**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**BÁO CÁO MÔN HỌC**

**TÍCH HỢP HỆ THỐNG**

Nhóm 3: **ULTRAESB**

***Danh sách sinh viên thực hiện:***

Lê Lương Tuấn Anh

Nguyễn Tuấn Anh

Lê Văn Duy

Bùi Ngọc Đăng

Phùng Đình Xuân

**HÀ NỘI – 2019**

## **Tích hợp hệ thống là gì?**

Trong kỹ thuật, hiểu đơn giản, tích hợp hệ thống (System Integration – SI) là kết nối một chuỗi các hệ thống con với những tính năng khác nhau vào một hệ thống lớn, đảm bảo các hệ thống con được gắn kết chặt chẽ với nhau như một thể thống nhất. Vận hành theo mục đích riêng có của từng doanh nghiệp. Là giải pháp đáp ứng mọi yêu cầu phức tạp nhất về vấn đề công nghệ một cách thông minh, hơn nữa là tùy biến theo nhu cầu.



Tích hợp hệ thống

Trong công nghệ thông tin, Tích hợp hệ thống giúp tích hợp các hệ thống con rời rạc, các phần mềm ứng dụng khác nhau bằng việc sử dụng các kỹ thuật kết nối như mạng máy tính, tích hợp ứng dụng, quản lý quy trình, lập trình... Tích hợp hệ thống cũng là quy trình giúp gia tăng giá trị và năng lực của hệ thống mẹ nhờ hợp lực tương tác giữa các hệ thống con.

## **Những lợi ích của tích hợp hệ thống.**

Tích hợp hệ thống giúp doanh nghiệp tối ưu chi phí nhờ khả năng tích hợp linh hoạt khi được lựa chọn công nghệ, thiết bị, dịch vụ phù hợp. Hơn nữa, tích hợp hệ thống còn giúp tối ưu hóa nhu cầu sử dụng, giúp doanh nghiệp hoạch định và đầu tư theo từng giai đoạn hoặc phân hệ nghiệp vụ tùy vào khả năng và mức nhu cầu của họ trong giai đoạn đó. Sự ứng dụng công nghệ hiện đại trong quá trình quản lý, sản xuất, giúp tinh gọn quy trình, rút ngắn thời gian quản trị hệ thống và ngăn chặn các rủi ro xâm hại từ những môi trường kinh doanh không lành mạnh. Góp phần làm tăng sức cạnh tranh của doanh nghiệp trên thị trường.

Để tích hợp hệ thống thành công, cốt lõi nằm ở năng lực của nhà tích hợp hệ thống. Khi các yếu tố phần mềm, phần cứng nền tảng là như nhau. Thì năng lực triển khai của kỹ sư tích hợp là nghệ thuật kết nối hệ thống rời rạc này thành khối sức mạnh hợp nhất.

Toàn Cầu là một trong những với đối tác kinh doanh của những tập đoàn công nghệ hàng đầu như: Juniper,Cisco, IBM, Hitachi, ORACLE, Intel Security, BMC Software, Dell, HP…v.v…



## **ULTRAESB là gì?**

**ESB là gì?**

ESB là một hệ thống giải quyết các vấn đề liên quan đến việc tích hợp mà không cần phải xóa bỏ những giải pháp có sẵn. Mục đích của ESB là làm cho việc tích hợp các ứng dụng và quy trình trở nên thuận tiện hơn bằng cách cung cấp một quy trình phân tán, điều hướng thông minh, bảo mật và có thể tự động chuyển đổi dữ liệu. Trong hệ thống ESB những dịch vụ trên là những dịch vụ nền tảng do đó các ứng dụng không cần phải thi hành riêng biệt những yêu cầu trên theo một cách thức riêng biệt của chúng.

ESB tập trung vào giải quyết những điểm yếu của những giải pháp có sẵn bằng cách tạo ra một nền tảng chuẩn cho việc tích hợp. Giải pháp point to point, yêu cầu cứ n thành phần tham gia hệ thống thì phải có n-1 interface để có thể giao tiếp được với các thành phần còn lại, được thay thế bằng giải pháp bus, mỗi thành phần chỉ yêu cầu có 1 interface để giao tiếp với bus và thông qua bus để giao tiếp với các thành phần còn lại. Một hệ thống ESB cung cấp việc giao tiếp phân tán, chuyển hướng, xử lý nghiệp vụ, ổn định và bảo mật. Nó cũng cung cấp các dịch vụ có khả năng plug-and-play bởi vì nó dựa trên chuẩn nên những dịch vụ plug-and-play này có thể được cung cấp bởi các công ty khác và hoạt động trên bus.

