ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

****

BÁO CÁO KHÓA LUẬN TỐT NGHIỆP

Ngành: Công nghệ Thông tin

Đề tài: Xây dựng hệ thống đồng bộ nội dung sản phẩm lên các sàn Thương mại điện tử

*Giảng viên hướng dẫn*: TS. Trần Trúc Mai

|  |  |
| --- | --- |
|  | **Sinh viên**: Nguyễn Văn Tùng  **Mã sinh viên**: 16021225  **Lớp**: QH-2016-I-CQ/CC |

***Hà Nội, tháng 06 năm 2020***

Trang phụ bìa

TÓM TẮT

**LỜI CẢM ƠN**

LỜI CAM ĐOAN

MỤC LỤC

[Chương 1. Đặt vấn đề 1](#_Toc43292218)

[1.1 Thương mại và Thương mại điện tử. 1](#_Toc43292219)

[1.2 Các sàn Thương mại điện tử lớn ở Việt Nam. 2](#_Toc43292220)

[1.2.1 Shopee VN – https://shopee.vn 2](#_Toc43292221)

[1.2.2 Tiki – https://tiki.vn 3](#_Toc43292222)

[1.2.3 Lazada VN – https://lazada.vn 4](#_Toc43292223)

[1.3 Công việc của một Nhà bán hàng và bài toán đồng bộ sản phẩm. 5](#_Toc43292224)

[1.4 Syncpro – Giải pháp đồng bộ sản phẩm lên các sàn Thương mại điện tử. 6](#_Toc43292225)

[Chương 2. API các sàn thương mại điện tử hỗ trợ 8](#_Toc43292226)

[2.1 API của Shopee. 8](#_Toc43292227)

[2.1.1 Cách sử dụng API của Shopee. 8](#_Toc43292228)

[2.1.2 Một số API quan trọng. 14](#_Toc43292229)

[2.2 API của Tiki. 15](#_Toc43292230)

[2.2.1 Cách sử dụng API của Tiki. 15](#_Toc43292231)

[2.2.2 Một số API quan trọng. 17](#_Toc43292232)

[2.3 API của Lazada. 18](#_Toc43292233)

[2.3.1 Cách sử dụng API của Lazada. 18](#_Toc43292234)

[2.3.2 Người bán hàng ủy quyền 22](#_Toc43292235)

[2.3.3 Một số API quan trọng. 23](#_Toc43292236)

[Chương 3. Thiết kế giải pháp 25](#_Toc43292237)

[3.1 Ý tưởng giải quyết bài toán. 25](#_Toc43292238)

[3.1.1 Bài toán đẩy sản phẩm lên các sàn thương mại điện tử. 25](#_Toc43292239)

[3.1.2 Bài toán đồng bộ nội dung sản phẩm giữa các sàn thương mại điện tử. 44](#_Toc43292240)

[3.2 Thiết kế kiến trúc hệ thống. 49](#_Toc43292241)

[3.2.1 Database. 50](#_Toc43292242)

[3.2.2 API Service. 50](#_Toc43292243)

[3.2.3 CRON Service. 50](#_Toc43292244)

[3.2.4 Job queue. 50](#_Toc43292245)

[3.2.5 Workers. 51](#_Toc43292246)

[3.2.6 Webapp. 51](#_Toc43292247)

[Chương 4. Triển khai giải pháp 52](#_Toc43292248)

[4.1 Triển khai Database. 52](#_Toc43292249)

[4.2 Cài đặt API Service. 55](#_Toc43292250)

[4.3 Cài đặt CRON Service. 57](#_Toc43292251)

[4.4 Triển khai Job queue. 59](#_Toc43292252)

[4.5 Cài đặt Workers. 61](#_Toc43292253)

[4.6 Cài đặt Webapp. 65](#_Toc43292254)

[Chương 5. Kết quả đạt được và hướng phát triển 67](#_Toc43292255)

[5.1 Kết quả đạt được. 67](#_Toc43292256)

[5.2 Hướng phát triển. 71](#_Toc43292257)

[5.2.1 Hỗ trợ thêm sản phẩm có nhiều biến thể. 71](#_Toc43292258)

[5.2.2 Hoàn thiện tính năng xác thực người dùng. 72](#_Toc43292259)

[5.2.3 Nghiên cứu phát triển tính năng đồng bộ số lượng sản phẩm. 72](#_Toc43292260)

[5.2.4 Mở rộng ra nhiều sàn thương mại điện tử hơn. 72](#_Toc43292261)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1. Top 5 sàn thương mại điện tử tại Việt Nam quý 1 năm 2020. 2](#_Toc43292170)

[Hình 1.2. Trang chủ Shopee.vn. 3](#_Toc43292171)

[Hình 1.3. Trang chủ Tiki.vn. 4](#_Toc43292172)

[Hình 1.4. Trang chủ Lazada.vn. 5](#_Toc43292173)

[Hình 1.5. Minh họa các tính năng của Syncpro. 7](#_Toc43292174)

[Hình 2.1. 6 bước để sử dụng API của Shopee. 8](#_Toc43292175)

[Hình 2.2. Đăng ký tài khoản nhà phát triển trên Shopee. 9](#_Toc43292176)

[Hình 2.3. Các thông tin cơ bản mà Shopee cần nhà phát triển cung cấp. 10](#_Toc43292177)

[Hình 2.4. Các thông tin cần thiết để tạo APP với Shopee. 11](#_Toc43292178)

[Hình 2.5. Các bước xác thực và ủy quyền với chủ shop trên Shopee. 12](#_Toc43292179)

[Hình 2.6. Chủ shop đăng nhập để ủy quyền trên Shopee. 13](#_Toc43292180)

[Hình 2.7. Luồng làm việc giữa chủ shop và Syncpro với Tiki API key. 16](#_Toc43292181)

[Hình 2.8. Minh họa cách lấy API key của Tiki. 16](#_Toc43292182)

[Hình 2.9. Thử gọi API của Tiki với Postman. 17](#_Toc43292183)

[Hình 2.10. 5 bước để sử dụng API của Lazada. 18](#_Toc43292184)

[Hình 2.11. Đăng ký tài khoản nhà phát triển trên Lazada. 19](#_Toc43292185)

[Hình 2.12. Các quyền truy cập API của Lazada. 20](#_Toc43292186)

[Hình 2.13. Các bước xác thực và ủy quyền với chủ shop trên Lazada. 22](#_Toc43292187)

[Hình 2.14. Chủ shop đăng nhập để ủy quyền trên Lazada. 23](#_Toc43292188)

[Hình 3.1. Các bước tạo sản phẩm trên Shopee. 27](#_Toc43292189)

[Hình 3.2. Bản đồ trạng thái của sản phẩm trên Shopee. 29](#_Toc43292190)

[Hình 3.3. Các bước yêu cầu tạo sản phẩm trên Tiki. 33](#_Toc43292191)

[Hình 3.4. Bản đồ trạng thái của sản phẩm trên Tiki. 35](#_Toc43292192)

[Hình 3.5. Form sản phẩm dạng XML của Lazada. 39](#_Toc43292193)

[Hình 3.6. Các bước tạo sản phẩm trên Lazada. 40](#_Toc43292194)

[Hình 3.7. Bản đồ trạng thái của sản phẩm trên Lazada. 43](#_Toc43292195)

[Hình 3.8. Sơ đồ kiến trúc hệ thống Syncpro. 49](#_Toc43292196)

[Hình 4.1. Tổ chức dữ liệu bên trong MongoDB. 53](#_Toc43292197)

[Hình 4.2. Cách thức hoạt động của Kue - Redis. 60](#_Toc43292198)

[Hình 4.3. Giao diện của Kue. 60](#_Toc43292199)

[Hình 5.1. Người bán hàng tạo sản phẩm với Syncpro. 67](#_Toc43292200)

[Hình 5.2. Sản phẩm xuất hiện trên Seller center của Shopee. 67](#_Toc43292201)

[Hình 5.3. Sản phẩm xuất hiện trên Seller center của Tiki. 68](#_Toc43292202)

[Hình 5.4. Sản phẩm xuất hiện trên Seller center của Lazada. 68](#_Toc43292203)

[Hình 5.5. Sản phẩm hiển thị trên sàn Shopee.vn. 68](#_Toc43292204)

[Hình 5.6. Sản phẩm hiển thị trên sàn Lazada.vn. 69](#_Toc43292205)

[Hình 5.7. Người bán hàng cập nhật thông tin sản phẩm với Syncpro. 69](#_Toc43292206)

[Hình 5.8. Thông tin sản phẩm đã thay đổi trên sàn Shopee.vn. 70](#_Toc43292207)

[Hình 5.9. Thông tin sản phẩm đã thay đổi trên Seller center của Tiki. 70](#_Toc43292208)

[Hình 5.10. Thông tin sản phẩm đã thay đổi trên sàn Lazada.vn. 71](#_Toc43292209)

BẢNG TRA CỨU THUẬT NGỮ

# Đặt vấn đề

## Thương mại và Thương mại điện tử.

**Trao đổi hàng hóa** là hoạt động trao đổi của cải, dịch vụ, kiến thức, tiền tệ, ... giữa hai hay nhiều đối tác và nhận lại một giá trị tương đương bằng tiền, hoặc bằng hàng hóa khác. Trong quá trình này, người bán là người cung cấp hàng hóa, người mua là người trả một giá trị tương đương bằng tiền hoặc hàng hóa khác để có được hàng hóa mà người bán cung cấp.

**Thương mại** bao gồm hoạt động trao đổi hàng hóa và tất cả các hoạt động giúp cho việc trao đổi hàng hóa diễn ra thuận lợi, suôn sẻ từ đầu đến cuối, ví dụ như: vận chuyển, thanh toán, bảo hành, lưu trữ kho, quảng cáo, ...

Trước đây, các hoạt động thương mại chủ yếu diễn ra theo hình thức truyền thống. Với hình thức này, người bán sẽ bày bán sản phẩm của mình tại các nơi bán hàng như siêu thị, khu chợ, gian hàng, ... Người mua sẽ phải tự mình đến các nơi bán để mua hàng. Do đó, người mua sẽ gặp gỡ trực tiếp với người bán, được thử trực tiếp sản phẩm, thanh toán trực tiếp bằng tiền (tiền mặt hoặc tiền ngân hàng) và ngay lập tức có được sản phẩm sau khi mua.

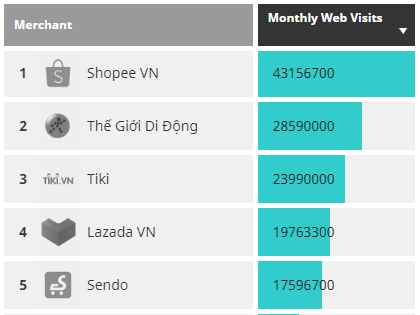
Nhiều năm trở lại đây, một hình thức thương mại mới xuất hiện và ngày càng nở rộ: **Thương mại điện tử**. Với hình thức này, người bán sẽ bày bán sản phẩm của mình tại các gian hàng ảo trên Internet (thường là các trang web bán hàng). Người mua sẽ tìm đến các gian hàng ảo tùy theo nhu cầu, chọn sản phẩm muốn mua trên đó, thanh toán và chờ hàng được giao đến tay. Người mua và người bán sẽ không gặp gỡ trực tiếp mà chỉ trao đổi qua Internet. Người mua không được thử sản phẩm vật lý. Hoạt động thanh toán cũng không diễn ra theo cách người mua chuyển tiền trực tiếp cho người bán mà sẽ do bên thứ ba đứng ra đảm bảo, ví dụ như các cổng thanh toán điện tử, các đơn vị giao hàng. Sau khi mua hàng, người mua sẽ phải chờ một khoảng thời gian để có được sản phẩm.

So với loại hình thương mại truyền thống, thương mại điện tử mang đến nhiều lợi ích to lớn. Lợi ích lớn nhất mà nó đem lại chính là sự tiết kiệm cả về chi phí, thời gian, công sức và tạo thuận lợi cho các bên giao dịch. Giao dịch bằng phương tiện điện tử nhanh hơn và rẻ hơn so với giao dịch truyền thống. Các bên có thể tiến hành giao dịch khi ở cách xa nhau, giữa thành phố với nông thôn, từ nước này sang nước kia, hay nói cách khác là không bị giới hạn bởi không gian địa lý. Với người bán, họ tiết kiệm được chi phí thuê gian hàng vật lý, chi phí thuê nhân lực bán hàng, mở rộng phạm vi tiếp cận người mua, tiết kiệm thời gian gặp mặt khi giao dịch... Với người mua, họ có thể ngồi tại nhà để đặt hàng, mua sắm nhiều loại hàng hóa, dịch vụ nhanh chóng và dễ dàng. Vì những thuận lợi mà thương mại điện tử tạo ra, nó cũng gián tiếp thúc đẩy sự cạnh tranh giữa các nhà bán hàng. Người mua thì ngày càng có thêm nhiều lựa chọn mua hàng với giá cả hợp lý hơn. Kinh tế nhờ đó cũng phát triển hơn.

Bất chấp những định kiến rằng độ tin cậy và an toàn thấp, tiềm ẩn nhiều rủi ro, lừa gạt, thương mại điện tử vẫn phát triển từng ngày và dần trở thành xu thế nổi bật, song hành cùng với hình thức thương mại truyền thống.

## Các sàn Thương mại điện tử lớn ở Việt Nam.

Ba sàn thương mại điện tử lớn ở Việt Nam phải kể đến là Shopee VN, Tiki, Lazada VN. Theo thống kê từ iPrice, vào quý 1 năm 2020, ba sàn này đứng trong top 5 sàn thương mại điện tử ở Việt Nam có nhiều lượt truy cập nhất (cùng với Thế giới di động và Sendo).



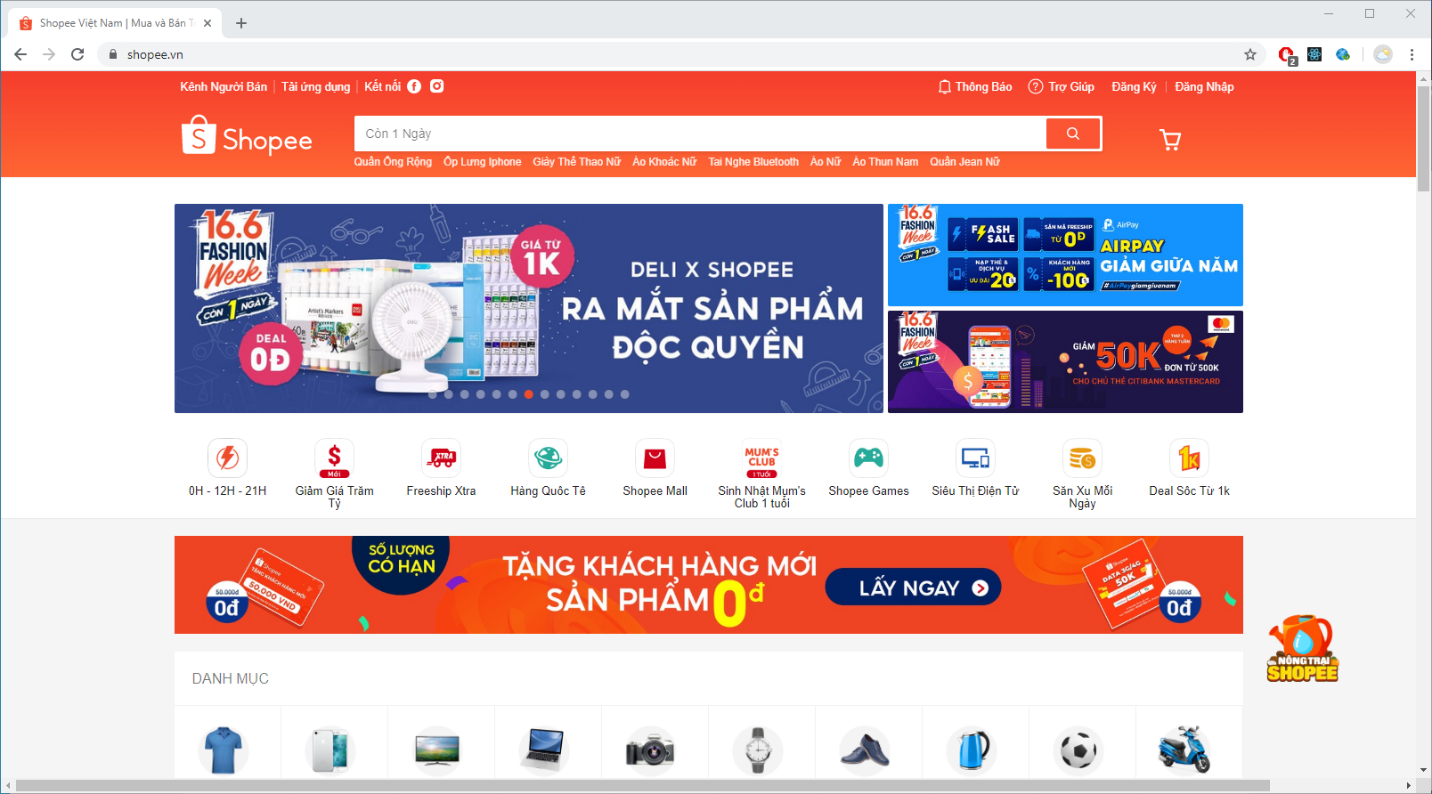
Hình .. Top 5 sàn thương mại điện tử tại Việt Nam quý 1 năm 2020.

### Shopee VN – <https://shopee.vn>

**Shopee** là nền tảng thương mại điện tử hàng đầu tại khu vực Đông Nam Á có trụ sở tại Singapore và trực thuộc công ty Sea (trước đây là Garena - chủ sở hữu các thương hiệu như: Garena , Foody, Now, Airpay) ra đời từ năm 2015 và tại thời điểm hiện tại đã có mặt trên tổng cộng 7 nước khu vực châu Á gồm: Singapore; Malaysia; Thái Lan; Đài Loan; Indonesia; Việt Nam, và Philippines.

Shopee ra đời nhằm tạo ra một sàn thương mại điện tử để cung cấp cho khách hàng trải nghiệm việc mua sắm trực tuyến một cách dễ dàng, an toàn và tiện lợi bởi quá trình thanh toán và vận chuyển nhanh chóng.

Bất kỳ ai cũng có thể tạo tài khoản và bán hàng trên Shopee mà không cần giấy phép kinh doanh. Bởi vậy các mặt hàng trên Shopee hết sức đa dạng và phong phú: từ Điện tử, Tiêu dùng đến Nhà cửa & Đời sống, Sức khỏe và sắc đẹp, Đồ chơi trẻ em, Thời trang và Thiết bị thể thao, ...



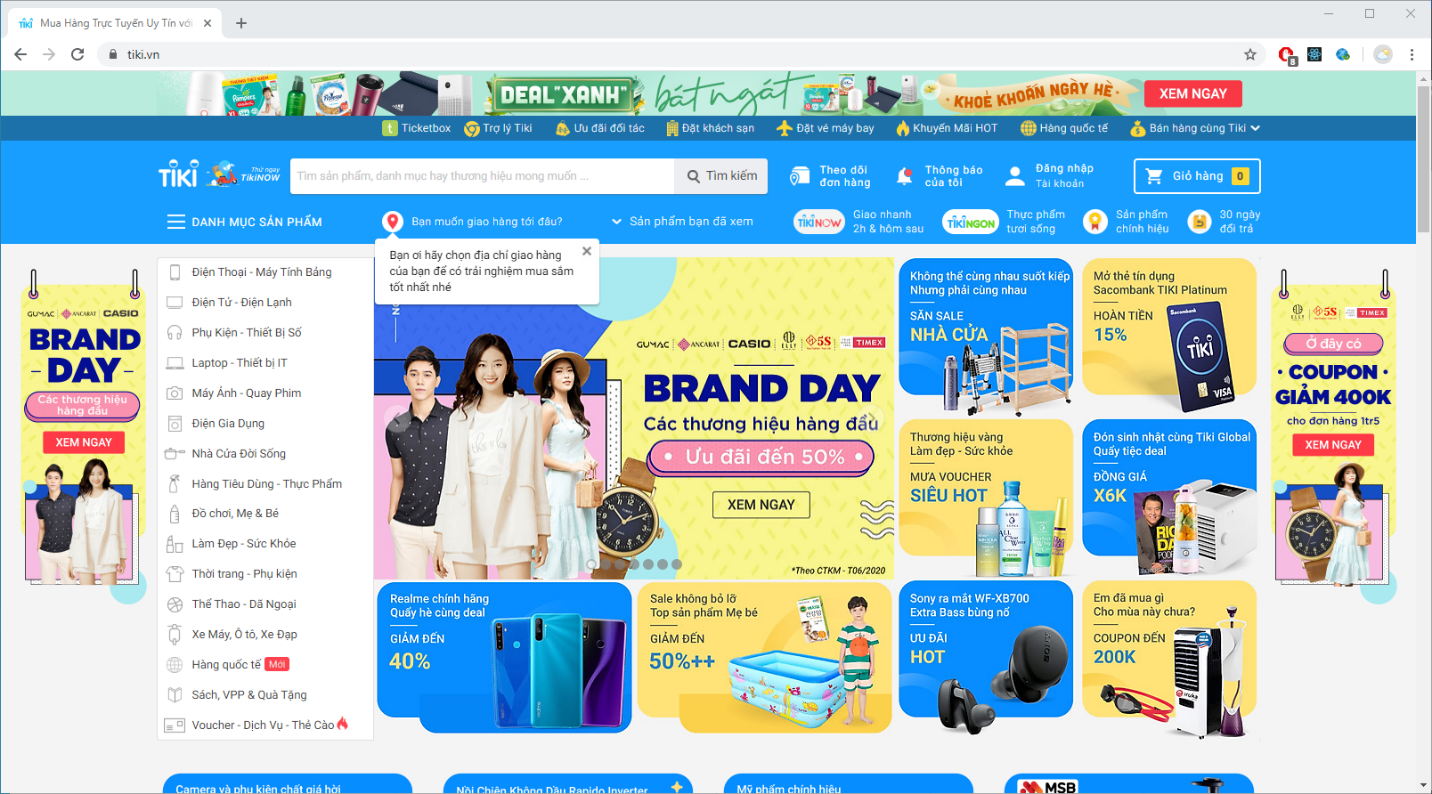
Hình .. Trang chủ Shopee.vn.

### Tiki – <https://tiki.vn>

**Tiki** là viết tắt của “Tìm kiếm & Tiết kiệm”. Hiện nay Tiki là trang mua sắm trực tuyến đáng tin cậy nhất tại thị trường Việt Nam. Người Việt quá quen với sàn thương mại điện tử này bởi cái tên Tiki xuất hiện trong rất nhiều MV của nhiều ca sĩ nổi tiếng.

Tiki được sáng lập bởi ông Trần Ngọc Thái Sơn vào tháng 03/2010. Tháng 3/2012, Chủ tịch kiêm CEO của Quỹ đầu tư CyberAgent Ventures Inc là ông Soichi Tajima đã quyết định đầu tư vào Tiki. Với việc góp vốn này, 20% cổ phẩn của công ty được nắm giữ bởi CyberAgent Ventures Inc. Vào năm 2016, Tiki.vn được đầu tư số tiền 384 tỷ đồng bởi Công ty Cổ phần VNG. Số tiền này tương ứng với 38% cổ phần. Năm 2018, Tiki tiếp tục được đầu tư khoảng 1000 tỷ đồng bởi nhà bán lẻ lớn nhất Trung Quốc là JD.com.

