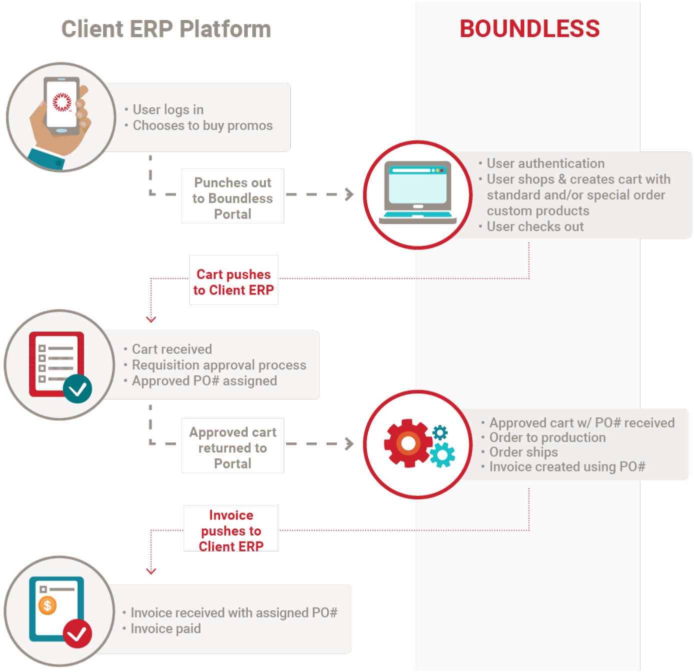
# PUNCHOUT WITH BOUNDLESS

## What is Punchout?

In general, a Punchout catalog (more accurately called a Punchout webstore) is a method for an organization's approved buyers to make purchases from a supplier's ecommerce platform from within the organization's procurement application or hosted e-procurement system – often called an Enterprise Resource Planning platform (ERP).

More specifically, at Boundless, a Punchout enabled Portal Store is a custom ecommerce webstore with the special ability to communicate directly with a procurement system via cXML (which is a commerce eXtensible Markup Language and a protocol for data exchange between applications) and return a pending purchase order back to Boundless without the buyer manually entering product information in the procurement system. A Punchout may also include the ability for the order's final invoice to be automatically sent back to the procurement system and be matched against the purchase order without the need for manual entry.

The overall goal is for an organization's users to be able to sign into their procurement system to automatically authenticate into their custom Boundless Portal Store to shop, add custom branded merchandise, promotional products, and/or print materials to a shopping cart and seamlessly send that cart back to your ERP to requisition an approved purchase order for Boundless to use to fulfill and invoice the order.


Client ERP Platform
BOUNDLESS
• User logs in
• Chooses to buy promos
Punches out to Boundless Portal
• User authentication
• User shops & creates cart with standard and/or special order custom products
• User checks out
Cart pushes to Client ERP
• Cart received
• Requisition approval process
• Approved PO# assigned
Approved cart returned to Portal
• Approved cart w/ PO# received
• Order to production
• Order ships
• Invoice created using PO#
Invoice pushes to Client ERP
• Invoice received with assigned PO#
• Invoice paid

## Why is Punchout with Boundless good for your organization?

A Punchout allows the approval process to reside where it belongs, and where it is already set up and maintained – within your procurement system. By utilizing a Punchout integration, your organization does not have to maintain any catalogs for products available on the Portal Store. Boundless maintains the account specific products, pricing, and feature sets so users can have access to everything the Boundless Portal platform has to offer while still maintaining the appropriate approval and purchase order requisition processes set forth in the procurement system.

### Highlights:

- Portal Store branded and customized specifically to your organization.
- Single Sign On into the Portal Store alleviating the need for additional login by the user.
- Cart moves seamlessly between the procurement system and the Boundless platforms.
- Quick and easy updates to product offerings without requiring client resources.
- Boundless can control the user experience in the Portal Store.
- Buyer access to real time order tracking, reporting, and historical data.
- Invoicing can be built into the automated process using a cXML integration, if desired.

## Setting up your Boundless Portal Store

Before anything else can happen, your dedicated Boundless account management team will work with your marketing/brand teams to merchandise your program and negotiate the pricing for your store. Depending on the number of products and their specializations, this could take 1-4 weeks.

