

**TẬP ĐOÀN CÔNG NGHIỆP VIỄN THÔNG QUÂN ĐỘI**

**TÀI LIỆU HƯỚNG DẪN THIẾT KẾ MẪU HÓA ĐƠN**

**(Dành cho nhân viên CNTT tỉnh/thành phố)**



**Hà Nội, 12/2018**

MỤC LỤC

[I. GIỚI THIỆU TỔNG QUAN 7](#_Toc31460941)

[1.1. Hóa đơn điện tử là gì? 7](#_Toc31460942)

[1.2. Quy trình thiết kế mẫu hóa đơn điện tử 7](#_Toc31460943)

[1.3. Cấu trúc một file khi thiết kế mẫu hóa đơn 8](#_Toc31460944)

[1.4. Ngôn ngữ XSL 8](#_Toc31460945)

[1.5. Tài liệu tham khảo. 8](#_Toc31460946)

[1.6 Địa chỉ FTP lấy file hóa đơn chuẩn của hệ thống 9](#_Toc31460947)

[II. CÁC TRƯỜNG DỮ LIỆU CHUẨN XML VÀ CÁCH ĐỌC DỮ LIỆU TỪ XSL 9](#_Toc31460948)

[2.1. Dữ liệu của người bán hàng (Seller) 9](#_Toc31460949)

[2.2. Dữ liệu liên quan đến người mua hàng (Buyer). 10](#_Toc31460950)

[2.3. Dữ liệu liên quan Hóa đơn 10](#_Toc31460951)

[2.4. Thanh toán. 11](#_Toc31460952)

[2.5. Hàng hóa. 11](#_Toc31460953)

[2.6. Thuế suất. 12](#_Toc31460954)

[2.7. Chỉ số đồng hồ (Dành cho đặc thù điện nước) 12](#_Toc31460955)

[2.8. Trường bổ sung (metadata) 13](#_Toc31460956)

[III. THIẾT KẾ HÓA ĐƠN 13](#_Toc31460957)

[3.1. Giới thiệu sơ lược về các file hóa đơn 13](#_Toc31460958)

[a. File view trực tiếp (invoice.xsl) 13](#_Toc31460959)

[b. File view pdf (invoice-fo.xsl và exchange-invoice-fo.xsl) 15](#_Toc31460960)

[3.2. File invoice-fo.xsl và exchange-invoice-fo.xsl 17](#_Toc31460961)

[a) Định dạng tiền trong hoá đơn 17](#_Toc31460962)

[b) Khai báo các style attribute để sử dụng chung 18](#_Toc31460963)

[c) Chèn dòng trống vào bảng hàng hóa 18](#_Toc31460964)

[d) Thiết lập kích thước khổ giấy PDF và phân trang 19](#_Toc31460965)

[e) Fill dữ liệu bảng hàng hóa. 21](#_Toc31460966)

[f) Lấy thuế suất 22](#_Toc31460967)

[g) Chữ ký số của người mua, người bán, chuyển đổi 24](#_Toc31460968)

[h) Font chữ của hóa đơn 28](#_Toc31460969)

[3.3. File invoice.xsl. 28](#_Toc31460970)

[a. Định dạng tiền trong hóa đơn 28](#_Toc31460971)

[b. Khai báo style dùng chung. 29](#_Toc31460972)

[c. Chèn số dòng trống ở bảng hàng hóa 29](#_Toc31460973)

[d. Thiết lập kích thước khổ giấy. 30](#_Toc31460974)

[e. Fill dữ liệu bảng hàng hóa 30](#_Toc31460975)

[f. Lấy thuế suất 31](#_Toc31460976)

[g. Chữ ký số của người mua, người bán 32](#_Toc31460977)

[h. Font chữ của hóa đơn 34](#_Toc31460978)

[} 34](#_Toc31460979)

[3.4 Một số khai báo và hàm hay sử dụng (if, choose, template, variable, contains, count, sum,....) 34](#_Toc31460980)

[IV. PHẦN MỀM THIẾT KẾ VÀ CÁCH XEM KẾT QUẢ CÁC FILE THIẾT KẾ. 36](#_Toc31460981)

[4.1. Các phần mềm có thể view và sửa code xsl 36](#_Toc31460982)

[4.2. Tool render từ xsl + xml sang pdf 36](#_Toc31460983)

[4.3. Xem kết quả 36](#_Toc31460984)

[a) File invoice-fo.xsl và exchange-invoice-fo.xsl 36](#_Toc31460985)

[b) file invoice.xsl 37](#_Toc31460986)

[V. MỘT SỐ TRƯỜNG HỢP THƯỜNG XUYÊN HỖ TRỢ 37](#_Toc31460987)

[CHÚ Ý: Link video hướng dẫn thiết kế trên Youtube các trường hợp thường xuyên hỗ trợ: 37](#_Toc31460988)

[Một số trường hợp ví dụ cụ thể 37](#_Toc31460989)

[1. Thêm bớt thông tin ngày ký trên hóa đơn 37](#_Toc31460990)

[a. Invoice-fo.xsl và exchange-invoice-fo.xsl 37](#_Toc31460991)

[b. Invoice.xsl 38](#_Toc31460992)

[2. Xóa 1 số dữ liệu: fax, website, email …. 40](#_Toc31460993)

[a. Invoice-fo và exchange-invoice-fo 40](#_Toc31460994)

[b. Invoice.xsl 40](#_Toc31460995)

[3. Thêm cột trong bảng hàng hóa (Thêm cột Số lô, Hạn dùng) 41](#_Toc31460996)

[4. Giãn cách trường Mã số thuế (Xuống dòng Mã hàng hóa) 42](#_Toc31460997)

[a. Invoice.xsl 42](#_Toc31460998)

[Mã số thuế 42](#_Toc31460999)

[Mã hàng hóa 42](#_Toc31461000)

[b. invoice-fo.xsl và exchange-invoice-fo.xsl 42](#_Toc31461001)

[Mã số thuế 42](#_Toc31461002)

[Mã hàng hóa 43](#_Toc31461003)

[5. Bỏ hiển thị 0 trên mẫu hóa đơn 43](#_Toc31461004)

[a. invoice-fo.xsl và exchange-invoice-fo.xsl 43](#_Toc31461005)

[b. invoice.xsl 43](#_Toc31461006)

[6. Chỉnh sửa thuế suất không chịu thuế (Hiển thị …\....%) 44](#_Toc31461007)

[a. Invoice-fo.xsl và exchange-invoice-fo.xsl 44](#_Toc31461008)

[b. Invoice.xsl 44](#_Toc31461009)

[7. Điều chỉnh vị trí, kích thước của logo và watermark. 45](#_Toc31461010)

[a. Logo 46](#_Toc31461011)

[b. watermark 46](#_Toc31461012)

[8. Xuống dòng dữ liệu theo ý muốn 47](#_Toc31461013)

[a. invoice-fo.xsl và exchange-invoice-fo.xsl 47](#_Toc31461014)

[b. invoice.xsl 48](#_Toc31461015)

[9. Gán cứng số lượng chữ số sau dấu thập phân 49](#_Toc31461016)

[a. invoice-fo.xsl và exchange-invoice-fo.xsl 49](#_Toc31461017)

[b. invoice.xsl 50](#_Toc31461018)

[10. Mẫu hóa đơn chi nhánh chứa thông tin của công ty 51](#_Toc31461019)

[11. Lấy 1 dữ liệu không chứa trong mảng 51](#_Toc31461020)

[12. Hiển thị thuế nhà thầu (xxx%) 52](#_Toc31461021)

[Bước 1. Nhập liệu 53](#_Toc31461022)

[Bước 2: Chỉnh sửa mẫu để hiển thị 53](#_Toc31461023)

[13. Bỏ giờ trong phần ngày chuyển đổi 54](#_Toc31461024)

[14. Sử dụng các trường có sẵn của hàng hóa để nhập liệu 54](#_Toc31461025)

[Bước 1: Nhập liệu 54](#_Toc31461026)

[Bước 2: Chỉnh sửa mẫu để hiển thị 55](#_Toc31461027)

[VI. KẾT THÚC 56](#_Toc31461028)

I. GIỚI THIỆU TỔNG QUAN

1.1. Hóa đơn điện tử là gì?

**Hoá đơn điện tử** là tập hợp các thông điệp dữ liệu điện tử về bán hàng hoá, cung ứng dịch vụ, được khởi tạo, lập, gửi, nhận, lưu trữ và quản lý bằng phương tiện điện tử

Hay nói theo cách khác HDDT là hóa đơn, nhưng không còn viết trên giấy nữa mà được khởi tạo bằng hình thức điện tử, sử dụng chữ ký điện tử thay cho chữ ký tay, đóng dấu như trước.

**Mẫu hóa đơn điện tử:** Dựa vào dữ liệu hóa đơn khi lập (thông tin người bán, người mua, hàng hóa, thông tin thanh toán ....) dạng xml sẽ được hệ thống render ra 2 dạng thể hiện pdf hoặc view trực tiếp. Về cơ bản mẫu hóa đơn sẽ chỉ lấy dữ liệu có sẵn trên xml lên và hiển thị (trừ những TH đặc biệt dựa vào xml thực hiện tính toán 1 số dữ liệu sao cho phù hợp: phí thuế khác)

***Mẫu hóa đơn chuẩn***: ***chỉ để xác định cách nhập liệu (hóa đơn bán hàng, hóa đơn GTGT, các loại hóa đơn khác, GTGT thuế dòng hàng hóa, thuế tổng, chiết khấu dòng, chiết khấu tổng, điện nước, Qrcode, .... ) cùng 1 số thiết kế cơ bản và danh sách trường động được khai báo mặc định.***

***Do đó, chỉ cần chọn 1 mẫu chuẩn có chung cách nhập liệu và mapping đúng danh sách trường động mong muốn là được. Còn sau đó có thể sử dụng các mẫu hóa đơn khác đã chỉnh sửa up mẫu lên phần ”Quản lý mẫu hóa đơn” theo quyền của CNTT chi nhánh.***

***File danh sách các mẫu hóa đơn chuẩn(Khai báo nhập liệu, trường động, trạng thái ...):*** ftp://10.60.109.43/GPDN/TAI\_LIEU/SINVOICE/TK%20HOA%20DON/CATALOG/ /DS\_MAUCHUAN\_HDDT.xlsx

***Mỗi mẫu hóa đơn chuẩn sẽ có phần chú thích. Tùy theo đặc thù của mỗi mẫu hóa đơn sẽ được chú thích tại phần này. Khi chọn mẫu chuẩn chú ý đến phần này để nắm được thông tin đặc thù hoặc cách nhập liệu để hướng dẫn KH tránh TH chọn sai mong muốn.***

Do vậy, về mặt dữ liệu trước tiên cần kiểm tra file xml xem dữ liệu các đối tượng mong muốn hiển thị đã đúng chưa sau khi kiểm tra dữ liệu chắc chắn đã đúng ta sẽ kiểm tra đến phần mẫu lấy dữ liệu hóa đơn xem đã mapping đúng trường dữ liệu, đúng đường dẫn lấy dữ liệu chưa nhé.

1.2. Quy trình thiết kế mẫu hóa đơn điện tử

|  |  |  |  |
| --- | --- | --- | --- |
| **Bước** | **Công việc** | **Tài liệu tham chiếu** | **Kết quả cần đạt được** |
| 1 | Viettel chi nhánh tiếp nhận yêu cầu khách hàng, phối hợp thống nhất mẫu hóa đơn. |  |  |
| 2 | Thiết kế mẫu hóa đơn theo yêu cầu khách hàng: |  |  |
| 3 | Code XSL để thiết kế mẫu hoá đơn theo yêu cầu của khách hàng |  |  |
| 4 | Gửi mẫu thiết kế vừa thiết kế xong lên VTS kiểm tra, phê duyệt dựa vào chức năng trên hệ thống |  |  |
| 5 | VTS kiểm tra mẫu phê duyệt hoặc từ chối phê duyệt (lý do từ chối) để Chi nhánh kiểm tra và chỉnh sửa |  |  |

1.3. Cấu trúc một file khi thiết kế mẫu hóa đơn

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên file** | **Mục đích** |
| 1 | exchange-invoice-fo.xsl | File thiết kế hóa đơn chuyển đổi pdf. |
| 2 | ban\_chuyen\_doi.pdf | Kết quả hóa đơn chuyển đổi. |
| 3 | invoice-fo.xsl | File thiết kế bản thể hiện của hóa đơn điện tử pdf |
| 4 | ban\_the\_hien.pdf | Kết quả bản thể hiện. |
| 5 | invoice.xsl | File thiết kế bản thể hiện hóa đơn view web. |
| 6 | data-fo.xml | Nguồn dữ liệu đầu vào cho bản thể hiện pdf, bản chuyển đổi pdf (dùng chung dữ liệu) |
| 7 | data-fo-nhieutrang.xml | Chính là data-fo.xml nhưng có nhiều hàng hóa để test nhiều trang (Đổi tên thành data-fo.xml khi muốn bản thể hiện hay chuyển đổi ở trạng thái nhiều trang) |
| 8 | data.xml | Nguồn dữ liệu đầu vào cho bản thể hiện (view web) |

1.4. Ngôn ngữ XSL

XSL là một dạng ngôn ngữ dựa trên nền tảng XML để tạo ra stylesheet nhằm mục đích định dạng nội dung XML.

1.5. Tài liệu tham khảo.

Để hiểu thêm về XML, XSLT, HTML... bạn có thể ghép thăm http://w3schools.sinsixx.com/xslfo/xslfo\_documents.asp.htm .Ở đấy có giải thích và các ví dụ cụ thể giúp bạn có thể nắm bắt một cách dễ dàng về các thuộc tính và cách sử dụng.

1.6 Địa chỉ FTP lấy file hóa đơn chuẩn của hệ thống

[**ftp://10.60.109.43/GPDN/TAI\_LIEU/SINVOICE/TK%20HOA%20DON/**](ftp://10.60.109.43/GPDN/TAI_LIEU/SINVOICE/TK%20HOA%20DON/DS%20MAU%20CHUAN%20HE%20THONG/)

Tại đây sẽ chia theo thư mục của các loại hóa đơn: HD BH, HD GTGT, PXK NB ……

II. CÁC TRƯỜNG DỮ LIỆU CHUẨN XML VÀ CÁCH ĐỌC DỮ LIỆU TỪ XSL

Tất cả các dữ liệu chuẩn của các đối tượng có thể tự điều chỉnh lên các vị trí như mong muốn. Cần dữ liệu nào thì lấy dữ liệu lên theo đúng đường dẫn chỉ định sẵn.

