hoạt động tốt trên Internet như Distributed Component Object Model (DCOM) and Common Object Request Broker Architecture (CORBA). Những công nghệ này không thành công vì chúng dựa vào truyền tin nhị phân, cách truyền tin XML mà SOAP sử dụng làm việc tốt hơn qua Internet.

XML được sử dụng để tạo ra các requests và nhận responses trong SOAP, nó có thể trở nên cực kỳ phức tạp. Trong một số ngôn ngữ lập trình, bạn cần phải xây dựng các requests đó bằng tay, điều này trở nên mơ hồ vì SOAP không dung nạp được các lỗi. Tuy nhiên, một vài ngôn ngữ khác có thể sử dụng các phím tắt mà SOAP cung cấp; Có thể giúp bạn giảm bớt effort cần thiết để tạo request và phân tích responses.

Một trong những tính năng quan trọng nhất của SOAP là tích hợp trong xử lý lỗi. Nếu có sự cố với requests của bạn, responses chứa thông tin lỗi mà bạn có thể sử dụng để

khắc phục sự cố. Bạn có thể không sở hữu các Web service, tính năng này đặc biệt vô cùng quan trọng; Nếu không bạn sẽ thắc mắc tại sao mọi việc lại không thành công. Các báo cáo lỗi thậm chí cung cấp mã code chuẩn để có thể tự động hoá một số nhiệm vụ xử lý lỗi trong mã của bạn.

Một tính năng khác khá thú vị của SOAP là bạn không nhất thiết phải sử dụng nó với giao thức HyperText Transfer Protocol (HTTP). Có một chỉ định rõ ràng để sử dụng SOAP qua Simple Mail Transfer Protocol (SMTP) và không có lý do gì bạn không thể sử dụng nó trên các phương thức khác.

### **Ưu điểm của SOAP:**

- Ngôn ngữ, nền tảng và phương tiện độc lập (REST yêu cầu sử dụng HTTP)
- Hoạt động tốt trong môi trường doanh nghiệp (REST giao tiếp thẳng trực tiếp)
- Tiêu chuẩn hoá
- Cung cấp khả năng mở rộng đáng kể trước khi xây dựng dưới dạng các tiêu chuẩn WS (Tiêu chuẩn Dịch vụ Web – Địa chỉ hóa (WS-Addressing))
- Tích hợp xử lý lỗi
- Tự động hóa khi sử dụng với một số sản phẩm ngôn ngữ

### **Nhược điểm của SOAP:**

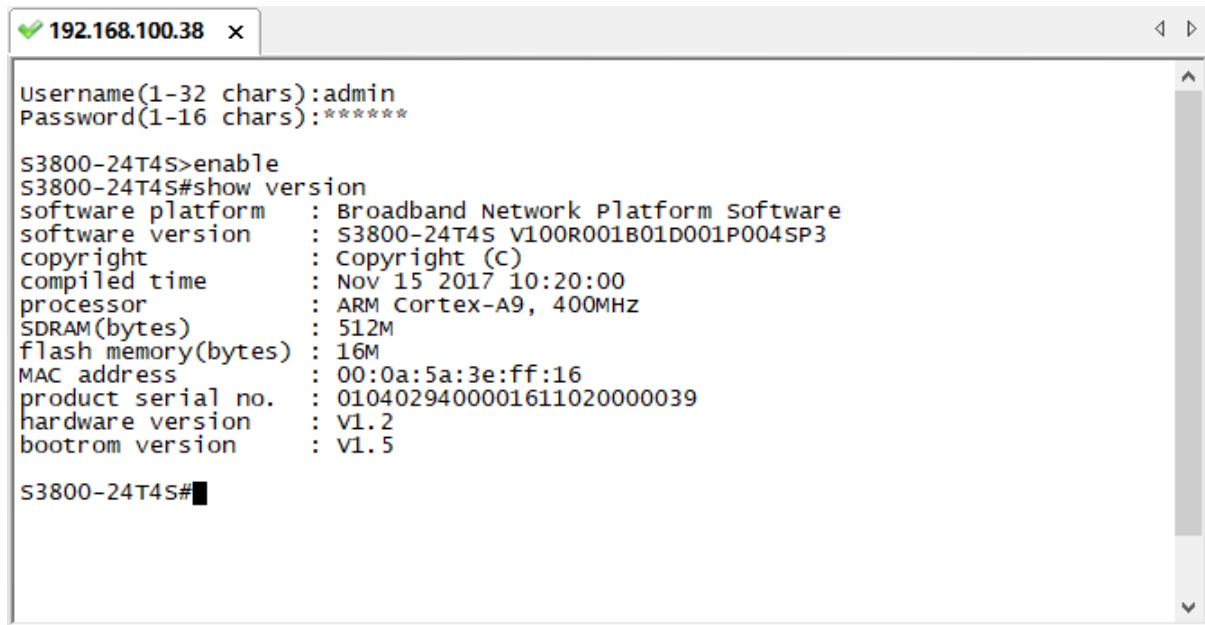
- Việc duy trì trạng thái (stateful) khiến tiêu tốn tài nguyên cho các metadata.
- Khó sử dụng và không phổ biến bằng REST trong các ứng dụng web hay mobile thông thường.
- Chỉ sử dụng XML