As soon as a small quantity (at least 10) have been sourced and configured in your Portal Store, the Boundless integration team, comprised of senior level software engineers, can pull that information down to our “preview” server. Our preview server is a test/staging/set up server that hosts a copy of your Boundless Portal store. Preview is used to test connectivity and integrations with third-party platforms like Procurement systems.

The preview/test environment differs from the production/live environment in that you can add $100K worth of products to your cart and “purchase” it all, but you will never be charged a dime. You can do this 100 times. Your test environment speaks to our test environment and we can mess around a LOT with no repercussions.

Once your Portal Store has been populated in preview, we can enable it for Punchout. This means we tell Portal to get ready to receive messages from your procurement system. But first we’ll need to ask you a few questions.

## Buyer Questionnaire

### Users

In your procurement system, your I/T team will maintain a list of users who are allowed to Punchout, they'll set user permissions and maintain an access control list (ACL). In many buyer catalogs, users are allowed into the system as long as certain criteria are met.

Our Portal is a *little* different. We've written a plugin called Tyson (get it? Punchout?) that sits on top of our e-commerce technology. A plugin is basically a mini-app that sits on top of a much larger app and allows it to perform extra functions. The extra functions that Tyson provides are a way to receive and to respond to messages (requests) from various procurement system platforms.

In order for your procurement system to allow users to Punchout to our platform, we need to know in advance who those users are so that we can locate them in our user directory and log them in.

For the above authentication requirement, we provide two options:

1. Send us an export of your access control list and we can create the users in advance (The user fields we would need would be: firstName, lastName, email, phoneNumber). We can do this on a schedule: quarterly, weekly, or nightly. ***Pros***: you know EXACTLY who is on this list and if they're not on the list, they can pretty much never Punchout. ***Cons***: if you add someone in your PROCUREMENT SYSTEM ACL, they won't be allowed to Punchout in Portal until the user list has been imported on our side. They'll get a User Unauthorized 401 exception.

**OR**

2. Allow us to dynamically create the contacts as they Punchout into our system (if they don't already exist in our user directory). This is the option that most buyer organizations choose. ***Pros***: no-one ever gets a User Unauthorized 401 exception and you never ever have to send us a user list. ***Cons***: Sometimes we only have limited information to create the user so, for instance, someone's first name may be stored as an email address until the SPT goes in and moderates the user. But that's a pretty small con, in general.

### Invoicing

Will your invoicing be via cXML? For every PO that you send back to Portal, we can send back an invoice in cXML format that your procurement system can then receive and push into your accounting system. This can be done on an automated nightly basis and we can also issue Credit Memos via cXML using the same feature.

If the invoicing is NOT via cXML, which accounting nodes – if any – are the ones which should be used for reporting or invoicing in the Punchout Order Request (PO) cXML?

For instance, here are a couple of nodes for items that came over in a Punchout Order Request from different buyer organizations, and highlighted are the values (accounting nodes) that they considered most important for reporting purposes:

```
<ItemOut isAdHoc=”yes” quantity=”981” lineNumber=”36”>
 <ItemID>
  <SupplierPartID>SVKNAT19050-SU</SupplierPartID>
 </ItemID>
 <ItemDetail>
  <UnitPrice>
   <Money alternateAmount=”” currency=”USD” alternateCurrency=””>107</Money>
  </UnitPrice>
  <Description xml:lang=”en”>SVK HAIL MARY GOAL POST MASS</Description>
  <UnitOfMeasure>PK</UnitOfMeasure>
  <Classification domain=”custom”>P12479412956</Classification>
  <LeadTime>0</LeadTime>
  <Extrinsic name=”Req. Line No.”>36</Extrinsic>
  <Extrinsic name=”Requester”>Michael Little</Extrinsic>
  <Extrinsic name=”PR No.”>PR7208</Extrinsic>
 </ItemDetail>
</ItemOut>
```