2.1. Dữ liệu của người bán hàng (Seller)

Đọc dữ liệu từ xsl: invoiceData/seller/X

Trong đó X :

* sellerLegalName //Tên (đăng ký kinh doanh trong trường hợp là doanh nghiệp) của người bán
* sellerTaxCode //Mã số thuế người bán được cấp bởi TCT Việt Nam
* sellerAddressLine //Địa chỉ bưu điện người bán
* sellerPostalCode //Mã bưu điện //04 (Hà Nội) 08 (HCM)
* sellerDistrictName //Tên Quận Huyện
* sellerCityName //Tên Tỉnh/Thành phố
* sellerCountryCode //Mã quốc gia
* sellerPhoneNumber //Số điện thoại người bán
* sellerFaxNumber //Số fax người bán
* sellerEmail //Địa chỉ thư điện tử người bán
* sellerBankName //Tên trụ sở chính ngân hàng nơi người bán mở tài khoản giao dịch
* sellerBankAccount //Tài khoản ngân hàng của người bán
* sellerContactPersonName //Tên người đại diện người bán
* sellerSignedPersonName //Tên người ký duyệt hóa đơn bằng chữ ký điện tử
* sellerSubmittedPersonName //Tên người gửi hóa đơn
* sellerWebsite //Địa chỉ web của người bán
* sellerBusinessLicenseNo; // Giấy phép kinh doanh
* sellerDescription;
* sellerParentSupplierTaxCode;
* sellerRepresentative;// Tên người đại diện
* sellerBankAccountOwner;// Chủ tài khoản
* sellerBankBranch;
* sellerRepresentativeIdType;// Loại giấy tờ của người đại diện
* sellerSearchInvoiceLink;// Link tra cứu hóa đơn
* sellerRepresentativeIdNo;// Số giấy tờ của người đại diện

2.2. Dữ liệu liên quan đến người mua hàng (Buyer).

Đọc dữ liệu từ xsl: invoiceData/buyer/Y

Trong đó Y :

* buyerDisplayName //Tên người mua
* buyerLegalName //Tên (đăng ký kinh doanh trong trường hợp là doanh nghiệp) của người mua
* buyerTaxCode //Mã số thuế người mua được cấp bởi TCT Việt Nam.
* buyerAddressLine //Địa chỉ bưu điện người mua
* buyerPostalCode //Mã bưu điện
* buyerDistrictName //Tên Quận Huyện
* buyerCityName //Tên Tỉnh/Thành phố
* buyerCountryCode //Mã quốc gia VN (Việt Nam)
* buyerPhoneNumber //Số điện thoại người mua
* buyerFaxNumber //Số fax người mua
* buyerEmail //Email người mua
* buyerBankName //Tên trụ sở chính ngân hàng nơi người mua mở tài khoản giao dịch
* buyerBankAccount //Tài khoản ngân hàng của người mua
* buyerIdType //loai giay to
* buyerIdNo //so giay to khach hang, co the la chung minh thu, so giay phep kinh doanh...
* buyerBirthDay; // Ngày sinh
* buyerCode; //Ma KH

2.3. Dữ liệu liên quan Hóa đơn

Đọc dữ liệu từ xsl: invoiceData /Z

Trong đó Z :

* invoiceType //Mã loại hóa đơn chỉ nhận các giá trị sau: 01GTKT, 02GTTT, 07KPTQ, 03XKNB, 04HGDL.
* templateCode //Mã mẫu hóa đơn, tuân thủ theo quy định ký hiệu mẫu hóa đơn
* invoiceSeries //Là “Ký hiệu hóa đơn”
* invoiceNumber;//Là Số hóa đơn phát sinh từ phần mềm phát sinh hóa đơn.
* invoiceName //Tên hóa đơn.
* invoiceIssuedDate //DateTime Ngày lập hóa đơn .
* contractDate //Ngày hợp đồng liên quan đến hóa đơn
* currencyCode //Mã tiền tệ dùng cho hóa đơn.
* exchangeRate //Tỷ giá ngoại tệ tại thời điểm lập hóa đơn quy đổi ra VNĐ
* totalAmountWithoutVAT // Tổng tiền hóa đơn chưa bao gồm VAT.
* totalVATAmount // Tổng tiền thuế trên toàn hóa đơn.
* totalAmountWithVAT //Tổng tiền trên hóa đơn đã bao gồm VAT.
* totalAmountWithVATFrn// Tổng tiền ngoại tệ của hóa đơn đã bao gồm VAT.
* totalAmountAfterDiscount // Tổng tiền hóa đơn sau chiết khấu.
* discountAmount // Tiền chiết khấu.

2.4. Thanh toán.

Đọc dữ liệu từ xsl: invoiceData /payments/payment/P

Trong đó P :

* paymentMethodName//Phương thức thanh toán

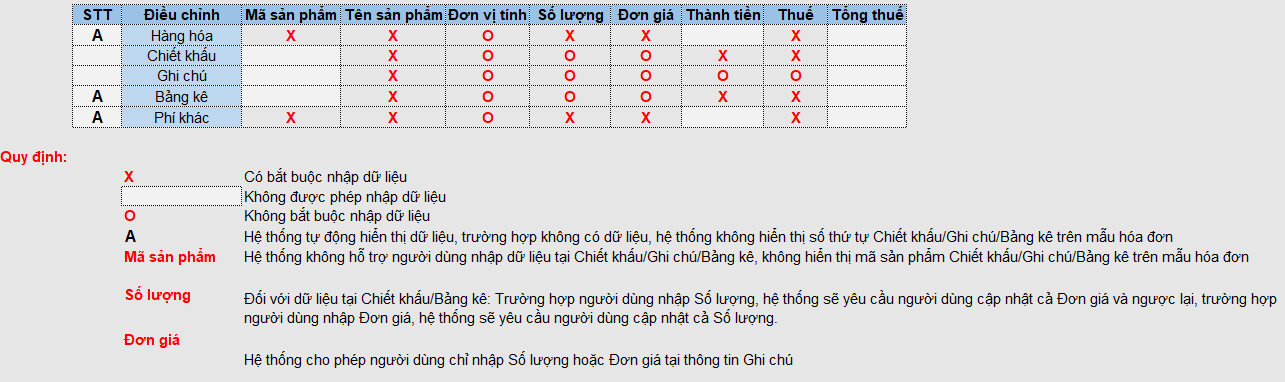
2.5. Hàng hóa.

Đọc dữ liệu từ xsl: invoiceData/items/item/I

Trong đó I :

* lineNumber //UnsignedInt Thứ tự dòng hóa đơn, bắt đầu từ 1
* itemCode //Mã hàng hóa, dịch vụ
* itemName //Tên hàng hóa, dịch vụ
* unitCode //Mã đơn vị tính TM: Tiền mặt CK: Chuyển khoản
* unitName //Tên đơn vị tính hàng hóa, dịch vụ
* selection // Loại hàng hóa, dịch vụ
* unitPrice // Decimal (15,2) Đơn giá
* quantity //Decimal (15,2) Số lượng
* itemTotalAmountWithoutVat // Là tổng tiền hàng hóa dịch vụ chưa có VAT
* itemTotalAmountWithVat //Tổng tiền sau thuế
* itemTotalAmountAfterDiscount // Tổng tiền sau chiết khấu
* vatPercentage;// Thuế suất của hàng hóa, dịch vụ
* vatAmount;// Tổng tiền thuế
* itemNote //Ghi chú bổ sung thông tin hàng hóa
* batchNo //So lo, chi co gia tri hien thi ve mat du lieu
* expDate //Han dung, chi co gia tri hien thi ve mat du lieu
* discount // % Chiết khấu 1
* discount2 // % Chiết khấu 2
* itemDiscount // Tiền chiết khấu

Dữ liệu hàng hóa sẽ phải đảm bảo theo đúng chuẩn hệ thống đáp ứng được như sau:



Dựa vào các loại hàng hóa để chọn giải pháp phù hợp cho KH khi cần điều chỉnh mẫu hiển thị như mong muốn: Phí khác, chiết khấu, ghi chú, bảng kê

Dữ liệu thì do hệ thống validate không liên quan đến mẫu hóa đơn. Mẫu hóa đơn chỉ lấy dữ liệu mà hệ thống ghi nhận được để view lên pdf.

2.6. Thuế suất.

Đọc dữ liệu từ xsl: invoiceData/invoiceTaxBreakdowns /invoiceTaxBreakdowns/I

Trong đó I :

* vatPercentage //Mức thuế: khai báo giá trị như sau 0%: 0 5%: 5 10%: 10 Không chịu thuế: -1, Không kê khai nộp thuế: -2
* vatTaxableAmount //Tổng tiền chịu thuế của mức thuế tương ứng
* vatTaxAmount //Tổng tiền thuế của mức thuế tương ứng

2.7. Chỉ số đồng hồ (Dành cho đặc thù điện nước)

***(Chú ý: Chỉ những mẫu hóa đơn chuẩn được khai báo là đặc thù điện nước thì mới có phần thông tin chỉ số trên hệ thống để nhập liệu)***

Đọc dữ liệu từ xsl: invoiceData/items/meterReading/I

Trong đó I :

* previousIndex // chỉ số tháng trước
* currentIndex // chỉ số tháng này
* factor // Hệ số
* amount // Tổng mức tiêu thụ
* meterName // Tên chỉ số

2.8. Trường bổ sung (metadata)

Đọc dữ liệu từ xsl: invoiceData/metadata/M

Trong đó M: Những trường không tồn tại ở 5 mục trên => cho vào trường bổ sung (tự đặt tên).

Ví dụ: Phương tiện vận chuyển.

File -fo.xsl: invoicedata/metadata/vehicle (Đường dẫn sử dụng trong file thiết kế).

File invoice.xsl: inv:invoicedata/inv:metadata/inv:vehicle.

**Chú ý**: ***Việc khai báo trường động hiện trạng đang gán với mẫu hóa đơn chuẩn. Do vậy, việc thêm trường động là rất hạn chế. Trước khi sử dụng, nên tìm hiểu các trường động được khai báo sẵn có cho các mẫu chuẩn, chọn 1 mẫu có các trường động tương đồng và tương đồng về cách nhập liệu để sử dụng lại các trường động. Thông tin cụ thể đã có trong tài liệu trên ftp:*** <ftp://10.60.109.43/GPDN/TAI_LIEU/SINVOICE/TK%20HOA%20DON/CATALOG/> . ***Sau khi khai báo thêm trường động mới, khi gửi yêu cầu duyệt mẫu hóa đơn cần ghi rõ nội dung: Cần thêm trường động: Tên – tag để người duyệt mẫu biết và thêm trên hệ thống khi đó người dùng mới có thể nhập dữ liệu được (VD: Thêm trường động: Ghi chú – invoiceNote, Số tham chiếu - refNo)***

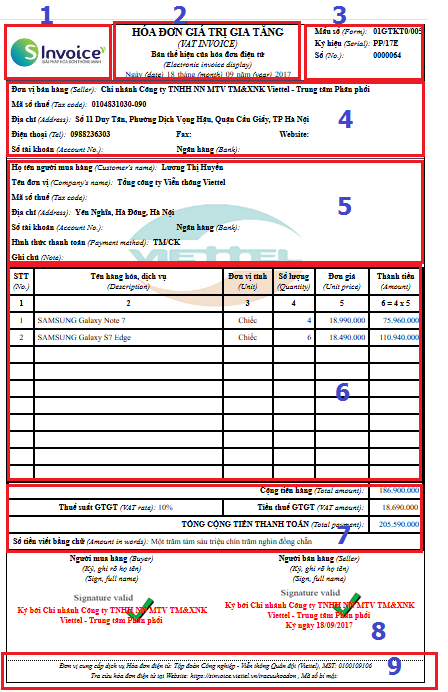
Trường động thêm chỉ thêm được ở dạng lập dữ liệu của hóa đơn, không thêm được ở trong phần hàng hóa và không tác dụng tham gia vào tính toán số liệu. Nếu cần thêm ở phần hàng hóa đọc nội dung tài liệu mục 5.14 để có giải pháp xử lý

III. THIẾT KẾ HÓA ĐƠN

3.1. Giới thiệu sơ lược về các file hóa đơn

a. File view trực tiếp (invoice.xsl)

Do phần view trực tiếp hóa đơn hiện tại chỉ là dạng liệt kê, không có định dạng khổ giấy hay format của hóa đơn nhiều trang, nên thông thường file invoice.xsl sẽ được chia bố cục gồm các phân vùng như sau và thông thường các bố cục này ngang hàng:



1. Thông tin logo
2. Thông tin loại hóa đơn (Hóa đơn Giá trị gia tăng, Hóa đơn bán hàng, ….) và thông tin ngày tháng của hóa đơn
3. Thông tin Mẫu số, ký hiệu và số hóa đơn

(3 phần này hiện đang nằm trên cùng 1 <tr> thuộc <table> lớn nhất bao quanh hóa đơn)

1. Thông tin người bán hàng (Dữ liệu seller)
2. Thông tin người mua hàng (Dữ liệu buyer)
3. Thông tin chi tiết hành hóa (Dữ liệu trong item)
4. Dữ liệu tổng hợp của hóa đơn

Là 1 phần thông tin trong invoiceData: Cộng tiền hàng, tổng cộng tiền thanh toán, chiết khấu…..

Phần thuế suất GTGT sẽ được lấy theo phần invoiceTaxBreakdowns

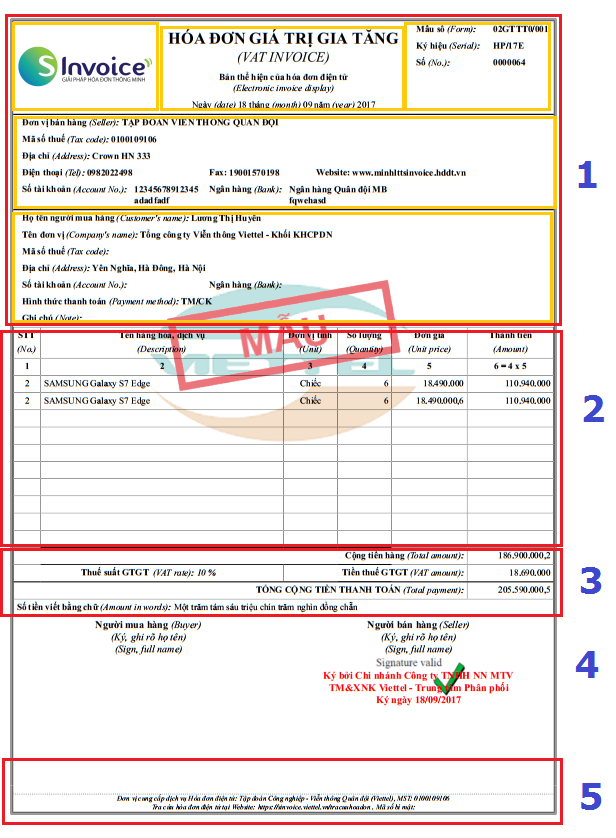
1. Phần hiển thị thông tin ký điện tử của hóa đơn
2. Phần hiển thị footer và thông tin dữ liệu tra cứu hóa đơn

b. File view pdf (invoice-fo.xsl và exchange-invoice-fo.xsl)

Đối với 2 file –fo.xsl, do ngoài việc hiển thị thông tin còn phải đảm bảo đúng theo thông tư 39 và thông tư 32 của BTC về việc hiển thị hóa đơn chuyển đổi từ hóa đơn điện tử sang hóa đơn giấy.

Trường hợp cơ sở kinh doanh sử dụng hóa đơn tự in, việc lập và in hóa đơn thực hiện trực tiếp từ phần mềm và số lượng hàng hóa, dịch vụ bán ra nhiều hơn số dòng của một trang hóa đơn, Cục thuế xem xét từng trường hợp cụ thể để chấp thuận cho cơ sở kinh doanh được sử dụng hóa đơn nhiều hơn một trang nếu trên phần đầu của trang sau của hóa đơn có hiển thị: **cùng số hóa đơn như của trang đầu (do hệ thống máy tính cấp tự động); cùng tên, địa chỉ, MST của người mua, người bán như trang đầu; cùng mẫu và ký hiệu hóa đơn như trang đầu; kèm theo ghi chú bằng tiếng Việt không dấu “tiep theo trang truoc – trang X/Y” (trong đó X là số thứ tự trang và Y là tổng số trang của hóa đơn đó)**.”

Do vậy, thiết kế file –fo.xsl cũng phải tuân thủ đúng việc này nên việc phân vùng sẽ có phần khác biệt



1. Phần header của hóa đơn (Lặp lại ở tất cả các trang đảm bảo theo thông tư 39)

Trong phần này chia ra các phân vùng tương ứng như bên invoice.xsl

1. Thông tin logo
2. Thông tin loại hóa đơn (Hóa đơn Giá trị gia tăng, Hóa đơn bán hàng, ….) và thông tin ngày tháng của hóa đơn
3. Thông tin Mẫu số, ký hiệu và số hóa đơn

(3 phần này hiện đang nằm trên cùng 1 <fo:table\_row> thuộc <fo:table-header> của <fo:table> lớn nhất bao quanh hóa đơn)

1. Thông tin người bán hàng (Dữ liệu seller)
2. Thông tin người mua hàng (Dữ liệu buyer)

(Các dữ liệu này tùy chỉnh dựa vào từng mẫu hóa đơn)

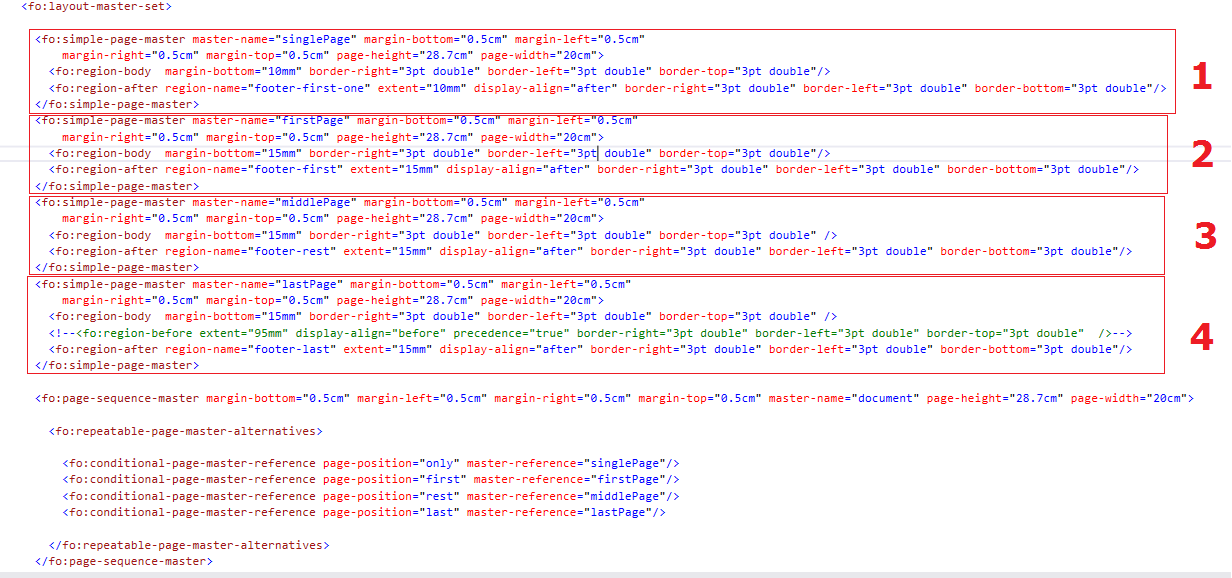
1. Thông tin chi tiết hành hóa (Dữ liệu trong item)
2. Dữ liệu tổng hợp của hóa đơn

Là 1 phần thông tin trong invoiceData: Cộng tiền hàng, tổng cộng tiền thanh toán, chiết khấu…..