## **5.RESTCONF-API**

Có 2 cách truyền thông mà các kỹ sư quản trị mạng thường sử dụng để cấu hình mạng đó là sử dụng **CLI (Command Line Interface)** và **SNMP (Simple Network Management Protocol)**

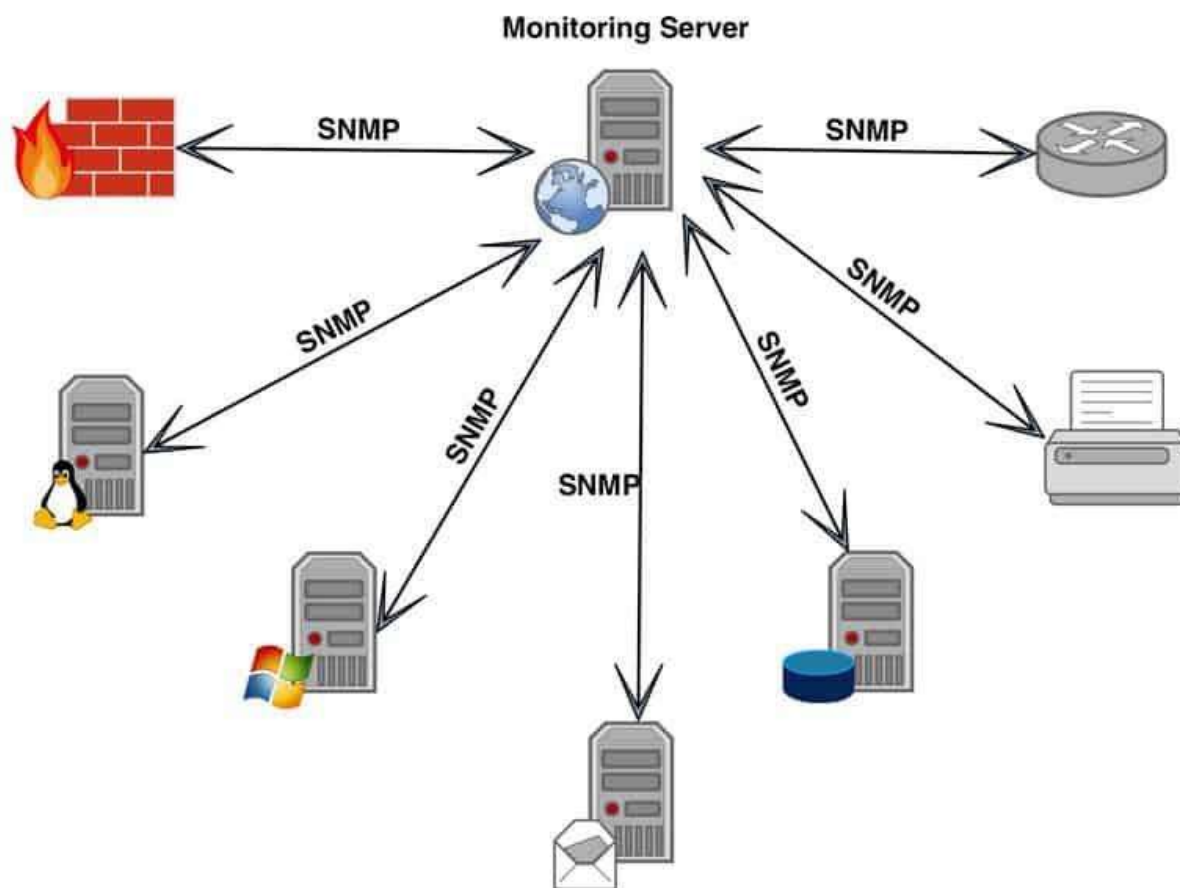
Nhưng ở cả cách trên đều có những điểm còn hạn chế như sau:

- Với CLI thì đây cách khá chậm và phức tạp. Quan trọng hơn hết là với cách này sẽ có sự đồng nhất giữa các thiết bị mạng từ các nhà cung cấp khác nhau (bởi vì không có tính đồng nhất trong câu lệnh). Ngoài ra nó rất hạn chế khi chỉ thao tác với số lượng nhỏ thiết bị mạng



```
192.168.100.38 x
Username(1-32 chars):admin
Password(1-16 chars):*****
S3800-24T4S>enable
S3800-24T4S#show version
software platform      : Broadband Network Platform Software
software version       : S3800-24T4S V100R001B01D001P004SP3
copyright              : Copyright (c)
compiled time          : Nov 15 2017 10:20:00
processor              : ARM Cortex-A9, 400MHZ
SDRAM(bytes)           : 512M
flash memory(bytes)    : 16M
MAC address            : 00:0a:5a:3e:ff:16
product serial no.     : 0104029400001611020000039
hardware version       : V1.2
bootrom version        : V1.5
S3800-24T4S#
```

- Với SNMP thì giúp việc quản lý thiết bị một cách dễ dàng hơn nhưng không đáng tin cậy, không đảm bảo tính bảo mật



Vì thế để khắc phục những vấn đề của cách quản trị mạng cũ thì network automation ra đời và được xem là xu thế của tương lai, có thể thay đổi cách làm việc của người quản trị mạng.

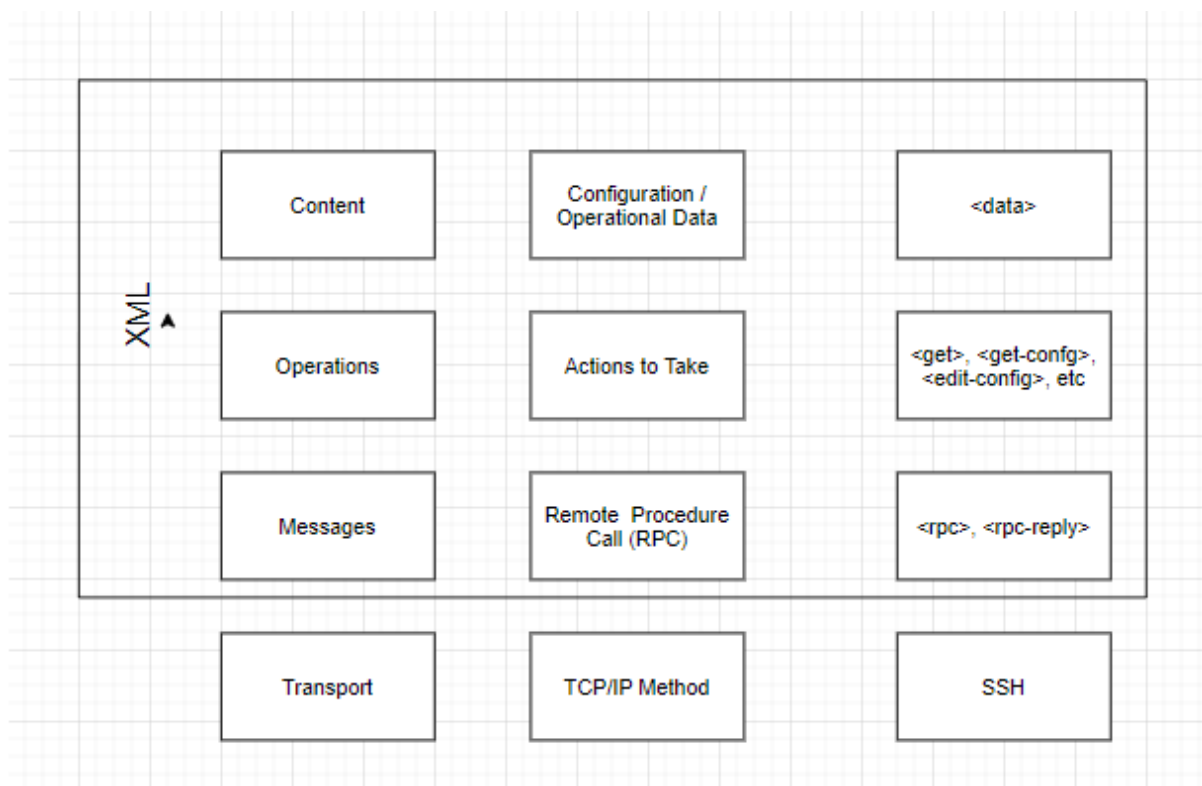
Với số lượng lớn thiết bị mạng lên đến hàng trăm hàng nghìn thiết bị, người quản trị mạng cần có những cách làm việc mới hiệu quả hơn tiết kiệm thời gian công sức hơn