And

```
<ItemOut quantity=”3” lineNumber=”1”>
 <ItemID>
  <SupplierPartID>3599</SupplierPartID>
  <SupplierPartAuxiliaryID>so:69018-soi:158775-sp:3599</SupplierPartAuxiliaryID>
 </ItemID>
 <ItemDetail>
  <UnitPrice>
   <Money currency=”USD”>16.91</Money>
  </UnitPrice>
  <Description xml:lang=”en”>PI-406 Black Golf UmbrellaDescription>
  <UnitOfMeasure>EA</UnitOfMeasure>
  <Classification domain=”UNSPROCUREMENT SYSTEMC”>unknown</Classification>
  <Extrinsic name=”hmis_contract_num” />
  <Extrinsic name=”sci_supplier_order_number” />
  <Extrinsic name=”supplierpartauxiliaryid” />
  <Extrinsic name=”supplier_production_code” />
  <Extrinsic name=”sales_item_id” />
 </ItemDetail>
 <Distribution>
  <Accounting name=”Local Advertising”>
   <Segment description=”Location” id=”48007” type=”Location” />
   <Segment description=”Account” id=”63010000” type=”Account” />
   <Segment description=”Dept” id=”2926” type=”Dept” />
  </Accounting>
  <Charge>
   <Money currency=”USD”>50.73</Money>
  </Charge>
 </Distribution>
</ItemOut>
```

## UNSPSC Codes

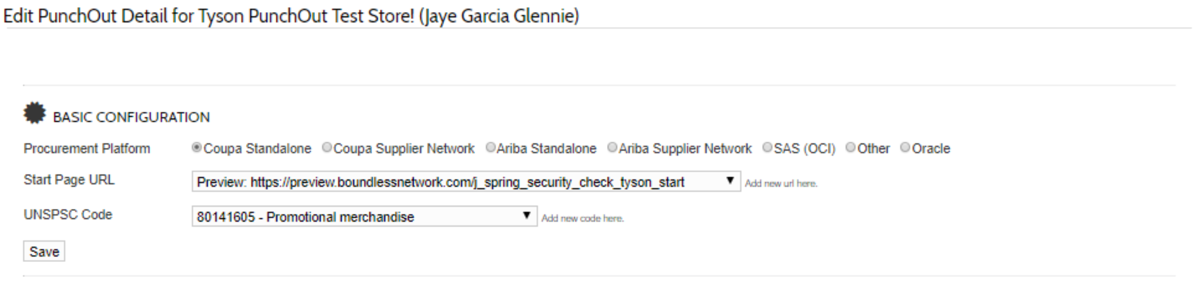
The United Nations Standard Products and Services Code (UNSPSC) is a taxonomy of products and services for use in eCommerce and is usually a required code in standard procurement systems. We have the most frequently used UNSCP codes already set up in our system for use, but can add additional codes if your procurement system required them. Please review the list below and let us know if you will you need additional UNSPSC codes so we may set them up. (NOTE: 80141605 – Promotional merchandise – is the most commonly used, and is thus our default UNSPSC code for all of our store products)

| | |
|---|---|
| 46181500 | Safety apparel |
| 53100000 | Clothing |
| 53103000 | T-shirts |
| 55101528 | Religious books |
| 80141605 | Promotional merchandise |
| 80141626 | Promotional program management service |
| 80171702 | Brand promotion and management service |
| 82121505 | Promotional or advertising printing |
| 82121507 | Stationery or business form printing |
| 93141811 | Promotional services |

## Portal Configuration

Once all of these questions are answered, the integration team will get the go-ahead to enable your Portal Store for Punchout. Our config screens look like this:

### Basic Configuration



Edit PunchOut Detail for Tyson PunchOut Test Store! (Jaye Garcia Glennie)

BASIC CONFIGURATION

Procurement Platform ● Coupa Standalone ○ Coupa Supplier Network ○ Ariba Standalone ○ Ariba Supplier Network ○ SAS (OCI) ○ Other ○ Oracle