Phần thuế suất GTGT sẽ được lấy theo phần invoiceTaxBreakdowns

1. Phần hiển thị thông tin ký điện tử của hóa đơn
2. Phần hiển thị footer và thông tin dữ liệu tra cứu hóa đơn

(Phần này đặc biệt hơn đối với file invoice.xsl Do cần khai báo nhiều định dạng với nhiều format nhiều trang. Hóa đơn chỉ có 1 trang (Khai báo tại phần khung 1), hóa đơn nhiều trang trang đầu (Khung 2), hóa đơn nhiều trang trang giữa (Khung 3), hóa đơn nhiều trang trang cuối cùng (Khung 4)). Phần này thông thường không cần điều chỉnh vì đã chuẩn với thông tư



3.2. File invoice-fo.xsl và exchange-invoice-fo.xsl

Các nội dung cần nắm trong cấu trúc file invoice-fo.xsl và exchange-invoice-fo.xsl

1. Định dạng tiền trong hoá đơn

Trường hợp 1: Đơn vị tiền là EURO hoặc VNĐ.

|  |
| --- |
| <xsl:function name="foo:custom-num-format">  <xsl:param name="num"/>  <xsl:param name="formats"/>  <xsl:param name="locale"/>  <xsl:choose>  <xsl:when test="string-length($num) = 0">  </xsl:when>  <xsl:when test="floor($num) = $num">  <xsl:value-of select="format-number($num, '###.##0', 'european')"/>  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="format-number($num, '###.##0,#########', 'european')"/>  </xsl:otherwise>  </xsl:choose>  </xsl:function> |

Trường hợp 2: Đơn vị tiền là USD

|  |
| --- |
| <xsl:function name="foo:custom-num-format">  <xsl:param name="num"/>  <xsl:param name="formats"/>  <xsl:param name="locale"/>  <xsl:choose>  <xsl:when test="string-length($num) = 0">  </xsl:when>  <xsl:when test="floor($num) = $num">  <xsl:value-of select="format-number($num, '###,##0', 'us')"/>  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="format-number($num, '###,##0.#########', 'us')"/>  </xsl:otherwise>  </xsl:choose>  </xsl:function> |

1. Khai báo các style attribute để sử dụng chung

|  |
| --- |
| <xsl:attribute-set name="labelNormal">  <xsl:attribute name="color">#000000</xsl:attribute>  <xsl:attribute name="margin-top">1mm</xsl:attribute>  <xsl:attribute name="margin-left">1mm</xsl:attribute>  <xsl:attribute name="margin-right">1mm</xsl:attribute>  <xsl:attribute name="margin-bottom">1mm</xsl:attribute>  </xsl:attribute-set>  <xsl:attribute-set name="labelItalic">  <xsl:attribute name="color">#000000</xsl:attribute>  <xsl:attribute name="font-style">italic</xsl:attribute>  <xsl:attribute name="margin-top">1mm</xsl:attribute>  <xsl:attribute name="margin-left">1mm</xsl:attribute>  <xsl:attribute name="margin-right">1mm</xsl:attribute>  <xsl:attribute name="margin-bottom">1mm</xsl:attribute>  </xsl:attribute-set> |

1. Chèn dòng trống vào bảng hàng hóa

|  |
| --- |
| <xsl:template name="loop">  <xsl:param name="var"></xsl:param>  <xsl:choose>  <xsl:when test="$var &lt; 10 and $var &gt; 0">  <fo:table-row keep-together.within-page="always">  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="center" xsl:use-attribute-sets="itemNormal" display-align="center">  <fo:leader />  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="left" xsl:use-attribute-sets="itemNormal" display-align="center">  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="center" xsl:use-attribute-sets="itemNormal" display-align="center">  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal" display-align="center">  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal" display-align="center">  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal" display-align="center">  </fo:block>  </fo:table-cell>  </fo:table-row>  <xsl:call-template name="loop">  <xsl:with-param name="var">  <xsl:number value="number($var)+1" />  </xsl:with-param>  </xsl:call-template>  </xsl:when>  <xsl:otherwise>  </xsl:otherwise>  </xsl:choose>  </xsl:template> |

Đoạn template này ý nghĩa là để chèn dòng trống vào bảng hàng hóa tối thiểu trên một trang. Số lượng fo:table-cell bằng đúng số cột của bảng hàng hóa.

<xsl:when test="$var &lt; 10 and $var &gt; 0">

Tối thiểu bao nhiêu dòng trên 1 trang. VD như trên là 10 dòng

1. Thiết lập kích thước khổ giấy PDF và phân trang

|  |
| --- |
| <fo:layout-master-set>  <fo:simple-page-master master-name="singlePage" margin-bottom="0.5cm" margin-left="0.5cm"  margin-right="0.5cm" margin-top="0.5cm" page-height="28.7cm" page-width="20cm">  <fo:region-body margin-bottom="10mm" border-right="3pt double" border-left="3pt double" border-top="3pt double"/>  <fo:region-after region-name="footer-first-one" extent="10mm" display-align="after" border-right="3pt double" border-left="3pt double" border-bottom="3pt double"/>  </fo:simple-page-master>  <fo:simple-page-master master-name="firstPage" margin-bottom="0.5cm" margin-left="0.5cm"  margin-right="0.5cm" margin-top="0.5cm" page-height="28.7cm" page-width="20cm">  <fo:region-body margin-bottom="15mm" border-right="3pt double" border-left="3pt double" border-top="3pt double"/>  <fo:region-after region-name="footer-first" extent="15mm" display-align="after" border-right="3pt double" border-left="3pt double" border-bottom="3pt double"/>  </fo:simple-page-master>  <fo:simple-page-master master-name="middlePage" margin-bottom="0.5cm" margin-left="0.5cm"  margin-right="0.5cm" margin-top="0.5cm" page-height="28.7cm" page-width="20cm">  <fo:region-body margin-bottom="15mm" border-right="3pt double" border-left="3pt double" border-top="3pt double" />  <fo:region-after region-name="footer-rest" extent="15mm" display-align="after" border-right="3pt double" border-left="3pt double" border-bottom="3pt double"/>  </fo:simple-page-master>  <fo:simple-page-master master-name="lastPage" margin-bottom="0.5cm" margin-left="0.5cm"  margin-right="0.5cm" margin-top="0.5cm" page-height="28.7cm" page-width="20cm">  <fo:region-body margin-bottom="15mm" border-right="3pt double" border-left="3pt double" border-top="3pt double" />  <fo:region-after region-name="footer-last" extent="15mm" display-align="after" border-right="3pt double" border-left="3pt double" border-bottom="3pt double"/>  </fo:simple-page-master>  <fo:page-sequence-master margin-bottom="0.5cm" margin-left="0.5cm" margin-right="0.5cm" margin-top="0.5cm" master-name="document" page-height="28.7cm" page-width="20cm">  <fo:repeatable-page-master-alternatives>  <fo:conditional-page-master-reference page-position="only" master-reference="singlePage"/>  <fo:conditional-page-master-reference page-position="first" master-reference="firstPage"/>  <fo:conditional-page-master-reference page-position="rest" master-reference="middlePage"/>  <fo:conditional-page-master-reference page-position="last" master-reference="lastPage"/>  </fo:repeatable-page-master-alternatives>  </fo:page-sequence-master>  </fo:layout-master-set> |

Ở trên đang là kích thước giấy A4 nếu muốn thiết kế kích thước A5 chỉ cần thay đổi 28.7cm -> 14.5cm. Đường viền (border) xung quanh (3pt double)

Chia 1 trang hóa đơn thành 2 vùng:

* region-body: Chứa toàn bộ thông tin của hóa đơn (thông tin người bán, người mua, thông tin về số hóa đơn được lặp lại ở các trang)
* region-after: Phần nhiều trang để đảm bảo theo thông tư 39 BTC

Phần phân trang được chia ra làm 4 case:

* singlePage: Hóa đơn chỉ có 1 trang sẽ lấy code trong phần này để hiển thị footer
* firstPage: Hóa đơn nhiều trang – Trang đầu tiên sẽ lấy code trong phần này hiển thị footer
* middlePage: Hóa đơn nhiều trang – Trang đầu tiên sẽ lấy code trong phần này hiển thị footer
* lastPage: Hóa đơn nhiều trang – Trang đầu tiên sẽ lấy code trong phần này hiển thị footer

Các phần footer sẽ được khai báo trong static-content đặt tên như bên dưới

|  |
| --- |
| <fo:static-content flow-name="footer-first-one">  <fo:block font-size="7pt" text-align="right">  <fo:block>  <fo:table table-layout="fixed" width="100%">  <fo:table-column column-width="100%"/>  <fo:table-body>  <fo:table-row>  <fo:table-cell>  <fo:block text-align="center">  <fo:inline font-size="7pt" xsl:use-attribute-sets="labelItalic">(Cần kiểm tra đối chiếu khi lập, giao nhận hóa đơn)</fo:inline>  </fo:block>  </fo:table-cell>  </fo:table-row>  <fo:table-row>  <fo:table-cell border-top="dotted">  <fo:block text-align="center">  <fo:inline font-size="7pt" xsl:use-attribute-sets="labelItalic">Đơn vị cung cấp dịch vụ Hóa đơn điện tử: Tập đoàn Công nghiệp - Viễn thông Quân đội (Viettel), MST: 0100109106 </fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:wrapper font-size="7pt" xsl:use-attribute-sets="labelItalic">Tra cứu hóa đơn điện tử tại Website: https://sinvoice.viettel.vn/tracuuhoadon. Mã số bí mật: </fo:wrapper>  <fo:wrapper font-size="7pt" xsl:use-attribute-sets="labelNormal">  <xsl:value-of select="invoiceData/customDefines/reservationCode"/>  </fo:wrapper>  </fo:block>  </fo:table-cell>  </fo:table-row>  </fo:table-body>  </fo:table>  </fo:block>  </fo:block>  </fo:static-content> |

1. Fill dữ liệu bảng hàng hóa.

|  |
| --- |
| <xsl:for-each select="invoiceData/items/item">  <fo:table-row keep-together.within-page="always">  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="center" xsl:use-attribute-sets="itemNormal">  <xsl:choose>  <xsl:when test="lineNumber &gt; 0">  <xsl:value-of select="lineNumber"/>  </xsl:when>  <xsl:otherwise>  <fo:leader />  </xsl:otherwise>  </xsl:choose>  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="left" xsl:use-attribute-sets="itemNormal" display-align="center">  <xsl:value-of select="itemName"/>  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="center" xsl:use-attribute-sets="itemNormal" display-align="center">  <xsl:value-of select="unitName"/>  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal" display-align="center">  <xsl:if test="quantity != 'null' and quantity &gt;= 0">  <xsl:value-of select="foo:custom-num-format(quantity, '###.##0', 'european')"/>  </xsl:if>  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal" display-align="center">  <xsl:if test="unitPrice != 'null' and unitPrice &gt;= 0">  <xsl:value-of select="foo:custom-num-format(unitPrice, '###.##0', 'european')"/>  </xsl:if>  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal" display-align="center">  <xsl:if test="itemTotalAmountWithoutVat != 'null' and itemTotalAmountWithoutVat &gt;= 0">  <xsl:value-of select="foo:custom-num-format(itemTotalAmountWithoutVat, '###.##0', 'european')"/>  </xsl:if>  </fo:block>  </fo:table-cell>  </fo:table-row>  </xsl:for-each>  <xsl:call-template name="loop">  <xsl:with-param name="var">  <xsl:value-of select="count(//invoiceData/items/item)"/>  </xsl:with-param>  </xsl:call-template> |

**Chú ý**: Phần call-template ”loop” chính là phần tự động tạo ra các dòng trắng mặc định khi số lượng hàng hóa của hóa đơn không bằng số dòng tối thiểu trên 1 trang.

Nếu không muốn hệ thống sinh các dòng trắng mà để có bao nhiêu hàng hóa hiển thị bấy nhiêu sẽ xóa bỏ phần call template này đi

1. Lấy thuế suất

|  |
| --- |
| <xsl:choose>  <xsl:when test="invoiceData/invoiceTaxBreakdowns/invoiceTaxBreakdowns != ''">  <xsl:for-each select="invoiceData/invoiceTaxBreakdowns/invoiceTaxBreakdowns">  <xsl:choose>  <xsl:when test="vatPercentage != 'null' and vatPercentage &gt;= 0">  <fo:table-row>  <fo:table-cell number-columns-spanned="2" xsl:use-attribute-sets="boxSmall">  <fo:block text-align="center" xsl:use-attribute-sets="itemNormal">  <fo:wrapper xsl:use-attribute-sets="labelBold">Thuế suất GTGT </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">  (VAT rate): <xsl:value-of select="vatPercentage"/> %  </fo:wrapper>  </fo:block>  </fo:table-cell>  <fo:table-cell number-columns-spanned="3" xsl:use-attribute-sets="boxSmall">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal">  <fo:wrapper xsl:use-attribute-sets="labelBold">Tiền thuế GTGT </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(VAT amount): </fo:wrapper>  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal">  <xsl:if test="vatTaxAmount != 'null' and vatTaxAmount &gt;= 0">  <xsl:value-of select="foo:custom-num-format(vatTaxAmount, '###.##0', 'european')"/>  </xsl:if>  </fo:block>  </fo:table-cell>  </fo:table-row>  </xsl:when>  <xsl:otherwise>  <fo:table-row>  <fo:table-cell number-columns-spanned="2" xsl:use-attribute-sets="boxSmall">  <fo:block text-align="center" xsl:use-attribute-sets="itemNormal">  <fo:wrapper xsl:use-attribute-sets="labelBold">Thuế suất GTGT </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(VAT rate): </fo:wrapper>  <fo:inline xsl:use-attribute-sets="labelNormal">  .....\.....%  </fo:inline>  </fo:block>  </fo:table-cell>  <fo:table-cell number-columns-spanned="3" xsl:use-attribute-sets="boxSmall">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal">  <fo:wrapper xsl:use-attribute-sets="labelBold">Tiền thuế GTGT </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(VAT amount): </fo:wrapper>  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal">  .....\.....  </fo:block>  </fo:table-cell>  </fo:table-row>  </xsl:otherwise>  </xsl:choose>  </xsl:for-each>  </xsl:when>  <xsl:otherwise>  <fo:table-row>  <fo:table-cell number-columns-spanned="2" xsl:use-attribute-sets="boxSmall">  <fo:block text-align="center" xsl:use-attribute-sets="itemNormal">  <fo:wrapper xsl:use-attribute-sets="labelBold">Thuế suất GTGT </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(VAT rate): </fo:wrapper>  <fo:inline xsl:use-attribute-sets="labelNormal">  ..........%  </fo:inline>  </fo:block>  </fo:table-cell>  <fo:table-cell number-columns-spanned="3" xsl:use-attribute-sets="boxSmall">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal">  <fo:wrapper xsl:use-attribute-sets="labelBold">Tiền thuế GTGT </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(VAT amount): </fo:wrapper>  </fo:block>  </fo:table-cell>  <fo:table-cell xsl:use-attribute-sets="boxSmall">  <fo:block text-align="right" xsl:use-attribute-sets="itemNormal">  </fo:block>  </fo:table-cell>  </fo:table-row>  </xsl:otherwise>  </xsl:choose> |