Với cách làm việc mới người quản trị mạng tương tác với app/bot controller (Ví dụ : cisco DNA center) và sau đó bot controller sẽ tự động tương tác với số lượng lớn thiết bị mạng như get information, put, sử dụng query... Nhờ đó Tận dụng khả năng tính toán và tự động hóa số lượng lớn của máy móc, điều mà con người không làm được. Vì bot controller nhanh hơn con người rất nhiều lần, cần có 1 phương pháp giao tiếp đặc biệt để

tương tác giữa controller và các thiết bị mạng thay vì dùng phương pháp cũ như SSA( Stealthwatch Administration) . Người quản trị mạng cần sử dụng 1 API được hỗ trợ/ hiểu bởi cả controller và thiết bị mạng. RESTCONF ( REST configuration) chuyên dụng trong cấu hình mạng, là API để tương tác giữa controller và thiết bị mạng ( networking devices)

Trước khi tìm hiểu về **RESTCONF** ta cần phải tìm hiểu về **NETCONF** trước tiên

**NETCONF** (Network Configuration Protocol) là giao thức quản trị mạng được phát triển và tiêu chuẩn hóa bởi IETF. **NETCONF** cung cấp những cơ chế để cài đặt, thao tác và xóa cấu hình của các thiết bị mạng. Sử dụng cơ chế SSH và XML để truyền tải dữ liệu. Data model để được biểu diễn dưới dạng XML để truyền tải từ Manager đến các Agent.



NETCONF bao gồm 4 layer trong đó 3 layer biểu diễn dưới dạng xml

- Content là nơi mô hình phác thảo cấu trúc của dữ liệu thể hiện nội dung của dữ liệu
- Operations là các phương thức tác động lên thiết bị
- Messages là nơi thông báo các RPC (Remote Procedure Call) được sử dụng
- Transport là nơi vận chuyển dữ liệu từ manager đến agent

NETCONF sử dụng một ngôn ngữ mô hình hóa gọi là YANG (Yet Another Next Generation) làm data model. Data bên dưới là các thiết bị mạng sử dụng Yang và sau đó mã hóa thành JSON để đưa lên các tầng trên. Ngôn ngữ này đọc lập với giao thức và có thể chuyển đổi thành bất kỳ dạng mã hóa nào mà giao thức cấu hình mạng hỗ trợ như là XML hoặc JSON.

Về cơ bản RESTCONF tương tự như NETCONF nhưng được cải tiến thêm có thể sử dụng Rest-API. Ngoài ra, so với NETCONF thì RESTCONF đơn giản hơn nhiều. Sau đây ra hãy xem qua các lớp của RESTCONF.

Content: Không giống như NETCONF (thường sử dụng nhiều XML), RESTCONF cho phép sử dụng cả XML và JSON, đặc biệt JSON là một trong những định dạng dữ liệu phổ biến nhất hiện nay

Operations: RESTCONF sử dụng các HTTP operations khác nhau để cung cấp CRUD (Create, Replace, Update, Delete) operations

Transport: Giao thức truyền tải là HTTP, cho phép sử dụng HTTPS. Cung cấp những lợi ích về bảo mật mà TLS mang lại

Ngoài ra Rest-API cũng là khái niệm quan trọng ta cần tìm hiểu

API (Application Programming Interface) là giao diện lập trình ứng dụng giúp tạo ra các phương thức kết nối với các thư viện và ứng dụng khác nhau. Còn REST là một dạng chuyển đổi cấu trúc dữ liệu, là một phong cách kiến trúc cho việc thiết kế các ứng dụng có kết nối. Nó sử dụng HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.





- Việc sử dụng các HTTP method và REST (Representational State Transfer) đã phổ biến làm cho RESTCONF dễ tiếp cận hơn với các nhà phát triển và giúp các nhà phát triển có nhiều lựa chọn về thư viện và các công cụ
- RESTCONF hỗ trợ 2 ngôn ngữ mã hóa dữ liệu là XML và JSON, so với chỉ mình XML của NETCONF
- RESTCONF phù hợp với hệ thống mạng có nhiều client cùng truy cập mọi lúc

## 5.2. Nhược điểm

- RESTCONF không hỗ trợ transaction trên mạng diện rộng mà chỉ hỗ trợ trong việc cấu hình mô hình D2D.
- RESTCONF phù hợp trong việc cấu hình một mạng gồm một cổng và một bộ điều phối, không phù hợp để cấu hình mạng gồm một bộ điều phối và nhiều thiết bị.