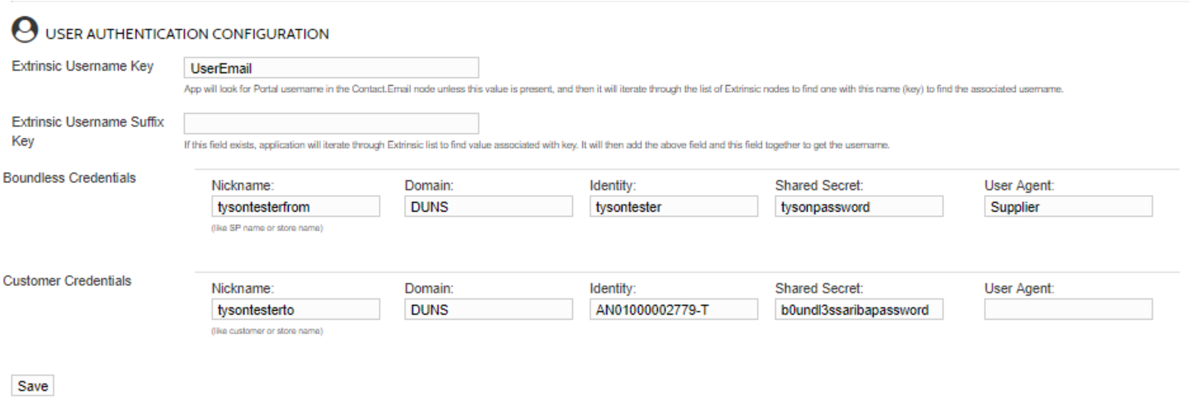
Start Page URL: Preview: https://preview.boundlessnetwork.com/j_spring_security_check_tyson_start ▼ Add new url here.

UNSPSC Code: 80141605 - Promotional merchandise ▼ Add new code here.

Save

This will tell Portal which procurement system will be communicating with this store, which environment information to send (preview/testing or production/live), and which UNSPSC code to apply to all of the store products by default, although your dedicated account management team can set individual store products to have a different UNSPSC code if needed.

## User Authentication Configuration



USER AUTHENTICATION CONFIGURATION

Extrinsic Username Key: UserEmail

App will look for Portal username in the Contact.Email node unless this value is present, and then it will iterate through the list of Extrinsic nodes to find one with this name (key) to find the associated username.

Extrinsic Username Suffix Key:

If this field exists, application will iterate through Extrinsic list to find value associated with key. It will then add the above field and this field together to get the username.

Boundless Credentials

| Nickname: | Domain: | Identity: | Shared Secret: | User Agent: |
|---|---|---|---|---|
| tysontesterfrom<br>(like SP name or store name) | DUNS | tysontester | tysonpassword | Supplier |

Customer Credentials

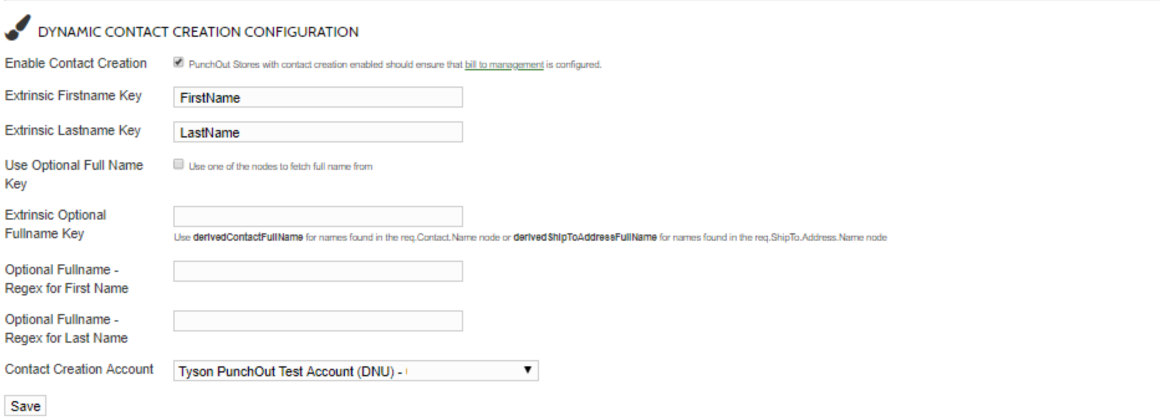
| Nickname: | Domain: | Identity: | Shared Secret: | User Agent: |
|---|---|---|---|---|
| tysontesterto<br>(like customer or store name) | DUNS | AN01000002779-T | b0undl3ssaribapassword | |

This is the BIG KAHUNA configuration set. This tells Portal what secret information your procurement system will be sending to us that will let in your users. In this particular configuration, you're the customer tysontester. When you send information to Portal for Punchout Setup Requests (the messages that you send to Portal to say, "Log me in, and let me in") and for Purchase Order Requests (the messages that you send to Portal to say, "Hey! Here's that order you wanted, and there's a PO in the basket now, fulfill my order!"), you also send over an identify and a shared secret – a passcode that we both agree on – so that we know it's really you. The Boundless Credentials feature when we send over messages to your procurement system.  Mainly when we send over invoices via cXML.