Phần lấy thuế suất đang lấy theo 3 điều kiện

* Thuế suất có giá trị >= 0: Tương ứng với thuế 0%, 5%, 10% ...
* Thuế suất có giá trị < 0: Tương ứng thuế ’Không chịu thuế’ (-2) hoặc không kê khai tính thuế (-1)
* Thuế suất không có giá trị: Tương ứng với mẫu hóa đơn trắng tải ở phần lập thông báo phát hành

1. Chữ ký số của người mua, người bán, chuyển đổi

|  |
| --- |
| Khai báo 1 template lấy thông tin theo chữ ký số  <xsl:template name="tokenize">  <xsl:param name="pText"/>  <xsl:if test="string-length($pText)">  <xsl:choose>  <xsl:when test="contains($pText,',')">  <xsl:variable name="text">  <xsl:value-of select="substring-before($pText, ',')"/>  </xsl:variable>  <xsl:choose>  <xsl:when test="substring($text,1,3) = 'CN='">  <xsl:value-of select="substring-after($text, 'CN=')"/>  </xsl:when>  <xsl:otherwise>  <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select=  "substring-after($pText, ',')"/>  </xsl:call-template>  </xsl:otherwise>  </xsl:choose>  </xsl:when>  <xsl:otherwise>  <xsl:if test="substring($pText,1,3) = 'CN='">  <xsl:value-of select="substring-after($pText, 'CN=')"/>  </xsl:if>  </xsl:otherwise>  </xsl:choose>  </xsl:if>  </xsl:template> |
| Sử dụng template đã khai báo để lấy dữ liệu  <fo:table-row>  <fo:table-cell number-columns-spanned="2">  <fo:block margin-top="1mm" font-size="10pt">  <fo:table table-layout="fixed" width="100%">  <fo:table-column column-width="50%"/>  <fo:table-column column-width="50%"/>  <fo:table-body>  <fo:table-row>  <fo:table-cell>  <fo:block text-align="center">  <fo:wrapper xsl:use-attribute-sets="labelBold">Người mua hàng </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(Buyer)</fo:wrapper>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Ký, ghi rõ họ tên)</fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Sign, full name)</fo:inline>  </fo:block>  <fo:block text-align="center" display-align="center">  <fo:block padding-top="7px" margin-top="7px">  </fo:block>  <fo:block xsl:use-attribute-sets="signature">  <xsl:if test="not((//\*[local-name()='X509SubjectName'])[2]) = false() and (//\*[local-name()='X509SubjectName'])[2] != ''">  <fo:block-container absolute-position="absolute" text-align="center" left="1mm" right="1mm" top="11mm">  <fo:block>  <fo:external-graphic content-height="scale-to-fit" content-width="scale-to-fit" max-height="1.6cm" max-width="100%" scaling="uniform" src="url('signature.png')"/>  </fo:block>  </fo:block-container>  <fo:block xsl:use-attribute-sets="signature">  Ký bởi <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="(//\*[local-name()='X509SubjectName'])[2]"/>  </xsl:call-template>  </fo:block>  </xsl:if>  </fo:block>  </fo:block>  </fo:table-cell>  <fo:table-cell>  <fo:block text-align="center">  <fo:wrapper xsl:use-attribute-sets="labelBold">Người bán hàng </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(Seller)</fo:wrapper>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Ký, ghi rõ họ tên)</fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Sign, full name)</fo:inline>  </fo:block>  <fo:block text-align="center" display-align="center">  <fo:block padding-top="7px" margin-top="7px">  </fo:block>  <fo:block-container absolute-position="absolute" text-align="center" left="1mm" right="1mm" top="11mm">  <fo:block>  <fo:external-graphic content-height="scale-to-fit" content-width="scale-to-fit" max-height="1.6cm" max-width="100%" scaling="uniform" src="url('signature.png')"/>  </fo:block>  </fo:block-container>  <fo:block xsl:use-attribute-sets="signature">  <xsl:if test="invoiceData/seller/sellerLegalName != 'null'">  <fo:block xsl:use-attribute-sets="signature">  Ký bởi <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="(//\*[local-name()='X509SubjectName'])[1]"/>  </xsl:call-template>  </fo:block>  <fo:block xsl:use-attribute-sets="signature">  Ký ngày  <xsl:if test="invoiceData/invoiceIssuedDate != 'null' and invoiceData/invoiceIssuedDate != ''">  <xsl:value-of select="concat(substring(invoiceData/invoiceIssuedDate, 9, 2),'/',substring(invoiceData/invoiceIssuedDate, 6, 2),'/',substring(invoiceData/invoiceIssuedDate, 1, 4))"/>  </xsl:if>  </fo:block>  </xsl:if>  </fo:block>  </fo:block>  </fo:table-cell>  </fo:table-row>  </fo:table-body>  </fo:table>  </fo:block>  </fo:table-cell>  </fo:table-row> |

Hóa đơn chuyển đổi sẽ có phần khác so với bản thể hiện như sau

* Chuyển bản thể hiện của hóa đơn điện tử (Electronic invoice display) thành HÓA ĐƠN CHUYỂN ĐỔI TỪ HÓA ĐƠN ĐIỆN TỬ (Invoice converted from electronic invoice display)
* Phần chữ ký thêm 1 cột “người chuyển đổi”. Lấy thông tin Người chuyển đổi và Ngày chuyển đổi như file mẫu

|  |
| --- |
| <fo:table-row>  <fo:table-cell number-columns-spanned="2">  <fo:block margin-top="1mm" font-size="10pt">  <fo:table table-layout="fixed" width="100%">  <fo:table-column column-width="32%"/>  <fo:table-column column-width="32%"/>  <fo:table-column column-width="36%"/>  <fo:table-body>  <fo:table-row>  <fo:table-cell>  <fo:block text-align="center">  <fo:wrapper xsl:use-attribute-sets="labelBold">Người chuyển đổi </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(Exchanged by)</fo:wrapper>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Ký, ghi rõ họ tên)</fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Sign, full name)</fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline>    </fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline>    </fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline>    </fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline>    </fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelNormal">${exchangeUser}</fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:wrapper xsl:use-attribute-sets="labelBold">Ngày chuyển đổi </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(Exchange date):</fo:wrapper>  <fo:block xsl:use-attribute-sets="labelNormal">${exchangeDay}/${exchangeMonth}/${exchangeYear}</fo:block>  </fo:block>  </fo:table-cell>  <fo:table-cell>  <fo:block text-align="center">  <fo:wrapper xsl:use-attribute-sets="labelBold">Người mua hàng </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(Buyer)</fo:wrapper>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Ký, ghi rõ họ tên)</fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Sign, full name)</fo:inline>  </fo:block>  <fo:block text-align="center" display-align="center">  <fo:block padding-top="7px" margin-top="7px">  </fo:block>  <fo:block xsl:use-attribute-sets="signature">  <xsl:if test="not((//\*[local-name()='X509SubjectName'])[2]) = false() and (//\*[local-name()='X509SubjectName'])[2] != ''">  <fo:block-container absolute-position="absolute" text-align="center" left="1mm" right="1mm" top="11mm">  <fo:block>  <fo:external-graphic content-height="scale-to-fit" content-width="scale-to-fit" max-height="1.6cm" max-width="100%" scaling="uniform" src="url('signature.png')"/>  </fo:block>  </fo:block-container>  <fo:block xsl:use-attribute-sets="signature">  Ký bởi <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="(//\*[local-name()='X509SubjectName'])[2]"/>  </xsl:call-template>  </fo:block>  </xsl:if>  </fo:block>  </fo:block>  </fo:table-cell>  <fo:table-cell>  <fo:block text-align="center">  <fo:wrapper xsl:use-attribute-sets="labelBold">Người bán hàng </fo:wrapper>  <fo:wrapper xsl:use-attribute-sets="labelItalic">(Seller)</fo:wrapper>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Ký, ghi rõ họ tên)</fo:inline>  </fo:block>  <fo:block text-align="center">  <fo:inline xsl:use-attribute-sets="labelItalic">(Sign, full name)</fo:inline>  </fo:block>  <fo:block text-align="center" display-align="center">  <fo:block padding-top="7px" margin-top="7px">  </fo:block>  <fo:block-container absolute-position="absolute" text-align="center" left="1mm" right="1mm" top="11mm">  <fo:block>  <fo:external-graphic content-height="scale-to-fit" content-width="scale-to-fit" max-height="1.6cm" max-width="100%" scaling="uniform" src="url('signature.png')"/>  </fo:block>  </fo:block-container>  <fo:block xsl:use-attribute-sets="signature">  <xsl:if test="invoiceData/seller/sellerLegalName != 'null'">  <fo:block xsl:use-attribute-sets="signature">  Ký bởi <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="(//\*[local-name()='X509SubjectName'])[1]"/>  </xsl:call-template>  </fo:block>  <fo:block xsl:use-attribute-sets="signature">  Ký ngày  <xsl:if test="invoiceData/invoiceIssuedDate != 'null' and invoiceData/invoiceIssuedDate != ''">  <xsl:value-of select="concat(substring(invoiceData/invoiceIssuedDate, 9, 2),'/',substring(invoiceData/invoiceIssuedDate, 6, 2),'/',substring(invoiceData/invoiceIssuedDate, 1, 4))"/>  </xsl:if>  </fo:block>  </xsl:if>  </fo:block>  </fo:block>  </fo:table-cell>  </fo:table-row>  </fo:table-body>  </fo:table>  </fo:block>  </fo:table-cell>  </fo:table-row> |

1. Font chữ của hóa đơn

|  |
| --- |
| <fo:root font-family="Times New Roman"> |

Hiện tại hệ thống đáp ứng 1 số các font chữ sau:

Arial, Times New Roman, Gothic, Malgun Gothic

Một số font đặc thù: Tiếng Hàn, Tiếng Nhật, .... ngoài font chính thêm các font đặc thù

|  |
| --- |
| <fo:root font-family="Times New Roman, Malgun Gothic"> |

3.3. File invoice.xsl.

1. Định dạng tiền trong hóa đơn

Trường hợp 1: Đơn vị tiền là EURO hoặc VNĐ

|  |
| --- |
| <xsl:decimal-format decimal-separator="," grouping-separator="."/> |
| <td align="right" class= "boxSmall itemNormal">  <xsl:if test="inv:quantity != 'null' and inv:quantity >= 0">  <xsl:value-of select="format-number(inv:quantity, '###.##0,#########')"/>  </xsl:if>  </td> |

Trường hợp 2: Đơn vị tiền là US

|  |
| --- |
| <xsl:decimal-format decimal-separator="." grouping-separator=","/> |
| <td align="right" class= "boxSmall itemNormal">  <xsl:if test="inv:quantity != 'null' and inv:quantity >= 0">  <xsl:value-of select="format-number(inv:quantity, '###,##0.#########')"/>  </xsl:if>  </td> |

1. Khai báo style dùng chung.

|  |
| --- |
| .labelNormal{  padding : 2px 2px 2px 2px  }  .labelItalic{  padding : 2px 2px 2px 2px;  font-style: italic;  color: #000000;  } |

1. Chèn số dòng trống ở bảng hàng hóa

|  |
| --- |
| <xsl:template name="loop">  <xsl:param name="var"></xsl:param>  <xsl:choose>  <xsl:when test="$var &lt; 10 and $var &gt; 0">  <tr>  <td align="center" class= "boxSmall itemNormal">  <font class="labelNormal" ></font>  </td>  <td align="left" class= "boxSmall itemNormal">  </td>  <td align="center" class= "boxSmall itemNormal">  </td>  <td align="right" class= "boxSmall itemNormal">  </td>  <td align="right" class= "boxSmall itemNormal">  </td>  <td align="right" class= "boxSmall itemNormal">  </td>  </tr>  <xsl:call-template name="loop">  <xsl:with-param name="var">  <xsl:number value="number($var)+1" />  </xsl:with-param>  </xsl:call-template>  </xsl:when>  <xsl:otherwise>  </xsl:otherwise>  </xsl:choose>  </xsl:template> |

Đoạn template này ý nghĩa là để chèn dòng trống vào bảng hàng hóa tối thiểu trên một trang. Số lượng <td> bằng đúng số cột của bảng hàng hóa.

<xsl:when test="$var &lt; 10 and $var &gt; 0">

Tối thiểu bao nhiêu dòng trên 1 trang. VD như trên là 10 dòng

1. Thiết lập kích thước khổ giấy.

Đối với bản view web trực tiếp không có phần định dạng theo khổ giấy. Có bao nhiêu hóa đơn sẽ fill theo dạng liệt kê.

1. Fill dữ liệu bảng hàng hóa

|  |
| --- |
| <xsl:for-each select="inv:invoiceData/inv:items/inv:item">  <tr>  <td align="center" class= "boxSmall itemNormal">  <xsl:choose>  <xsl:when test="inv:lineNumber > 0">  <xsl:value-of select="inv:lineNumber"/>  </xsl:when>  <xsl:otherwise>  <font class="labelNormal" ></font>  </xsl:otherwise>  </xsl:choose>  </td>  <td align="left" class= "boxSmall itemNormal">  <xsl:value-of select="inv:itemName"/>  </td>  <td align="center" class= "boxSmall itemNormal">  <xsl:value-of select="inv:unitName"/>  </td>  <td align="right" class= "boxSmall itemNormal">  <xsl:if test="inv:quantity != 'null' and inv:quantity >= 0">  <xsl:value-of select="format-number(inv:quantity, '###.##0,#########')"/>  </xsl:if>  </td>  <td align="right" class= "boxSmall itemNormal">  <xsl:if test="inv:unitPrice != 'null' and inv:unitPrice >= 0">  <xsl:value-of select="format-number(inv:unitPrice, '###.##0,#########')"/>  </xsl:if>  </td>  <td align="right" class= "boxSmall itemNormal">  <xsl:if test="inv:itemTotalAmountWithoutVat != 'null' and inv:itemTotalAmountWithoutVat >= 0">  <xsl:value-of select="format-number(inv:itemTotalAmountWithoutVat, '###.##0,#########')"/>  </xsl:if>  </td>  </tr>  </xsl:for-each>  <xsl:call-template name="loop">  <xsl:with-param name="var">  <xsl:value-of select="count(//inv:invoiceData/inv:items/inv:item)"/>  </xsl:with-param>  </xsl:call-template> |

**Chú ý**: Phần call-template ”loop” chính là phần tự động tạo ra các dòng trắng mặc định khi số lượng hàng hóa của hóa đơn không bằng số dòng tối thiểu trên 1 trang.