Ban đầu, Tiki chỉ là website trực tuyến bán sách tiếng Anh. Đến tháng 8/2012, Tiki.vn là nhà sách trực tuyến số 1 trên thị trường Việt Nam. Trải qua 10 năm phát triển, hiện nay, ngoài bán sách, Tiki.vn còn bán hầu hết các sản phẩm thuộc 9 ngành hàng khác. Bao gồm: Điện Thoại – Máy Tính Bảng, Thiết Bị Số – Phụ Kiện Số, Điện Gia Dụng, Nhà Cửa Đời Sống, Làm Đẹp – Sức Khỏe, Thiết Bị Văn Phòng Phẩm, Đồ Chơi – Đồ Lưu Niệm, Mẹ & Bé, Thể thao.



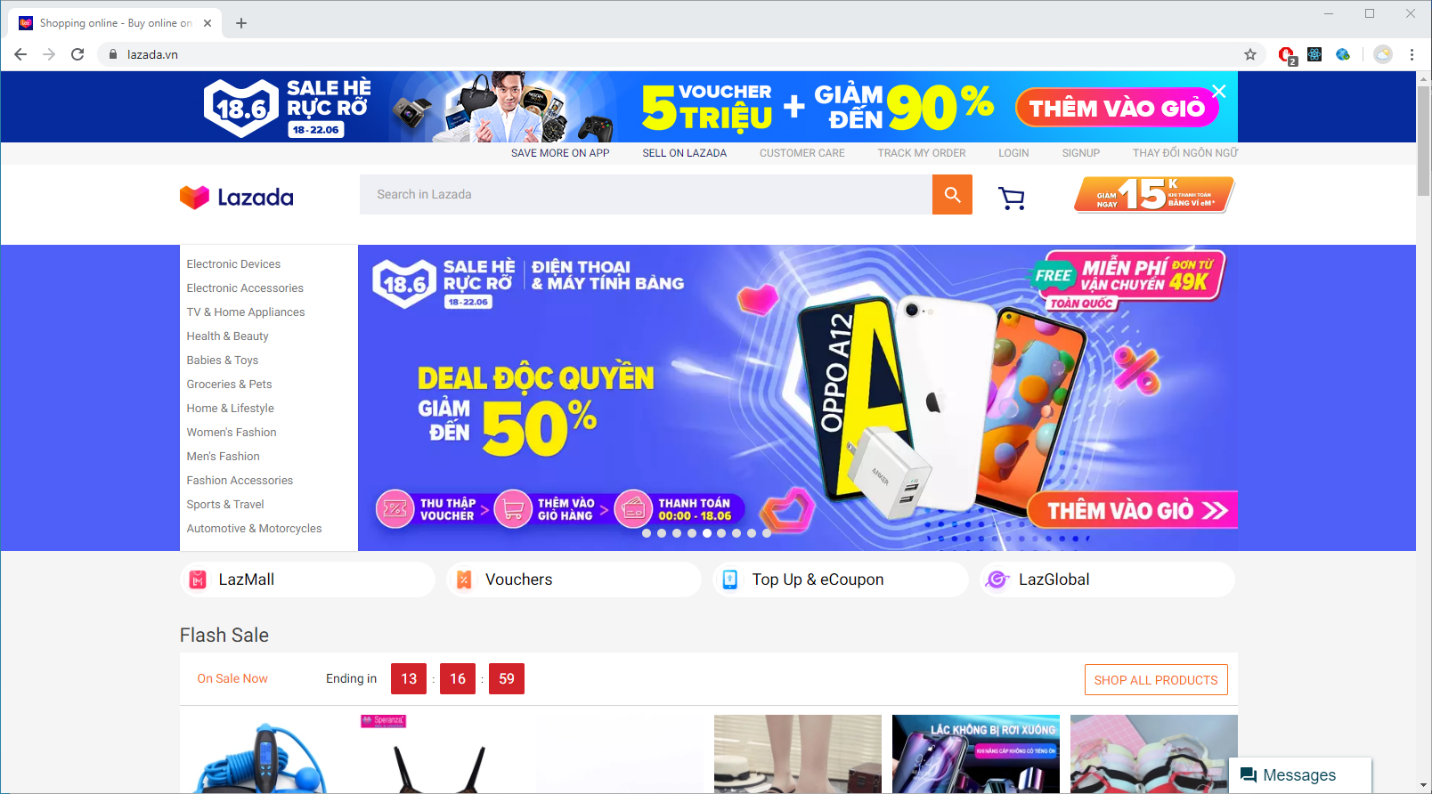
Hình .. Trang chủ Tiki.vn.

### Lazada VN – <https://lazada.vn>

Lazada Group là một công ty thương mại điện tử tư nhân của Đức được thành lập bởi Rocket Internet vào năm 2011. Tính đến 2014, Lazada Group đã hoạt động tại Indonesia, Malaysia, Philippines, Singapore, Thái Lan và Việt Nam. Năm 2015, Lazada Group được tập đoàn Alibaba của tỷ phú Trung Quốc Jack Ma mua lại.

Lazada Việt Nam là một phần của Lazada Group được ra đời vào tháng 3 năm 2012, và cho đến thời điểm hiện tại thì Lazada tại Việt Nam đang là một trong những sàn giao dịch thương mại điện tử lớn nhất.

Lazada ngoài việc cung cấp các sản phẩm, dịch vụ thì đa phần các sản phẩm trên lazada đều là của các cá nhân, doanh nghiệp, công ty đăng bán. Sản phẩm trên Lazada rất đa dạng và phong phú. Hầu hết tất cả các mặt hàng có khả năng giao được đều có mặt trên đây như: Nội thất, Điện thoại - máy tính bảng, Thời trang và phụ kiện, Sản phẩm chăm sóc sức khỏe, Làm đẹp, Đồ chơi và Đồ dùng thể thao, ...



Hình .. Trang chủ Lazada.vn.

## Công việc của một Nhà bán hàng và bài toán đồng bộ sản phẩm.

Đối với nhà bán hàng, mục tiêu hàng đầu là bán được sản phẩm. Vì vậy, nhà bán hàng thường sẽ đăng thông tin sản phẩm của mình lên nhiều sàn thương mại điện tử để tiếp cận được nhiều khách hàng hơn. Giả sử muốn bán một chiếc Iphone X, nhà bán hàng sẽ lên trang người bán của Lazada, điền thông tin về chiếc Iphone X, phát hành sản phẩm. Sau đó nhà bán hàng lại tiếp tục lên trang người bán của Tiki, điền thông tin chiếc Iphone X, phát hành sản phẩm. Sau đó, nhà bán hàng lại tiếp tục lên trang người bán của Shopee, điền thông tin chiếc Iphone X, phát hành sản phẩm. Thao tác cứ lặp đi lặp lại như vậy cho nhiều sàn thương mại điện tử khác nhau.

Nếu chỉ có từ 5 đến 10 sản phẩm thì mọi chuyện vẫn ổn. Nhưng một nhà bán hàng thường sẽ có cả trăm đến cả nghìn sản phẩm. Thời gian và công sức bỏ ra để đăng thông tin của tất cả sản phẩm lên các sàn thương mại điện tử khác nhau là không hề nhỏ.

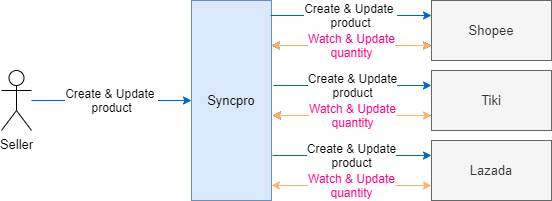
Hơn thế, không chỉ phải đăng sản phẩm lên các sàn thương mại điện tử, nhà bán hàng còn phải đồng bộ số lượng sản phẩm giữa các sàn với nhau để đảm bảo số sản phẩm sẵn có trên các sàn luôn khớp với số lượng hàng còn trong kho, tránh tình trạng có đơn đặt hàng nhưng số hàng trong kho không đủ, phải hủy đơn hàng. Việc đồng bộ số lượng cũng mất nhiều thời gian và công sức không kém. Nhà bán hàng mỗi khi bán được sản phẩm thì phải kiểm tra xem trong kho còn bao nhiêu sản phẩm nữa, sau đó cập nhật số lượng này lên các sàn.

Giả sử, nhà bán hàng có 10 chiếc Iphone X, bán trên cả Tiki và Lazada. Cùng lúc trên Tiki có đơn mua 1 chiếc, trên Lazada cũng có đơn mua 1 chiếc, trong kho lúc này sẽ chỉ còn lại 8 chiếc Iphone, trong khi trên Tiki và Lazada đang hiển thị còn lại 9 chiếc. Nhà bán hàng sẽ phải cập nhật số lượng 8 chiếc lên cả Tiki và Lazada.

## Syncpro – Giải pháp đồng bộ sản phẩm lên các sàn Thương mại điện tử.

Ý tưởng xây dựng ứng dụng Syncpro bắt nguồn từ mong muốn giải quyết bài toán đồng bộ sản phẩm, giúp cho nhà bán hàng giảm bớt thời gian và công sức để đăng tải và đồng bộ nội dung sản phẩm lên nhiều sàn thương mại điện tử. Ứng dụng cũng cung cấp một nơi để nhà bán hàng quản lý tập trung các sản phẩm được phát hành.

Sử dụng Syncpro, nhà bán hàng chỉ cần điền thông tin sản phẩm ở một nơi duy nhất, bấm phát hành là sản phẩm sẽ được tự động đăng lên các sản thương mại điện tử. Khi số lượng sản phẩm còn lại ở một sàn thay đổi, số lượng còn lại thực tế sẽ được tính toán và tự động đồng bộ lên tất cả các sàn. Nhà bán hàng muốn thay đổi nội dung sản phẩm cũng chỉ cần thay đổi nội dung ở một nơi duy nhất, bấm cập nhật là những thay đổi về sản phẩm sẽ được cập nhật cho tất cả.



Hình .. Minh họa các tính năng của Syncpro.

# API các sàn thương mại điện tử hỗ trợ

Để thực hiện được mục tiêu đăng tải và cập nhật nội dung sản phẩm lên các sàn thương mại điện tử, Syncpro sử dụng **API** được các sàn hỗ trợ. Cả ba sàn thương mại điện tử Shopee, Tiki, Lazada đều có hỗ trợ đầy đủ API.

## API của Shopee.

Shopee cung cấp một bộ API đầy đủ cho bất kì nhà phát triển nào muốn xây dựng một nền tảng tích hợp với Shopee. Với bộ API này, nhà phát triển có thể: quản lý sản phẩm, quản lý đơn hàng, quản lý các tài khoản chủ shop được liên kết, ... Trang tài liệu API của Shopee nằm ở địa chỉ <https://open.shopee.com/documents>.

### Cách sử dụng API của Shopee.

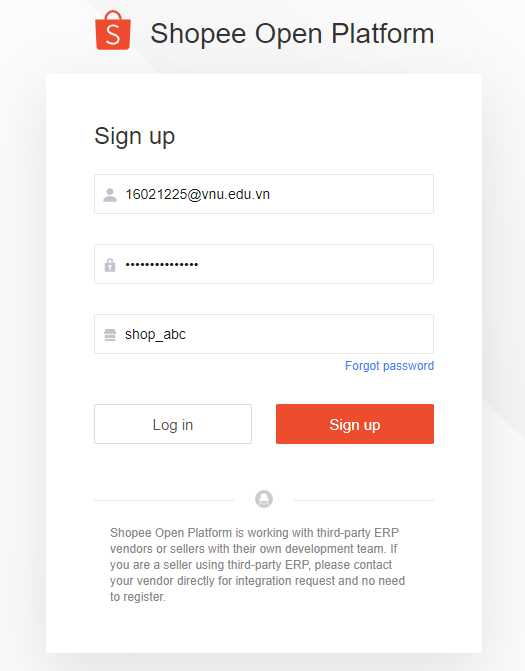
Để sử dụng được bộ API, nhà phát triển phải trải qua 6 bước, được ghi rõ trong tài liệu của Shopee:



Hình .. 6 bước để sử dụng API của Shopee.

***Bước 1: Đăng ký làm nhà phát triển.***

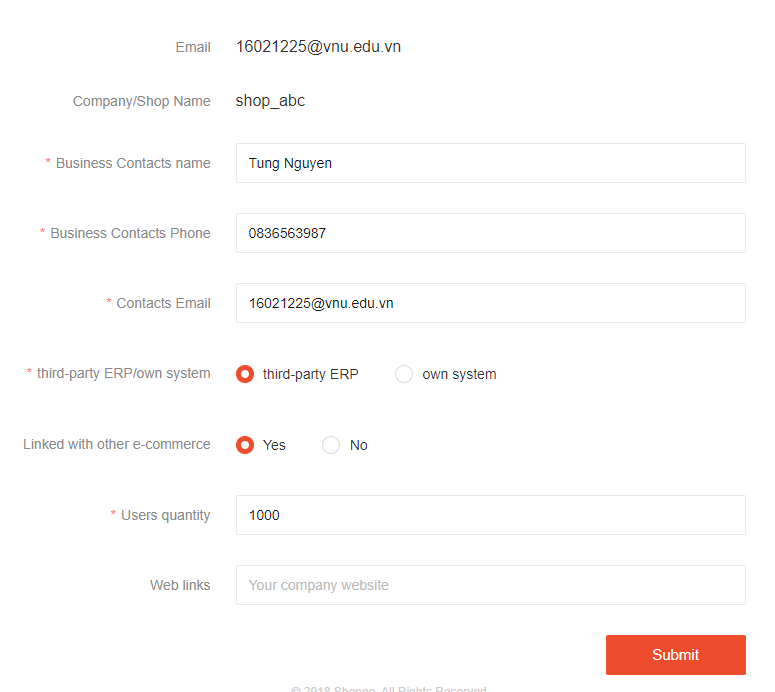
Nhà phát triển truy cập vào trang <https://open.shopee.com/signup> để đăng ký tài khoản. Nhà phát triển phải cung cấp các thông tin: Email, Mật khẩu, Tên công ty/Tên cửa hàng.



Hình .. Đăng ký tài khoản nhà phát triển trên Shopee.

***Bước 2: Cung cấp các thông tin cơ bản.***

Sau khi đăng ký thành công, nhà phát triển phải điền các thông tin cơ bản, bao gồm: Tên, Số điện thoại, Loại hệ thống, Lượng người dùng, ...

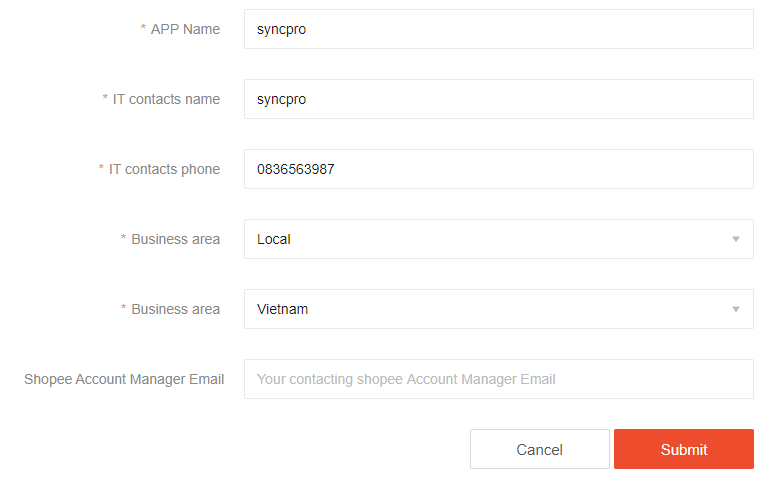


Hình .. Các thông tin cơ bản mà Shopee cần nhà phát triển cung cấp.

Chú ý với Loại hệ thống, chọn third-party ERP.

***Bước 3. Tạo APP.***

Muốn sử dụng API của Shopee, nhà phát triển phải tạo một APP. Mỗi APP đều có định nghĩa rõ ràng về quyền hạn và phạm vi hoạt động của nó. Để tạo APP, nhà phát triển phải cung cấp một số thông tin, trong đó có một thông tin quan trọng là Phạm vi hoạt động. Với Syncpro, phạm vi hoạt động giới hạn trong Việt Nam.



Hình .. Các thông tin cần thiết để tạo APP với Shopee.

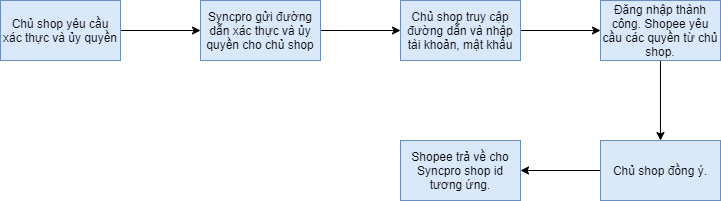
***Bước 4. Lấy Partner ID và Key.***

Sau khi APP được hệ thống Shopee kiểm duyệt và chấp thuận (mất khá nhiều thời gian), Shopee sẽ cung cấp cho APP đó một Partner ID và một Key để định danh. Nhà phát triển vào xem thông in của APP, mục Overview để lấy hai thông tin này.

Partner ID và Key rất quan trọng, vì mọi API của Shopee đều cần chúng như là bắt buộc. Không có Partner ID và Key thì không thể gọi API.

***Bước 5. Chủ shop xác thực và ủy quyền.***

Syncpro sử dụng quyền của các chủ shop để đăng tải sản phẩm, lấy thông tin sản phẩm, ..., vì vậy cần phải được các chủ shop ủy quyền. Cơ chế ủy quyền trên Shopee diễn ra như sau:



Hình .. Các bước xác thực và ủy quyền với chủ shop trên Shopee.

Đường dẫn xác thực và ủy quyền có dạng:

[https://partner.shopeemobile.com/api/v1/shop/auth\_partner?**id**=xxxxxx&**token**=xxxxxxxxxxxx&**redirect**=xxxxxx.com](https://partner.shopeemobile.com/api/v1/shop/auth_partner?id=xxxxxx&token=xxxxxxxxxxxx&redirect=xxxxxx.com)

Trong đó:

* **id** là Partner ID mà Shopee cung cấp cho APP Syncpro.
* **token** là một chuỗi đặc biệt được sinh ra bằng quy tắc của Shopee (sẽ được đề cập dưới đây).
* **redirect** là địa chỉ mà Shopee sẽ chuyển hướng đến sau khi quá trình xác thực và ủy quyền thành công.

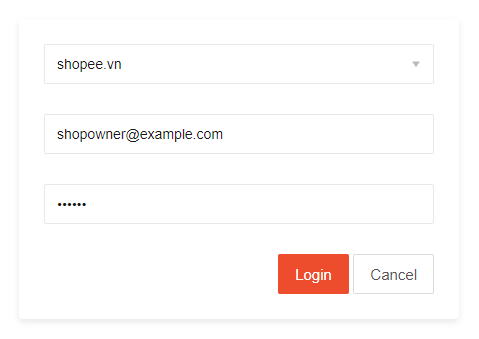
Các bước để sinh ra token hợp lệ như sau:

1. Đem Key mà Shopee cung cấp cho APP Syncpro nối với địa chỉ redirect để được một chuỗi duy nhất. VD:

9b754aba01a5d719cc70c57782941ae6ff90fcc687282908ee480a364901d181https://www.baidu.com

1. Băm chuỗi thu được với thuật toán **SHA256** (không phải HMAC-SHA256).
2. Encode kết quả băm dưới dạng hex để thu được token hoàn chỉnh.

Chủ shop bấm vào đường dẫn sẽ được đưa đến một trang đăng nhập có địa chỉ dạng <https://open.shopee.com/authorize?auth_shop=true&id=xxxxxx&random=xxxxxxxxxx>.



Hình .. Chủ shop đăng nhập để ủy quyền trên Shopee.

Tại đây, chủ shop nhập tài khoản bán hàng trên Shopee của mình vào, sau đó đăng nhập. Đăng nhập thành công, Shopee sẽ hiển thị một giao diện yêu cầu chủ shop xác nhận đồng ý cấp các quyền cho Syncpro. Nếu chủ shop đồng ý, Shopee sẽ chuyển hướng người dùng về địa chỉ redirect và Syncpro sẽ nhận được shop id của chủ shop. Shop id này có tác dụng như một con dấu xác nhận Syncpro có đủ tư cách truy cập vào dữ liệu của chủ shop.

***Bước 6. Gọi API và sẵn sàng triển khai.***

Khi đã có đủ **Partner ID**, **Key** và **shop id**, nhà phát triển có thể sử dụng tất cả các API mà Shopee hỗ trợ. Tuy nhiên, vẫn còn một số quy tắc khi gọi API Shopee đề ra mà nhà phát triển cần lưu ý:

1. Tất cả các API của Shopee đều phải được gọi bằng phương thức HTTP POST.
2. Tất cả các API khi gọi nên có HTTP header

 Content-Type: application/json.

1. HTTP body phải ở dạng serialized JSON. Ba trường sau luôn phải có trong mọi request:

partner\_id: Chính là Partner ID mà Shopee cung cấp.

shopid: Chính là shop id có được nhờ chủ shop ủy quyền. Một APP được nhiều chủ shop ủy quyền thì sẽ có nhiều shop id, vì vậy phải chắc chắn API đang gọi sử dụng đúng shop id.

timestamp: Thời điểm hiện tại dưới dạng UNIX timestamp (số giây từ ngày 01/01/1970 đến nay).

1. Mọi HTTP request đều phải được xác thực với một chuỗi xác thực đặt trong HTTP header Authorization. Chuỗi xác thực này được tính theo quy tắc của Shopee.

Chuỗi xác thực là chữ ký thể hiện HTTP request đến Shopee là hợp lệ. Cách tính chuỗi xác thực của Shopee như sau:

1. Lấy base URL của request (bao gồm phương thức https:// hay http:// cùng với địa chỉ của request, không có query hay bookmark) và body của request là dạng JSON nhưng đã được serialize thành chuỗi, đem nối với nhau tạo thành một chuỗi duy nhất, với dấu phân cách là |. VD:

https://partner.shopeemobile.com/api/v1/orders/detail|{"ordersn":"160726152598865","shopid":61299,"partner\_id":1,"timestamp":1470198856}

1. Băm chuỗi thu được với thuật toán **HMAC-SHA256**, key sử dụng để băm chính là Key mà Shopee cung cấp khi tạo APP.
2. Encode kết quả băm dưới dạng hex để thu được chuỗi xác thực hoàn chỉnh.

Với cách tính này, mỗi request sẽ có một chuỗi xác thực khác nhau. Việc cài đặt chính xác cách tính của Shopee rất quan trọng, vì nếu không có chuỗi xác thực đúng thì sẽ không gọi được API.

### Một số API quan trọng.

Syncpro chú trọng vào các API thuộc mục quản lý sản phẩm của Shopee. Dưới đây chỉ liệt kê tên và chức năng của một số API quan trọng. Cách gọi và định dạng dữ liệu của mỗi API đã được viết rất rõ trong tài liệu của Shopee.

**GetCategoriesByCountry**: Đây là API cho phép lấy tất cả các Danh mục ngành hàng hiện có trên Shopee theo từng quốc gia. Từ danh sách danh mục này có thể xây dựng được một cây danh mục hoàn chỉnh. Trong cây danh mục, các lá là các Danh mục ngành hàng sơ cấp (Primary Category). Vì sản phẩm luôn phải được gắn với một Danh mục ngành hàng sơ cấp nên việc lấy tất cả danh mục ngành hàng là rất cần thiết.

**GetAttributes**: Trên Shopee, mỗi Danh mục ngành hàng sơ cấp sẽ có một số thuộc tính đặc biệt tương ứng với nó. Các thuộc tính đặc biệt này là cần thiết để tạo sản phẩm. Ví dụ: sản phẩm thuộc danh mục Laptop sẽ có thông tin về CPU, RAM, Thương hiệu, v.v... API này chính là để lấy về các thuộc tính đặc biệt đó.