The username keys tell us where in the Punchout Setup Request (the login message) to find the username of the buyer trying to Punchout. So, if this is a POSR, highlighted is where this store's buyer usernames will be located.

```
<cXML payloadID=”1517.6951661@app514.snv.ariba.com” xml:lang=”en” version=”1.2.034”
timestamp=”2018-02-01T06:18:33-08:00”>
 <Header>
  <From>
   <Credential domain=”NetworkId”>
    <Identity>AN01000002779-T-T</Identity>
   </Credential>
  </From>
  <To>
   <Credential domain=”transactionnetworkid”>
    <Identity>an01003151181-t</Identity>
   </Credential>
  </To>
  <Sender>
   <Credential domain=”AribaNetworkUserId”>
    <Identity>sysadmin@ariba.com</Identity>
    <SharedSecret>b0undl3ssaribapassword</SharedSecret>
   </Credential>
   <UserAgent>Buyer 14s2</UserAgent>
  </Sender>
 </Header>
 <Request deploymentMode=”test”>
  <PunchoutSetupRequest operation=”create”>
   <BuyerCookie>1P9RDG4ARIUED</BuyerCookie>
   <Extrinsic name=”UserEmail”>tysontest@bn.com</Extrinsic>
   <Extrinsic name=”UniqueName”>tysontester87364</Extrinsic>
   <Extrinsic name=”FirstName”>Tyson</Extrinsic>
   <Extrinsic name=”LastName”>Tester</Extrinsic>
   <BrowserFormPost>
    <URL>httprocurement system://s1-2.ariba.com/Buyer/Punchout?client=HTML.237 </URL>
   </BrowserFormPost>
  </PunchoutSetupRequest>
 </Request>
</cXML>
```

## Dynamic Contact Creation



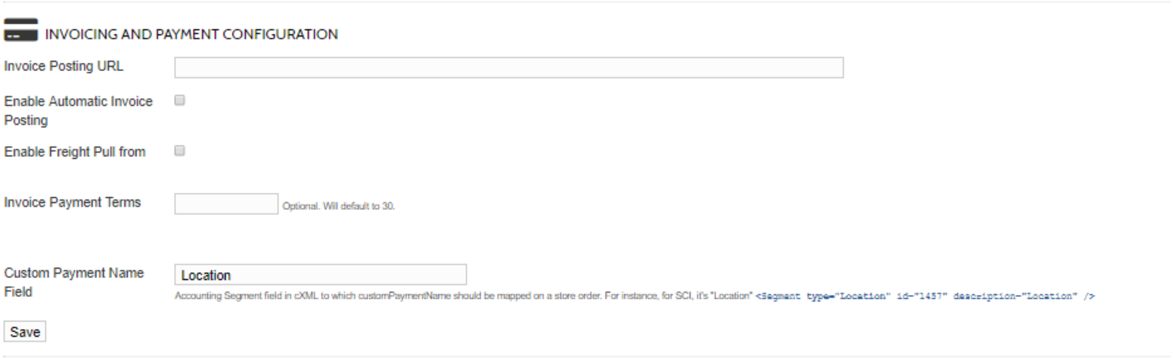
DYNAMIC CONTACT CREATION CONFIGURATION

| | |
|---|---|
| Enable Contact Creation | ☑ PunchOut Stores with contact creation enabled should ensure that bill to management is configured. |
| Extrinsic Firstname Key | FirstName |
| Extrinsic Lastname Key | LastName |
| Use Optional Full Name Key | □ Use one of the nodes to fetch full name from |
| Extrinsic Optional Fullname Key | Use **derivedContactFullName** for names found in the req.Contact.Name node or **derivedShipToAddressFullName** for names found in the req.ShipTo.Address.Name node |
| Optional Fullname - Regex for First Name | |
| Optional Fullname - Regex for Last Name | |
| Contact Creation Account | Tyson PunchOut Test Account (DNU) - ▼ |

Do you remember when we addressed how users can get created automatically? Here’s the configuration set that takes care of that piece. In the cXML code above, you can see that not only is there an Extrinsic node that holds UserEmail, but also there are nodes that hold FirstName and LastName. We use those to create the user on our end.