Nếu không muốn hệ thống sinh các dòng trắng mà để có bao nhiêu hàng hóa hiển thị bấy nhiêu sẽ xóa bỏ phần call template này đi

1. Lấy thuế suất

|  |
| --- |
| <xsl:choose>  <xsl:when test="inv:invoiceData/inv:invoiceTaxBreakdowns/inv:invoiceTaxBreakdowns != 'null'">  <xsl:for-each select="inv:invoiceData/inv:invoiceTaxBreakdowns/inv:invoiceTaxBreakdowns">  <tr>  <td align="center" colspan="2" class= "boxSmall labelNormal">  <font class="labelBold">Thuế suất GTGT</font>  <font class="labelItalic">(VAT rate):</font>  <font class="labelNormal">  <xsl:choose>  <xsl:when test="inv:vatPercentage != 'null' and inv:vatPercentage >= 0">  <xsl:value-of select="inv:vatPercentage"/>%  </xsl:when>  <xsl:otherwise>  .....\.....%  </xsl:otherwise>  </xsl:choose>  </font>  </td>  <td align="right" colspan="3" class= "boxSmall labelNormal">  <font style= "" class="labelBold">Tiền thuế GTGT</font>  <font style= "" class="labelItalic">(VAT amount):</font>  </td>  <xsl:choose>  <xsl:when test="inv:vatPercentage != 'null' and inv:vatPercentage >= 0">  <td align="right" colspan="1" class= "boxSmall itemNormal">  <xsl:if test="inv:vatTaxAmount != 'null' and inv:vatTaxAmount >= 0">  <font class = "itemNormal">  <xsl:value-of select="format-number(inv:vatTaxAmount, '###.##0,#########')"/>  </font>  </xsl:if>  </td>  </xsl:when>  <xsl:otherwise>  <td align="right" colspan="1" class= "boxSmall itemNormal">  .....\.....  </td>  </xsl:otherwise>  </xsl:choose>  </tr>  </xsl:for-each>  </xsl:when>  <xsl:otherwise>  <tr>  <td align="center" colspan="2" class= "boxSmall labelNormal">  <font class="labelBold">Thuế suất GTGT</font>  <font class="labelItalic">(VAT rate):</font>  <font class="labelNormal">  ..........%  </font>  </td>  <td align="right" colspan="3" class= "boxSmall labelNormal">  <font style= "" class="labelBold">Tiền thuế GTGT</font>  <font style= "" class="labelItalic">(VAT amount):</font>  </td>  <td align="right" colspan="1" class= "boxSmall itemNormal">  </td>  </tr>  </xsl:otherwise>  </xsl:choose> |

Phần lấy thuế suất đang lấy theo 3 điều kiện

* Thuế suất có giá trị >= 0: Tương ứng với thuế 0%, 5%, 10% ...
* Thuế suất có giá trị < 0: Tương ứng thuế ’Không chịu thuế’ (-2) hoặc không kê khai tính thuế (-1)
* Thuế suất không có giá trị: Tương ứng với mẫu hóa đơn trắng tải ở phần lập thông báo phát hành

1. Chữ ký số của người mua, người bán

|  |
| --- |
| Khai báo 1 template lấy thông tin theo chữ ký số  <xsl:template name="tokenize">  <xsl:param name="pText"/>  <xsl:if test="string-length($pText)">  <xsl:choose>  <xsl:when test="contains($pText,',')">  <xsl:variable name="text">  <xsl:value-of select="substring-before($pText, ',')"/>  </xsl:variable>  <xsl:choose>  <xsl:when test="substring($text,1,3) = 'CN='">  <xsl:value-of select="substring-after($text, 'CN=')"/>  </xsl:when>  <xsl:otherwise>  <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select=  "substring-after($pText, ',')"/>  </xsl:call-template>  </xsl:otherwise>  </xsl:choose>  </xsl:when>  <xsl:otherwise>  <xsl:if test="substring($pText,1,3) = 'CN='">  <xsl:value-of select="substring-after($pText, 'CN=')"/>  </xsl:if>  </xsl:otherwise>  </xsl:choose>  </xsl:if>  </xsl:template> |
| Sử dụng template đã khai báo để lấy dữ liệu  <tr>  <td colspan="3">  <table width="100%">  <tr>  <td align="center" width="50%" style="vertical-align:top">  <font class="labelBold" text-align="top">Người mua hàng</font>  <font class="labelItalic" >(Buyer)</font>  <br/>  <font class="labelItalic" >(Ký, ghi rõ họ tên)</font>  <br/>  <font class="labelItalic" >(Sign, full name)</font>  </td>  <td align="center" width="50%">  <font class="labelBold" >Người bán hàng</font>  <font class="labelItalic" >(Seller)</font>  <br/>  <font class="labelItalic" >(Ký, ghi rõ họ tên)</font>  <br/>  <font class="labelItalic" >(Sign, full name)</font>  </td>  </tr>  <tr>  <td align="center">  <xsl:if test="not((//\*[local-name()='X509SubjectName'])[2]) = false() and (//\*[local-name()='X509SubjectName'])[2] != ''">  <div class="BG">  <div style="height: 30px" ></div>  <font class="labelBold" style="font-weight:bold;color: #FF0000;word-wrap:break-word">  Ký bởi <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="(//\*[local-name()='X509SubjectName'])[2]"/>  </xsl:call-template>  </font>  <div style="height: 10px" ></div>  </div>  </xsl:if>  </td>  <td align="center" width="50%">  <div class="BG">  <div style="height: 30px" ></div>  <xsl:if test="inv:invoiceData/inv:seller/inv:sellerLegalName != 'null'">  <font class="labelBold" style="font-weight:bold;color: #FF0000;word-wrap:break-word">  Ký bởi <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="(//\*[local-name()='X509SubjectName'])[1]"/>  </xsl:call-template>  </font>  <br/>  <font class="labelBold" style="font-weight:bold;color: #FF0000;word-wrap:break-word">  Ký ngày  <xsl:if test="inv:invoiceData/inv:invoiceIssuedDate != 'null' and inv:invoiceData/inv:invoiceIssuedDate != ''">  <xsl:value-of select="concat(substring(inv:invoiceData/inv:invoiceIssuedDate, 9, 2),'/',substring(inv:invoiceData/inv:invoiceIssuedDate, 6, 2),'/',substring(inv:invoiceData/inv:invoiceIssuedDate, 1, 4))"/>  </xsl:if>  </font>  </xsl:if>  <div style="height: 10px" ></div>  </div>  </td>  </tr>  </table>  </td>  </tr> |

1. Font chữ của hóa đơn

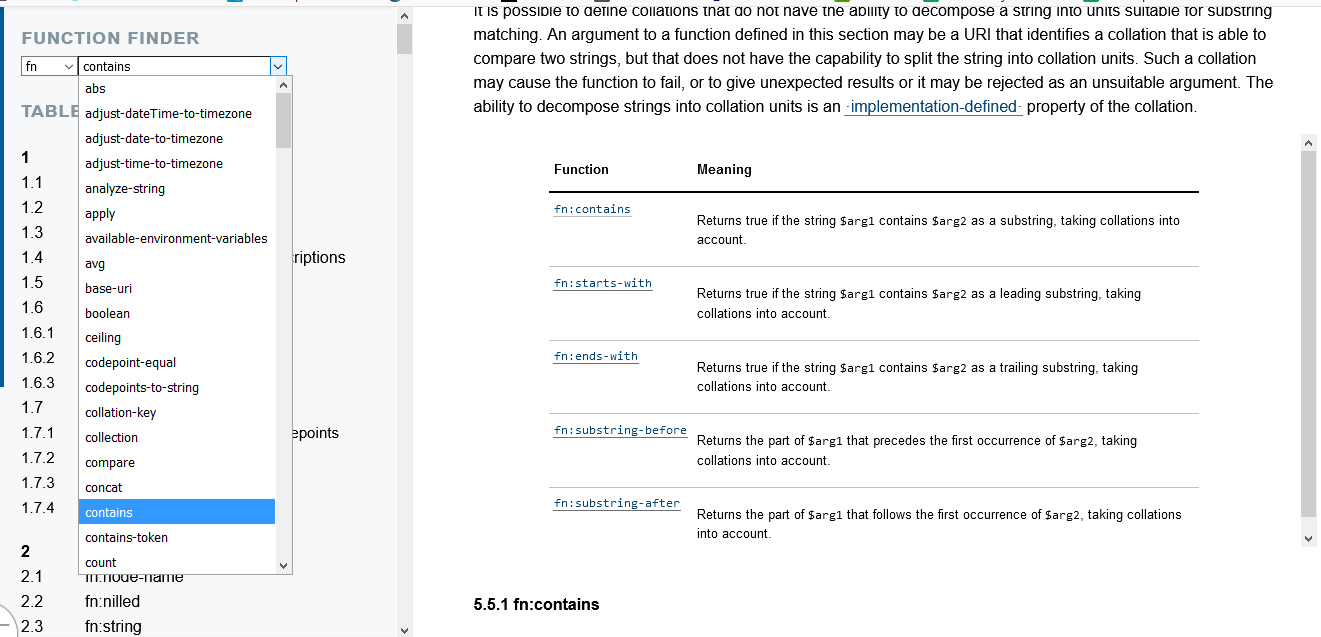
|  |
| --- |
| .serif {  font-family: "Times New Roman";  } |

Hiện tại hệ thống đáp ứng 1 số các font chữ sau:

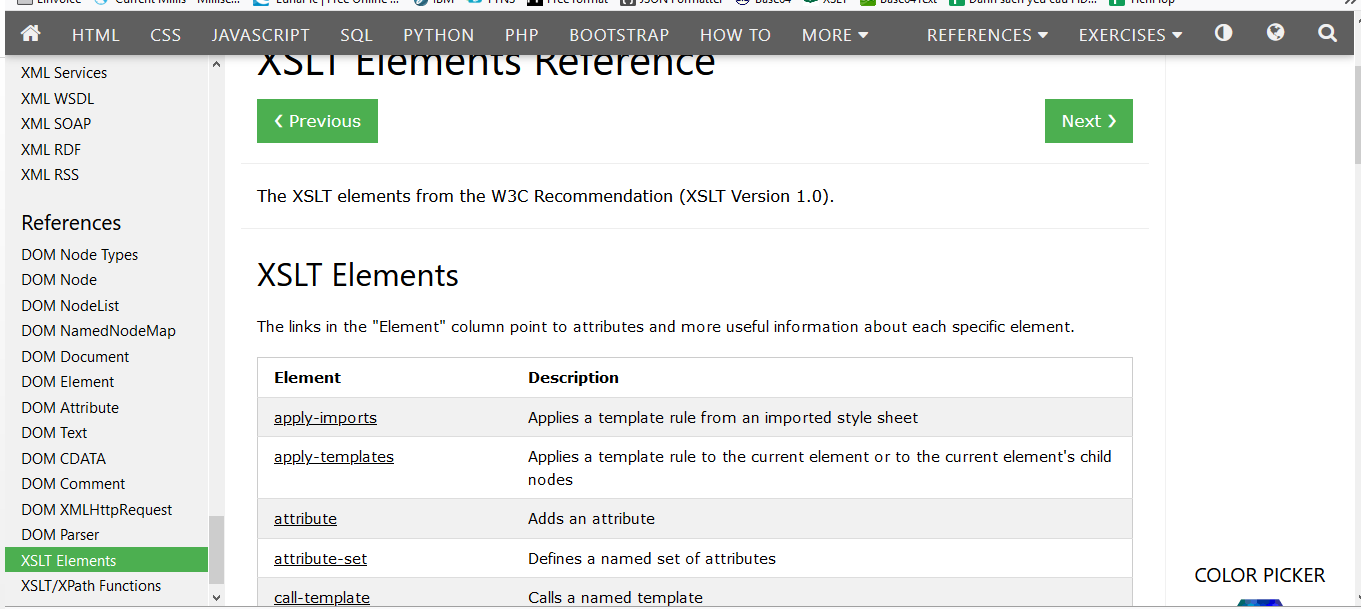
Arial, Times New Roman, Gothic, Malgun Gothic

3.4 Một số khai báo và hàm hay sử dụng (if, choose, template, variable, contains, count, sum,....)

Tham khảo các hàm có thể sử dụng ở trang sau: <https://www.w3.org/TR/xpath-functions/#func-not>



Các khai báo có thể sử dụng trang sau: <https://www.w3schools.com/xml/xsl_elementref.asp>



Tóm tắt cách sử dụng và 1 số ví dụ để dễ hình dung về các dùng

|  |  |  |
| --- | --- | --- |
|  | Không điều kiện | Có điều kiện theo dữ liệu |
| If | <xsl:if test="*điều kiện*">   Thực hiện tác vụ  </xsl:if> | Nếu loại tiền tệ != VND thì hiển thị thêm loại tiền tệ vào sau các số tiền  <xsl:if test="//invoiceData/currencyCode != 'VND'">  <xsl:value-of select="//invoiceData/currencyCode"/>  </xsl:if> |
| choose | <xsl:choose>   <xsl:when test="*điều kiện1*">  Thực hiện tác vụ   </xsl:when>  <xsl:when test="*điều kiện2*">  Thực hiện tác vụ   </xsl:when>   <xsl:otherwise>  Thực hiện tác vụ    </xsl:otherwise> </xsl:choose> | Nếu USD thì gán cứng 2 số sau số thập phâ, còn lại sẽ hiển thị tự động theo cấu hình  <xsl:choose>  <xsl:when test="//invoiceData/currencyCode = 'USD'">  <xsl:value-of select="format-number(unitPrice, '###.##0,00', 'european')"/>  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="foo:custom-num-format(unitPrice, '###.##0', 'european')"/>  </xsl:otherwise>  </xsl:choose> |
| Variable | <xsl:variable name="Tên(viết liền không dấu)" select="điều kiện">    <!-- Content:template -->  </xsl:variable>  Cách gọi đến biến đã khai báo  <xsl:value-of select="$Tên"/> | Khai báo 1 biến lấy giá trị tổng tiền thuế hàng hóa dạng phí khác  <xsl:variable name="sumtotalOtherTaxAmount" select="sum(//invoiceData/items/item[selection = 5]/vatAmount)" />  Gọi biến đã khai báo  <xsl:value-of select="foo:custom-num-format(vatTaxAmount - $sumtotalOtherTaxAmount, '###.##0', 'european')"/> |
| Contains | contains(Bien1 as xs:string?, Bien2 as xs:string?) as xs:boolean  Kiểm tra xem Bien2 có bao gồm trong Bien1 không.  Nếu có thì return true  Nếu không thì return false | Kiểm tra ghi chú xem có chứa XXX không  (Phục vụ cho thuế nhà thầu)  <xsl:choose>  <xsl:when test="contains(//invoiceData/metadata/invoiceNote,'XXX')">  xxx  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="vatPercentage"/>  </xsl:otherwise>  </xsl:choose> |
| Sum | Tính tổng số lượng các mặt hàng  <xsl:if test="sum(//invoiceData/items/item/quantity) &gt;= 0">  <xsl:value-of select="foo:custom-num-format(sum(//invoiceData/items/item/quantity), '###.##0', 'european')"/>  </xsl:if> | Tính tổng số lượng các mặt hàng loại là hàng hóa  <xsl:if test="sum(//invoiceData/items/item[not(selection) or selection = 1]/quantity) &gt;= 0">  <xsl:value-of select="foo:custom-num-format(sum(//invoiceData/items/item[not(selection) or selection = 1]/quantity), '###.##0', 'european')"/>  </xsl:if> |
| count | Đếm số lượng hàng hóa  <xsl:call-template name="loop">  <xsl:with-param name="var">  <xsl:value-of select="count(//invoiceData/items/item)"/>  </xsl:with-param>  </xsl:call-template> | Đếm số lượng hàng hóa dạng != loại phí khác  <xsl:call-template name="loop">  <xsl:with-param name="var">  <xsl:value-of select="count(//invoiceData/items/item[not(selection) or selection != '5'])"/>  </xsl:with-param>  </xsl:call-template> |
|  |  |  |

IV. PHẦN MỀM THIẾT KẾ VÀ CÁCH XEM KẾT QUẢ CÁC FILE THIẾT KẾ.

4.1. Các phần mềm có thể view và sửa code xsl

* Notepad
* Notepad++
* Visual studio ....

4.2. Tool render từ xsl + xml sang pdf

Có thể tải tool render hỗ trợ việc chỉnh sửa mẫu hóa đơn trực quan và xác định lỗi bằng những cách sau:

- Tải trên link nội bộ: <ftp://10.60.109.43/GPDN/TAI_LIEU/SINVOICE/TK%20HOA%20DON/Tool_view_pdf>

- Tải trên link public: <https://drive.google.com/file/d/1ytiUuDbFiknQGkZ22rqB8YclW2rSeOhS/view>

4.3. Xem kết quả

a) File invoice-fo.xsl và exchange-invoice-fo.xsl

B1: Copy thư mục tool về ổ public để tránh lỗi không tìm thấy đường link fop/fop.xml

B2: Chạy file pdf\_render\_tool.jar

**(Để chạy được file này, yêu cầu máy cần cài tối thiểu java 7 trở lên)**

B3: Copy link thư mục chứa 3 file code xsl + xml vào textbox: **Thư mục**

B4: Nếu cầu thay logo, watermark thì copy 2 file này vào thư mục Tool\_view\_pdf\fop được lưu tương ứng trên máy

B5: Nhấn button **Tạo file**

B6: Kiểm tra file ban\_the\_hien.pdf và ban\_chuyen\_doi.pdf trong thư mục chứa file code để xem thể hiện của hóa đơn sau khi chỉnh sửa mẫu được render.