**Add**: Đây là API tạo sản phẩm trên Shopee.

**Delete**: Đây là API xóa sản phẩm trên Shopee.

**UnlistItem**: Đây là API ẩn/hiển thị sản phẩm trong danh sách các sản phẩm đang được bán.

**GetItemDetail**: Đây là API lấy về tất cả thông tin chi tiết của sản phẩm.

**UpdateItem**, **UpdateItemImg**, **UpdatePrice**, **UpdateStock**: Chức năng của 4 API này theo thứ tự là cập nhật nội dung, cập nhật ảnh, cập nhật giá, cập nhật số lượng. Cả 4 API này mới tạo nên chức năng cập nhật sản phẩm hoàn chỉnh.

## API của Tiki.

Dù là nền tảng thương mại điện tử của Việt Nam nhưng Tiki cũng thể hiện là một nền tảng hết sức chuyên nghiệp khi hỗ trợ đầy đủ API cùng với một trang tài liệu Tiếng Anh chi tiết. Với API của Tiki, nhà phát triển cũng có thể quản lý sản phẩm, quản lý đơn hàng, ... giống như Shopee. Tài liệu API của Tiki nằm tại địa chỉ <https://open.tiki.vn>.

### Cách sử dụng API của Tiki.

Tiki không yêu cầu nhà phát triển phải đăng ký APP mà sử dụng hình thức API key làm cơ chế xác thực. Với cơ chế này, mỗi chủ shop sau khi đăng ký bán hàng thành công trên Seller center của Tiki sẽ được cấp một chuỗi gọi là “tham số kết nối” – bản chất là API key. Chủ shop nếu muốn sử dụng dịch vụ của nhà phát triển thì sẽ phải đưa “tham số kết nối” này cho phía nhà phát triển. Với “tham số kết nối” mà chủ shop đưa, nhà phát triển có đầy đủ quyền truy cập vào dữ liệu sản phẩm, đơn hàng, ... của chủ shop trên Tiki.

Cơ chế xác thực bằng API key tuy đơn giản nhưng bộc lộ nhiều khuyết điểm, ví dụ như tính bảo mật thấp (vì API key dễ lộ), cứng nhắc trong việc phân quyền, thời gian tồn tại lâu, ... Hiện tại, Tiki đang dần chuyển sang mô hình Oauth2 (mô hình xác thực và ủy quyền giống như Shopee) và dự kiến đến cuối năm 2020 sẽ không hỗ trợ hình thức API key đã cũ.

Tuy nhiên, hiện tại Syncpro vẫn đang sử dụng hình thức xác thực bằng API key. Vì vậy, tài liệu này sẽ chỉ mô tả chi tiết về cách thức sử dụng Tiki API với API key mà không mô tả về cách thức sử dụng với Oauth2.

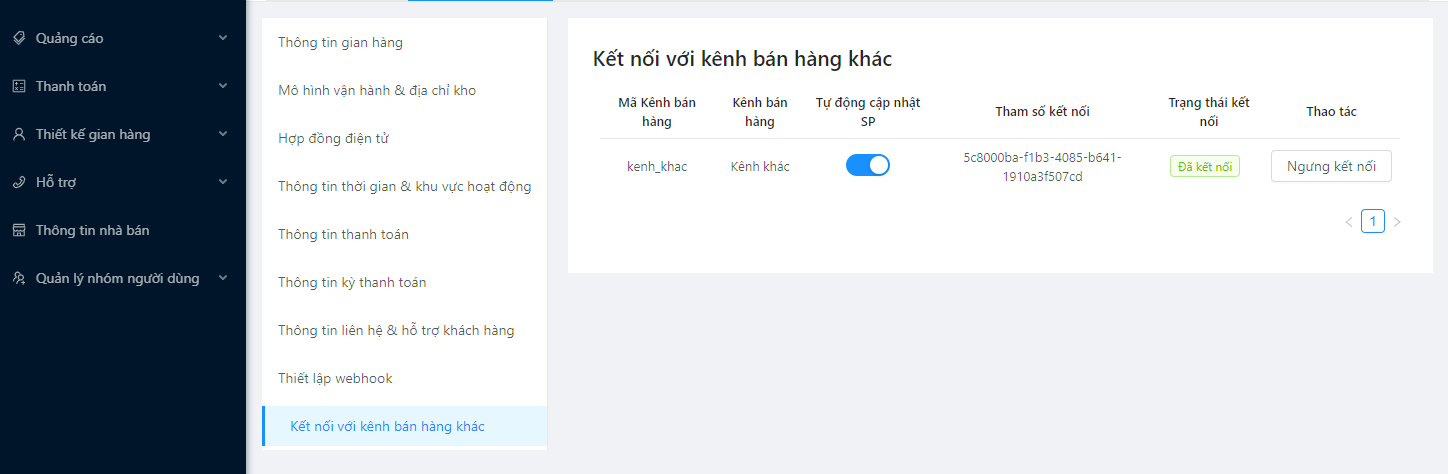
Với hình thức xác thực bằng API key, việc bắt tay giữa chủ shop và Syncpro diễn ra hết sức đơn giản:



Hình .. Luồng làm việc giữa chủ shop và Syncpro với Tiki API key.

***Lấy Tiki API key***

Sau khi đăng ký thành công tài khoản bán hàng trên Tiki Seller center, chủ shop sẽ được Tiki cấp cho API key. Để lấy API key này, chủ shop vào mục **Thông tin nhà bán → Kết nối với kênh bán hàng khác**,nhìn vào cột **Tham số kết nối** để thấy API key.



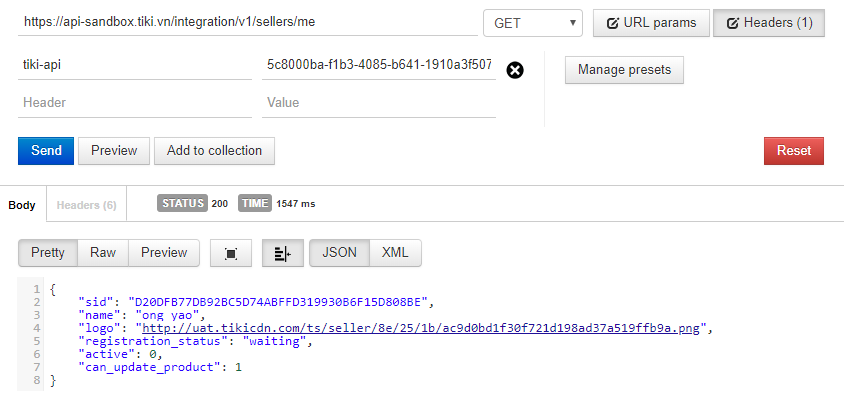
Hình .. Minh họa cách lấy API key của Tiki.

***Ủy quyền cho nhà phát triển.***

Muốn gọi được các API truy cập vào dữ liệu của chủ shop, nhà phát triển phải được chủ shop ủy quyền. Bằng cách này hay cách khác, nhà phát triển phải xin được API key từ phía chủ shop.

***Gọi API và sẵn sàng triển khai.***

Sau khi có được sự đồng ý và API key từ phía chủ shop, nhà phát triển có thể sử dụng tất cả các API mà Tiki hỗ trợ. Việc gọi API rất đơn giản: chỉ cần đưa API key vào trong HTTP header tiki-api. Tiki không yêu cầu tính toán chuỗi xác thực phức tạp như Shopee. Do đó, việc gọi thử API có thể dễ dàng tiến hành trên Postman:



Hình .. Thử gọi API của Tiki với Postman.

### Một số API quan trọng.

Dưới đây chỉ liệt kê tên và chức năng của một số API quan trọng. Cách gọi và định dạng dữ liệu của mỗi API đã được viết rất rõ trong tài liệu của Tiki.

**Get categories**: Đây là API lấy về tất cả các danh mục ngành hàng hiện có trên Tiki. Từ danh sách danh mục này có thể xây dựng được một cây danh mục hoàn chỉnh. Trong cây danh mục, các lá là các Danh mục ngành hàng sơ cấp (Primary Category). Sản phẩm luôn phải được gắn với một Danh mục ngành hàng sơ cấp.

**Get category detail**: Trên Tiki, mỗi Danh mục ngành hàng sơ cấp sẽ có một số thuộc tính đặc biệt tương ứng với nó. Các thuộc tính đặc biệt này là cần thiết để tạo sản phẩm. API này lấy về thông tin chi tiết của Danh mục ngành hàng sơ cấp và các thuộc tính đặc biệt tương ứng với nó.

**Create Product Request**: API này dùng để tạo ra một yêu cầu tạo sản phẩm. Tiki định nghĩa rằng nếu người bán muốn tạo một sản phẩm trên hệ thống, cái mà họ tạo ra không phải là sản phẩm mà là một yêu cầu tạo sản phẩm. Yêu cầu đó lần lượt trải qua các bước kiểm duyệt của Tiki. Nếu yêu cầu vượt qua được tất cả các bước kiểm duyệt thì Tiki mới tạo sản phẩm trên hệ thống cho họ.

**Tracking a product request**: API này dùng để theo dõi yêu cầu tạo sản phẩm, nhờ đó có thể biết được trạng thái của yêu cầu tạo sản phẩm: đang kiểm duyệt, đã chấp thuận hay đã loại bỏ.

**Update variant price/quantity/active**: Cập nhật giá, số lượng, trạng thái hiển thị của sản phẩm ngay tức thì. Trạng thái hiển thị quyết định sản phẩm có được hiển thị trên danh sách các mặt hàng được bán hay không.

**Update product market\_price/image/images**: Cập nhật giá thị trường (giá gốc khi chưa có giảm giá), hình đại diện, danh sách hình ảnh của sản phẩm. Các cập nhật này cần được Tiki phê duyệt.

**Get a product**: API lấy về tất cả thông tin chi tiết của sản phẩm.

**Delete a product request**: API xóa một yêu cầu tạo sản phẩm khỏi hệ thống của Tiki.

## API của Lazada.

Lazada là một trong những “ông trùm” thương mại điện tử ở Đông Nam Á nên việc có hỗ trợ API là điều tất yếu. Bộ API của Lazada rất đầy đủ, có cả tài liệu tham khảo và hướng dẫn chi tiết. Với API của Lazada, nhà phát triển có thể: quản lý sản phẩm, quản lý đơn hàng, quản lý mã giảm giá, ... Trang hướng dẫn và tài liệu API của Lazada nằm ở địa chỉ: <https://open.lazada.com>.

### Cách sử dụng API của Lazada.

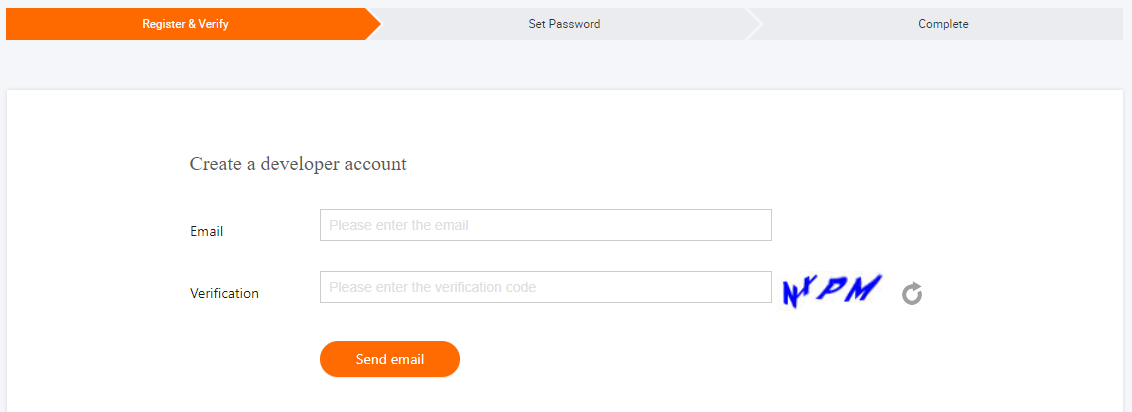
Để sử dụng API, nhà phát triển phải thực hiện 5 bước theo hướng dẫn của Lazada:



Hình .. 5 bước để sử dụng API của Lazada.

***Bước 1. Đăng ký làm nhà phát triển.***

Để đăng ký, nhà phát triển vào trang <https://uac.lazada.com/register> và điền các thông tin cần thiết theo các bước của form đăng ký. Thông tin cần thiết gồm có Email và Mật khẩu.



Hình .. Đăng ký tài khoản nhà phát triển trên Lazada.

Sau khi đăng ký thành công, nhà phát triển phải vào <https://uac.lazada.com/profile> để cập nhật các thông tin cá nhân của mình. Lazada yêu cầu các thông tin: Tên nhà phát triển, Loại hình (Cá nhân, Công ty), Địa chỉ, Chứng minh nhân dân hoặc căn cước, Có là nhà bán hàng trên Lazada hay không. Chú ý với thông tin Có là nhà bán hàng trên Lazada hay không, phải chọn là Không. Lazada cần 2 đến 3 ngày để kiểm duyệt thông tin cá nhân của nhà phát triển.

***Bước 2. Đăng ký Lazada application.***

Sau khi thông tin cá nhân đã được kiểm duyệt, nhà phát triển cần tạo Lazada application (giống APP bên Shopee). Tại trang <https://open.lazada.com> nhà phát triển chọn **App Console** → **Create** và điền vào các thông tin để tạo application. Chú ý với thông tin App Category cần chọn ERP system, và chỉ những nhà phát triển không phải là nhà bán hàng mới có thể chọn. Một thông tin cần lưu ý nữa là địa chỉ Callback URL của app. Địa chỉ này có vai trò quan trọng trong việc xác thực và ủy quyền từ phía chủ shop.

Application sau khi tạo ra sẽ ở trạng thái **pending**, chờ quản trị viên của Lazada xác minh. Application đã được xác minh sẽ chuyển sang trạng thái **active**.

***Bước 3. Lấy App Key và App Secret.***

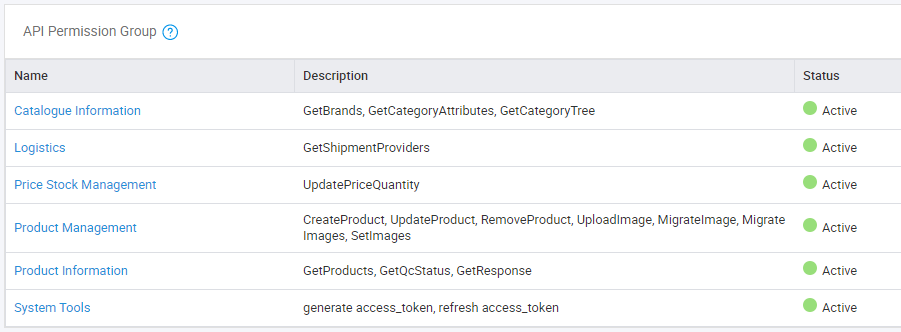
Khi application đã chuyển sang trạng thái active, nó sẽ được Lazada cấp cho App Key và App Secret. Nhà phát triển vào App Console, chọn Manage, tab App Overview để lấy App Key và App Secret.

Cũng giống như Partner ID và Key của Shopee, App Key và App Secret của Lazada rất quan trọng vì mọi API của Lazada đều yêu cầu hai thông tin này.

***Bước 4. Yêu cầu quyền truy cập API.***

Lazada chia bộ API của mình thành các nhóm API, mỗi nhóm là một tập các API cùng liên quan đến một chức năng nhất định. Application của nhà phát triển muốn gọi API nào thì phải xin quyền truy cập vào nhóm API chứa API đó. Để xin quyền truy cập, trong mục App Overview phần API Permission Group bấm Apply vào các nhóm API mong muốn và cung cấp lý do.

Syncpro cần xin quyền truy cập vào các nhóm API: **Catalogue Information**, **Logistics**, **Price Stock Management**, **Product Management**, **Product Information**, **System Tools**. Sau khi được cấp quyền truy cập, trạng thái của nhóm API chuyển sang Active.



Hình .. Các quyền truy cập API của Lazada.

***Bước 5. Gọi API và sẵn sàng triển khai.***

Phần lớn API của Lazada được gọi bằng phương thức HTTP GET, nhưng trong trường hợp payload của request quá lớn, API đó có thể được gọi bằng phương thức HTTP POST. Nếu gọi bằng phương thức GET, nội dung payload sẽ được truyền vào chuỗi query. Nếu gọi bằng phương thức POST, nội dung payload sẽ được truyền vào body với định dạng x-www-form-urlencoded. Ví dụ: API tạo sản phẩm có thể được gọi bằng cả hai phương thức, nhưng nên gọi bằng HTTP POST vì nội dung sản phẩm là một payload cỡ lớn.

Lazada có những quy định phải tuân thủ khi gọi API request:

* Lựa chọn đúng điểm cuối để gọi API. Có 6 điểm cuối tương ứng với 6 quốc gia mà Lazada đang hoạt động. Điểm cuối tại Việt Nam là <https://api.lazada.vn/rest>.
* Một số tham số nhất định phải có trong payload của mỗi request, đó là:
  + app\_key: là App Key mà Lazada cung cấp.
  + timestamp: thời điểm gọi request theo định dạng epoch millis (số milli-giây từ ngày 01/01/1970).
  + sign\_method: thuật toán sử dụng để sinh chữ ký, hãy để là “sha256”.
  + sign: chữ ký của request, được sinh theo cách thức mà Lazada định nghĩa.
* Đối với các API truy cập vào dữ liệu của người bán hàng (chủ shop) thì trong payload của request phải có thêm access\_token chứa token của chủ shop đó (xem phần [2.3.2](#_Người_bán_hàng)).

***Sinh chữ ký cho API request***

Chữ ký dùng để xác thực request hợp lệ hay không, vì vậy việc đảm bảo sinh chữ ký theo đúng quy trình rất quan trọng. Lazada sẽ không thực hiện yêu cầu và trả về lỗi với những request sai chữ ký.

Các bước sinh chữ ký của Lazada như sau:

1. Sắp xếp các tham số trong payload (ngoại trừ trường sign) theo thứ tự trên bảng mã ASCII.

Ví dụ: foo=1, bar=2, foo\_bar=3, foobar=4 sẽ được sắp xếp thành bar=2, foo=1, foo\_bar=3, foobar=4.

1. Nối các tham số và giá trị của chúng thành một chuỗi duy nhất.

Ví dụ: bar2foo1foo\_bar3foobar4.

1. Thêm tên API vào đầu chuỗi ở bước 2. Tên của API tham khảo trong tài liệu của Lazada.

Ví dụ, với tên API là “/test/api”: /test/apibar2foo1foo\_bar3foobar4.

1. Encode chuỗi ở bước 3 theo định dạng UTF-8, sau đó băm với thuật toán HMAC-SHA256 sử dụng key là App Secret mà Lazada cung cấp.
2. Encode kết quả dưới dạng HEX để thu được chữ ký hoàn chỉnh.

Ví dụ, với chuỗi: /order/getaccess\_tokentestapp\_key123456or der\_id1234sign\_methodsha256timestamp1517820392000 và App Secret là helloworld thì chữ ký thu được là 4190D32361CFB9581350222F345CB77 F3B19F0E31D162316848A2C1FFD5FAB4A.

Với cách tính này, mỗi API request sẽ có một chữ ký khác nhau.

### Người bán hàng ủy quyền

Nhà phát triển muốn truy cập vào dữ liệu của chủ shop thì phải được sự đồng ý của chủ shop. Kết quả của việc bắt tay suôn sẻ giữa đôi bên là một chuỗi xác thực (access token) được nhà phát triển nắm giữ để có thể gọi những API lên quan đến dữ liệu riêng của chủ shop như API sản phẩm, API đơn hàng, ...

Mô hình xác thực và ủy quyền trên Lazada gần giống với Shopee. Các bước tuần tự như sau:



Hình .. Các bước xác thực và ủy quyền với chủ shop trên Lazada.

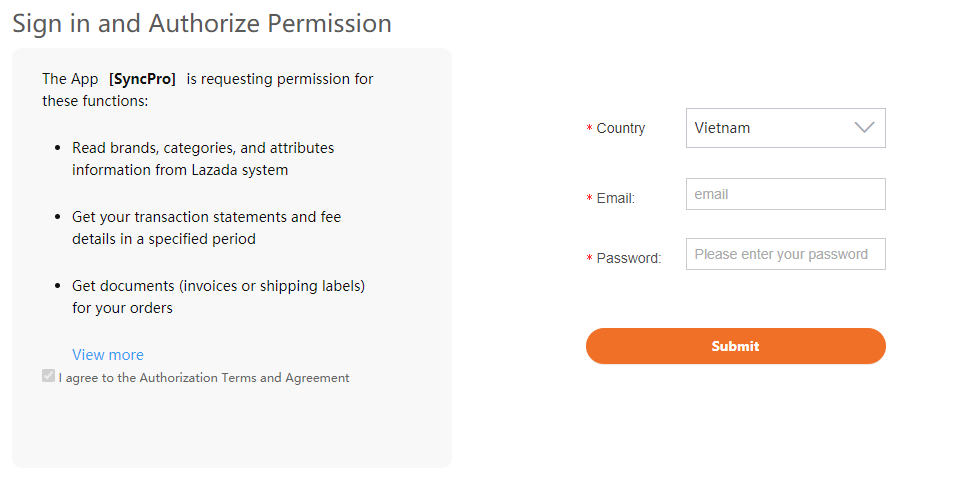
Đường dẫn xác thực và ủy quyền có dạng:

[https://auth.lazada.com/oauth/authorize?**response\_type**=code&**force\_auth**=true&**redirect\_uri**=https://xxxxxx.com&**client\_id**=xxxxxx](https://auth.lazada.com/oauth/authorize?response_type=code&force_auth=true&redirect_uri=https://xxxxxx.com&client_id=xxxxxx)

Ở đây, các tham số cần để ý là:

* **response\_type**: phải là code
* **force\_auth**: để là true.
* **client\_id**: là App Key mà Lazada cung cấp.
* **redirect\_uri**: địa chỉ này phải trùng với địa chỉ Callback URL của Lazada application mà nhà phát triển đã cài đặt trước đó. Đây là địa chỉ mà Lazada sẽ trả về authorization code khi chủ shop đăng nhập thành công.

Chủ shop bấm vào đường dẫn sẽ được đưa đến một trang đăng nhập của Lazada có giao diện như sau:



Hình .. Chủ shop đăng nhập để ủy quyền trên Lazada.

Tại đây, chủ shop thực hiện đăng nhập bằng tài khoản bán hàng trên Lazada của mình. Đăng nhập thành công, Lazada sẽ gửi một authorization code đến địa chỉ **redirect\_uri** thiết lập trước đó. Nhà phát triển sẽ sử dụng authorization code này để lấy access token. Authorization code chỉ có tác dụng trong 30 phút, vì thế nên lấy access token càng sớm càng tốt.

Access token của Lazada chỉ có tác dụng trong 7 ngày. Nếu muốn sử dụng trong thời gian lâu hơn, phải “làm mới” nó. Khi gọi API lấy access token, Lazada sẽ trả về thêm một **refresh token** có tác dụng trong 30 ngày. Nhà phát triển sẽ sử dụng refresh token này, gọi một API khác để làm mới access token. Bản chất của việc làm mới là Lazada sẽ cấp lại một cặp access token và refresh token khác. Thời hạn sử dụng của access token mới sẽ lại là 7 ngày kể từ thời điểm gọi API làm mới. Tuy nhiên, refresh token mới thì không tăng thời hạn sử dụng. Vậy nên lâu nhất là sau 30 ngày, chủ shop phải thực hiện đăng nhập lại.

### Một số API quan trọng.

Dưới đây chỉ liệt kê tên và chức năng của một số API quan trọng. Cách gọi và định dạng dữ liệu của mỗi API đã được viết rất rõ trong tài liệu của Lazada.