There are a lot of different ways procurement system platforms can send over user information, so we have lots of ways we can capture it. Our integration team can talk about Regexes if you want, but you really don’t want, trust us.

## Invoicing



INVOICING AND PAYMENT CONFIGURATION

| | |
|---|---|
| Invoice Posting URL | |
| Enable Automatic Invoice Posting | □ |
| Enable Freight Pull from | □ |
| Invoice Payment Terms | Optional. Will default to 30. |
| Custom Payment Name Field | Location<br>Accounting Segment field in cXML to which customPaymentName should be mapped on a store order. For instance, for SCI, it's "Location" <Segment type="Location" id="1457" description="Location" /> |

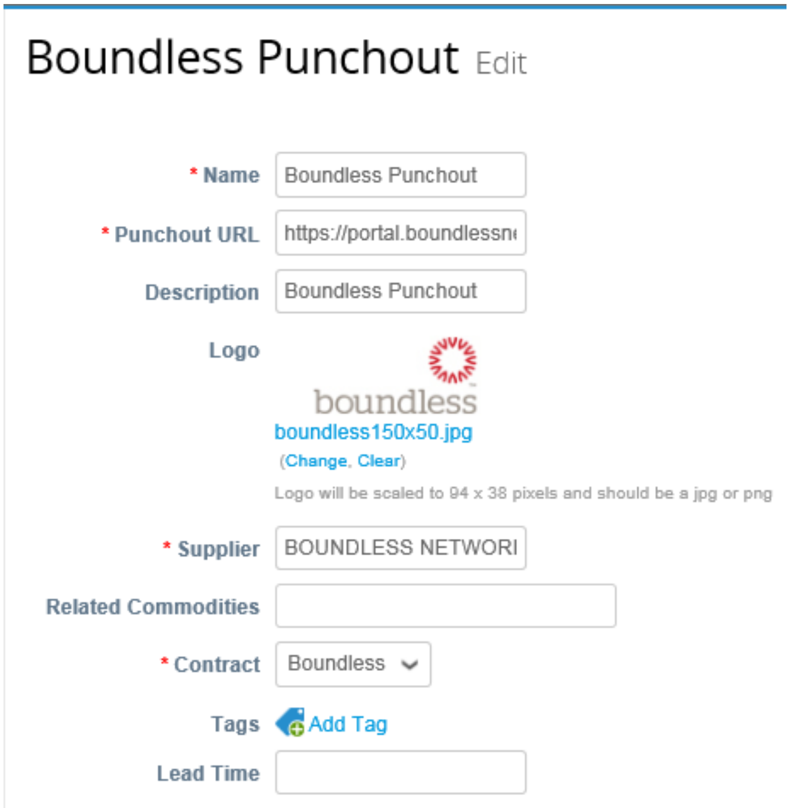
Some clients would prefer that we send over invoices via cXML and here’s where we’d set up that configuration.

Now let's look at how these configuration screens MIGHT look in YOUR procurement system.

## Your Procurement System Setup

So, here are the sibling screens you and your IT team will probably see that relate to the Boundless Punchout configuration screens.