**Chú ý**:

Mở file .log trong thư mục logs sửa hết những WARNING (1 số Waring về border và padding của fo:region-xxx là chấp nhận được) hoặc Lỗi (ERROR) nếu có.

b) file invoice.xsl

Click chuột phải vào file “data.xml” -> open with “firefox”, ”IE” hoặc tool xem file xml

Hướng dẫn check lỗi trên tool view đã mô tả chi tiết trong video:

<https://www.youtube.com/watch?v=lamjZXY512w&t=2s>

V. MỘT SỐ TRƯỜNG HỢP THƯỜNG XUYÊN HỖ TRỢ

CHÚ Ý: Link video hướng dẫn thiết kế trên Youtube các trường hợp thường xuyên hỗ trợ:

<https://www.youtube.com/channel/UC2MmM9PgIP6SavMhTSrvKZQ>

Một số trường hợp ví dụ cụ thể

1. Thêm bớt thông tin ngày ký trên hóa đơn

a. Invoice-fo.xsl và exchange-invoice-fo.xsl

Tìm kiếm đến phần chữ ký của người bán hàng



Copy câu lệnh sau vào dưới block của thông tin Ký bởi



|  |
| --- |
| <fo:block xsl:use-attribute-sets="signature">  Ký ngày  <xsl:if test="invoiceData/invoiceIssuedDate != 'null' and invoiceData/invoiceIssuedDate != ''">  <xsl:value-of select="concat(substring(invoiceData/invoiceIssuedDate, 9, 2),'/',substring(invoiceData/invoiceIssuedDate, 6, 2),'/',substring(invoiceData/invoiceIssuedDate, 1, 4))"/>  </xsl:if>  </fo:block> |

Sau khi Copy sẽ như sau:



b. Invoice.xsl

Tìm đến vị trí chữ ký của người bán hàng



Copy đoạnh lệnh sau xuống dưới khối Ký bởi



|  |
| --- |
| <br/>  <font class="labelBold" style="font-weight:bold;color: #FF0000;word-wrap:break-word">  Ký ngày  <xsl:if test="inv:invoiceData/inv:invoiceIssuedDate != 'null' and inv:invoiceData/inv:invoiceIssuedDate != ''">  <xsl:value-of select="concat(substring(inv:invoiceData/inv:invoiceIssuedDate, 9, 2),'/',substring(inv:invoiceData/inv:invoiceIssuedDate, 6, 2),'/',substring(inv:invoiceData/inv:invoiceIssuedDate, 1, 4))"/>  </xsl:if>  </font> |



2. Xóa 1 số dữ liệu: fax, website, email ….

a. Invoice-fo và exchange-invoice-fo

Tìm đến vị trí của trường muốn xóa



Cần kiểm tra xem cùng table-row với trường muốn xóa còn có table-cell nào không

Nếu không có – Xóa bình thường

Nếu có – Sau khi xóa cần gán lại các độ rộng phù hợp cho table-column sau khi đã xóa trường cần thiết

Sau khi xóa sẽ như sau



b. Invoice.xsl

Tìm đến vị trí và xóa

Set lại width của các <td> cùng <tr>



Sau khi xóa sẽ được như sau



3. Thêm cột trong bảng hàng hóa (Thêm cột Số lô, Hạn dùng)

Thực hiện theo các bước sau :

- Thêm cell vào header

- Thêm cell và row dữ liệu

- Thêm cell vào row trắng tự sinh

- Thêm cell vào các row tổng hợp

(Tham khảo trong video để nắm rõ từng bước)

4. Giãn cách trường Mã số thuế (Xuống dòng Mã hàng hóa)

Trường hợp MST cần giãn cách các ký tự VD: 0100109106 -> 0 1 0 0 1 0 9 1 0 6

Hoặc các trường hàng hóa trong bảng (Mã hàng, tên hàng, số lô, hạn dùng) viết liền không có dấu cách dẫn đến việc không xác định được 1 từ nên không tự động xuống dòng thì cần thực hiện chèn ký tự nhận diện vào để giúp cho việc tự động xuống dòng.

Trong các trường hợp này ta thực hiện chỉnh sửa như sau:

- Khai báo 1 template (Khai báo chung giữa invocie.xsl, invoice-fo.xsl, exchange-invoie-fo.xsl)

|  |
| --- |
| <xsl:template name="zero\_width\_space\_1">  <xsl:param name="data"/>  <xsl:param name="counter" select="0"/>  <xsl:param name="chr" select="'&#8203;'"/>  <xsl:choose>  <xsl:when test="$counter &lt;= string-length($data)">  <xsl:value-of select='concat(substring($data,$counter,1),$chr)'/>  <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="$data"/>  <xsl:with-param name="counter" select="$counter+1"/>  <xsl:with-param name="chr" select="$chr"/>  </xsl:call-template>  </xsl:when>  <xsl:otherwise>  </xsl:otherwise>  </xsl:choose>  </xsl:template> |

* Call template tại vị trí get dữ liệu Mã số thuế

1. Invoice.xsl

Mã số thuế

|  |
| --- |
| <td align="center"  <font class="labelNormalHeader" >Mã số thuế: </font>  <font class="labelBoldHeader">  <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="inv:invoiceData/inv:seller/inv:sellerTaxCode"/>  <xsl:with-param name="counter" select="0"/>  <xsl:with-param name="chr" select="'&#160;'"/>  </xsl:call-template>  </font>  </td> |

Mã hàng hóa (Hoặc các trường tương ứng)

|  |
| --- |
| <td align="left" class= "boxSmall itemNormal">  <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="inv:itemCode"/>  <xsl:with-param name="counter" select="0"/>  <xsl:with-param name="chr" select="'&#8203;'"/>  </xsl:call-template>  </td> |

1. invoice-fo.xsl và exchange-invoice-fo.xsl

Mã số thuế

|  |
| --- |
| <fo:block font-size="10pt" text-align="center" xsl:use-attribute-sets="labelNormalHeader">  <fo:wrapper xsl:use-attribute-sets="labelNormalHeader">Mã số thuế: </fo:wrapper>  <fo:inline xsl:use-attribute-sets="labelBoldHeader">  <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="invoiceData/seller/sellerTaxCode"/>  <xsl:with-param name="counter" select="0"/>  <xsl:with-param name="chr" select="'&#160;'"/>  </xsl:call-template>  </fo:inline>  </fo:block> |

Mã hàng hóa (Hoặc các trường tương ứng)

|  |
| --- |
| <fo:table-cell xsl:use-attribute-sets="boxSmall" display-align="center">  <fo:block text-align="left" xsl:use-attribute-sets="itemNormal">  <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="itemCode"/>  <xsl:with-param name="counter" select="0"/>  <xsl:with-param name="chr" select="'&#8203;'"/>  </xsl:call-template>  </fo:block>  </fo:table-cell> |

5. Bỏ hiển thị 0 trên mẫu hóa đơn

Với các dữ liệu kiểu số như:

số lượng - quantity,

đơn giá – unitPirce,

thành tiền – itemTotalAmountWithoutVat

…

Trong bảng hàng hóa, thông thường sẽ hiển thị theo điều kiện >= 0 hoặc &gt;=

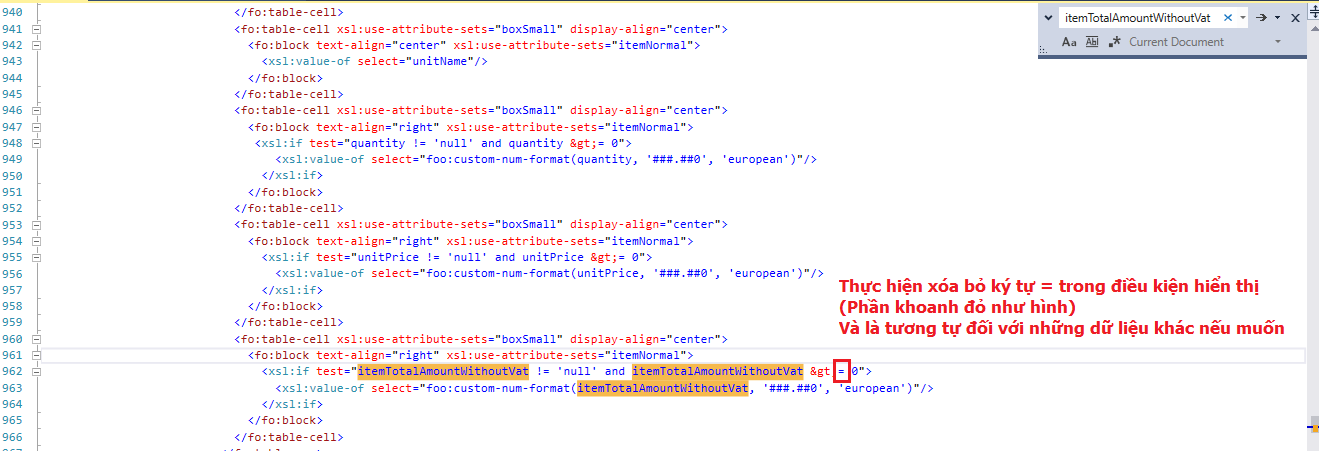
Tùy vào nhu cầu của KH, nếu KH không muốn hiển thị dữ liệu = 0 thì chỉ cần bỏ điều kiện đi tại trường dữ liệu mong muốn.

VD: Bỏ phần hiển thị 0 ở cột thành tiền

a. invoice-fo.xsl và exchange-invoice-fo.xsl

Đầu tiên tìm kiếm đến vị trí cột thành tiền theo key: itemTotalAmountWithoutVat

Tiếp theo xóa bỏ điều kiện =



b. invoice.xsl

Đầu tiên tìm kiếm đến vị trí cột thành tiền theo key: itemTotalAmountWithoutVat

Tiếp theo xóa bỏ điều kiện =



6. Chỉnh sửa thuế suất không chịu thuế (Hiển thị …\....%)

Theo thông tư 39 của BTC đã nêu:

2.1. Tổ chức nộp thuế theo phương pháp khấu trừ thuế bán hàng hóa, dịch vụ thuộc đối tượng không chịu thuế GTGT, đối tượng được miễn thuế GTGT thì sử dụng hóa đơn GTGT, trên hóa đơn GTGT chỉ ghi dòng giá bán là giá thanh toán, dòng thuế suất, số thuế GTGT không ghi và gạch bỏ.

Để đáp ứng yêu cầu thông tư 39. Các mẫu hóa đơn chuẩn hiện tại đã có phần hiển thị này. Tuy nhiên, 1 số khách hàng cũ sử dụng mẫu chưa được chuẩn hóa sẽ hiển thị thuế -1, -2%. Th này điều chỉnh đơn giản như sau (Thông thường chỉ cần điều chỉnh file invoice.xsl):

a. Invoice-fo.xsl và exchange-invoice-fo.xsl

b. Invoice.xsl

Tìm đến vị trí hiển thị thuế với key: vatPercentage.

Copy đoạn code bên dưới đè vào phần hiển thị thuế như sau:

|  |
| --- |
| <xsl:choose>  <xsl:when test="inv:vatPercentage != 'null' and inv:vatPercentage >= 0">  <xsl:value-of select="inv:vatPercentage"/>%  </xsl:when>  <xsl:otherwise>  .....\.....%  </xsl:otherwise>  </xsl:choose> |

Hình ảnh mẫu hóa đơn hiển thị -1, -2%



Sau khi copy phần code bên trên sẽ được như sau:



7. Điều chỉnh vị trí, kích thước của logo và watermark.

**Chú ý**: Hình ảnh logo, watermark sẽ quyết định 1 phần hiệu năng tải bản pdf và view web hóa đơn. Do vậy không nhất thiết phải để hình ảnh to, nên để hình ảnh vừa phải (logo <300px và watermark <1500px).

Hiển thị logo và watermark đều cần xác định được vùng hiển thị, chiều cao tối đa, .... Ảnh hưởng đến hiển thị ngoài phụ thuộc vào code, còn phụ thuộc vào chính hình ảnh logo, watermark: Như viền trắng bao quanh, hình ảnh vỡ viền,....

Dưới đây sẽ hướng dẫn về phần chỉnh sửa code phần hiển thị logo watermark. Ngoài những giá trị hướng dẫn bên dưới không thực hiện chỉnh các thông số còn lại

1. Logo

Invoice-fo.xsl và exchange-invoice-fo.xsl

|  |
| --- |
| <fo:block margin-left="3px">  <fo:external-graphic content-height="scale-to-fit" max-height="3cm" content-width="scale-to-fit" max-width="100%" scaling="uniform" src="url('logo.png')">  </fo:external-graphic>  </fo:block> |

Text-align: Phần khai báo vị trí hiển thị trong khung. Bao gồm các giá trị: center –giữa, left – bên trái, right – bên phải

Max-height: chiều cao tối đa của ảnh logo. Điền 1 con số phù hợp: 90px, 2.5cm .....

Invoice.xsl

|  |
| --- |
| <td align="center" width = "24%">  <img src="logo.png" style="max-height: 90px; max-width: 100%;" align="middle"/>  </td> |

Align: Phần khai báo vị trí hiển thị trong khung. Bao gồm các giá trị: center –giữa, left – bên trái, right – bên phải

Max-height: chiều cao tối đa của ảnh logo. Điền 1 con số phù hợp: 90px, 2.5cm .....

**Chú ý**: Đối với TH khi chọn mẫu chuẩn không chọn logo thì hệ thống sẽ tự động xóa phần logo.png trong file invoice.xsl. Nếu sau này KH cần thêm logo, chỉ cần vào phần invoice.xsl thêm lại tên logo.png trong thẻ <img src="logo.png" style="max-height: 90px; max-width: 100%;" align="middle"/> và add logo lên phần quản lý mẫu hóa đơn như bình thường

b. watermark

invoice-fo.xsl và exchange-invoice-fo.xsl

|  |
| --- |
| <fo:block-container absolute-position="fixed" text-align="center" display-align="before" top="80mm" left="10mm" right="10mm">  <fo:block>  <fo:external-graphic content-height="scale-to-fit" content-width="scale-to-fit" max-height="16cm" max-width="100%" scaling="uniform" src="url('watermark.png')"/>  </fo:block>  </fo:block-container> |

Top: xác định vị trí hình ảnh watermark bắt đầu hiển thị tính từ điểm trên cùng của khổ giấy. Điền 1 giá trị số vào. Muốn hiển thị càng cao thì thông số càng bé: 10mm, 100mm 45mm.....

Left: xác định vị trí hình ảnh watermark cách viền trái bao nhiêu. Thông thường để 10mm để tránh hình ảnh đè lên viền.

Right: xác định vị trí hình ảnh watermark cách viền phải bao nhiêu. Thông thường để 10mm để tránh hình ảnh đè lên viền.

Max-height: chiều cao tối da mà watermark có thể hiển thị. Điền 1 giá trị số vào: 15cm, 110mm ...

invoice.xsl

|  |
| --- |
| <table id='section-to-print' ALIGN="center" class = "serif boxLarge" style="background-image:url(watermark.png); background-repeat:no-repeat;background-position: center 300px;"> |

Background-position: Xác định vị trí hình ảnh watermark bắt đầu hiển thị. Điền 1 giá trị số vào: 350px ....

Đối với việc gán theo size cụ thể (Do ảnh quá to hoặc quá bé) thì sửa như sau:

|  |
| --- |
| <table id='section-to-print' ALIGN="center" class = "serif boxLarge" style="font-size: 10pt; background-image:url(watermark.png); background-repeat:no-repeat;background-position: center 300px; background-size:350px auto"> |

Giá trị 350px chỉnh tùy thuộc theo ảnh của KH chỉnh sửa sao cho phù hợp

8. Xuống dòng dữ liệu theo ý muốn

Do hiện tại hệ thống chưa cho phép nhập liệu theo kiểu nhiều dòng, do đó, nhiều dữ liệu khi muốn xuống dòng thì cần thông qua 1 số bước sau để có thể hiển thị được dữ liệu dạng nhiều dòng:

* B1: Lựa chọn 1 ký tự để phân biệt xuống dòng khi hiển thị lên mẫu hóa đơn (Thông thường các giá trị này phải là ký tự đặc biệt, trong dữ liệu k xuất hiện để tránh TH xuống dòng k đúng mục đích: VD: số điện thoại, số tài khoản xuống dòng bằng ký tự ;......)
* B2: Điều chỉnh mẫu hóa đơn phần hiển thị dữ liệu theo hướng dẫn bên dưới

1. invoice-fo.xsl và exchange-invoice-fo.xsl

|  |
| --- |
| <xsl:choose>  <xsl:when test="contains(DULIEU,'KYTU')">  <xsl:value-of select="replace(DULIEU,'KYTU','&#x2028;')"/>  </xsl:when>  <xsl:otherwise>  <xsl:value-of select=" DULIEU "/>  </xsl:otherwise>  </xsl:choose> |

**DULIEU**: Nguồn lấy dữ liệu và xử lý:

VD: invoiceData/seller/sellerBankAccount, invoiceData/seller/sellerLegalName, ….