**GetCategoryTree**: API cho phép lấy về các Danh mục ngành hàng hiện có trên Lazada. Từ các danh mục lấy về, có thể tổng hợp thành Cây danh mục hoàn chỉnh. Trong Cây danh mục, các lá là các Danh mục ngành hàng sơ cấp. Mọi sản phẩm đều phải thuộc một Danh mục ngành hàng sơ cấp nào đó.

**GetBrands**: API lấy về các Thương hiệu trên Lazada. Thương hiệu là bắt buộc phải có khi tạo sản phẩm.

**GetCategoryAttributes**: API lấy về các thuộc tính đặc biệt ứng với một Danh mục ngành hàng sơ cấp. Các thuộc tính đặc biệt này là cần thiết để tạo sản phẩm trên Lazada. Một số thuộc tính đặc biệt là bắt buộc.

**CreateProduct**: API tạo sản phẩm trên Lazada.

**GetProductItem**: API lấy về thông tin chi tiết của một sản phẩm.

**GetQcStatus**: API lấy về thông tin kiểm duyệt chất lượng nội dung sản phẩm.

**RemoveProduct**: API xóa sản phẩm khỏi Lazada.

**UpdateProduct**: API cập nhật mọi thông tin của sản phẩm (nội dung, hình ảnh, giá, số lượng, v.v...).

**Generate access\_token**: API lấy access token từ authorization code.

**Refresh access\_token**: API làm mới các token, sử dụng refresh token.

# Thiết kế giải pháp

## Ý tưởng giải quyết bài toán.

Bài toán đồng bộ sản phẩm bao gồm hai vấn đề chính: **Đẩy sản phẩm lên từng sàn thương mại điện tử** và **Đồng bộ nội dung giữa các sàn**. Đẩy sản phẩm phải đảm bảo sản phẩm xuất hiện trên các sàn và Đồng bộ nội dung phải đảm bảo nội dung sản phẩm trên các sàn giống nhau. Thêm nữa, để đảm bảo việc gọi API luôn được trơn tru, phải để ý đến việc làm mới token theo chu kỳ (đối với sàn Lazada).

Bài toán đồng bộ sản phẩm thực sự là một bài toán khó vì trên thực tế, đa phần sản phẩm trên các sàn thương mại điện tử là sản phẩm có nhiều biến thể (VD: một đôi giày có nhiều kích cỡ) nên việc đồng bộ là khá khó khăn. Vì vậy, với phiên bản đầu tiên này của Syncpro, ý tưởng nêu trong tài liệu xin phép được **giải bài toán trong trường hợp cơ bản nhất: một sản phẩm chỉ có một biến thể**. Các sản phẩm đa biến thể sẽ được xử lý trong những phiên bản tiếp theo.

### Bài toán đẩy sản phẩm lên các sàn thương mại điện tử.

Bài toán đẩy sản phẩm không chỉ đơn thuần là gọi API tạo sản phẩm mà còn rất nhiều vấn đề khác cần giải quyết. Muốn đẩy được sản phẩm lên, phải nghiên cứu kỹ lưỡng form sản phẩm của từng sàn thương mại điện tử. Từ việc nghiên cứu form sản phẩm, biết được định dạng dữ liệu của từng trường; biết được trường nào bắt buộc, trường nào không bắt buộc; tìm ra các trường bắt buộc của sản phẩm mà cần phải gọi API mới có được giá trị để điền vào trường đó, sau đó tìm phương án giải quyết với các trường như vậy. Sau khi gọi API tạo sản phẩm thành công, phải theo dõi sản phẩm đó xem nó có được chấp nhận hay đã bị từ chối. Nếu sản phẩm bị từ chối, phải có phương án để người dùng tạo lại. Nói tóm lại, phải làm tất cả hành động cần thiết để đảm bảo sản phẩm xuất hiện trên hệ thống của các sàn.

#### Đẩy sản phẩm lên Shopee.

Để tạo sản phẩm trên Shopee, sử dụng API **Add item**. API này yêu cầu form sản phẩm dạng JSON với các trường như sau:

* **category\_id**: ID Danh mục ngành hàng sơ cấp của sản phẩm, bắt buộc phải có. Để có ID của Danh mục ngành hàng sơ cấp, phải sử dụng API **GetCategoriesBy-Country** để lấy về các Danh mục ngành hàng.
* **name**: Tên của sản phẩm, dạng chữ, bắt buộc.
* **description**: Mô tả về sản phẩm, dạng chữ. Không giống với đa số các sàn thương mại điện tử khác, Shopee không hỗ trợ mô tả sản phẩm bằng HTML mà chỉ hỗ trợ dạng chữ bình thường. Trường này là bắt buộc.
* **price**: Giá bán của sản phẩm, dạng số tự nhiên, bắt buộc.
* **stock**: Số lượng sản phẩm trong kho, dạng số tự nhiên, bắt buộc.
* **item\_sku**: SKU mà người bán đặt cho sản phẩm, dạng chữ, bắt buộc.
* **variations**: Các biến thể sản phẩm. Shopee không khuyên dùng trường này để tạo sản phẩm đa biến thể mà có API để tạo sản phẩm đa biến thể riêng. Nên bỏ qua trường này.
* **images**: Các hỉnh ảnh minh họa cho sản phẩm, dạng mảng của các object, bắt buộc. Mỗi object có cấu trúc:
  + **url**: Địa chỉ của hình ảnh trên internet, dạng chữ.
* **attributes**: Các thuộc tính riêng của sản phẩm, dạng mảng của các object, bắt buộc. Mỗi object có cấu trúc:
  + **attributes\_id**: ID của thuộc tính, dạng số tự nhiên.
  + **value**: Giá trị của thuộc tính, dạng chữ.

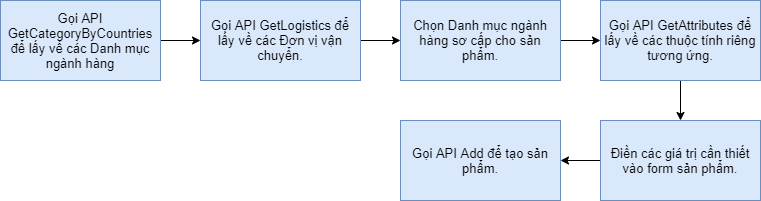
Để có được ID của thuộc tính, phải gọi API **GetAttributes** để lấy về các thuộc tính tương ứng với ID Danh mục ngành hàng sơ cấp của sản phẩm. Một số thuộc tính là bắt buộc.

* **logistics**: Các đơn vị giao hàng hỗ trợ, dạng mảng của các object. Mỗi object có cấu trúc:
  + **logistic\_id**: ID của đơn vị vận chuyển, dạng số tự nhiên.
  + **enabled**: Có cho phép đơn vị vận chuyển hoạt động hay không. Giá trị dạng True – False. Hãy để là True.
  + **is\_free**: Có miễn phí giao hàng đối với đơn vị vận chuyển này hay không. Giá trị dạng True – False.

Để có được ID của đơn vị vận chuyển, phải gọi API **GetLogistics** để lấy về thông tin của tất cả các đơn vị vận chuyển hiện có trên Shopee.

* **weight**: Cân nặng của gói hàng (kg), dạng số thập phân, bắt buộc.
* **package\_length**: Chiều dài của gói hàng (cm), dạng số tự nhiên, không bắt buộc.
* **package\_width**: Chiều rộng của gói hàng (cm), dạng số tự nhiên, không bắt buộc.
* **package\_height**: Chiều cao của gói hàng (cm), dạng số tự nhiên, không bắt buộc.
* **days\_to\_ship**: Số ngày đảm bảo sẽ giao hàng xong, dạng số tự nhiên, không bắt buộc. Với hàng đặt trước, hãy điền giá trị từ 7 – 30. Với hàng không phải đặt trước, hãy bỏ qua trường này.
* **wholesales**: Các định mức bán sỉ sản phẩm, dạng mảng của các object, không bắt buộc. Mỗi object có cấu trúc:
  + **min**: Số lượng bán sỉ tối thiểu, dạng số tự nhiên, bắt buộc.
  + **max**: Số lượng bán sỉ tối đa, dạng số tự nhiên, bắt buộc.
  + **unit\_price**: Giá cho mỗi sản phẩm khi bán sỉ, dạng số thập phân, bắt buộc.
* **condition**: NEW nếu sản phẩm là hàng mới, USED nếu đã qua sử dụng, không bắt buộc.
* **status**: NORMAL nếu muốn sản phẩm được hiển thị trên Shopee, UNLIST nếu không muốn, không bắt buộc.
* **is\_pre\_order**: Hàng có phải đặt trước hay không. Giá trị dạng True – False, không bắt buộc.

Nghiên cứu form sản phẩm của Shopee cho thấy rằng: nhiều trường thì người bán có thể tự điền giá trị, nhưng nhiều trường thì phải điền theo các giá trị mà Shopee đã định trước. Do đó, trước khi điền vào form sản phẩm, phải sử dụng API lấy về các giá trị định trước này. Cụ thể với Shopee là lấy về giá trị cho các trường **category\_id**, **attributes**, **logistics**. Các bước tuần tự để tạo sản phẩm trên Shopee như sau:



Hình .. Các bước tạo sản phẩm trên Shopee.

Trên thực tế, nếu mỗi lần tạo sản phẩm lại phải gọi các API của Shopee để lấy dữ liệu thì không hợp lý. Thứ nhất là tốc độ lấy dữ liệu chậm vì phải lấy qua internet. Thứ hai là có thể bị Shopee giới hạn số request được gọi trong một khoảng thời gian (một giây, một giờ, một ngày, ...). Thứ ba là các dữ liệu về Danh mục ngành hàng, Thuộc tính, Đơn vị vận chuyển rất hiếm khi thay đổi nên việc gọi API để lấy dữ liệu mỗi lần tạo sản phẩm là không cần thiết.

Giải pháp cho vấn đề này là crawl tất cả các dữ liệu trên, lưu trữ vào một database cục bộ và cập nhật chúng theo một chu kỳ vừa phải (một lần trong khoảng 1 – 2 tuần). Tất cả các Danh mục ngành hàng, các Thuộc tính, các Đơn vị vận chuyển sẽ được lưu trữ trong các bảng tương ứng của database. Mỗi lần tạo sản phẩm, chỉ cần truy vấn dữ liệu trong database mà không phải gọi API của Shopee nữa.

Khi gọi API Add và tạo được sản phẩm trên Shopee là bước đầu của bài toán đẩy sản phẩm đã thành công. Công việc tiếp theo, khó khăn hơn, đó là quản lý trạng thái của sản phẩm đã tạo. Quản lý trạng thái ở đây nghĩa là nắm bắt tất cả các trạng thái có thể của sản phẩm và có hành động xử lý đối với từng trạng thái mà sản phẩm đạt tới.

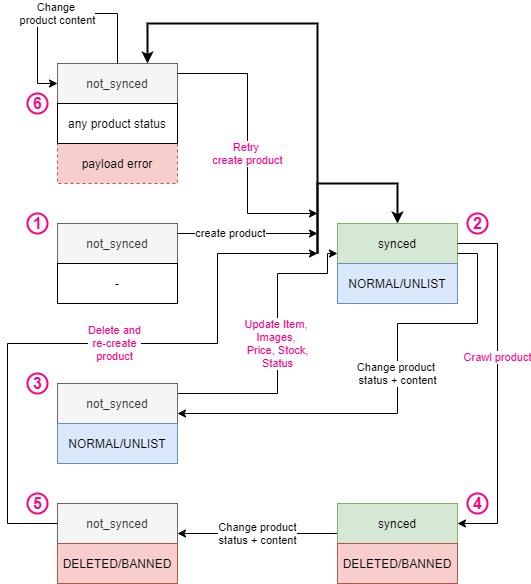
Để giải quyết vấn đề quản lý trạng thái, một **bản đồ trạng thái của sản phẩm** sẽ được xây dựng. Bản đồ trạng thái của sản phẩm là tấm bản đồ với điểm dừng là các trạng thái và đường đi là các hành động xử lý. Nhìn vào tấm bản đồ đó sẽ thấy được tất cả các trạng thái có thể có của một sản phẩm và biết được với từng trạng thái thì nên làm gì để xử lý.

Một sản phẩm trên hệ thống của Syncpro đối với một sàn thương mại điện tử sẽ có một trong hai trạng thái: **synced** và **not\_synced**. Sản phẩm đạt trạng thái synced khi mà nội dung mới nhất của nó đã được cập nhật lên sàn thương mại điện tử tương ứng. Ngược lại, khi mà nội dung mới nhất của sản phẩm chưa được cập nhật lên sàn thương mại điện tử tương ứng thì trạng thái của sản phẩm (đối với sàn thương mại điện tử đó) sẽ là not\_synced.

Một sản phẩm trên Shopee có thể rơi vào một trong các trạng thái: **NORMAL**, **UNLIST**, **DELETED**, **BANNED**. Ý nghĩa của các trạng thái như sau:

* **NORMAL**: sản phẩm hợp lệ và được hiển thị bình thường.
* **UNLIST**: sản phẩm hợp lệ nhưng nhà bán hàng ẩn đi, không cho hiển thị.
* **DELETED**: sản phẩm đã bị nhà bán hàng xóa bỏ.
* **BANNED**: sản phẩm bị cấm hiển thị vì có thông tin không hợp lệ.

Từ việc nghiên cứu các trạng thái của sản phẩm trên Shopee kết hợp với trạng thái synced/not\_synced trên Syncpro, bản đồ trạng thái của sản phẩm được xây dựng như sau:



Hình .. Bản đồ trạng thái của sản phẩm trên Shopee.

Theo những gì được vẽ trên bản đồ, một sản phẩm có thể trải qua các bước:

1. Sản phẩm mới được tạo ra trên Syncpro có trạng thái là **not\_synced** và chưa có trạng thái trên Shopee. Sản phẩm sẽ được đồng bộ lên Shopee bằng API tạo sản phẩm.
2. Sản phẩm được đồng bộ thành công lên Shopee có trạng thái **synced-NORMAL/UNLIST**. Tại đây sản phẩm có thể rẽ sang 2 hướng:
   1. Sản phẩm được người bán thay đổi nội dung và chuyển sang trạng thái **not\_synced-NORMAL/UNLIST**.
   2. Sản phẩm được theo dõi thường xuyên, phát hiện ra bị chuyển sang trạng thái **synced-DELETED/BANNED**.
3. Sản phẩm ở trạng thái **not\_synced-NORMAL/UNLIST** thì sẽ được cập nhật nội dung mới nhất lên Shopee.
4. Sản phẩm ở trạng thái **synced-DELETED/BANNED** thì người bán phải thay đổi nội dung để chuyển sản phẩm sang trạng thái **not\_synced-DELETED/BANNED**.
5. Sản phẩm ở trạng thái **not\_synced-DELETED/BANNED** thì sẽ được xóa đi tạo lại trên Shopee.
6. Nếu việc tạo sản phẩm thất bại thì lỗi sẽ được ghi lại với sản phẩm đó. Người bán hàng nên xem lỗi và sửa lại nội dung sản phẩm vì đa phần lỗi không tạo được sản phẩm là do thông tin trong form sản phẩm không hợp lệ. Sản phẩm có lỗi sẽ được tự động thử tạo lại vào lúc khác. Sau khi thử tạo lại, sản phẩm có thể rẽ 2 hướng:
   1. Tạo lại thành công, sản phẩm được xóa lỗi và có trạng thái **synced-NORMAL/UNLIST**, sẽ được xử lý ở bước 2.
   2. Tạo lại thất bại, sản phẩm được ghi lỗi mới và sẽ được xử lý lại ở bước 6.

Trên bản đồ, các hành động xử lý với chữ màu đỏ là các hành động sẽ được chạy tự động theo chu kỳ. Trong hệ thống của Syncpro, nó sẽ chạy dưới dạng các công việc (job) tự động. Khi một job chạy, nó sẽ tìm trong database những sản phẩm có trạng thái phù hợp để thực thi thao tác với sản phẩm đó. Các hành động với chữ màu đen là các hành động phát sinh do tác động của con người.

#### Đẩy sản phẩm lên Tiki.

Trên Tiki, không có API tạo sản phẩm trực tiếp mà phải sử dụng API yêu cầu tạo sản phẩm. Định dạng form sản phẩm của API này cũng là JSON với các trường yêu cầu gồm có:

* **category\_id**: ID Danh mục ngành hàng sơ cấp của sản phẩm, bắt buộc phải có. Để có ID của Danh mục ngành hàng sơ cấp, phải sử dụng API **Get categories** để lấy về các Danh mục ngành hàng.
* **name**: Tên của sản phẩm, dạng chữ, bắt buộc.
* **description**: Mô tả về sản phẩm, dạng chữ và có hỗ trợ HTML, bắt buộc.
* **market\_price**: Giá thị trường của sản phẩm khi chưa giảm giá, dạng số tự nhiên, không bắt buộc.
* **attributes**: Các thuộc tính riêng của sản phẩm, dạng object, bắt buộc. Khóa của object là tên các thuộc tính, giá trị của mỗi khóa là giá trị tương ứng của thuộc tính đó. Ví dụ:

{

"bulky": 1,

"origin": "my",

"product\_top\_features": "White bracelet watch featuring rhinestone-accented bezel and mother-of-pearl dial with sparkling Mickey Mouse design\n37-mm metal case with glass dial window\nQuartz movement with analog display\nMetal alloy bracelet with jewelry-clasp closure\nNot water resistant\n",

"brand": "Disney",

"case\_diameter": "37 millimeters",

"filter\_case\_diameter": "37 millimeters",

"brand\_origin": "viet nam",

"require\_expiry\_date": 1,

"product\_weight": 1,

"product\_width": 10,

"product\_length": 20,

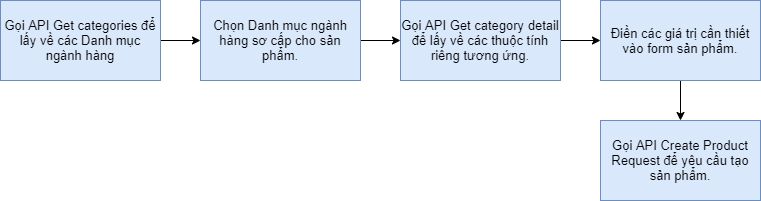
"product\_height": 5,

}

Để có được tên các thuộc tính, phải gọi API **Get category detail** lấy về các thuộc tính tương ứng với ID Danh mục ngành hàng sơ cấp của sản phẩm. Một số thuộc tính là bắt buộc. Một số thuộc tính thì sản phẩm thuộc Danh mục ngành hàng nào cũng sẽ có:

* **brand**: Thương hiệu của sản phẩm, dạng chữ, bắt buộc.
* **product\_top\_features**: Mô tả ngắn về các tính năng nổi trội của sản phẩm, dạng chữ, không hỗ trợ HTML, bắt buộc. Các tính năng nổi trội ngăn cách bởi dấu xuống dòng \n.
* **product\_weight**: Cân nặng của gói hàng (kg), dạng số thập phân, bắt buộc.
* **product\_width**: Chiều rộng của gói hàng (cm), dạng số tự nhiên, bắt buộc.
* **product\_length**: Chiều dài của gói hàng (cm), dạng số tự nhiên, bắt buộc.
* **product\_height**: Chiều cao của gói hàng (cm), dạng số tự nhiên, bắt buộc.
* **image**: Địa chỉ hình ảnh đại diện của sản phẩm, dạng chữ, bắt buộc.
* **images**: Một mảng các địa chỉ hình ảnh minh họa cho sản phẩm, bắt buộc.
* **option\_attributes**: Mảng các thuộc tính làm nên biến thể của sản phẩm, hỗ trợ nhiều nhất là 2 thuộc tính. Ví dụ: [“size”, “color”]. Với sản phẩm chỉ có 1 biến thể, phải để trường này là mảng rỗng chứ không được bỏ trường này và cũng không được để null.
* **variants**: Các biến thể sản phẩm, dạng mảng của các object, bắt buộc. Mỗi object có cấu trúc:
  + **sku**: SKU mà người bán đặt cho sản phẩm, dạng chữ, bắt buộc.
  + **price**: Giá bán của sản phẩm, dạng số tự nhiên, bắt buộc.
  + **market\_price**: Giá thị trường của sản phẩm, dạng số tự nhiên, không bắt buộc. Giá trị này sẽ ghi đè lên giá trị trường **market\_price** bên ngoài.
  + **option1**: Giá trị của thuộc tính đầu tiên làm nên biến thể. VD: “M”, “XL”. Giá trị này là bắt buộc nếu mảng option\_attributes bên ngoài có phần tử đầu tiên.
  + **option2**: Giá trị của thuộc tính thứ hai làm nên biến thể. VD: “green”, “blue”. Giá trị này là bắt buộc nếu mảng option\_attributes bên ngoài có phẩn tử thứ hai.
  + **inventory\_type**: Một trong các giá trị **instock**, **backorder**, **seller\_backorder**, **drop\_ship**, **cross\_border**. Ý nghĩa của các giá trị này tham khảo tại <http://open.tiki.vn/#inventory-type>. Trường này không bắt buộc.
  + **quantity**: Số lượng hàng trong kho, dạng số tự nhiên, bắt buộc.
  + **supplier**: ID nhà cung cấp. Nếu **inventory\_type** là một trong các giá trị **backorder**, **drop\_ship**, **cross\_border** thì trường này là bắt buộc. Nhà bán hàng phải liên hệ với Tiki để có được ID nhà cung cấp này.
  + **name**: Tên của sản phẩm, dạng chữ, không bắt buộc. Giá trị này ghi đè lên giá trị trường **name** bên ngoài.
  + **description**: Mô tả về sản phẩm, dạng chữ có hỗ trợ HTML, không bắt buộc. Giá trị trường này ghi đè lên giá trị trường **description** bên ngoài.
  + **attributes**: Các thuộc tính riêng của sản phẩm, dạng object, không bắt buộc. Giá trị trường này sẽ được kết hợp với giá trị trường **attributes** bên ngoài.
  + **image**: Địa chỉ hình đại diện cho sản phẩm, dạng chữ, không bắt buộc. Giá trị trường này ghi đè lên giá trị trường **image** bên ngoài.
  + **images**: Mảng các địa chỉ hình ảnh minh họa cho sản phẩm, không bắt buộc. Giá trị trường này ghi đè lên giá trị trường **images** bên ngoài.

Về cơ bản, form sản phẩm của Tiki khá giống Shopee. Đa số các trường cho phép người bán tự điền giá trị, nhưng cũng có một số trường phải dùng API để lấy về bộ giá trị mới có thể điền vào được. Cụ thể ở đây là lấy về giá trị cho các trường **category\_id** và **attributes**. Để yêu cầu tạo sản phẩm trên Tiki, phải làm tuần tự các bước sau:



Hình .. Các bước yêu cầu tạo sản phẩm trên Tiki.

Danh mục ngành hàng và các Thuộc tính riêng trên Tiki cũng là những dữ liệu hiếm khi thay đổi. Vì vậy, giải pháp tốt hơn vẫn là crawl hết Danh mục ngành hàng và Thuộc tính riêng về một database cục bộ rồi cập nhật chúng một lần trong khoảng 1 – 2 tuần. Mỗi khi tạo sản phẩm, những dữ liệu này sẽ được lấy trong database cục bộ thay vì phải gọi API để lấy trên Tiki, gây mất thời gian mà lại không hiệu quả.