## 5.3. So sánh với các giao thức khác

	SNMP	NETCONF	RESTConf
Chuẩn hóa	IETF	IETF	IETF
Cấu trúc dữ liệu	MIBs	YANG Models	YANG models
Ngôn ngữ mô hình hóa dữ liệu	SMI	YANG	YANG
Ngôn ngữ mã hóa dữ liệu	BER	XML	XML, JSON
Tài nguyên	OIDs	Paths	URLs

Ngăn xếp giao vận	UDP	SSH	SSL
		TCP	HTTP
			TCP

## 6. TẦM QUAN TRỌNG CỦA API

API là khớp nối giữa các thành phần phần mềm. Giả sử bạn có một tính năng cần cung cấp cho module khác phần mềm khác thì bạn sẽ mở một API để tác giả của module/ phần mềm truy cập vào.

Ví dụ, trên các thiết bị điện toán thì hệ điều hành là phần mềm duy nhất có khả năng truy cập tới các thiết bị phần cứng. Do đó, hệ điều hành sẽ phải cung cấp API để ghi file, đọc file, đọc dữ liệu.... Mỗi ứng dụng khi hoạt động sẽ gọi tới API tương ứng của hệ điều hành.

Khi một phần mềm gọi tới API, phần mềm gọi có thể cung cấp dữ liệu đầu vào và đòi hỏi dữ liệu đầu ra từ API hoặc không, nhưng trong mọi trường hợp, phần mềm gọi để có thể tiếp tục hoạt động thì nó cần phần mềm cung cấp phải thực hiện những gì đã cam kết qua API.

## 7. ƯU, NHƯỢC ĐIỂM CỦA API

API đã trở thành một công cụ đắc lực để tối ưu hóa trải nghiệm của người dùng trên các phần mềm. Giao diện này sở hữu rất nhiều ưu điểm như:

- Kết nối mọi lúc mọi nơi, chỉ cần có kết nối internet
- Giao tiếp API được thực hiện 2 chiều nên các thông tin nhận được có độ tin cậy cao
- Đem đến những giải pháp phát triển phần mềm mới cho các nhà lập trình
- Sở hữu cấu hình đơn giản
- Mã nguồn mở
- Hỗ trợ đầy đủ các thành phần MVC như: model binder, IoC container, dependency injection, action result, filter, unit test, routing, controller,...
- Thân thiện với người sử dụng

Tuy nhiên, giao diện này cũng có một số nhược điểm nhất định như chi phí phát triển, vận hành cao và đòi hỏi kiến thức chuyên sâu.

## 8. MỨC ĐỘ BẢO MẬT CỦA API

Dữ liệu trong thiết bị của bạn không bao giờ được chuyển hoàn toàn đến máy chủ. Máy chủ cũng chỉ cung cấp lại đúng những thông tin mà bạn yêu cầu. Vì vậy, khó có thể xảy ra những lỗ hổng bảo mật trong hệ thống này.

## 9. PHÂN CHIA CÔNG VIỆC TRONG NHÓM

**Lê Thành Vinh:** Tìm hiểu về khái niệm và cách hoạt động cơ bản của Restconf-API, làm slide phần RestAPI và phần SOAP

**Trương Hoàng Long:** Tìm hiểu về ưu điểm và nhược điểm của Restconf-API và so sánh Restconf-API với các giao thức khác, làm slide phần Restconf-API

**Ngô Trung Kiên:** Tìm hiểu khái niệm, ưu nhược điểm phần SOAP, tìm hiểu cách cài đặt RESTCONF

**Nguyễn Đức Hải:** Tìm hiểu về khái niệm, ưu nhược điểm RestAPI cũng như phần phân API, viết document phần tương ứng

## TÀI LIỆU THAM KHẢO

SOAP: <https://blog.smartbear.com/apis/understanding-soap-and-rest-basics/>

API: <http://dbahire.com/api-la-gi>

<https://viblo.asia/p/tim-hieu-kien-thuc-co-ban-ve-api-maGK7A4Mlj2>

<https://viblo.asia/p/api-la-gi-tat-tan-tat-ve-api-V3m5WXdwKO7>

<https://www.totolink.vn/article/482-api-la-gi-nhung-dieu-ban-can-biet-ve-api.html>

<https://viblo.asia/p/co-ban-ve-api-testing-phan-2-RQqKLQmbZ7z>