**UltraESB** là một [xe buýt dịch vụ doanh nghiệp](https://en.wikipedia.org/wiki/Enterprise_service_bus) nhẹ (ESB) có khả năng hỗ trợ nhiều phương tiện vận chuyển và định dạng tin nhắn nguyên bản. Nó cho phép các thông điệp được trung gian thông qua các ngôn ngữ kịch bản Java hoặc JSR 223 thông qua API, và là ESB đầu tiên yêu cầu hỗ trợ cho việc ủy ​​quyền Zero-Copy của các tin nhắn có khả năng IO không chặn.Năm 2013, công ty Fortune chọn là nền tảng Quản lý API và ESB số một gần đây dựa trên hiệu suất và tính ổn định của nó.

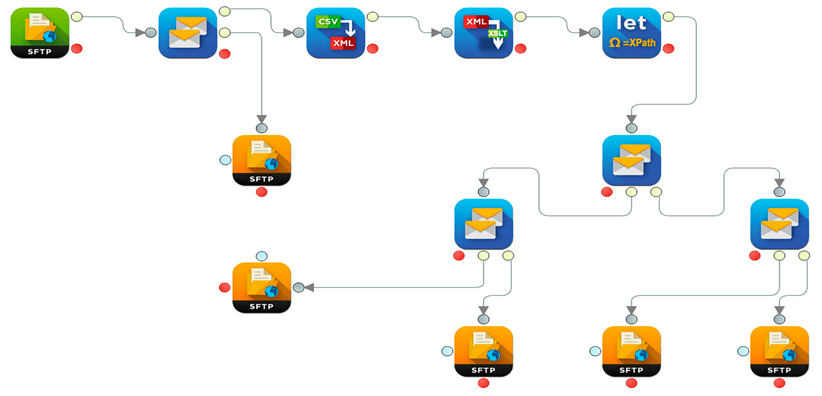
Phiên bản ban đầu được xuất bản vào tháng 1 năm 2010 và mã này sau đó đã được mở nguồn theo Giấy phép Công cộng Chung ( [AGPL](https://en.wikipedia.org/wiki/Affero_General_Public_License) ) được OSI phê duyệt vào ngày 24 tháng 8 năm 2010. Xây dựng hàng đêm liên tục để đảm bảo hiệu suất của mã phát triển mới nhất.

## **Tính năng nổi bật ULTRAESB**

* Sử dụng bất kỳ IDE nào - Cho phép người dùng chọn bất kỳ IDE nào để phát triển làm giảm đáng kể đường cong học tập để phát triển
* Khả năng kiểm tra đơn vị, cùng với độ bao phủ mã gần 50% [[6]](https://en.wikipedia.org/wiki/UltraESB#cite_note-6)
* Dễ dàng mở rộng thông qua các thư viện của bên thứ ba, đậu được cấu hình Spring và mã tùy chỉnh
* Hỗ trợ số lượng định dạng và chuyển đổi thông báo [[7]](https://en.wikipedia.org/wiki/UltraESB#cite_note-7)
* Hỗ trợ tải trọng và giao thức như vận chuyển AS2, HL7, SOAP, REST, FIX như MLLP / S, JMS, AMQP, FTP / S, SFTP và WS-Security, XACML và JTA XA
* Tải xuống trọng lượng rất nhẹ ~ 42MB để phân phối hoàn chỉnh (Phiên bản tối thiểu <9MB)
* Hỗ trợ phân cụm và chuyển đổi dự phòng cho tính sẵn sàng cao
* Các số liệu tích hợp và quản lý dựa trên JMX, CLI và Web

***Hỗ trợ nhiều giao thức truyền thông và định dạng tin nhắn:***

|  |  |  |  |
| --- | --- | --- | --- |
| Transports :  -HTTP/S  -JMS  -Email (POP3/IMAP/SMTP)  -AMQP  -File/SFTP/FTP/  FTPS/Samba  -WebSockets  -TCP/S  -MLLP/S | Protocols:  -REST  -SOAP  -Hessian  -FastInfoset  -AS2  -Protocol Buffers  -SWIFT  -FIX | Payloads:  -XML/POX  -JSON  -Text/String  -HL7  -Map  -Binary  -SWIFT  -EDI | Features:  -WS-Security  -XSLT/XQuery  -Data Services  -Transactional Support  -AuthN and AuthR  -Caching  -Throttling |



UltraESB chạy với một bộ kết nối và bộ xử lý trên khung công tác Project-X. Hàng chục Trình kết nối và Bộ xử lý có sẵn trong hộp trong kho kết nối / bộ xử lý.