### Adding Boundless as a Supplier



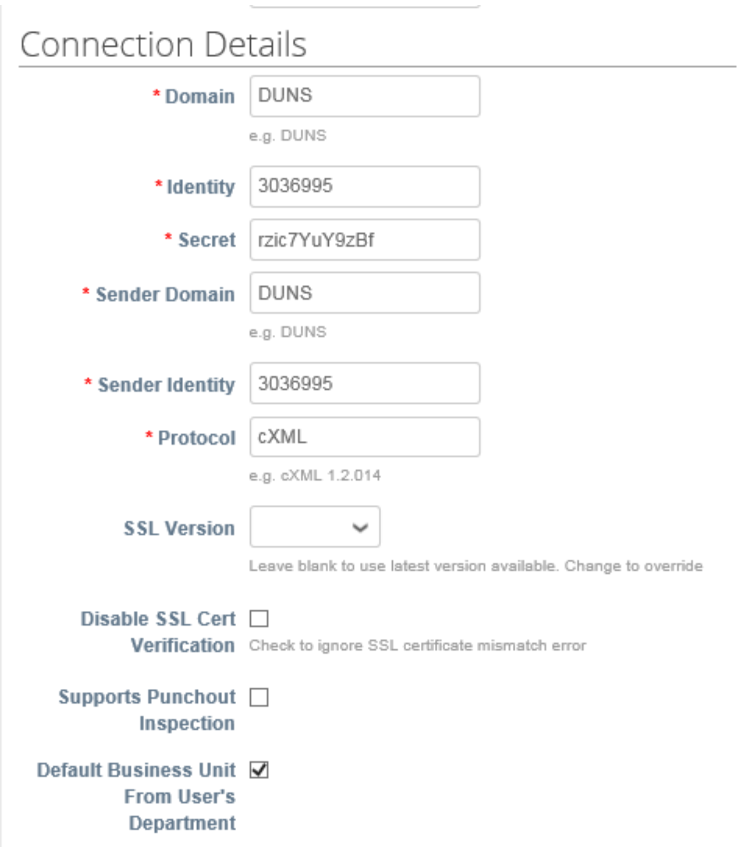
Boundless Punchout Edit

| | |
|---|---|
| * Name | Boundless Punchout |
| * Punchout URL | https://portal.boundlessn |
| Description | Boundless Punchout |
| Logo | boundless<br>boundless150x50.jpg<br>(Change, Clear)<br>Logo will be scaled to 94 x 38 pixels and should be a jpg or png |
| * Supplier | BOUNDLESS NETWORI |
| Related Commodities | |
| * Contract | Boundless |
| Tags | Add Tag |
| Lead Time | |

We will send you a logo to upload and the Punchout URL to which login requests can be sent.

## Punchout Setup Request

If you look closely, you can see that this looks a lot like the Boundless ***User Authentication Configuration*** screens. The values above should match the values that Boundless enters on our end.



Connection Details

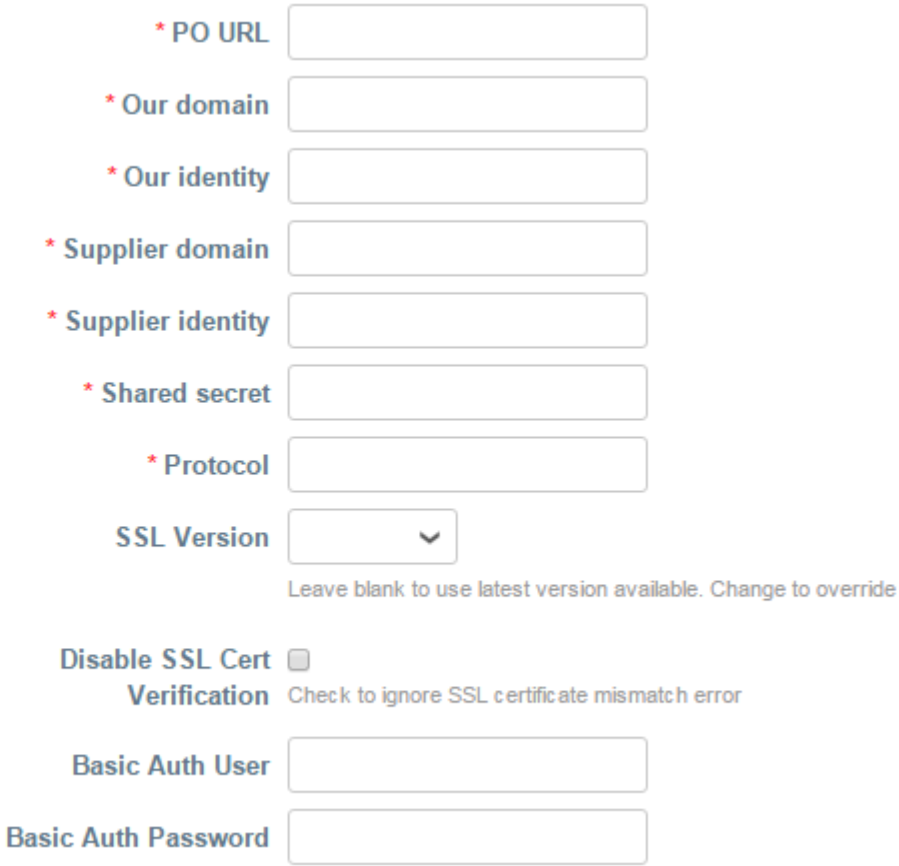
| | |
|---|---|
| * Domain | DUNS<br>e.g. DUNS |
| * Identity | 3036995 |
| * Secret | rzic7YuY9zBf |
| * Sender Domain | DUNS<br>e.g. DUNS |
| * Sender Identity | 3036995 |
| * Protocol | cXML<br>e.g. cXML 1.2.014 |
| SSL Version | <br>Leave blank to use latest version available. Change to override |
| Disable SSL Cert Verification | □ Check to ignore SSL certificate mismatch error |
| Supports Punchout Inspection | □ |
| Default Business Unit From User's Department | ☑ |

## PO Configuration

To setup how POs get routed to Portal from your procurement system also mirror the ***User Authentication Configuration*** screens.