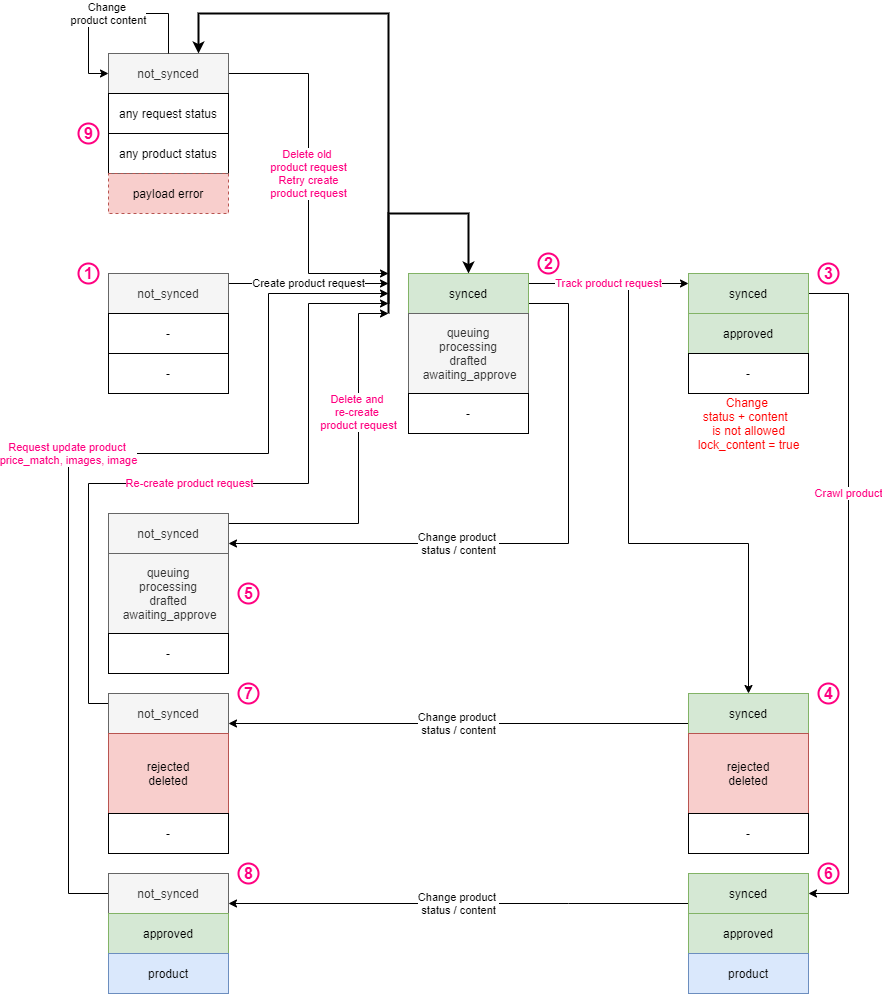
Sau khi gọi API yêu cầu tạo sản phẩm thành công, việc tiếp theo cần làm là theo dõi và quản lý trạng thái của yêu cầu cũng như theo dõi và quản lý trạng thái của sản phẩm. Giống như bên Shopee, muốn quản lý được trạng thái của sản phẩm, phải xây dựng được **Bản đồ trạng thái của sản phẩm** trên Tiki.

Hai trạng thái **synced** và **not\_synced** của sản phẩm trên hệ thống Syncpro đối với Tiki vẫn được sử dụng với ý nghĩa tương tự như đối với Shopee.

Trên Tiki, một yêu cầu tạo sản phẩm có thể rơi vào một trong các trạng thái: **queuing**, **processing**, **drafted**, **bot\_awaiting\_approve**, **md\_awaiting\_approve**, **awaiting\_appr-ove**, **approved**, **rejected**, **deleted**. Ý nghĩa của các trạng thái như sau:

* **queuing**: yêu cầu sản phẩm mới tạo, đang đợi xử lý.
* **processing**: yêu cầu sản phẩm đang được Tiki format lại theo định dạng của Tiki.
* **drafted**: yêu cầu sản phẩm sẵn sàng được kiểm định.
* **bot\_awaiting\_approve**: yêu cầu sản phẩm đang được kiểm định bởi bot của Tiki.
* **md\_awaiting\_approve**: yêu cầu sản phẩm đang được kiểm định các giá trị bắt buộc.
* **awaiting\_approve**: yêu cầu sản phẩm đang được kiểm định nội dung bởi đội kiểm định của Tiki.
* **approved**: yêu cầu sản phẩm được chấp thuận, sản phẩm được tạo thành công.
* **rejected**: yêu cầu sản phẩm bị bác bỏ.
* **deleted**: yêu cầu sản phẩm đã bị xóa, không còn trên hệ thống của Tiki.

Từ việc nghiên cứu các trạng thái của yêu cầu sản phẩm trên Tiki kết hợp với trạng thái synced/not\_synced trên Syncpro, bản đồ trạng thái của sản phẩm được xây dựng như sau:



Hình .. Bản đồ trạng thái của sản phẩm trên Tiki.

Để tiện cho việc ký hiệu, các trạng thái của yêu cầu tạo sản phẩm trên Tiki sẽ được chia làm 3 nhóm:

* **WAITING**: queuing, processing, drafted, bot\_awaiting\_approve, md\_awaiting\_ap-prove, awaiting\_approve.
* **FAIL**: rejected, deleted.
* **SUCCESS**: approved.

Theo những gì được vẽ trên bản đồ, một sản phẩm có thể trải qua các bước:

1. Sản phẩm mới được tạo ra trên Syncpro có trạng thái là **not\_synced** và chưa có trạng thái trên Tiki. Sản phẩm sẽ được đồng bộ lên Tiki bằng API yêu cầu tạo sản phẩm.
2. Sản phẩm được đồng bộ thành công lên Tiki có trạng thái **synced-WAITING**. Tại đây sản phẩm có thể rẽ sang 3 hướng:
   1. Sản phẩm được theo dõi thường xuyên, phát hiện ra được chuyển sang trạng thái **synced-SUCCESS**.
   2. Sản phẩm được theo dõi thường xuyên, phát hiện ra bị chuyển sang trạng thái **synced-FAIL**.
   3. Sản phẩm được người bán thay đổi nội dung và chuyển sang trạng thái **not\_synced-WAITING**.
3. Sản phẩm ở trạng thái **synced-SUCCESS** nghĩa là yêu cầu tạo sản phẩm đã được Tiki kiểm duyệt và chấp nhận, sản phẩm đã được tạo thành công. Để tránh việc người bán hàng sửa đổi nội dung sản phẩm lúc này gây ra các trạng thái không mong muốn, sản phẩm sẽ tạm thời bị khóa nội dung. Thông tin chi tiết của sản phẩm trên Tiki được lấy về. Sản phẩm chuyển sang trạng thái **synced-SUCCESS có thông tin sản phẩm đính kèm**.
4. Sản phẩm ở trạng thái **synced-FAIL** tức là yêu cầu tạo sản phẩm không được chấp nhận. Người bán phải thay đổi nội dung để chuyển trạng thái sản phẩm sang **not\_synced-FAIL**.
5. Sản phẩm ở trạng thái **not\_synced-WAITING** thì yêu cầu tạo sản phẩm sẽ bị xóa đi và tạo lại yêu cầu khác.
6. Sản phẩm ở trạng thái **synced-SUCCESS có thông tin sản phẩm đính kèm**, nếu người bán hàng thay đổi nội dung sản phẩm thì trạng thái chuyển sang **not\_synced-SUCCESS có thông tin sản phẩm đính kèm**.
7. Sản phẩm ở trạng thái **not\_synced-FAIL** sẽ được tạo yêu cầu sản phẩm khác.
8. Sản phẩm ở trạng thái **not\_synced-SUCCESS có thông tin sản phẩm đính kèm** thì sẽ tạo yêu cầu cập nhật nội dung mới nhất lên Tiki.
9. Nếu việc yêu cầu tạo sản phẩm hay yêu cầu cập nhật sản phẩm thất bại (ngay lúc gọi API) thì lỗi sẽ được ghi lại đối với sản phẩm đó. Lúc này người bán nên xem xét lại nội dung sản phẩm và sửa lại vì đa phẩn lỗi là do nội dung sản phẩm không hợp lệ. Với các sản phẩm có lỗi, yêu cầu tạo sản phẩm hay cập nhật sản phẩm sẽ tự động được xóa đi và tạo lại vào lúc khác. Sau khi thử tạo lại yêu cầu, sản phẩm có thể đi theo 2 hướng:
   1. Tạo lại thành công, sản phẩm được xóa lỗi và có trạng thái **synced-WAITING**, sẽ được xử lý ở bước 2.
   2. Tạo lại thất bại, sản phẩm được ghi lỗi mới và sẽ được xử lý lại ở bước 9.

Giống như bản đồ trạng thái sản phẩm trên Shopee, các hành động xử lý với chữ màu đỏ là các hành động sẽ được chạy tự động theo chu kỳ dưới dạng các job. Khi một job chạy, nó sẽ tìm trong database những sản phẩm có trạng thái phù hợp để thực thi thao tác với sản phẩm đó. Các hành động với chữ màu đen là các hành động phát sinh do tác động của con người.

#### Đẩy sản phẩm lên Lazada.

Lazada sử dụng API **CreateProduct** để tạo sản phẩm. Không giống hai sàn thương mại điện tử còn lại, API tạo sản phẩm của Lazada sử dụng form sản phẩm định dạng XML. Cấu trúc của form sản phẩm như sau:

* Thẻ gốc (root) là thẻ **Request**.
* Bên trong thẻ **Request** là các thẻ con: **PrimaryCategory**, **Attributes**, **Skus**.
* Thẻ **PrimaryCategory**: chứa ID Danh mục ngành hàng sơ cấp của sản phẩm, bắt buộc. Để có ID Danh mục ngành hàng sơ cấp, phải sử dụng API **GetCategoryTree** để lấy về Cây danh mục ngành hàng.
* Thẻ **Attributes**: chứa các thuộc tính riêng của sản phẩm. Mỗi thuộc tính là một thẻ con của thẻ Attributes. Tên thẻ con là tên của thuộc tính. Giá trị của thẻ con là giá trị của thuộc tính.

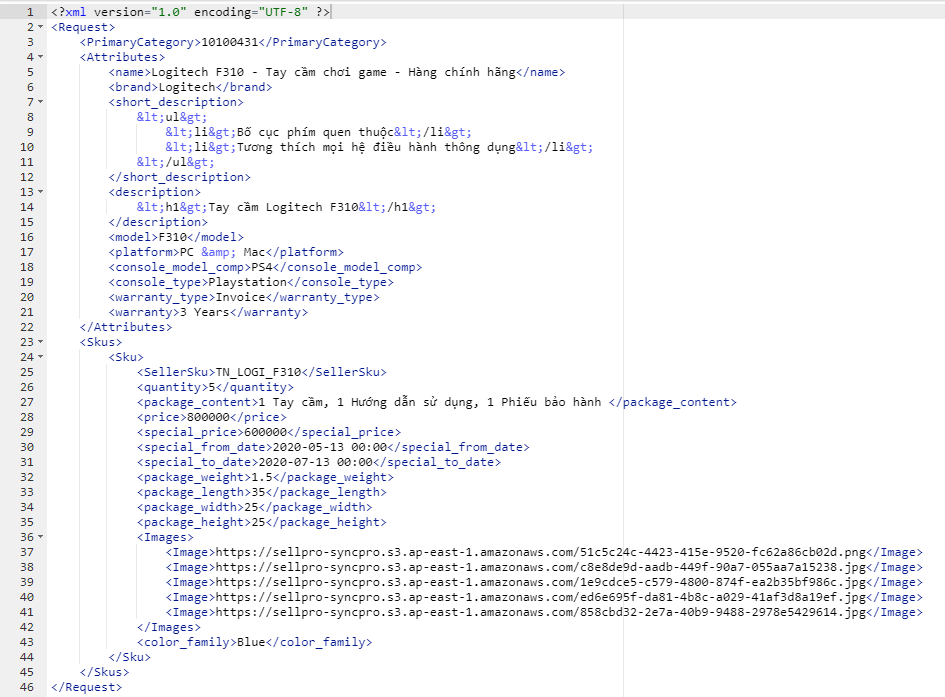
Cácthuộc tính riêng là khác nhau với mỗi Danh mục ngành hàng sơ cấp. Để có được tên các thuộc tính, phải gọi API **GetCategoryAttributes** và truyền vào ID của Danh mục ngành hàng. API này trả về một danh sách nhiều thuộc tính, nhưng trong đó chỉ có các thuộc tính có giá trị **attribute\_type** là “normal” thì mới hợp lệ khi nằm trong thẻ **Attributes**, vì vậy cần lọc ra các thuộc tính đó.

Dù với mỗi Danh mục ngành hàng sơ cấp lại có các thuộc tính riêng khác nhau, nhưng có một số thuộc tính thì sản phẩm thuộc Danh mục ngành hàng nào cũng sẽ có:

* **name**: Tên của sản phẩm, dạng chữ, bắt buộc.
* **brand**: Thương hiệu của sản phẩm, dạng chữ, bắt buộc.
* **short\_description**: Mô tả ngắn về các tính năng nổi trội của sản phẩm, dạng chữ và có hỗ trợ HTML, bắt buộc.
* **description**: Mô tả sản phẩm, dạng chữ và có hỗ trợ HTML, bắt buộc.
* Thẻ **Skus**: Trong thẻ **Skus** chứa nhiều thẻ con **Sku**, mỗi thẻ con **Sku** là một biến thể của sản phẩm. Các thẻ con nằm trong thẻ con **Sku** là các thuộc tính của biến thể. Giá trị của các thẻ con đó là giá trị của các thuộc tính. Thông thường, các thuộc tính của một biến thể là:
  + **SellerSku**: SKU mà người bán đặt cho sản phẩm, dạng chữ, bắt buộc.
  + **quantity**: Số lượng sản phẩm trong kho, dạng số tự nhiên, bắt buộc.
  + **package\_content**: Mô tả thành phần bên trong gói hàng, dạng chữ, bắt buộc.
  + **price**: Giá thị trường của sản phẩm khi chưa giảm giá, dạng số tự nhiên, bắt buộc.
  + **special\_price**: Giá bán của sản phẩm sau khi đã giảm giá, dạng số tự nhiên, bắt buộc.
  + **special\_from\_date**: Ngày bắt đầu giảm giá, định dạng YYYY-MM-DD HH:mm.
  + **special\_to\_date**: Ngày kết thúc giảm giá, định dạng YYYY-MM-DD HH:mm.
  + **package\_weight**: Cân nặng của gói hàng (kg), dạng số thập phân, bắt buộc.
  + **package\_length**: Chiều dài của gói hàng (cm), dạng số tự nhiên, bắt buộc.
  + **package\_width**: Chiều rộng của gói hàng (cm), dạng số tự nhiên, bắt buộc.
  + **package\_height**: Chiều cao của gói hàng (cm), dạng số tự nhiên, bắt buộc.
  + **Images**: Bên trong thẻ **Images** là nhiều thẻ con **Image**. Mỗi thẻ con **Image** này mang giá trị là địa chỉ url hình ảnh minh họa cho sản phẩm. Mặc định hình ảnh đầu tiên là avatar của sản phẩm.

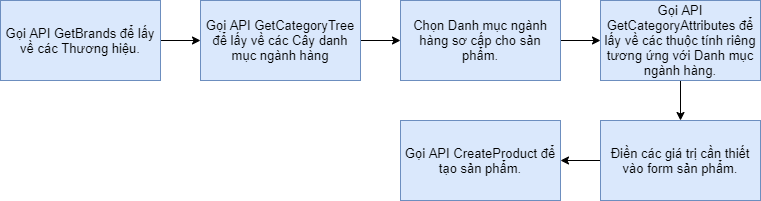
Còn một số thuộc tính khác nữa phụ thuộc vào việc sản phẩm thuộc Danh mục ngành hàng sơ cấp nào. Việc lấy tên các thuộc tính này cũng sử dụng API **GetCa-tegoryAttributes**, chỉ khác là phải lọc ra các thuộc tính có giá trị **attribute\_type** là “sku” thay vì lọc ra các thuộc tính có giá trị **attribute\_type** là “normal”.

Dưới đây là ví dụ một form sản phẩm hoàn chỉnh của Lazada:



Hình .. Form sản phẩm dạng XML của Lazada.

Tuy có form sản phẩm định dạng XML khác với Shopee và Tiki, song về thông tin cần cung cấp để tạo sản phẩm thì cơ bản Lazada giống với hai sàn thương mại điện tử còn lại. Trong form sản phẩm có nhiều trường mà người bán tự điền giá trị và cũng có nhiều trường thì phải dùng API để lấy về bộ giá trị mới điền vào được. Cụ thể ở đây đó là các trường **PrimaryCategory**, **Attributes**, **brand**, **Sku**. Các bước tuần tự để tạo sản phẩm trên Lazada như sau:



Hình .. Các bước tạo sản phẩm trên Lazada.

Trên Lazada hiện tại có hơn 100.000 thương hiệu khác nhau và không có cách nào để lấy tất cả số thương hiệu đó về trong một lần gọi API **GetBrands**, do API này chỉ cho phép lấy về tối đa 1000 thương hiệu mỗi lần gọi. Vì vậy, cách làm duy nhất là crawl tất cả các thương hiệu vào database cục bộ của Syncpro và cập nhật chúng một lần trong khoảng 1 – 2 tuần. Khi người bán hàng chọn thương hiệu cho sản phẩm, vì số lượng thương hiệu là quá lớn nên không thể hiển thị hết ra cho người bán chọn mà sẽ cần phải làm tính năng gợi ý: người bán gõ vào từ khóa tên thương hiệu, bên dưới hiển thị ra danh sách các thương hiệu được tìm thấy với từ khóa đó cho người bán chọn.

Các dữ liệu về **CategoryTree**, **CategoryAttributes** thì có thể gọi API của Lazada để lấy về đầy đủ mỗi lần tạo sản phẩm. Tuy nhiên, như đã phân tích ở mục [3.1.1.1](#_Đẩy_sản_phẩm), cách làm này là không hợp lý. Các dữ liệu ít thay đổi như thế này nên được crawl hết về database cục bộ để khi tạo sản phẩm thì chỉ cần truy vấn dữ liệu trong database mà không phải gọi API của Lazada nữa. Việc cập nhật dữ liệu sẽ được tiến hành một lần trong khoàng 1 – 2 tuần.

Sau khi gọi được API tạo sản phẩm trên Lazada, việc tiếp theo cần làm là quản lý trạng thái sản phẩm trên Lazada. Cũng giống như hai sàn thương mại điện tử kia, để quản lý trạng thái sản phẩm trên Lazada, cần phải xây dựng **Bản đồ trạng thái của sản phẩm trên Lazada**.

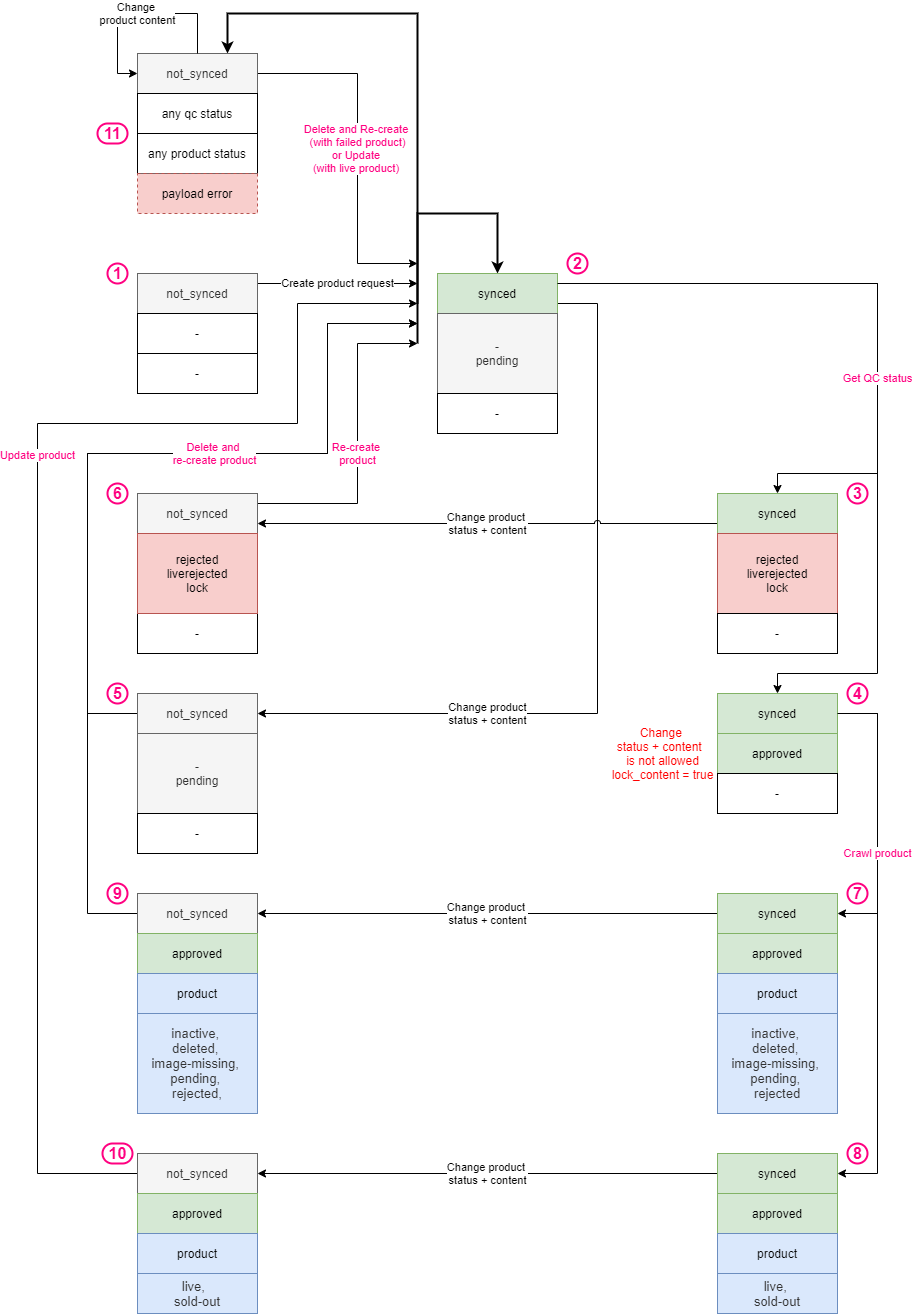
Hai trạng thái **synced** và **not\_synced** của sản phẩm trên hệ thống Syncpro đối với Lazada vẫn được sử dụng với ý nghĩa tương tự như đối với Shopee và Tiki.

Sản phẩm trên Lazada có một loại trạng thái gọi là Trạng thái QC (**QC status**). Đây là trạng thái cho biết kết quả kiểm duyệt của Lazada đối với sản phẩm đó. QC status có thể là một trong số những trạng thái sau: **pending**, **rejected**, **liverejected**, **lock**, **approved**. Nếu sản phẩm có QC status là **pending** tức là sản phẩm đó đang được kiểm duyệt. Nếu sản phẩm có QC status là **approved** thì có nghĩa là nó đã vượt qua vòng kiểm duyệt. Ngược lại, nếu QC status rơi vào một trong ba trạng thái còn lại thì có nghĩa là sản phẩm đã không vượt qua được vòng kiểm duyệt.

Khi và chỉ QC status chuyển sang **approved** thì lúc đó sản phẩm sẽ có thêm một trạng thái nữa gọi là trạng thái hoạt động (**product status**). Trạng thái hoạt động là trạng thái phản ánh tình trạng hoạt động của sản phẩm trên sàn Lazada. Trạng thái hoạt động có thể là một trong số những trạng thái: **inactive**, **deleted**, **image-missing**, **pending**, **rejected**, **live**, **sold-out**. Ý nghĩa của từng trạng thái như sau:

* **inactive**: Sản phẩm không hiển thị trên sàn Lazada.
* **deleted**: Sản phẩm đã bị xóa.
* **image-missing**: Sản phẩm thiếu ảnh minh họa.
* **pending**: Sản phẩm đang chờ được cập nhật phiên bản mới nhất.
* **rejected**: Sản phẩm bị loại bỏ do vi phạm chính sách.
* **live**: Sản phẩm đang được hiển thị bình thường và còn hàng.
* **sold-out**: Sản phẩm đã hết hàng.