## **Hỗ trợ công cụ**

### ***Công cụ thiết kế và phát triển***

UltraESB được trang bị các công cụ phát triển khác nhau để thử nghiệm, phát triển và triển khai.

1. [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) , [Eclipse](https://en.wikipedia.org/wiki/Eclipse_(software)) và [NetBeans](https://en.wikipedia.org/wiki/NetBeans) là IDE
2. Plugin cho IntelliJ IDEA
3. UTerm - Một dòng lệnh tương tác và tiện ích quản trị tập lệnh
4. Hỗ trợ quản lý / giám sát dựa trên JMX và tích hợp với giải pháp giám sát nguồn mở [Zabbix](https://en.wikipedia.org/wiki/Zabbix)

### ***Công cụ quản lý và giám sát***

Các công cụ quản lý và giám sát được cung cấp với tất cả các bản phân phối khác với mức tối thiểu (Phân phối hoàn toàn & phân phối nguồn).

1. UConsole - Bảng điều khiển quản lý và giám sát dựa trên web
2. IMonitor - thực thi như một ứng dụng web độc lập và cho phép quản lý và giám sát một nút hoặc một cụm ESB

## **Cách sử dụng ULTRAESB để tích hợp**

***Các connector được sử dụng:***

### **NIO HTTP Ingress Connector**

NIO HTTP Ingress Connector được sử dụng để nhận tin nhắn HTTP từ máy khách từ xa bằng cách lắng nghe một cổng cụ thể trên đường dẫn dịch vụ cụ thể và đưa các tin nhắn đó vào công cụ ESB. NIO HTTP Ingress Connector hỗ trợ HTTP RFC 2616.

NIO HTTP Ingress Connector cần thư mục gắn trên đĩa RAM để hoạt động. Đảm bảo thư mục / tmp / ram đã được tạo với các quyền thích hợp và nó được gắn trên ram.

Để sử dụng NIO HTTP Ingress Connector, trước tiên bạn phải chọn phụ thuộc Trình kết nối HTTP NIO từ danh sách trình kết nối khi bạn đang tạo một dự án Ultra trống. Nếu bạn đã tạo một dự án, bạn có thể thêm phụ thuộc NIO HTTP Ingress Connector thông qua Đăng ký thành phần. Từ menu Công cụ, chọn Ultra Studio → Đăng ký thành phần và từ danh sách Trình kết nối, chọn phụ thuộc NIO HTTP Ingress Connector .

Cổng ra

|  |  |
| --- | --- |
| Bộ xử lý | Thông báo nhận được tới NIO HTTP Ingress Connector từ Máy khách HTTP sẽ được phát ra từ cổng ra này |
| Về ngoại lệ | Thông báo sẽ được phát ra từ cổng ra này nếu Trình kết nối Ingress không xử lý tải trọng thư |
| Đầu vào | Thông báo được gửi tới cổng ra này sẽ được gửi dưới dạng phản hồi cho Máy khách HTTP đã khởi tạo yêu cầu |

### 

### **Slack Egress Connector**

Slack Egress Connector được sử dụng để gửi tin nhắn từ ESB đến nhóm Slack bằng WebHook đã đăng ký.

Cổng ra

|  |  |
| --- | --- |
| Về ngoại lệ | Thông báo sẽ được phát ra từ cổng ra này nếu phần tử xử lý không gửi được tin nhắn đến hệ thống bên ngoài |
| Bộ xử lý đáp ứng | Thông điệp phản hồi nhận được sẽ được chuyển đến cổng phát. Bạn có thể sử dụng cổng này để kết nối bất kỳ thành phần xử lý hoặc trình kết nối nào được cho là xử lý / tiêu thụ phản hồi. |

## 

Cổng phụ

|  |  |
| --- | --- |
| Hoạt động kết nối | Cổng này được sử dụng để kết nối các yếu tố hoạt động với Đầu nối ra. Theo mặc định, người dùng không phải kết nối bất kỳ yếu tố hoạt động nào và hoạt động kết nối mặc định sẽ được sử dụng. |

### 

### **NIO HTTPS Egress Connector**

Đây đơn giản là phiên bản bảo mật của NIO HTTPS Egress Connector . Đó là, trình kết nối đầu ra này sử dụng các kết nối HTTP được mã hóa để gửi tin nhắn đến các điểm cuối khác đang chấp nhận kết nối HTTPS. Để đạt được điều đó, trình kết nối này có thể được cấu hình để sử dụng bảo mật SSL (Lớp cổng bảo mật) và TLS (Bảo mật lớp vận chuyển) .