**KYTU**: Đây là dữ liệu đánh dấu việc xuống dòng.

VD: số tài khoản sử dụng dấu ; để xuống dòng

&#x2028; Ký tự xuống dòng

1 số ví dụ cụ thể:

Xuống dòng số tài khoản cách nhau bằng dấu ;

|  |
| --- |
| <xsl:choose>  <xsl:when test="contains(invoiceData/seller/sellerBankAccount,';')">  <xsl:value-of select="replace(invoiceData/seller/sellerBankAccount,';','&#x2028;')"/>  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="invoiceData/seller/sellerBankAccount"/>  </xsl:otherwise>  </xsl:choose> |

Xuống dòng tên người bán hàng bàng chữ “CHI”

|  |
| --- |
| <xsl:choose>  <xsl:when test="contains(invoiceData/seller/sellerLegalName,'CHI')">  <xsl:value-of select="replace(invoiceData/seller/sellerLegalName,'CHI','&#x2028;CHI')"/>  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="invoiceData/seller/sellerLegalName "/>  </xsl:otherwise>  </xsl:choose> |

1. invoice.xsl

Kiểm tra xem file đã khai báo template chưa. Nếu chưa thì khai báo bằng cách copy template này vào. Có rồi thì k cần thêm vì thêm sẽ báo lỗi trùng tên

|  |
| --- |
| <xsl:template name="string-replace-all">  <xsl:param name="text"/>  <xsl:param name="replace"/>  <xsl:param name="by"/>  <xsl:param name="spl"/>  <xsl:choose>  <xsl:when test="contains($text,$replace)">  <xsl:value-of select="substring-before($text,$replace)"/>  <xsl:value-of select="$by"/>  <br/>  <xsl:value-of select="$spl"/>  <xsl:call-template name="string-replace-all">  <xsl:with-param name="text" select="substring-after($text,$replace)"/>  <xsl:with-param name="replace" select="$replace"/>  <xsl:with-param name="by" select="$by"/>  <xsl:with-param name="spl" select="$spl"/>  </xsl:call-template>  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="$text"/>  </xsl:otherwise>  </xsl:choose>  </xsl:template> |

Phần lấy dữ liệu hiển thị

|  |
| --- |
| <xsl:variable name="FeatureInfo" select="DULIEU" />  <xsl:call-template name="string-replace-all">  <xsl:with-param name="text" select="$FeatureInfo"/>  <xsl:with-param name="replace" select="'KYTU'"/>  <xsl:with-param name="by" select="'&#160;'"/>  <xsl:with-param name="spl" select="''"/>  </xsl:call-template> |

**DULIEU**: Nguồn lấy dữ liệu và xử lý:

VD: inv:invoiceData/inv:seller/inv:sellerBankAccount, …

**KYTU**: Đây là dữ liệu đánh dấu việc xuống dòng.

VD: số tài khoản sử dụng dấu ; để xuống dòng

1 số ví dụ cụ thể:

Xuống dòng số tài khoản cách nhau bằng dấu ;

|  |
| --- |
| <xsl:variable name="FeatureInfo" select="inv:invoiceData/inv:seller/inv:sellerBankAccount" />  <xsl:call-template name="string-replace-all">  <xsl:with-param name="text" select="$FeatureInfo"/>  <xsl:with-param name="replace" select="';'"/>  <xsl:with-param name="by" select="'&#160;'"/>  <xsl:with-param name="spl" select="''"/>  </xsl:call-template> |

Xuống dòng tên người bán hàng bàng chữ “CHI”

|  |
| --- |
| <xsl:variable name="FeatureInfo" select="inv:invoiceData/inv:seller/inv:sellerLegalName" />  <xsl:call-template name="string-replace-all">  <xsl:with-param name="text" select="$FeatureInfo"/>  <xsl:with-param name="replace" select="'CHI'"/>  <xsl:with-param name="by" select="'&#160;'"/>  <xsl:with-param name="spl" select="'CHI'"/>  </xsl:call-template> |

9. Gán cứng số lượng chữ số sau dấu thập phân

1. invoice-fo.xsl và exchange-invoice-fo.xsl

Hiển thị các số hiện ta đa phần đang sử dụng function tự khai báo để hiển thị.

|  |
| --- |
| <xsl:if test="quantity != 'null' and quantity &gt;= 0">  <xsl:value-of select="foo:custom-num-format(quantity, '###.##0', 'european')"/>  </xsl:if> |

Tuy nhiên, đối với các trường hợp cần gán cứng 1 số chữ số sau đấu thập phân, cần điều chỉnh về function format mặc định của number như sau

* Định dạng european

|  |
| --- |
| <xsl:if test="quantity != 'null' and quantity &gt;= 0">  <xsl:value-of select="format-number(quantity, '###.##0,0000', 'european')"/>  </xsl:if> |

* Định dạng us

|  |
| --- |
| <xsl:if test="quantity != 'null' and quantity &gt;= 0">  <xsl:value-of select="format-number(quantity, '###,##0.0000', us)"/>  </xsl:if> |

Chú ý: Tùy vào định dạng hiển thị . và , cho số đã đăng ký thuế để chỉnh format về cho đúng.

Số chữ số 0 sau dấu thập phân sẽ là số chữ số sẽ luôn hiển thị mặc định trên mẫu sau phần thập phân (Nếu không có giá trị sẽ hiển thị thành 0 để đủ số lượng ký tự đã khai báo)

VD:

Gán cứng 4 số sau số thập phân:

18,12 -> 18,1200 (đối với european) hoặc 18.12 -> 18.1200 (đối với us)

1. invoice.xsl

Hiển thị bản web sẽ đơn giản hơn vì chỉ sử dụng format-number chuẩn mặc định.

|  |
| --- |
| <xsl:if test="inv:quantity != 'null' and inv:quantity >= 0">  <xsl:value-of select="format-number(inv:quantity, '###.##0,#########')"/>  </xsl:if> |

Chuyển sang định dạng gán cứng chỉ cần chuyển # -> 0 theo số chữ số muốn gán cứng

* Định dạng european

|  |
| --- |
| <xsl:if test="inv:quantity != 'null' and inv:quantity >= 0">  <xsl:value-of select="format-number(inv:quantity, '###.##0,0000')"/>  </xsl:if> |

* Định dạng us

|  |
| --- |
| <xsl:if test="inv:quantity != 'null' and inv:quantity >= 0">  <xsl:value-of select="format-number(inv:quantity, '###,##0.0000')"/>  </xsl:if> |

VD:

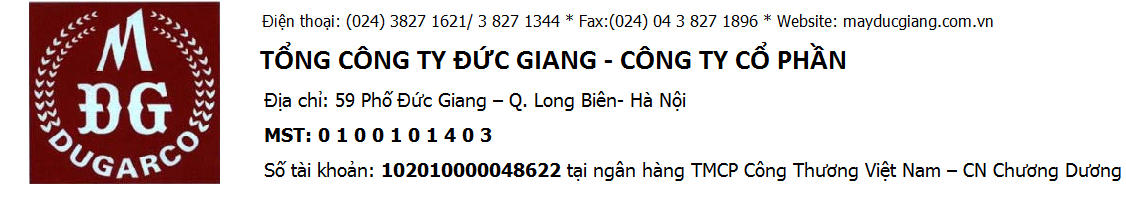
Gán cứng 4 số sau số thập phân:

18,12 -> 18,1200 (đối với european) hoặc 18.12 -> 18.1200 (đối với us)

10. Mẫu hóa đơn chi nhánh chứa thông tin của công ty

**Cách xử lý**: Thông thường trường hợp này có 2 hướng xử lý như sau:

* Thiết kế logo + thông tin công ty thành 1 hình ảnh như ví dụ:



* Gán cứng giá trị thông tin công ty trên mẫu hóa đơn.

Còn phần lấy dữ liệu theo người bán hàng sẽ đi theo user và lấy thông tin từ seller (thông tin người bán hàng). Nếu muốn xuất hóa đơn lấy được thông tin chi nhánh cần

* Tạo danh mục chi nhánh hoặc cửa hàng
* Lập thông báo phát hành cho chi nhánh hoặc cửa hàng
* Tạo user cho chi nhánh hoặc cửa hàng
* Đăng nhập bằng user chi nhánh hoặc cửa hàng để xuất hóa đơn

**Chú ý**: Không nên sử dụng trường động để nhập liệu cho cửa hàng (Những trường hợp trước đây là do hệ thống chưa có phần tạo danh mục chi nhánh không có mã số thuế nên bất đắc dĩ mới sử dụng cách thêm trường động) vì có 1 số bất tiện sau:

* Cần nhập dữ liệu trường động khi lập từng hóa đơn
* Không quản lý được theo dạng công ty chi nhánh: Tự động lấy dữ liệu khi xuất hóa đơn theo từng user, Phân quyền xuất hóa đơn, xuất báo cáo theo từng chi nhánh, ...

11. Lấy 1 dữ liệu không chứa trong mảng

Thông thường khi sử dụng vòng for-each ta sẽ dùng để lặp lại 1 mảng dữ liệu nào đó (VD: Mảng hàng hóa, mảng thuế, mảng chỉ số đồng hồ công tơ, .....) và hiển thị các thông tin trong mảng đó lên.

Tuy nhiên, có nhiều trường hợp ta cần lấy thêm 1 số giá trị ngoài mảng để phục vụ việc tính toán, điều kiện hoặc hiển thị.

**Ví dụ chi tiết**. Hiển thị thông tin tỷ giá, loại tiền tệ thì ta cần thêm tiền tố // trước đường link lấy dữ liệu. Mục đích để xác định là lấy từ root chứ k lấy theo link trong for-each.

**So sánh cách lấy giá trị của các trường trong và ngoài mảng:**

* Đối với các trường của mảng (VD: itemName, quantity, unitPrice, ....) chỉ cần gọi trực tiếp đến tên trường. Vì link gốc đã gán vào for-each (<xsl:for-each select="invoiceData/items/item">)
* Đối với các trường ngoài mảng đang đứng thì cần thêm // vào trước đường link cụ thể (<xsl:value-of select="//invoiceData/currencyCode"/>)



12. Hiển thị thuế nhà thầu (xxx%)

Đối với các doanh nghiệp bình thường thì việc thuế suất GTGT được yêu cầu nhập trong các giá trị: Không kê khai tính thuế, không chịu thuế, 0%, 5%, 10%

Tuy nhiên một số doanh nghiệp có mã số thuế nhà thầu đang tính tắt bằng cách là thuế suất = 5%/ (1 - 5%) => 5.26316

Nhưng thực tế theo thông tư 103/2014/TT-BTC

Doanh thu tính thuế GTGT = Doanh thu chưa bao gồm thuế GTGT / (1 - Tỷ lệ % để tính thuế GTGT trên doanh thu)

<https://thuvienphapluat.vn/van-ban/Thuong-mai/Thong-tu-103-2014-TT-BTC-huong-dan-thuc-hien-nghia-vu-thue-to-chuc-ca-nhan-nuoc-ngoai-kinh-doanh-Viet-Nam-243595.aspx>

Để đáp ứng nhu cầu hiển thị thuế nhà thầu cho các doanh nghiệp này, hiện tại hệ thống đưa ra các xử lý để hiển thị như sau

Bước 1. Nhập liệu

* Đối với hóa đơn thuế tổng: Thêm trường động ghi chú (metadata/invoiceNote) nếu chưa có. Nhập giá trị XXX vào phần ghi chú, thuế suất vẫn nhập bình thường. TH thuế xuất nhập giá trị khác giá trị chuẩn thì sử dụng việc lập hóa đơn theo file để nhập liệu.
* Đối với hóa đơn thuế dòng. Sử dụng cột ghi chú của hóa đơn (itemNote) có sẵn để nhập liệu. Nhập giá trị XXX vào cột ghi chú của hàng hóa có thuế nhà thầu. Thuế suất nhập bình thường. TH thuế xuất nhập giá trị khác giá trị chuẩn thì sử dụng lập hóa đơn theo file để nhập liệu.

Bước 2: Chỉnh sửa mẫu để hiển thị

* Đối với mẫu hóa đơn thuế tổng (Tham khảo mẫu chuẩn 039):

invoie-fo.xsl và exchange-invoice-fo.xsl

|  |
| --- |
| <xsl:choose>  <xsl:when test="contains(//invoiceData/metadata/invoiceNote,'XXX')">  xxx  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="vatPercentage"/>  </xsl:otherwise>  </xsl:choose> % |

Invoice.xsl

|  |
| --- |
| <xsl:choose>  <xsl:when test="inv:vatPercentage != 'null' and inv:vatPercentage >= 0">  <xsl:choose>  <xsl:when test="contains(//inv:invoiceData/inv:metadata/inv:invoiceNote,'XXX')">  xxx  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="inv:vatPercentage"/>  </xsl:otherwise>  </xsl:choose> %  </xsl:when>  <xsl:otherwise>  .....\.....%  </xsl:otherwise>  </xsl:choose> |

* Đối với hóa đơn thuế dòng (Tham khảo mẫu chuẩn 214):

invoice-fo.xsl và exchange-invoice-fo.xsl

|  |
| --- |
| <xsl:if test="vatPercentage != 'null' and vatPercentage &gt;= 0">  <xsl:choose>  <xsl:when test="itemNote = 'XXX'">  xxx  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="foo:custom-num-format(vatPercentage, '###.##0', 'european')"/>  </xsl:otherwise>  </xsl:choose>  </xsl:if> |

invoice.xsl

|  |
| --- |
| <xsl:if test="inv:vatPercentage != 'null' and inv:vatPercentage >= 0">  <xsl:choose>  <xsl:when test="inv:itemNote = 'XXX'">  xxx  </xsl:when>  <xsl:otherwise>  <xsl:value-of select="format-number(inv:vatPercentage, '###.##0,#########')"/>  </xsl:otherwise>  </xsl:choose>  </xsl:if> |

13. Bỏ giờ trong phần ngày chuyển đổi

Chuẩn hệ thống khi chuyển đổi sẽ hiển thị cả thông tin ngày và giờ.

VD: 03/12/2019 15:33

Đối với những KH không muốn hiển thị giờ thì thực hiện sửa code như hướng dẫn bên dưới

Chỉ cần điều chỉnh tại file exchange-invoice-fo.xsl. Thông tin giờ sẽ được sinh vào phần tham số exchangeYear, do vậy, ta chỉ cần bỏ phần giờ trong tham số này là được.

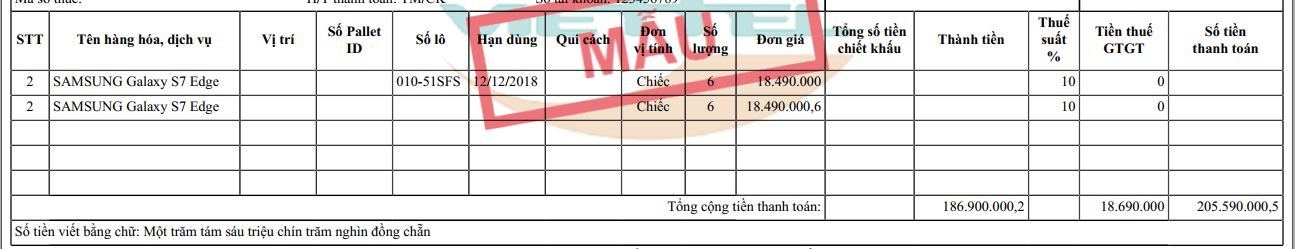
|  |
| --- |
| <xsl:variable name="year">    <xsl:value-of select="substring-before('${exchangeYear}', ' ')"/>  </xsl:variable>  <fo:block xsl:use-attribute-sets="labelNormal">${exchangeDay}/${exchangeMonth}/${year}</fo:block> |

14. Sử dụng các trường có sẵn của hàng hóa để nhập liệu

Hiện tại bảng hàng hóa có các trường chuẩn nhập liệu như phần 2.5 trong tài liệu. Tuy nhiên, một số khách hàng do đặc thù nghiệp vụ, cần nhiều hơn các số lượng cột cần nhập liệu. Lúc đó ta tận dụng các cột có sẵn, đưa các chuẩn nhập liệu riêng để có thể đáp ứng được (Tham khảo mẫu chuẩn 234)

Bước 1: Nhập liệu

VD yêu cầu của KH hiển thị thêm các giá trị: Vị trí, số Pallet, số lô, hạn dùng, quy cách



Trong khi đó mình chỉ có thể sử dụng được số lô (batchNo), hạn dùng (expDate), ghi chú (itemNote)

Do vậy, nếu theo chuẩn dữ liệu thì chỉ đáp ứng được 2 cột Số lô, hạn dùng có trường nhập liệu.