Từ việc nghiên cứu các trạng thái của sản phẩm trên Lazada kết hợp với trạng thái synced/not\_synced trên Syncpro, bản đồ trạng thái của sản phẩm được xây dựng như sau:



Hình .. Bản đồ trạng thái của sản phẩm trên Lazada.

Để tiện cho việc ký hiệu, trạng thái QC của sản phẩm được chia làm 3 nhóm:

* **QC\_PENDING**: pending.
* **QC\_YES**: approved.
* **QC\_NO**: rejected, liverejected, lock.

Trạng thái hoạt động của sản phẩm thì được chia làm 2 nhóm:

* **PR\_ACT**: live, sold-out.
* **PR\_INACT**: inactive, deleted, image-missing, pending, rejected.

Theo những gì được vẽ trên bản đồ, một sản phẩm có thể trải qua các bước:

1. Sản phẩm mới được tạo ra trên Syncpro có trạng thái là **not\_synced** và chưa có trạng thái trên Lazada. Sản phẩm sẽ được đồng bộ lên Tiki bằng API tạo sản phẩm.
2. Sản phẩm được đồng bộ thành công lên Lazada có trạng thái **synced-QC\_PENDING**. Tại đây sản phẩm có thể rẽ sang 3 hướng:
   1. Sản phẩm được theo dõi trạng thái QC thường xuyên, phát hiện ra nó không vượt qua được vòng kiểm duyệt và trạng thái bị chuyển sang **synced-QC\_NO**.
   2. Sản phẩm được theo dõi trạng thái QC thường xuyên, phát hiện ra nó đã vượt qua được vòng kiểm duyệt và trạng thái chuyển sang **synced-QC\_YES**.
   3. Sản phẩm được người bán thay đổi nội dung và chuyển sang trạng thái **not\_synced-QC\_PENDING**.
3. Sản phẩm ở trạng thái **synced-QC\_NO** thì người bán cần thay đổi nội dung cho phù hợp. Trạng thái sản phẩm sau đó chuyển sang **not\_synced-QC\_NO**.
4. Sản phẩm ở trạng thái **synced-QC\_YES** thì tiếp tục được theo dõi trạng thái hoạt động trên Lazada. Tại đây, sản phẩm có thể rẽ sang 2 hướng:
   1. Sản phẩm không hoạt động bình thường trên Lazada, trạng thái sản phẩm là **synced-QC\_YES-PR\_INACT**.
   2. Sản phẩm hoạt động bình thường trên Lazada, trạng thái sản phẩm là **synced-QC\_YES-PR\_ACT**.
5. Sản phẩm ở trạng thái **not\_synced-QC\_PENDING** thì xóa sản phẩm trên Lazada và tạo lại sản phẩm khác.
6. Sản phẩm ở trạng thái **not\_synced-QC\_NO** thì tạo lại sản phẩm đó trên Lazada.
7. Sản phẩm ở trạng thái **synced-QC\_YES-PR\_INACT** cần được người bán sửa đổi lại nội dung cho phù hợp. Trạng thái sản phẩm sau đó chuyển sang **not\_synced-QC\_YES-PR\_INACT**.
8. Sản phẩm ở trạng thái **synced-QC\_YES-PR\_ACT** tức là đang ở trạng thái đẹp nhất của bản đồ. Tuy nhiên, tại đây người bán có thể sửa đổi nội dung sản phẩm khiến cho trạng thái sản phẩm chuyển sang **not\_synced-QC\_YES-PR\_ACT**.
9. Sản phẩm ở trạng thái **not\_synced-QC\_YES-PR\_INACT** thì xóa sản phẩm trên Lazada và tạo lại sản phẩm khác.
10. Sản phẩm ở trạng thái **not\_synced-QC\_YES-PR\_ACT** thì cập nhật nội dung mới nhất của sản phẩm lên Lazada.
11. Nếu việc tạo sản phẩm hay cập nhật nội dung sản phẩm thất bại (khi gọi API) thì lỗi sẽ được ghi lại đối với sản phẩm đó. Lúc này người bán nên xem xét lại nội dung sản phẩm và sửa lại vì đa phẩn lỗi là do nội dung sản phẩm không hợp lệ. Các sản phẩm có lỗi sẽ được xóa đi trên Lazada và tạo lại (đối với thao tác tạo sản phẩm không thành công) hoặc sẽ được cập nhật lại (đối với thao tác cập nhật sản phẩm không thành công). Sau khi được thử tạo lại/cập nhật lại, sản phẩm có thể đi theo 2 hướng:
    1. Tạo lại/Cập nhật lại thành công, sản phẩm được xóa lỗi và có trạng thái **synced-QC\_PENDING**, sẽ được xử lý ở bước 2.
    2. Tạo lại/Cập nhật lại thất bại, sản phẩm được ghi lỗi mới và sẽ được xử lý lại ở bước 11.

Giống với hai sàn thương mại điện tử kia, trên bản đồ, các hành động xử lý với chữ màu đỏ là các hành động sẽ được chạy tự động theo chu kỳ dưới dạng các job. Khi một job chạy, nó sẽ tìm trong database những sản phẩm có trạng thái phù hợp để thực thi thao tác với sản phẩm đó. Các hành động với chữ màu đen là các hành động phát sinh do tác động của con người.

### Bài toán đồng bộ nội dung sản phẩm giữa các sàn thương mại điện tử.

Giải bài toán đồng bộ nội dung là phải đảm bảo nội dung sản phẩm giữa các sàn thương mại điện tử luôn luôn đồng nhất. Để giải bài toán này, có hai vấn đề cần quan tâm. Thứ nhất, khi người bán hàng tạo hay cập nhật một sản phẩm, phải làm sao để nội dung của sản phẩm đó trên tất cả các sàn là giống nhau. Thứ hai, khi số lượng sản phẩm ở một sàn thay đổi, phải cập nhật được sự thay đổi đó lên các sàn còn lại.

#### Vấn đề đồng bộ nội dung khi tạo hay cập nhật sản phẩm.

Để giải quyết vấn đề đồng bộ nội dung khi tạo hay cập nhật sản phẩm, cần phải làm ba việc. Đầu tiên, cần phải chắt lọc ra những thông tin của sản phẩm là chung cho tất cả các sàn thương mại điện tử (ví dụ: tên sản phẩm, mô tả sản phẩm, giá bán, ...), tạm gọi là các “thông tin chung” của sản phẩm. Sau khi đã xác định được một tập các thông tin chung như vậy, với mỗi thông tin chung, với mỗi sàn thương mại điện tử, phải xác định trường nào trong form sản phẩm là trường mang giá trị của thông tin đó. Cuối cùng, phải tìm hiểu định dạng dữ liệu của mỗi trường mang giá trị để có phương án uốn nắn lại giá trị của thông tin chung cho phù hợp.

Sau khi nghiên cứu kĩ lưỡng form sản phẩm của cả Shopee, Tiki và Lazada, các thông tin chung có thể đồng bộ được ở ba sàn này gồm có:

* Tên sản phẩm.
* Thương hiệu của sản phẩm (chỉ có thể đồng bộ được với Lazada và Tiki).
* Giá thị trường của sản phẩm.
* Giá bán của sản phẩm.
* Các hình ảnh minh họa sản phẩm.
* Mô tả về sản phẩm.
* Mô tả ngắn về tính năng nổi bật của sản phẩm.
* Cân nặng của gói hàng.
* Chiều rộng của gói hàng.
* Chiều dài của gói hàng.
* Chiều cao của gói hàng.
* SKU mà người bán đặt cho sản phẩm.
* Số lượng hàng trong kho.

Công việc tiếp theo là tìm trong form sản phẩm của từng sàn trường mang giá trị thông tin chung. Kết quả nằm trong bảng sau:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Shopee** | **Tiki** | **Lazada** |
| Tên | name | name | Request → Attributes → name |
| Thương hiệu |  | attributes → brands | Request → Attributes → brand |
| Giá thị trường |  | market\_price | Request → Skus → Sku → price |
| Giá bán | price | variants → price | Request → Skus → Sku → special\_price |
| Hình ảnh minh họa | images | images | Request → Skus → Sku → Images |
| Mô tả | description | description | Request → Attributes → description |
| Tính năng nổi bật |  | attributes → product\_top\_features | Request → Attributes → short\_description |
| Cân nặng gói hàng | weight | attributes → product\_weight | Request → Skus → Sku → package\_weight |
| Chiều rộng gói hàng | package\_widht | attributes → product\_width | Request → Skus → Sku → package\_width |
| Chiều dài gói hàng | package\_length | attributes → product\_length | Request → Skus → Sku → package\_length |
| Chiều cao gói hàng | package\_height | attributes → product\_height | Request → Skus → Sku → package\_height |
| SKU người bán đặt | item\_sku | variants → sku | Request → Skus → Sku → SellerSku |
| Số lượng | stock | variangs → quantity | Request → Skus → Sku → quantity |

Việc cuối cùng là định dạng lại giá trị của thông tin chung sao cho phù hợp với form sản phẩm của từng sàn thương mại điện tử. Bước này rất quan trọng vì cùng một thông tin chung nhưng định dạng để biểu thị thông tin chung đó trong form sản phẩm của mỗi sàn là khác nhau. Ví dụ: Lazada cho phép mô tả sản phẩm là HTML nhưng Shopee thì chỉ hỗ trợ mô tả sản phẩm với chữ thuần (plain text), do đó phải định dạng mô tả sản phẩm sang HTML để phù hợp với Lazada và sang chữ thuần để phù hợp với Shopee. Kết quả nghiên cứu các phương thức định dạng nằm trong bảng sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Định dạng mặc định** | **Định dạng sang Shopee** | **Định dạng sang Tiki** | **Định dạng sang Lazada** |
| Tên | Chữ thuần | Giữ nguyên | Giữ nguyên | Giữ nguyên |
| Thương hiệu | Chữ thuần |  | Giữ nguyên | Giữ nguyên |
| Giá thị trường | Số tự nhiên |  | Giữ nguyên | Giữ nguyên |
| Giá bán | Số tự nhiên | Giữ nguyên | Giữ nguyên | Giữ nguyên |
| Hình ảnh minh họa | Mảng các địa chỉ url | Chuyển mảng địa chỉ url thành mảng object dạng {url: url} | Giữ nguyên | Chuyển mảng các địa chỉ url thành nội dung XML của thẻ Images |
| Mô tả | HTML | Chuyển HTML sang chữ thuần. | Giữ nguyên | Giữ nguyên |
| Tính năng nổi bật | HTML |  | Chuyển HTML sang chữ thuần | Giữ nguyên |
| Cân nặng gói hàng | Số thập phân | Giữ nguyên | Giữ nguyên | Giữ nguyên |
| Chiều rộng gói hàng | Số tự nhiên | Giữ nguyên | Giữ nguyên | Giữ nguyên |
| Chiều dài gói hàng | Số tự nhiên | Giữ nguyên | Giữ nguyên | Giữ nguyên |
| Chiều cao gói hàng | Số tự nhiên | Giữ nguyên | Giữ nguyên | Giữ nguyên |
| SKU người bán đặt | Chữ thuần | Giữ nguyên | Giữ nguyên | Giữ nguyên |
| Số lượng | Số tự nhiên | Giữ nguyên | Giữ nguyên | Giữ nguyên |

Như vậy, vấn đề đồng bộ nội dung khi tạo hay cập nhật sản phẩm cơ bản đã có lời giải.

#### Vấn đề đồng bộ số lượng sản phẩm.

Để giải quyết vấn đề đồng bộ số lượng sản phẩm, cần phải làm ba việc: nắm bắt sự thay đổi số lượng của sản phẩm trên các sàn thương mại điện tử, tính toán lại số lượng sản phẩm thực tế, cập nhật số lượng sản phẩm đã tính lại lên các sàn.

Điểm mấu chốt để giải quyết vấn đề này là làm sao nắm bắt được sự thay đổi về số lượng của sản phẩm trên các sàn thương mại điện tử. Cách tiếp cận ngây thơ nhất là cứ theo một chu kỳ cố định (khoảng vài phút) thì lại lấy thông tin của sản phẩm từ tất cả các sàn về, đối chiếu số lượng với dữ liệu của sản phẩm trong database để nhận ra sự thay đổi. Cách này hoạt động, nhưng rất tốn tài nguyên vì phải theo dõi thông tin sản phẩm thường xuyên.

Cách tiếp cận thứ hai đó là dựa vào sự hỗ trợ của các sàn thương mại điện tử về webhook. Webhook giúp cho nhà phát triển nhận được các sự kiện từ hệ thống của một sàn thương mại điện tử thông qua một Địa chỉ URL gọi lại (Callback URL) được nhà phát triển cài đặt trước đó. Trong trường hợp này, webhook sẽ được sử dụng để lắng nghe sự kiện có đơn hàng phát sinh đối với một sản phẩm. Khi có sự kiện đơn hàng phát sinh xảy ra thì mới đi lấy thông tin sản phẩm về kiểm tra. Webhook hiệu quả vì giúp tránh được việc phải theo dõi thông tin sản phẩm thường xuyên. Tuy nhiên, chỉ mới có Shopee là hỗ trợ webhook đầy đủ; Tiki thì hỗ trợ webhook rất hạn chế; Lazada thậm chí còn không hỗ trợ webhook. Vì vậy, hiện tại chỉ có thể dùng webhook cho sàn Shopee, còn với Tiki và Lazada, vẫn phải dùng cách tiếp cận ngây thơ.

Để tính toán số lượng sản phẩm còn lại thực tế, cần xây dựng một công thức tính tổng quát. Giả sử một sản phẩm đang được được đồng bộ lên n sàn thương mại điện tử. Tại thời điểm t1, số lượng sản phẩm còn lại thực tế là st1 và đã được đồng bộ lên tất cả n sàn. Tại thời điểm t2 sau thời điểm t1, theo dõi số lượng sản phẩm trên n sàn thu được kết quả lần lượt là m1, m2, m3, ..., mn. Công thức để tính số lượng sản phẩm còn lại thực tế tại thời điểm t2 là:

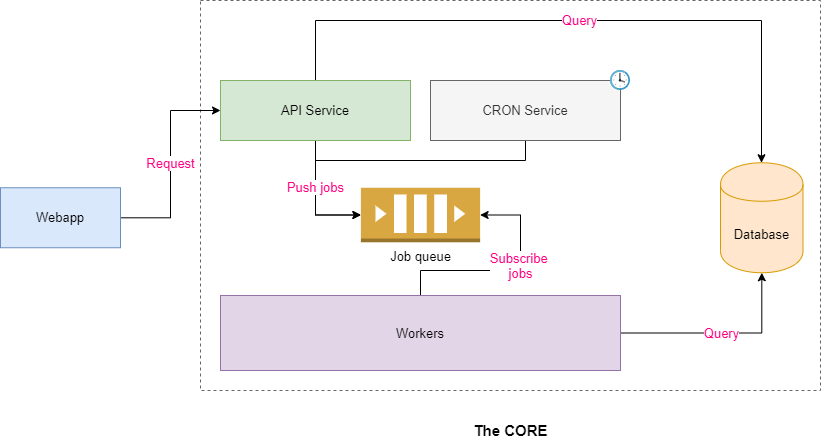
st2 = (m1 + m2 + m3 + ... + mn)– (n-1)\*st1

Sau khi tính được số lượng sản phẩm còn lại thực tế, cần phải cập nhật giá trị này lên tất cả các sàn thương mại điện tử. Vấn đề này đã được giải quyết trong bài toán đẩy sản phẩm: chỉ cần chỉnh sửa số lượng và đổi trạng thái của sản phẩm đối với tất cả các sàn thành **not\_synced**.

Như vậy, vấn đề đồng bộ số lượng sản phẩm về cơ bản chưa có một lời giải tối ưu trọn vẹn.

## Thiết kế kiến trúc hệ thống.

Hệ thống Syncpro được thiết kế theo kiến trúc **Job Queue** với sơ đồ như sau:



Hình .. Sơ đồ kiến trúc hệ thống Syncpro.

Các thành phần chính của hệ thống bao gồm: **Database**, **API Service**, **CRON Service**, **Job queue**, **Workers** và **Webapp**. Các thành phần này có sự phối hợp chặt chẽ với nhau để cùng thực hiện công việc. Dưới đây là trình bày chi tiết về vai trò của từng thành phần trong hệ thống.

### Database.

**Database** hay **cơ sở dữ liệu** là nơi lưu trữ tất cả dữ liệu của hệ thống. Bởi vậy, nó là thành phần cốt lõi và cực kỳ quan trọng của hệ thống. Database có vai trò là nơi để các thành phần khác trong hệ thống ghi và truy vấn dữ liệu. Dữ liệu trong **Database** gồm có dữ liệu người dùng, dữ liệu sản phẩm, dữ liệu về các danh mục ngành hàng, các thuộc tính, thương hiệu sử dụng để tạo sản phẩm, v.v...

### API Service.

**API Service** làm nhiệm vụ định hình các tính năng của hệ thống thành các đầu API và cung cấp chúng cho **Webapp**. **API Service** có thể thao tác trực tiếp với **Database** và cũng có thể tạo − đẩy job vào **Job queue**.

### CRON Service.

**CRON Service** làm nhiệm vụ lập lịch chạy các job tự động, ví dụ như các job cập nhật Danh mục ngành hàng, các job cập nhật Thuộc tính sản phẩm, ...

Nếu một job A có tần suất chạy là một lần một ngày thì cứ sau đúng một ngày, **CRON Service** sẽ đẩy một job A vào trong **Job queue**.Nếu một job B có tần suất chạy là một lần một tuần thì cứ sau đúng một tuần, **CRON Service** sẽ đẩy một job B vào trong **Job queue**. Nhiệm vụ của **CRON Service** là đảm bảo các job tự động sẽ luôn được sinh ra đúng thời điểm.

**CRON Service** không thao tác trực tiếp với **Database**.

### Job queue.

**Job queue** là nơi lưu trữ, quản lý và điều phối các job. Tại đây, các job được tổ chức theo cấu trúc dữ liệu hàng đợi (queue): job được đẩy vào ở một đầu và được lấy ra thực hiện ở đầu kia. Vậy nên nó mới có tên gọi là **Job queue**.

Chức năng lưu trữ job thể hiện ở việc nó lưu lại tất cả các job được đẩy vào cho đến khi job đó được lấy ra thực hiện. Chức năng quản lý cho phép chỉnh sửa hoặc xóa bỏ một job còn đang ở trong hàng đợi. Chức năng điều phối là chức năng quan trọng nhất: quyết định job nào trong hàng đợi sẽ được chạy tiếp theo. Hàng đợi bên trong **Job queue** không phải là hàng đợi đơn giản hoạt động theo kiểu First In First Out. Mỗi job trong hàng đợi khi đẩy vào đều có một độ ưu tiên nhất định. Job nào có độ ưu tiên cao hơn sẽ được ưu tiên thực hiện trước.

Các job được đẩy vào **Job queue** bởi hai thành phần: **API Service** và **CRON Service**.

### Workers.

**Workers** là thành phần xử lý chính trong hệ thống. Tất cả các job ở bên trong **Job queue** đều được lấy ra thực hiện bởi **Workers**.

Khi chạy, **Workers** sẽ lắng nghe ở đầu ra của **Job queue**. Nếu thấy phía đầu ra có job sẵn sàng, nó sẽ lấy job đó ra để thực hiện. Sau khi thực hiện xong, **Workers** sẽ báo lại kết quả về cho **Job queue** để **Job queue** có hành động xử lý thích hợp.

### Webapp.

**Webapp** là thành phần bên ngoài cùng của hệ thống, là bộ mặt của hệ thống, nơi người sử dụng nhìn thấy và chạm vào được các tính năng. **Webapp** đóng vai trò cầu nối giữa người sử dụng và các thành phần bên trong lõi hệ thống (**The CORE**). **Webapp** một mặt cung cấp giao diện đồ họa thân thiện cho người sử dụng. Mặt còn lại, nó sử dụng các đầu API mà **API Service** cung cấp để thao tác với phần lõi.

# Triển khai giải pháp

Tại bước triển khai giải pháp, từng thành phần của hệ thống sẽ được cài đặt để hiện thực hóa ý tưởng giải quyết bài toán. Theo kiến trúc hệ thống đã thiết kế thì phải cài đặt 6 thành phần, đó là: **Database**, **API Service**, **CRON Service**, **Job queue**, **Workers**, **Webapp**.

Ngôn ngữ lập trình được chọn để triển khai toàn bộ hệ thống từ front-end đến back-end là ngôn ngữ **JavaScript**. Lý do ngôn ngữ này được lựa chọn vì thứ nhất, nó là ngôn ngữ đa năng, có thể sử dụng để lập trình cho cả front-end, cả back-end (với Node JS) và cả mobile app (với React Native).Thứ hai, JavaScript có cộng đồng nhà phát triển hết sức đông đảo nên nếu có khúc mắc, người sử dụng luôn tìm được sự hỗ trợ dễ dàng. Thứ ba, kho package của JavaScript cực kỳ phong phú và có một hệ thống quản lý package khoa học. Rất nhiều giải pháp cho các vấn đề cơ bản đều đã được cài đặt thành các package JavaScript, sẵn sàng cho người sử dụng tải về dùng, rất thuận tiện. Cuối cùng, JavaScript là ngôn ngữ đơn luồng, xử lý bất đồng bộ, rất phù hợp cho việc cài đặt mô hình Job Queue của Syncpro.

## Triển khai Database.

Database của Syncpro sử dụng hệ quản trị cơ sở dữ liệu **MongoDB**. MongoDB là hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) nên không có các bảng và khóa ngoài. Đơn vị dữ liệu trong MongoDB là các **document**. Một document có chứa các trường (field) và các giá trị tương ứng. Một **collection** bao gồm nhiều document và các document thường có chung cấu trúc. Một **database** thường có nhiều collection. MongoDB thường quản trị nhiều database.



Hình .. Tổ chức dữ liệu bên trong MongoDB.

MongoDB rất phù hợp làm giải pháp cho bài toán của Syncpro. Thứ nhất, vì các dữ liệu sản phẩm không có cấu trúc rõ ràng, sản phẩm thuộc danh mục ngành hàng này sẽ có những trường thông tin khác với sản phẩm thuộc danh mục ngành hàng khác, nên việc thiết kế được một lược đồ cơ sở dữ liệu quan hệ cho bài toán này là rất vất vả. Với MongoDB, cấu trúc của một document thay đổi linh hoạt nên không cần phải xác định tất cả các trường thông tin cần lưu trữ ngay từ đầu. Thứ hai, MongoDB lưu trữ dữ liệu dưới dạng các tệp Binary JSON nên nó làm việc rất tốt với ngôn ngữ lập trình JavaScript, vì trong JavaScript, JSON đã được hỗ trợ sẵn.

Công cụ được sử dụng để thao tác với MongoDB từ trong code JavaScript (cụ thể là Node JS) là **Mongoose ODM**. Một số tính năng nổi bật của Mongoose ODM có thể kể đến như:

* Giúp định hình các document chặt chẽ hơn với Schema (lược đồ).
* Định kiểu dữ liệu cho các trường trong document, kiểm tra kiểu dữ liệu của các trường trước khi lưu document vào database.
* Xây dựng câu truy vấn dễ dàng, trả về kết quả dạng JSON thân thiện.