Ngoài các cấu hình thông thường cần phải được thực hiện trong NIO HTTPS Egress Connector,nó yêu cầu một số cấu hình quan trọng khác phải được thực hiện. Ví dụ: vì trình kết nối này đang gửi tin nhắn đến các điểm cuối khác, chúng tôi phải định cấu hình cửa hàng ủy thác để nói rằng đây là các chứng chỉ của các điểm cuối sẽ được tin cậy khi giao tiếp. Tất cả những cấu hình cần thiết được mô tả trong các phần dưới đây.

Cổng ra

|  |  |
| --- | --- |
| Về ngoại lệ | Thông báo sẽ được phát ra từ cổng ra này nếu phần tử xử lý không thể xử lý tải trọng do một số lý do |
| Bộ xử lý đáp ứng | Thông điệp phản hồi nhận được sẽ được chuyển đến cổng này. Bạn có thể sử dụng cổng này để kết nối bất kỳ phần tử hoặc đầu nối cần thiết nào được cho là xử lý / tiêu thụ phản hồi. |

## 

Cổng phụ

|  |  |
| --- | --- |
| Hoạt động kết nối | Cổng này được sử dụng để kết nối các yếu tố hoạt động với Đầu nối ra. Theo mặc định, người dùng không phải kết nối bất kỳ yếu tố hoạt động nào và hoạt động kết nối mặc định sẽ được sử dụng. |

Các Processors được sử dụng:

**Add Query Parameter**

Add Query Parameterlà một thành phần xử lý có thể được sử dụng để thêm một tham số truy vấn vào thông báo luồng tích hợp được cho là gửi ra bằng trình kết nối đầu ra HTTP.

**HTTP Method Setter**

HTTP Method Setter thiết lập phương thức HTTP được định cấu hình cho các cuộc gọi HTTP đang diễn ra. Tất cả các loại phương thức HTTP được phép đặt bằng phần tử xử lý này.

**String Payload Setter**

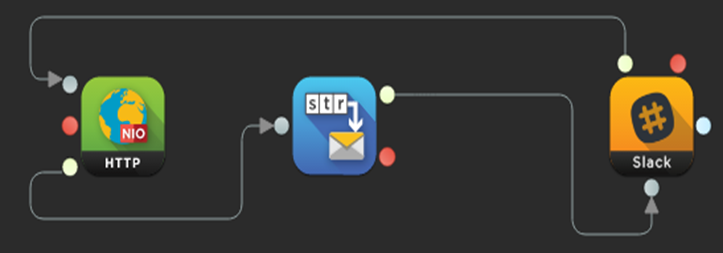
String Payload Setterxử lý có thể được sử dụng để đặt tải trọng thư từ một chuỗi được cấu hình sẵn. Chuỗi này có thể chứa nội dung động có thể xuất phát từ các thuộc tính thông báo, biến phạm vi, tiêu đề vận chuyển, id tin nhắn, dấu thời gian hiện tại và tải trọng hiện tại.

…

## **Cấu trúc hệ thống hoạt động:**

7.1 Tích hợp Slack-Trello

**Nhận thông báo từ trello webhooks và gửi tin nhắn tới slack.**



Bước đầu tiên ta cần làm là sử dụng webhooks của trello và slack.

Tạo ra webhooks url slack trỏ đến chanel mà ta muốn tích hợp. Tương tự với trello và cấu hình url nhận thông báo phù hợp

Sử dung ngrok để public lochost nhận thông báo từ trello.

Để nhận thông báo từ trello webhooks ta cần xây dựng 1 đường dẫn url để trello có thể gửi thông báo về. Ta sử dụng HTTP Ingress Connecter để lắng nghe các request. Khi HTTP Ingress Connecter nhận được thông báo từ trello ta truyền thông báo này sang String Payload Setter. String Payload Setter có nhiệm vụ xử lý chuỗi và gửi tin nhắn này sang Slack. Khi đó ta sẽ nhận được 1 tin nhắn trên chanel mà ta đã thiết lập cấu hình của slack webhooks.

**Nhận thông báo từ Slack Command và tạo thẻ trên trello.**

Áp dụng webhooks mà ta đã xây dựng ở trên. Tương tự như phần trên ta tạo ra 1 HTTP Ingress Connector để lắng nghe thông báo từ slack. Khi ta gọi đến Slack command thông báo sẽ đươc gửi về từ đó ta sẽ bổ sung các param cần thiết để khởi tạo thẻ trên trello như id, apikey, token … Sau đó ta sẽ gửi request đến trello qua HTTPS Egress để tạo ra thẻ mới theo các thông tin mà ta muốn.