Còn cột: Vị trí, số Pallet, Qui cách ta cần tìm cách để có thể nhập liệu dựa vào các trường chuẩn. Ở đây ta sử dụng trường ghi chú hàng hóa (itemNote) để nhập liệu theo 1 chuẩn tự đưa ra (Tùy từng người sẽ có 1 chuẩn riêng hoặc thảo luận với khách hàng để đưa ra 1 cách nhập liệu đơn giản nhất mà có thể sử dụng các hàm để lấy dữ liệu hiển thị lên sau khi nhập liệu theo chuẩn đã đưa ra) như sau.

VD: Đưa ra chuẩn nhập liệu cho trường ghi chú hàng hóa: Trường vị trí sẽ nhập liệu là VT=giá trị dữ liệu vị trí|ID=giá trị dữ liệu số pallet ID|QC=giá trị dữ liệu qui cách.

Dữ liệu sau khi nhập đúng chuẩn đưa ra như sau: VT=Vị trí 1|ID=123|QC=Đơn lẻ

Bước 2: Chỉnh sửa mẫu để hiển thị

Sau khi thống nhất được chuẩn nhập liệu và có dữ liệu như yêu cầu, ta thực hiện sửa code để lấy đúng giá trị cho từng cột lên mẫu hóa đơn.

* Khai báo 1 template để thực hiện lấy giá trị theo tham số VT, ID, QC như sau

|  |
| --- |
| <xsl:template name="tokenize">  <xsl:param name="pText"/>  <xsl:param name="subText"/>  <xsl:param name="sText"/>  <xsl:if test="string-length($pText)">  <xsl:choose>  <xsl:when test="contains($pText,$subText)">  <xsl:variable name="text">  <xsl:value-of select="substring-before($pText, $subText)"/>  </xsl:variable>  <xsl:choose>  <xsl:when test="substring($text,1,3) = $sText">  <xsl:value-of select="substring-after($text, $sText)"/>  </xsl:when>  <xsl:otherwise>  <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select= "substring-after($pText, $subText)"/>  <xsl:with-param name="subText" select="$subText"/>  <xsl:with-param name="sText" select="$sText"/>  </xsl:call-template>  </xsl:otherwise>  </xsl:choose>  </xsl:when>  <xsl:otherwise>  <xsl:if test="substring($pText,1,3) = $sText">  <xsl:value-of select="substring-after($pText, $sText)"/>  </xsl:if>  </xsl:otherwise>  </xsl:choose>  </xsl:if>  </xsl:template> |

* Lấy dữ liệu cho 1 cột cụ thể

File invoice-fo.xsl và exchange-invoice-fo.xsl

|  |
| --- |
| <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="itemNote"/>  <xsl:with-param name="subText" select="'|'"/>  <xsl:with-param name="sText" select="'VT='"/>  </xsl:call-template> |

File invoice.xsl

|  |
| --- |
| <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="inv:itemNote"/>  <xsl:with-param name="subText" select="'|'"/>  <xsl:with-param name="sText" select="'VT='"/>  </xsl:call-template> |

Lấy giá trị chuẩn nào thì nhập liệu vào tham số đó

pText: Nguồn lấy dữ liệu, như thống nhất sẽ lấy dữ liệu từ itemNote

subText:Ký tự chia dữ liệu, như thống nhất sẽ lấy ký tự | để phân cách các dữ liệu nhập liệu

sText: Giá trị để xác định cột nào, như thống nhất VT= là vị trí, ID= là giá trị pallet, QC= là giá trị quy cách

Như vậy tùy vào cột Vị trí, Số Pallet ID, Quy cách mà ta sẽ truyền tham số cho đúng theo từng cột

**Chú ý**: Do bên trên mình đang sửa chung vào template lấy thông tin chữ ký từ chứng thư số, nên mình phải sửa lại cả phần lấy thông tin chứng thư số để tránh bị sai sót. Nếu không muốn động chạm đến dữ liệu đang có, có thể khai báo template mới với 1 tên khác với những tên đã có và sử dụng bình thường.

|  |
| --- |
| <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="(//\*[local-name()='X509SubjectName'])[1]"/>  <xsl:with-param name="subText" select="','"/>  <xsl:with-param name="sText" select="'CN='"/>  </xsl:call-template> |

15. Sai thông tin Ký bởi (Không lấy được thông tin hoặc lấy thông tin không đầy đủ)

a. Trường hợp Không lấy được thông tin Ký bởi

- Lý do: Khai báo template lấy theo ký số bị sai, chưa theo template chuẩn

- Cách xử lý: Kiểm tra lại template khai báo cách lấy dữ liệu theo CTS. Nếu template khai báo không phải template chuẩn như mục 3.2 phần g và 3.3 phần g

|  |
| --- |
| Khai báo template  <xsl:template name="tokenize">  <xsl:param name="pText"/>  <xsl:if test="string-length($pText)">  <xsl:choose>  <xsl:when test="contains($pText,',')">  <xsl:variable name="text">  <xsl:value-of select="substring-before($pText, ',')"/>  </xsl:variable>  <xsl:choose>  <xsl:when test="substring($text,1,3) = 'CN='">  <xsl:value-of select="substring-after($text, 'CN=')"/>  </xsl:when>  <xsl:otherwise>  <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select=  "substring-after($pText, ',')"/>  </xsl:call-template>  </xsl:otherwise>  </xsl:choose>  </xsl:when>  <xsl:otherwise>  <xsl:if test="substring($pText,1,3) = 'CN='">  <xsl:value-of select="substring-after($pText, 'CN=')"/>  </xsl:if>  </xsl:otherwise>  </xsl:choose>  </xsl:if>  </xsl:template> |

b. Trường hợp lấy thông tin không đầy đủ

- Lý do: Trong thông tin Tên CTS có dứa dấu ,

- Cách xử lý: Chuyển phần thông tin Ký bởi lấy theo template lấy thông tin CTS sang lấy theo phần thông tin sellerInfo

|  |  |
| --- | --- |
| Ký bởi theo CTS | Ký bởi theo sellerInfo |
| <fo:block xsl:use-attribute-sets="signature">  Ký bởi <xsl:call-template name="tokenize">  <xsl:with-param name="pText" select="(//\*[local-name()='X509SubjectName'])[1]"/> </xsl:call-template>  </fo:block> | <fo:block xsl:use-attribute-sets="signature">  Ký bởi <xsl:value-of select="invoiceData/seller/sellerLegalName"/> </fo:block> |

16. Sử dụng các loại hàng hóa để tận dụng nhập liệu lên hóa đơn

Đối với các loại hàng hóa, hiện tại hệ thống cho phép nhập các loại sau:

* Null hoặc 1- Hàng Hóa (Bắt buộc phải nhập số lượng, đơn giá)
* 2: Ghi chú (Không sinh STT và không cộng tiền vào tổng tiền thanh toán)
* 3: Chiết khấu (Không cần nhập số lượng, đơn giá và thêm isIncreaseItem = false để xác định giảm tiền)
* 4: Bảng kê (Không cần nhập số lượng, đơn giá, chỉ cần nhập thành tiền)
* 5: Phí khác (Bắt buộc nhập số lượng, đơn giá)

Ta có thể sử dụng các trường hợp này cho các Khách hàng đặc thù để việc nhập liệu và hiển thị lên hóa đơn được chính xác.

1. Ghi chú

* Các lấy dữ liệu các hàng hóa dạng ghi chú:

|  |
| --- |
| <xsl:for-each select="invoiceData/items/item[selection = 2]">  Hiển thị tùy ý các giá trị của hàng hóa dạng ghi chú ở đây  </xsl:for-each> |

* Đối với các dòng hàng hóa chỉ mang tính chất làm rõ thông tin, không có tác dụng cho việc tính toán vào thành tiền của hóa đơn ta sẽ sử dụng loại ghi chú.

Phần hiển thị ghi chú do không thực hiện vào việc tính toán tiền nên không cần phải sử lý gì nhiều về mặt tính toán trên mẫu hóa đơn.

1. Phí khác cho KH đặc thù (Khách sạn, nhà hàng, máy bay, thu phí dịch vụ ....)

* Cách lấy dữ liệu hàng hóa dạng Phí khác:

|  |
| --- |
| <xsl:for-each select="invoiceData/items/item[selection = 5]">  Hiển thị tùy ý các giá trị của hàng hóa dạng ghi chú ở đây  </xsl:for-each> |

* Đối với phí khác trước thuế: Yêu cầu bắt buộc sử dụng mẫu thuế Tổng

Tham khảo 1 số mẫu hóa đơn chuẩn sau: 01GTKT0\_113, 01GTKT0\_140, 01GTKT0\_143, 01GTKT0\_183, 01GTKT0\_226, 01GTKT0\_233, 01GTKT0\_246, 01GTKT0\_247, 01GTKT0\_262

Hướng dẫn xử lý mẫu hóa đơn theo link hướng dẫn: <https://www.youtube.com/watch?v=8vrULYR48Rw>

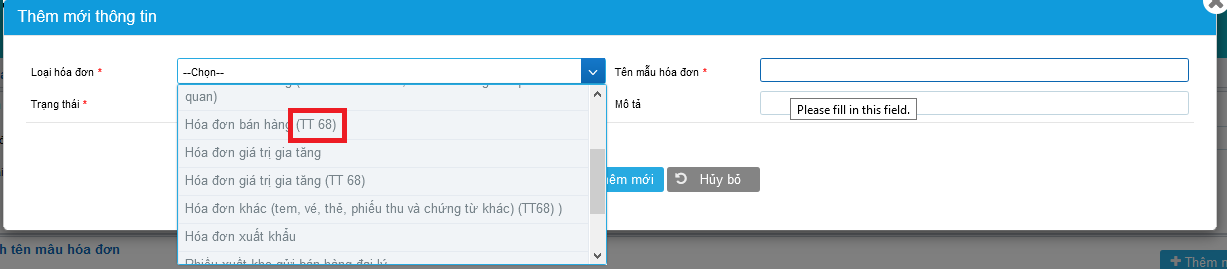
* Đối với phí khác sau thuế: Yêu cầu bắt buộc sử dụng mẫu thuế Dòng

Tham khảo 1 số mẫu hóa đơn sau: 01GTKT0\_107, 01GTKT0\_130, 01GTKT0\_148, 01GTKT0\_164, 01GTKT0\_196, 01GTKT0\_230, 01GTKT0\_231, 01GTKT0\_245

Hướng dẫn xử lý mẫu hóa đơn theo link hướng dẫn: <https://www.youtube.com/watch?v=SXtHl91fN6Y&t=332s>

17. Mẫu hóa đơn TT68

* Để sử dụng được mẫu hóa đơn theo TT68, yêu cầu bắt buộc đầu tiên phải chọn mẫu chuẩn theo TT68 trước (Khai báo tên mẫu hóa đơn chọn loại TT68, tương ứng sẽ chọn được mẫu chuẩn TT68). Để hệ thống tự động sinh dữ liệu theo chuẩn của thuế.



* Về mẫu hóa đơn thì có thể tùy chỉnh theo yêu cầu của KH tuy nhiên vẫn phải đảm bảo đúng yêu cầu theo TT68: Không hiển thị mẫu hóa đơn, Ký hiệu hóa hiển thị bao gồm ký hiệu mẫu hóa đơn và ký hiệu hóa đơn.
* So sánh giữa TT32 và TT68

Có thể sử dụng mẫu của TT32 cho TT68 (Với điều kiện đã chọn mẫu chuẩn theo TT68) và sửa lại 1 số thông tin theo chuẩn TT68 như sau (Tham khảo các hóa đơn chuẩn TT68\_1, TT68\_2, TT68\_3, TT68\_4)

|  |  |  |
| --- | --- | --- |
| Tiêu chí | TT32 | TT68 |
| Mẫu số | Yêu cầu bắt buộc hiển thị  <xsl:value-of select="invoiceData/templateCode"/> | Không hiển thị |
| Ký hiệu | Lấy theo đúng ký hiệu đăng ký  <fo:block xsl:use-attribute-sets="itemBold">  <xsl:value-of select="invoiceData/invoiceSeries"/>  </fo:block> | Lấy 2 giá trị Loại hóa đơn và ký hiệu hóa đơn ghép lại để hiển thị  <fo:block xsl:use-attribute-sets="itemBold">  <xsl:value-of select="invoiceData/invoiceType"/>  <xsl:value-of select="invoiceData/invoiceSeries"/>  </fo:block> |

Các giá trị còn lại vẫn lấy bình thường như TT32

18. Mở rộng phần xuống dòng khi dữ liệu quá dài

* 1. Đối với dữ liệu không có khoảng trắng (Dữ liệu viết liền mạch)

Khai báo 1 template

|  |
| --- |
| <xsl:template name="zero\_width\_space\_1">  <xsl:param name="data"/>  <xsl:param name="position"/>  <xsl:param name="counter"/>  <xsl:param name="replace"/>  <xsl:choose>  <xsl:when test="$position &lt;= string-length($data)">  <xsl:value-of select='concat(substring($data,$position,$counter),$replace)'/>  <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="$data"/>  <xsl:with-param name="position" select="$position + $counter"/>  <xsl:with-param name="counter" select="$counter"/>  <xsl:with-param name="replace" select="$replace"/>  </xsl:call-template>  </xsl:when>  <xsl:otherwise>  </xsl:otherwise>  </xsl:choose>  </xsl:template> |

Trong đó:

Data: Chuỗi dữ liệu truyền vào

Position: Vị trí bắt đầu – Thường để = 0

Counter: Số lượng ký tự, tùy từng mong muốn số ký tự để chèn ký tự thay thế

Replace: Ký tự thay thế

Gọi dữ liệu

|  |
| --- |
| <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="expDate"/>  <xsl:with-param name="position" select="0"/>  <xsl:with-param name="counter" select="15"/>  <xsl:with-param name="replace" select="'&#8203;'"/>  </xsl:call-template> |

* 1. Đối với dữ liệu có khoảng trắng

Khai báo 2 template

|  |
| --- |
| <xsl:template name="zero\_width\_space\_1">  <xsl:param name="data"/>  <xsl:param name="position"/>  <xsl:param name="counter"/>  <xsl:param name="replace"/>  <xsl:choose>  <xsl:when test="$position &lt;= string-length($data)">  <xsl:value-of select='concat(substring($data,$position,$counter),$replace)'/>  <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="$data"/>  <xsl:with-param name="position" select="$position + $counter"/>  <xsl:with-param name="counter" select="$counter"/>  <xsl:with-param name="replace" select="$replace"/>  </xsl:call-template>  </xsl:when>  <xsl:otherwise>  </xsl:otherwise>  </xsl:choose>  </xsl:template>  <xsl:template name="enter\_text">  <xsl:param name="text"/>  <xsl:param name="replace"/>  <xsl:choose>  <xsl:when test="contains($text,' ')">  <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="substring-before($text,' ')"/>  <xsl:with-param name="position" select="0"/>  <xsl:with-param name="counter" select="20"/>  <xsl:with-param name="replace" select="$replace"/>  </xsl:call-template>  <xsl:value-of select="' '"/>  <xsl:call-template name="enter\_text">  <xsl:with-param name="text" select="substring-after($text,' ')"/>  <xsl:with-param name="replace" select="$replace"/>  </xsl:call-template>  </xsl:when>  <xsl:otherwise>  <xsl:call-template name="zero\_width\_space\_1">  <xsl:with-param name="data" select="$text"/>  <xsl:with-param name="position" select="0"/>  <xsl:with-param name="counter" select="20"/>  <xsl:with-param name="replace" select="$replace"/>  </xsl:call-template>  </xsl:otherwise>  </xsl:choose>  </xsl:template> |

Trong đó:

Template zero\_width\_space\_1 tương tự như phần không có khoảng trắng

Data: Chuỗi dữ liệu truyền vào

Position: Vị trí bắt đầu – Thường để = 0

Counter: Số lượng ký tự, tùy từng mong muốn số ký tự để chèn ký tự thay thế

Replace: Ký tự thay thế

Template enter\_text để xử lý thêm phần có khoảng trắng để không replace nhầm

Gọi dữ liệu

|  |
| --- |
| <xsl:call-template name="enter\_text">  <xsl:with-param name="text" select="expDate"/>  <xsl:with-param name="replace" select="'&#8203;'"/>  </xsl:call-template> |

VI. KẾT THÚC