Mongoose ODM sử dụng các Schema (lược đồ) và kiểu dữ liệu để giúp cho cấu trúc document có thêm tính chặt chẽ nhưng vẫn giữ được sự mềm dẻo vốn có. Schema có thể được sửa đổi dễ dàng mà không làm ảnh hưởng đến dữ liệu.

Có hai thành phần trong hệ thống cùng thao tác với Database là API Service và Workers. Nếu cả hai cùng cài Mongoose ODM và định nghĩa Schema cho riêng mình thì Schema của thành phần này phải là bản sao Schema của thành phần kia. Khi sửa Schema thì sẽ phải sửa ở cả hai chỗ cùng lúc. Nếu sửa Schema ở API Service mà quên không sửa Schema ở Workers hoặc ngược lại thì sẽ gây ra hiện tượng không nhất quán về cấu trúc dữ liệu. Cách này vừa mất công sức lại vừa rủi ro cao.

Giải pháp tốt hơn để giải quyết vấn đề này đó là viết riêng một Node JS package làm đầu mối thao tác với Database. Package có tên là @syncpro/mongo. Bên trong nó sử dụng Mongoose ODM để kết nối và thao tác với Database. Nó cũng định nghĩa ra bộ Schema chuẩn cho hệ thống Syncpro. API Service và Workers sẽ phải cài package này và sử dụng bộ Schema chuẩn đã định nghĩa. Khi cần sửa đổi Schema, chỉ cần sửa đổi trong package này, gắn cho nó số hiệu phiên bản mới và cập nhật nó ở những nơi nó được sử dụng.

Mã nguồn của package nằm tại địa chỉ: <https://gitlab.com/syncpro/mongo>.

Dưới đây là các collection hiện có trong Database và ý nghĩa của chúng:

* **ShopeeLogistics**: Lưu trữ thông tin tất cả Đơn vị vận chuyển trên Shopee.
* **ShopeeCategories**: Lưu trữ thông tin tất cả Danh mục ngành hàng trên Shopee theo dạng phẳng.
* **ShopeeCategoryTree**: Lưu trữ thông tin tất cả Danh mục ngành hàng trên Shopee theo dạng cây.
* **ShopeeAttributes**: Lưu trữ tất cả các Thuộc tính riêng của sản phẩm tương ứng với Danh mục ngành hàng trên Shopee.
* **TikiCategories**: Lưu trữ thông tin tất cả Danh mục ngành hàng trên Tiki theo dạng phẳng.
* **TikiCategoryTree**: Lưu trữ thông tin tất cả Danh mục ngành hàng trên Tiki theo dạng cây.
* **TikiProductForms**: Lưu trữ các Thuộc tính riêng của sản phẩm tương ứng với Danh mục ngành hàng trên Tiki.
* **LazadaBrands**: Lưu trữ tất cả Thương hiệu hiện có trên Lazada.
* **LazadaCategories**: Lưu trữ tất cả Danh mục ngành hàng trên Lazada theo dạng phẳng.
* **LazadaCategoryTree**: Lưu trữ tất cả Danh mục ngành hàng của Lazada theo dạng cây.
* **LazadaProductForm**: Lưu trữ các Thuộc tính riêng của sản phẩm tương ứng với Danh mục ngành hàng trên Lazada.
* **Products**: Lưu trữ thông tin tất cả sản phẩm được quản lý bởi Syncpro.
* **Users**: Lưu trữ thông tin tất cả người dùng của Syncpro.

## Cài đặt API Service.

API Service được cài đặt và chạy trên nền tảng **Node JS**, sử dụng framework **Express**.

**Node JS** là một nền tảng (platform) được xây dựng trên V8 JavaScript Engine của Chrome. Node JS không phải là một ngôn ngữ lập trình, không phải là một framework. Nó là môi trường để thực thi các kịch bản được viết bằng ngôn ngữ JavaScript bên phía server. Phần lõi của Node JS được phát triển bằng ngôn ngữ C/C++ nên có tốc độ thực thi nhanh. Do được xây dựng trên V8 JavaScript Engine nên code JavaScript chạy trên Node JS cũng giống như JavaScript chạy trên trình duyệt: hướng sự kiện, đơn luồng và bất đồng bộ. Node JS là miễn phí, mã nguồn mở và có một hệ sinh thái các package vô cùng phong phú - **npm**. Trang chủ của Node JS: <https://nodejs.org>.

**Express** là một framework dùng để xây dựng các ứng dụng web chạy trên Node JS. Express cung cấp các cơ chế:

* Cho phép nhà phát triển viết các hàm xử lý HTTP request với nhiều phương thức (GET, POST, PUT, DELETE) và nhiều tuyến (routes).
* Tích hợp với các “động cơ vẽ khung nhìn” (view rendering engine) để tạo khung nhìn (view) bằng cách bơm dữ liệu vào các khuôn mẫu (view template).
* Cấu hình một số cài đặt thông thường của một ứng dụng web như số cổng sử dụng để kết nối, v.v...
* Cho phép nhà phát triển chèn thêm các hành động tiền xử lý HTTP request bằng cách chèn các middleware tại bất kỳ điểm nào trong luồng xử lý.

Express nhẹ và dễ sử dụng. Nó đặc biệt phù hợp cho việc xây dựng các dịch vụ API như API Service của Syncpro. Trang chủ của Express: <https://expressjs.com>.

Dưới đây là các đầu API đã được cài đặt trong API Service:

* GET /products: Lấy về danh sách sản phẩm của người dùng trên Syncpro.
* GET /products/:product\_id: Lấy về thông tin chi tiết của một sản phẩm trên Syncpro.
* POST /products: Phát hành sản phẩm mới lên nhiều sàn thương mại điện tử cùng lúc.
* PUT /products/:product\_id: Chỉnh sửa sản phẩm trên nhiều sàn thương mại điện tử cùng lúc.
* GET /shopee/category-tree: Lấy về Cây danh mục ngành hàng của Shopee.
* GET /shopee/logistics: Lấy về danh sách Đơn vị vận chuyển của Shopee.
* GET /shopee/product-form/:category\_id: Lấy về các thuộc tính sản phẩm tương ứng với một Danh mục ngành hàng của Shopee.
* GET /tiki/category-tree: Lấy về Cây danh mục ngành hàng của Tiki.
* GET /tiki/product-form/:category\_id: Lấy về các thuộc tính sản phẩm tương ứng với một Danh mục ngành hàng của Tiki.
* GET /lazada/brands: Tìm kiếm Thương hiệu của Lazada với từ khóa.
* GET /lazada/category-tree: Lấy về Cây danh mục ngành hàng của Lazada.
* GET /product-form/:category\_id: Lấy về các thuộc tính sản phẩm tương ứng với một Danh mục ngành hàng của Lazada.
* POST /static: Tải các tệp tĩnh (hình ảnh, ...) lên hệ thống Syncpro, sử dụng để cung cấp URL hình ảnh cho việc tạo sản phẩm.

Khi xử lý các HTTP request, API Service có thể đẩy một số job vào trong Job queue để cho Workers xử lý bất đồng bộ, còn mình thì nhanh chóng trả về response cho phía Webapp. Các job mà API Service đẩy vào trong Job queue gồm có:

* **shopee\_create\_product**: Job này được đưa vào queue khi gọi API phát hành sản phẩm mới cùng với tùy chọn đẩy sản phẩm lên sàn Shopee.
* **tiki\_create\_product**: Job này được đưa vào queue khi gọi API phát hành sản phẩm mới cùng với tùy chọn đẩy sản phẩm lên sàn Tiki.
* **lazada\_create\_product**: Job này được đưa vào queue khi gọi API phát hành sản phẩm mới cùng với tùy chọn đẩy sản phẩm lên sàn Lazada.

Mã nguồn của API Service nằm tại địa chỉ: <https://gitlab.com/syncpro/api-service>.

## Cài đặt CRON Service.

CRON Service cũng được cài đặt và chạy trên nền tảng Node JS. CRON Service sử dụng package **cron** của npm. Package này sử dụng biểu thức hẹn giờ gần giống với biểu thức cron của hệ điều hành Linux. Thông tin chi tiết hơn và hướng dẫn sử dụng package nằm tại địa chỉ: <https://www.npmjs.com/package/cron>.

Các job, ý nghĩa và lịch chạy của từng job được liệt kê trong bảng sau:

|  |  |  |
| --- | --- | --- |
| **Tên job** | **Ý nghĩa** | **Lịch chạy** |
| shopee\_crawl\_categories | Cập nhật các Danh mục ngành hàng của Shopee. | 0h sáng mỗi ngày |
| shopee\_crawl\_attributes | Cập nhật các Thuộc tính sản phẩm của Shopee. | 0h sáng, 2 ngày một lần. |
| shopee\_crawl\_logistics | Cập nhật thông tin các Đơn vị vận chuyển trên Shopee. | 0h sáng mỗi ngày |
| shopee\_crawl\_product | Theo dõi thông tin sản phẩm trên Shopee, lấy thông tin sản phẩm về database. | 5 phút một lần |
| shopee\_update\_product | Cập nhật nội dung lên Shopee cho các sản phẩm đang có trạng thái phù hợp. | 5 phút một lần |
| shopee\_delete\_recreate\_product | Xóa sản phẩm và tạo lại trên Shopee cho các sản phẩm đang có trạng thái phù hợp. | 5 phút một lần |
| shopee\_retry\_create\_product | Thử tạo lại sản phẩm trên Shopee cho các sản phẩm bị lỗi. | 5 phút một lần |
| lazada\_refresh\_token | Làm tươi access token của Lazada | 0h sáng, 6 ngày một lần |
| lazada\_crawl\_brands | Cập nhật các Thương hiệu trên Lazada | 1h sáng thứ Bảy hằng tuần |
| lazada\_crawl\_categories | Cập nhật các Danh mục ngành hàng của Lazada. | 0h sáng mỗi ngày |
| lazada\_crawl\_product\_forms | Cập nhật các Thuộc tính sản phẩm của Lazada. | 0h sáng, 2 ngày một lần |
| lazada\_get\_product\_qc\_status | Theo dõi QC status của sản phẩm trên Lazada. | 5 phút một lần |
| lazada\_crawl\_product | Theo dõi thông tin sản phẩm trên Lazada, lấy thông tin sản phẩm về database. | 5 phút một lần |
| lazada\_retry\_create\_or\_update\_product | Thử tạo lại sản phẩm hoặc cập nhật lại sản phẩm lên Lazada đối với các sản phẩm bị lỗi. | 5 phút một lần |
| lazada\_delete\_recreate\_product | Xóa sản phẩm và tạo lại trên Lazada đối với các sản phẩm đang có trạng thái phù hợp. | 5 phút một lần |
| lazada\_recreate\_product | Tạo lại sản phẩm trên Lazada đối với các sản phẩm đang có trạng thái phù hợp. | 5 phút một lần |
| lazada\_update\_product | Cập nhật nội dung sản phẩm lên Lazada đối với các sản phẩm đang có trạng thái phù hợp. | 5 phút một lần |
| tiki\_crawl\_categories | Cập nhật các Danh mục ngành hàng của Tiki. | 0h sáng mỗi ngày |
| tiki\_crawl\_product\_forms | Cập nhật các Thuộc tính sản phẩm của Tiki. | 0h sáng, 2 ngày một lần |
| tiki\_track\_product\_requests | Theo dõi yêu cầu tạo sản phẩm trên Tiki. | 5 phút một lần |
| tiki\_crawl\_product\_by\_seller\_sku | Theo dõi thông tin sản phẩm trên Tiki, lấy thông tin sản phẩm về database với những sản phẩm được tạo thành công. | 5 phút một lần |
| tiki\_retry\_create\_product\_request | Thử tạo lại yêu cầu tạo sản phẩm trên Tiki với những sản phẩm bị lỗi. | 5 phút một lần |
| tiki\_delete\_recreate\_product\_request | Xóa yêu cầu tạo sản phẩm và tạo lại yêu cầu khác trên Tiki với những sản phẩm đang có trạng thái phù hợp. | 5 phút một lần |
| tiki\_recreate\_product\_request | Tạo lại yêu cầu tạo sản phẩm trên Tiki với những sản phẩm đang có trạng thái phù hợp. | 5 phút một lần |
| tiki\_request\_update\_products | Yêu cầu cập nhật nội dung sản phẩm trên Tiki với những sản phẩm đang có trạng thái phù hợp. | 5 phút một lần |

Mã nguồn của CRON Service nằm tại địa chỉ: <https://gitlab.com/syncpro/crontab>.

## Triển khai Job queue.

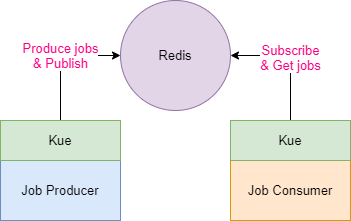
Job queue của Syncpro sử dụng package **Kue**. Kue là hàng đợi công việc có độ ưu tiên được xây dựng để chạy trên Node JS, sử dụng **Redis** làm cơ sở dữ liệu lưu trữ job và làm kênh truyền sự kiện. Trang github của Kue: <https://github.com/Automattic/kue>.

Để sử dụng được Kue, phải cài đặt Redis. Redis là công cụ dùng để lưu trữ các dữ liệu có cấu trúc vào trong bộ nhớ RAM của máy tính. Dữ liệu được lưu vào dưới dạng key-value và được truy vấn ra bằng key. Redis thường được sử dụng như một cơ sở dữ liệu có tốc độ đọc ghi rất cao. Các cấu trúc dữ liệu được Redis hỗ trợ gồm có: chuỗi (strings), mã hash (hashes), danh sách (lists), tập hợp (sets), v.v...

Một tính năng nữa của Redis đó là được sử dụng làm kênh giao tiếp giữa một bên phát sự kiện (publisher) và một bên thu sự kiện (subscriber). Redis cho phép tạo ra các kênh giao tiếp. Khi một publisher phát một sự kiện vào trong một kênh, tất cả các subscriber đang lắng nghe kênh đó đều nhận được sự kiện mà publisher phát ra. Cơ chế này có tên gọi là pub-sub (viết tắt của publish-subscribe).

Trang chủ của Redis: [https://redis.io](https://redis.io/).

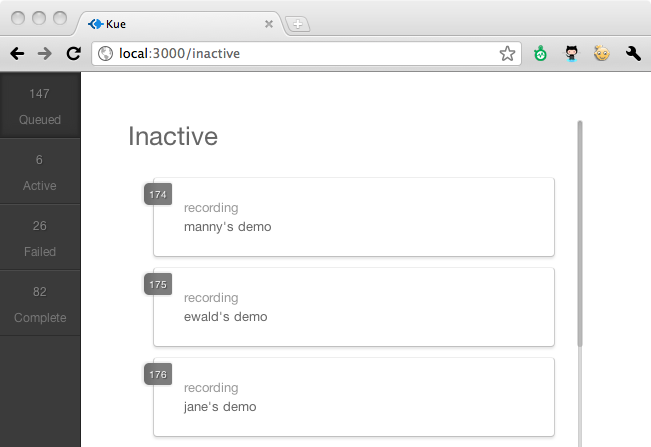
Kue dựa vào khả năng lưu trữ và cơ chế pub-sub của Redis để cài đặt chức năng của mình. Nếu có job được đẩy vào, nó sẽ lưu trữ job đó trong Redis và truyền sự kiện này cho các bên tiêu thụ job thông qua Redis. Các bên tiêu thụ job nhận được sự kiện này sẽ lấy job từ trong Redis ra để thực hiện.



Hình .. Cách thức hoạt động của Kue - Redis.

Khi sử dụng Kue, bên phía tạo job phải cài Kue và bên phía lấy job ra thực hiện cũng phải cài Kue. Kue bên phía tạo job và Kue bên phía thực hiện job phải cùng kết nối tới một Redis server.

Ngoài khả năng đẩy job vào và lấy job ra, Kue còn cho phép tùy chỉnh các job, ví dụ như cho phép đặt độ ưu tiên của job, đặt số lần thử lại nếu job đó bị thất bại. Kue còn có cả giao diện trực quan để theo dõi tình trạng và thao tác với các job trong hàng đợi.



Hình .. Giao diện của Kue.

## Cài đặt Workers.

Workers được cài đặt và chạy trên nền tảng Node JS. Nó lắng nghe Job queue để lấy job ra thực hiện. Các job được phân biệt với nhau bằng tên. Workers ánh xạ mỗi job với một hàm xử lý. Khi nhận được job, nó sẽ chạy hàm xử lý tương ứng với job đó.

Tất cả các tác vụ xử lý chính của hệ thống đều được đẩy vào Job queue dưới dạng các job, và vì thế đều được thực hiện bởi Workers. Từ việc crawl danh mục ngành hàng, crawl thuộc tính đến việc tạo sản phẩm, cập nhật sản phẩm, v.v... Có thể nói rằng, nếu Database là trái tim của hệ thống thì Workers chính là bộ não của hệ thống.

Bảng bên dưới liệt kê các job và hành động xử lý tương ứng của Workers đã được cài đặt trong hàm xử lý thực hiện job đó. Chú ý: trong bảng này có sử dụng từ **“Trạng thái phù hợp”** để chỉ những trạng thái của sản phẩm là phù hợp để thực hiện hành động xử lý. Xin hãy tham khảo các **Bản đồ trạng thái của sản phẩm** ở mục [3.1.1](#_Bài_toán_đẩy) để biết với mỗi hành động xử lý, những trạng thái sản phẩm nào là phù hợp.

|  |  |
| --- | --- |
| **Tên job** | **Hành động xử lý** |
| shopee\_crawl\_categories | 1. Gọi API **GetCategoriesByCountry** của Shopee lấy về các Danh mục ngành hàng (dạng phẳng). 2. Lưu các Danh mục ngành hàng vào collection **ShopeeCategories**. 3. Từ các Danh mục ngành hàng dạng phẳng, xây dựng Cây Danh mục ngành hàng. 4. Lưu Cây danh mục ngành hàng vào collection **ShopeeCategoryTree**. |
| shopee\_crawl\_logistics | 1. Gọi API **GetLogistics** của Shopee lấy về thông tin các Đơn vị vận chuyển. 2. Lưu các đơn vị vận chuyển vào collection **ShopeeLogistics**. |
| shopee\_crawl\_attributes | 1. Truy vấn collection **ShopeeCategories** lấy ra các Danh mục ngành hàng sơ cấp. 2. Với mỗi danh mục ngành hàng sơ cấp:    1. Gọi API **GetAttributes** để lấy về các thuộc tính sản phẩm.    2. Lưu các thuộc tính vào collection **ShopeeAttributes**. |
| shopee\_create\_product | 1. Tìm sản phẩm trong database bằng ID sản phẩm có trong job data. 2. Gọi API **Add** để tạo sản phẩm với form sản phẩm được tìm thấy. |
| shopee\_crawl\_product | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm, gọi API **GetItemDetail** lấy về thông tin chi tiết của sản phẩm, lưu vào collection **Products**. |
| shopee\_update\_product | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm, gọi các API **UpdateItem**, **UpdateItemImg**, **UpdatePrice**, **UpdateStock** để cập nhật toàn bộ nội dung sản phẩm lên Shopee. |
| shopee\_delete\_recreate\_product | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy:    1. Gọi API **Delete** để xóa sản phẩm trên Shopee.    2. Gọi API **Add** để tạo lại sản phẩm với nội dung mới trên Shopee. |
| shopee\_retry\_create\_product | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi API **Add** để thử tạo lại sản phẩm bị lỗi. |
| tiki\_crawl\_categories | 1. Gọi API **Get categories** của Tiki để lấy về các Danh mục ngành hàng dạng phẳng. 2. Lưu các Danh mục ngành hàng vào collection **TikiCategories**. 3. Dựng Cây danh mục ngành hàng từ các Danh mục ngành hàng dạng phẳng. 4. Lưu Cây danh mục ngành hàng vào collection **TikiCategoryTree**. |
| tiki\_crawl\_product\_forms | 1. Truy vấn collection **TikiCategories** lấy ra các Danh mục ngành hàng sơ cấp. 2. Với mỗi Danh mục tìm thấy:    1. Gọi API **Get category detail** của Tiki để lấy về các thuộc tính sản phẩm.    2. Lưu các thuộc tính sản phẩm vào collection **TikiProductForms**. |
| tiki\_create\_product | 1. Tìm sản phẩm trong database bằng ID sản phẩm có trong job data. 2. Gọi API **Create Product Request** để tạo yêu cầu sản phẩm với form sản phẩm được tìm thấy. |
| tiki\_track\_product\_requests | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi API **Tracking a product request** lấy về thông tin yêu cầu sản phẩm trên Tiki, lưu vào collection **Products**. |
| tiki\_crawl\_product\_by\_seller\_sku | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi API **Get a product** lấy về thông tin chi tiết của sản phẩm, lưu vào collection **Products**. |
| tiki\_delete\_recreate\_product\_request | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy:    1. Gọi API **Delete a product request** để xóa yêu cầu sản phẩm trên Tiki.    2. Gọi API **Create Product Request** để tạo lại yêu cầu sản phẩm với nội dung mới trên Tiki. |
| tiki\_recreate\_product\_request | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi API **Create Product Request** để tạo lại yêu cầu sản phẩm trên Tiki. |
| tiki\_request\_update\_products | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi các API **Update variant price/quantity/active**, **Update product market\_price/image/images** để yêu cầu cập nhật thông tin mới nhất của sản phẩm lên Tiki. |
| tiki\_retry\_create\_product\_request | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, xóa yêu cầu sản phẩm cũ (nếu có) rồi gọi API **Create Product Request** để thử tạo lại yêu cầu sản phẩm khác. |
| lazada\_refresh\_token | 1. Truy vấn collection **Users** những người dùng đang có Lazada access token. 2. Với mỗi người dùng, gọi API **Refresh access\_token** để lấy access token và refresh token mới, lưu vào dữ liệu người dùng đó. |
| lazada\_crawl\_categories | 1. Gọi API **GetCategoryTree** để lấy về Cây danh mục ngành hàng của Lazada. 2. Lưu Cây danh mục ngành hàng vào collection **LazadaCategoryTree**. 3. Duỗi phẳng Cây danh mục ngành hàng để có các Danh mục ngành hàng dạng phẳng. 4. Lưu các Danh mục ngành hàng dạng phẳng vào collection **LazadaCategories**. |
| lazada\_crawl\_brands | 1. Gọi API **GetBrands** lấy về các Thương hiệu trên Lazada. 2. Lưu các Thương hiệu vào collection **LazadaBrands**. |
| lazada\_create\_product | 1. Tìm sản phẩm trong database bằng ID sản phẩm có trong job data. 2. Gọi API **CreateProduct** để tạo sản phẩm với form sản phẩm được tìm thấy. |
| lazada\_crawl\_product\_forms | 1. Truy vấn collection **LazadaCategories** để lấy ra các Danh mục ngành hàng sơ cấp. 2. Với mỗi Danh mục ngành hàng sơ cấp:    1. Gọi API **GetCategoryAttributes** để lấy về các thuộc tính sản phẩm.    2. Lưu các thuộc tính sản phẩm vào collection **LazadaProductForm**. |
| lazada\_get\_access\_token | 1. Lấy authorization code từ job data. 2. Gọi API **Generate access\_token** để lấy access token từ authorization code. |
| lazada\_get\_product\_qc\_status | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi API **GetQcStatus** lấy về thông tin QC sản phẩm, lưu vào collection **Products**. |
| lazada\_crawl\_product | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi API **GetProductItem** lấy về thông tin chi tiết của sản phẩm trên Lazada, lưu vào collection **Products**. |
| lazada\_delete\_recreate\_product | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy:    1. Gọi API **RemoveProduct** để xóa sản phẩm trên sàn Lazada.    2. Gọi API **CreateProduct** để tạo sản phẩm với nội dung mới trên sàn Lazada. |
| lazada\_recreate\_product | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi API **CreateProduct** để tạo lại sản phẩm trên sàn Lazada. |
| lazada\_update\_product | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi API **UpdateProduct** để cập nhật nội dung sản phẩm lên Lazada. |
| lazada\_retry\_create\_or\_update\_product | 1. Tìm trong database các sản phẩm đang có trạng thái phù hợp. 2. Với mỗi sản phẩm tìm thấy, gọi API **RemoveProduct** xóa sản phẩm cũ (nếu sản phẩm đó không được hiển thị trên Lazada) rồi gọi API **CreateProduct** để thử tạo lại sản phẩm mới. |

Mã nguồn của Workers nằm tại địa chỉ: <https://gitlab.com/syncpro/workers>.

## Cài đặt Webapp.

Webapp của Syncpro được xây dựng với **React JS**. React JS là một thư viện JavaScript mã nguồn mở, được sử dụng để làm nên những giao diện web tương tác tốt với người sử dụng. Nó đặc biệt thích hợp để xây dựng nên các ứng dụng đơn trang (Single Page Application) với khả năng thay đổi dữ liệu hiển thị trên màn hình mà không cần phải tải lại cả trang web. React JS có một tư tưởng thiết kế rất rõ ràng. Nó cũng rất dễ học, dễ sử dụng, và có hiệu năng tốt. Trên thực tế, React JS đang là một trong những công nghệ làm front-end tốt nhất hiện nay.

Webapp của Syncpro gồm các trang:

* **Trang tạo sản phẩm mới**: Là trang dành cho người bán hàng phát hành sản phẩm mới lên nhiều sàn thương mại điện tử.
* **Trang danh sách sản phẩm**: Là trang để người bán hàng xem danh sách các sản phẩm đã được phát hành với Syncpro và trạng thái của từng sản phẩm.
* **Trang chỉnh sửa sản phẩm**: Là trang dành cho người bán hàng chỉnh sửa thông tin sản phẩm và cập nhật lên nhiều sàn thương mại điện tử.

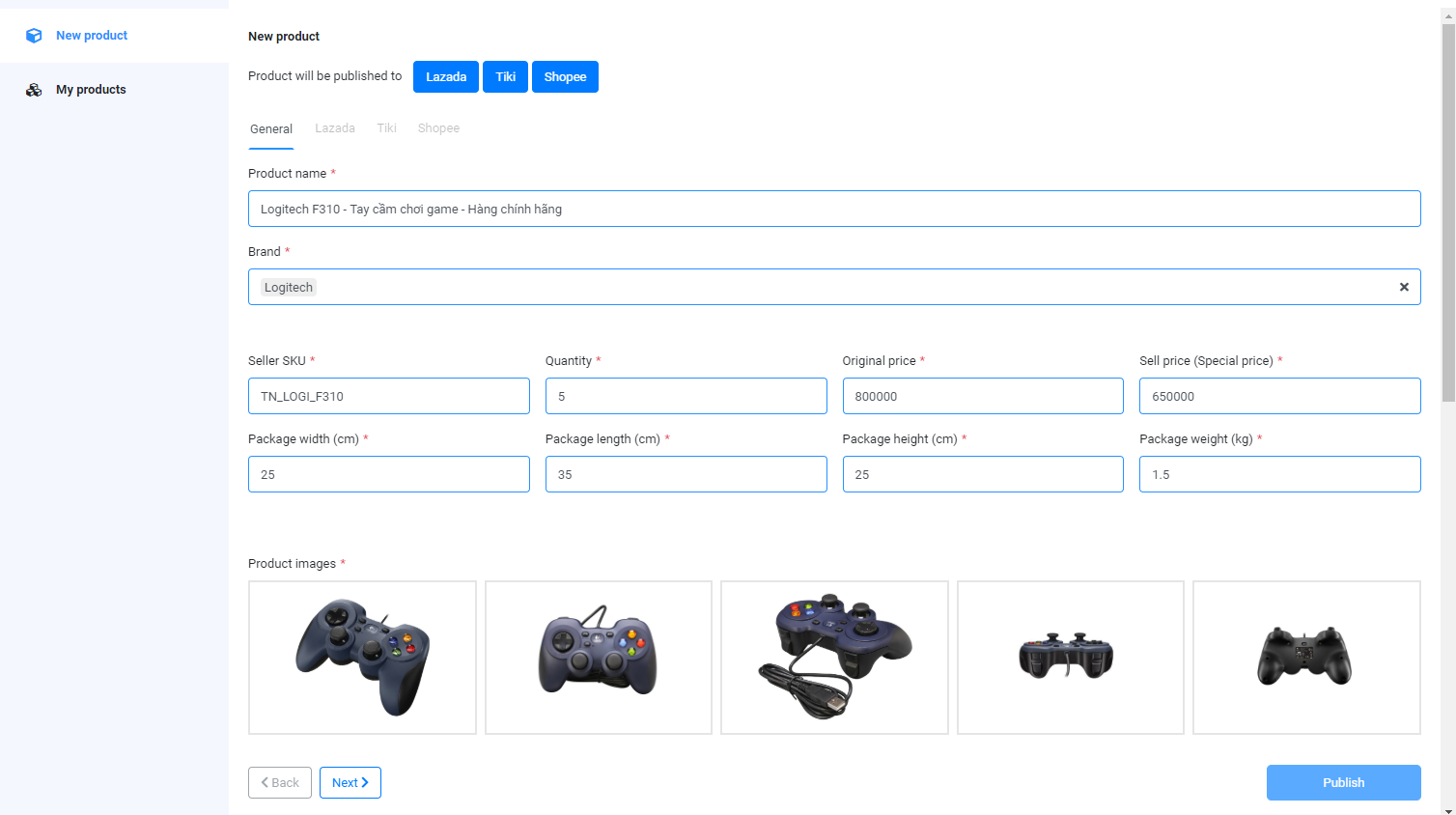
Mã nguồn của Syncpro Webapp nằm tại địa chỉ: <https://gitlab.com/syncpro/webapp>.

# Kết quả đạt được và hướng phát triển

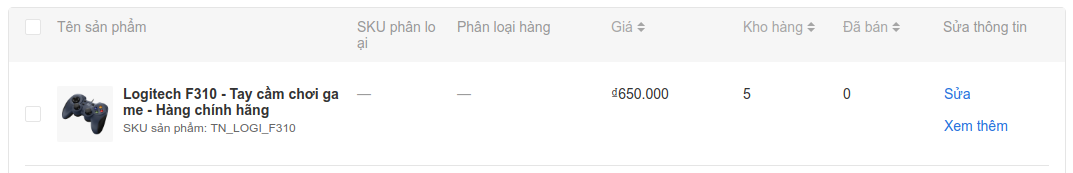
## Kết quả đạt được.

Syncpro hiện tại đã có thể đồng bộ nội dung sản phẩm cho ba sàn Shopee, Tiki, Lazada với các sản phẩm một biến thể. Người bán hàng có thể tạo và cập nhật nội dung sản phẩm, mọi thay đổi sẽ được Syncpro đồng bộ lên các sàn thương mại điện tử.

Dưới đây là một số hình minh họa việc tạo sản phẩm với Syncpro:



Hình .. Người bán hàng tạo sản phẩm với Syncpro.



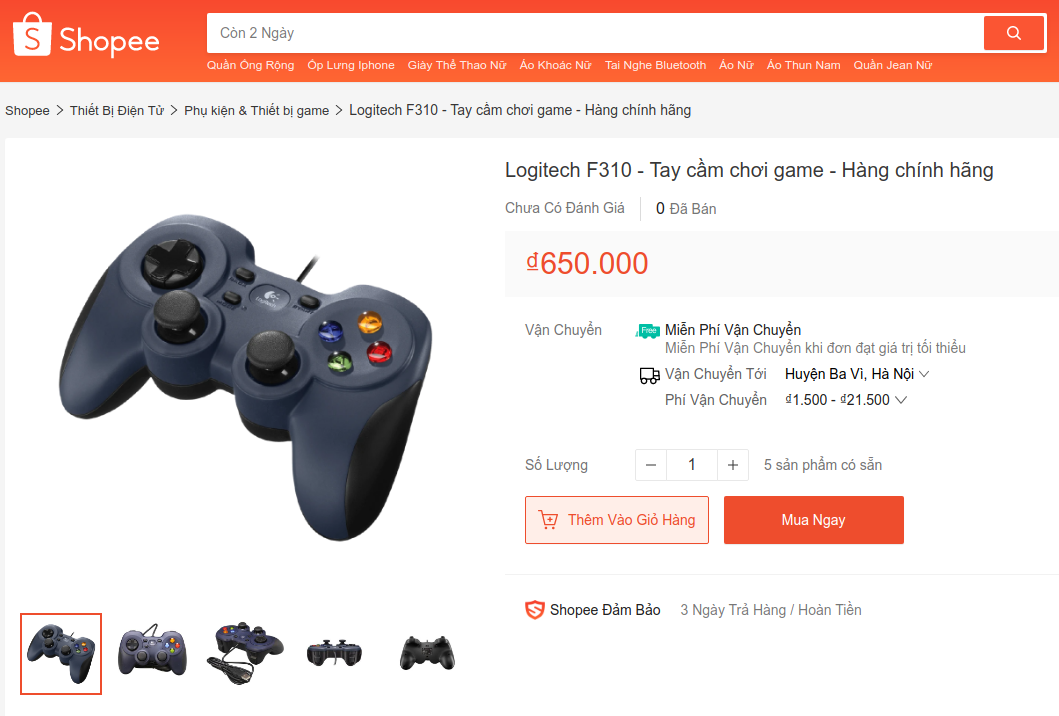
Hình .. Sản phẩm xuất hiện trên Seller center của Shopee.



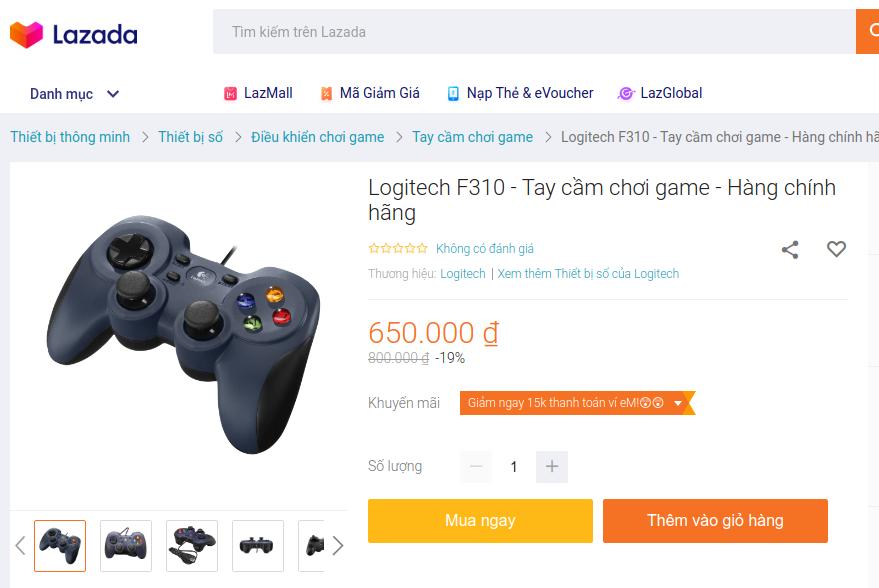
Hình .. Sản phẩm xuất hiện trên Seller center của Tiki.



Hình .. Sản phẩm xuất hiện trên Seller center của Lazada.

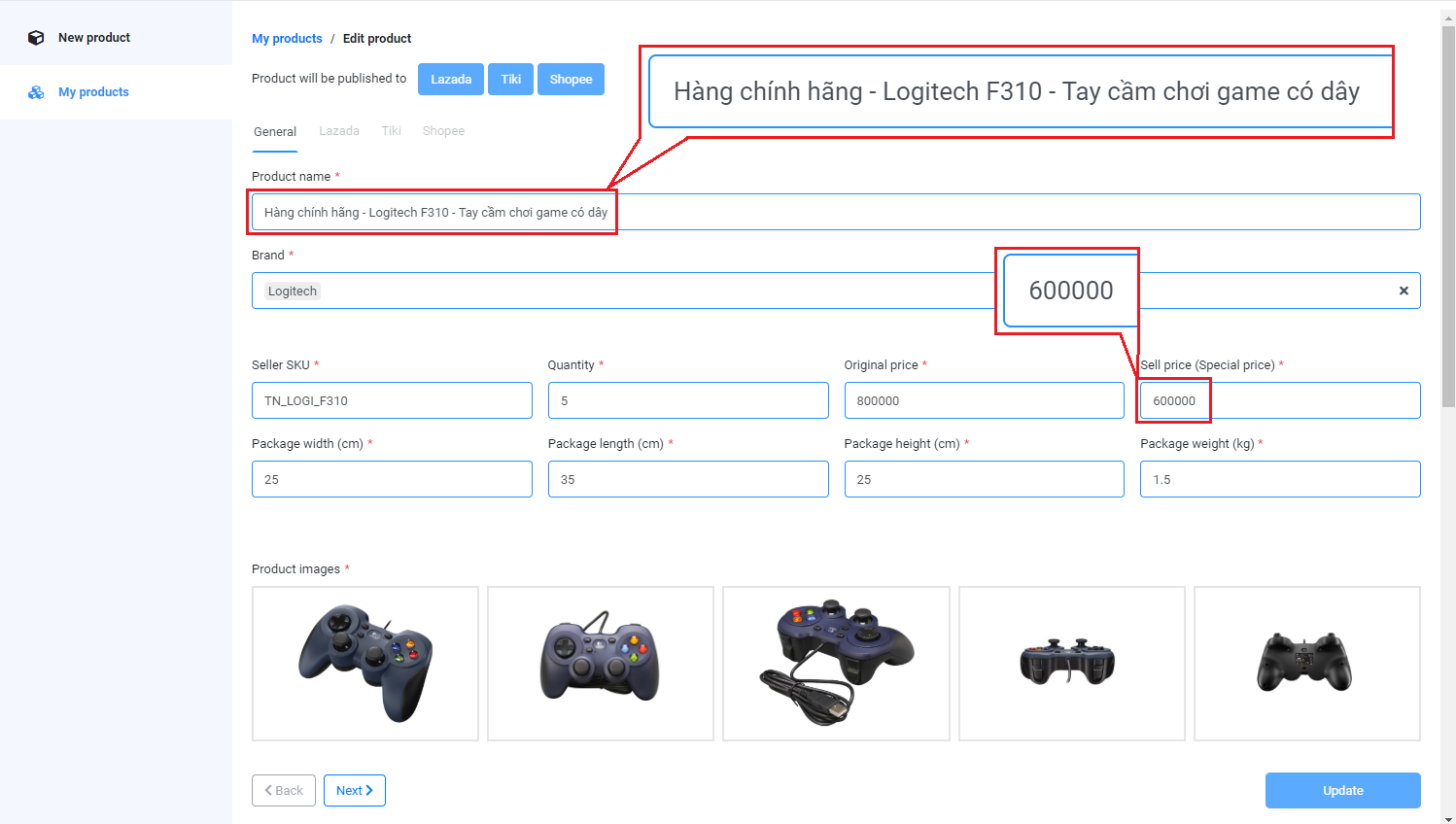


Hình .. Sản phẩm hiển thị trên sàn Shopee.vn.

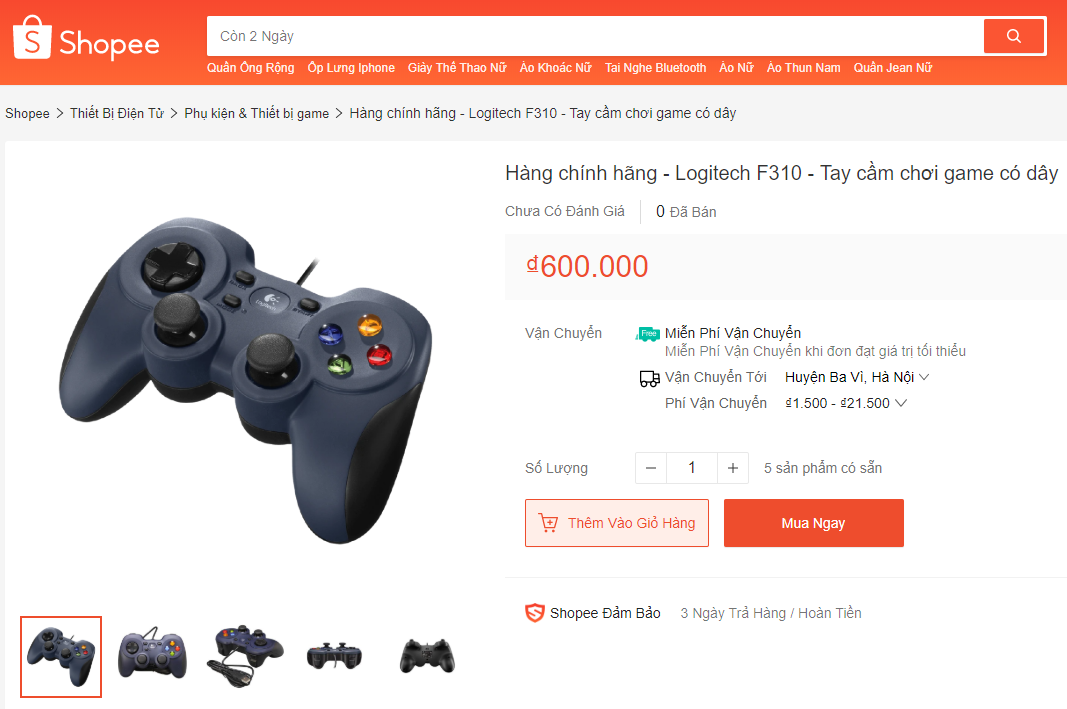


Hình .. Sản phẩm hiển thị trên sàn Lazada.vn.

Dưới đây là một số hình minh họa việc cập nhật thông tin sản phẩm với Syncpro:



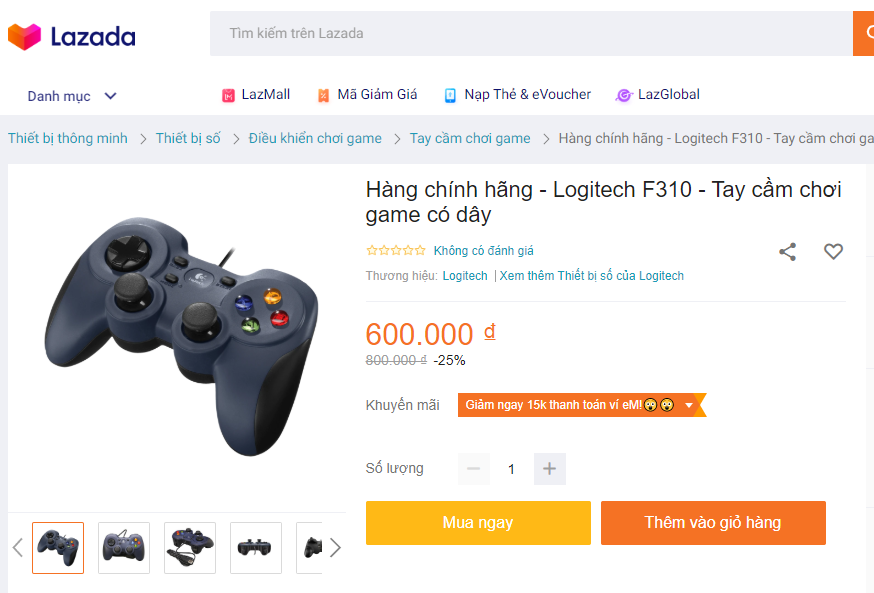
Hình .. Người bán hàng cập nhật thông tin sản phẩm với Syncpro.



Hình .. Thông tin sản phẩm đã thay đổi trên sàn Shopee.vn.



Hình .. Thông tin sản phẩm đã thay đổi trên Seller center của Tiki.



Hình .. Thông tin sản phẩm đã thay đổi trên sàn Lazada.vn.

## Hướng phát triển.

Về cơ bản, Syncpro đã làm được nhiệm vụ cốt lõi của mình, đó là đồng bộ nội dung sản phẩm lên ba sàn thương mại điện tử: Shopee, Tiki và Lazada. Tuy nhiên, để có thể trở thành một sản phẩm đủ chín, một sản phẩm thực sự sẵn sàng được đem ra sử dụng ngoài đời thì Syncpro vẫn còn thiếu nhiều tính năng cần thiết nữa. Trước mắt Syncpro vẫn còn một chặng đường rất dài để đi. Trong phiên bản tiếp theo của phiên bản này, Syncpro sẽ chỉ tập trung vào những tính năng quan trọng nhất, thiết thực nhất.

### Hỗ trợ thêm sản phẩm có nhiều biến thể.

Trên thực tế, phần nhiều sản phẩm trên các sàn thương mại điện tử đều là sản phẩm có nhiều biến thể. Một chiếc áo thì có nhiều kích cỡ; một chiếc điện thoại thì có nhiều phiên bản bộ nhớ; một đôi giày thì phân biệt giày nam, giày nữ; v.v... Những sản phẩm như vậy mới thực sự là những sản phẩm gây tốn công sức của người bán hàng khi phải nhập đi nhập lại thông tin trên các sàn thương mại điện tử. Việc thiếu đi sự hỗ trợ đồng bộ thông tin cho các sản phẩm này quả là một thiếu sót.

Trong tương lai gần, Syncpro nhất định sẽ hỗ trợ đồng bộ nội dung các sản phẩm có nhiều biến thể, hoàn thiện hơn lời giải cho bài toán đồng bộ sản phẩm và mang đến người bán hàng những trải nghiệm tốt hơn.

### Hoàn thiện tính năng xác thực người dùng.

Một hệ thống được thiết kế cho nhiều người sử dụng thì bao giờ cũng phải có cơ chế xác thực người dùng. Cơ chế này đóng vai trò rất quan trọng vì nó bảo vệ các tài nguyên chung của hệ thống khỏi sự truy cập trái phép. Tuy nhiên, do ban đầu chỉ tập trung vào xử lý các vấn đề cốt lõi nên hiện tại Syncpro chưa được cài đặt bất kỳ phương pháp nào để nhận biết những người có đủ quyền truy cập. Muốn trở thành một sản phẩm sẵn sàng được sử dụng trong thực tế, Syncpro cần phải có cơ chế xác thực người sử dụng cho riêng mình.

### Nghiên cứu phát triển tính năng đồng bộ số lượng sản phẩm.

Giải pháp đồng bộ số lượng sản phẩm hiện tại vẫn chưa được cài đặt vào hệ thống Syncpro vì cách tiếp cận ngây thơ được đánh giá là không hiệu quả. Trong phiên bản tiếp theo, Syncpro sẽ thử cài đặt giải pháp sử dụng cách tiếp cận ngây thơ và ước tính số sản phẩm tối đa mà hệ thống có thể xử lý, từ đó tìm cách tối ưu lại. Song song với việc tối ưu giải pháp hiện có, Syncpro cũng nghiên cứu tìm giải pháp mới cho bài toán đồng bộ số lượng sản phẩm.

### Mở rộng ra nhiều sàn thương mại điện tử hơn.

Ngoài Shopee, Tiki và Lazada, tại Việt Nam vẫn còn nhiều sàn thương mại điện tử rất có tiềm năng. Tiêu biểu trong số đó có sàn **Sendo** của FPT (địa chỉ: <https://sendo.vn>), cũng là một sàn thương mại điện tử lớn. Sau khi giải quyết trọn vẹn bài toán đồng bộ sản phẩm với Shopee, Tiki và Lazada, Syncpro sẽ chinh phục sàn thương mại điện tử Sendo, mang đến sự hỗ trợ tốt hơn cho các nhà bán hàng.

TÀI LIỆU THAM KHẢO