Although in principle you could have different shared secrets and identities for this setup, we re-use the same one for each message type.

Boundless will send you PO URLs for both testing and production.



* PO URL

* Our domain

* Our identity

* Supplier domain

* Supplier identity

* Shared secret

* Protocol

SSL Version

Leave blank to use latest version available. Change to override

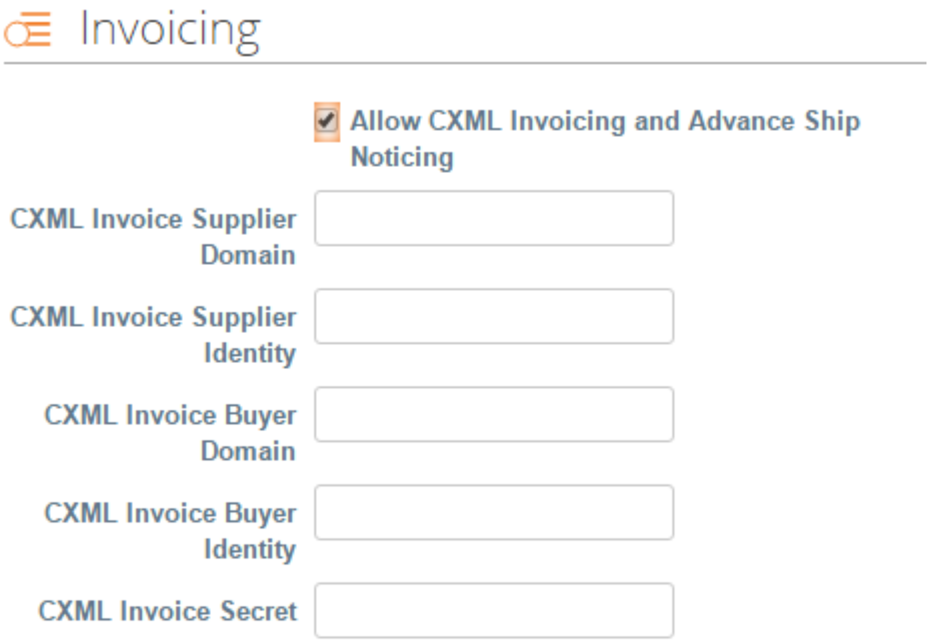
Disable SSL Cert Verification □
Check to ignore SSL certificate mismatch error

Basic Auth User

Basic Auth Password

## Invoicing

If you choose to have invoices sent via cXML from the Boundless Portal back to your procurement system, you'd use screens like this to tell your procurement system which secret code Portal will be using in order to access your system.



Invoicing

☑ Allow CXML Invoicing and Advance Ship Noticing

| | |
|---|---|
| CXML Invoice Supplier Domain | |
| CXML Invoice Supplier Identity | |
| CXML Invoice Buyer Domain | |
| CXML Invoice Buyer Identity | |
| CXML Invoice Secret | |

# What happens in testing?

We know, we know. That was REALLY tedious. But it doesn't take very long and then we move on to meatier pastures.

## Someone Punches Out

In your test system, a test user will Punchout into our test system (Portal on preview). They will go to your procurement system, login, find Boundless as a supplier, and press the Boundless button.

boundless

This will send over a Punchout Setup Request which is that cXML that you saw above.

In the most amazing, fairy-filled world, the POSR just WORKS! You get logged in under the covers and on your browser, Portal shows up with your store loaded and raring to go.

In the REAL world, there is usually a bit of fiddling that happens behind the scenes between our I/T and your I/T especially if dynamic contact creation is enabled. But this also doesn't take long, and Boundless will assign you a Very Smart Person (VSP) to get you logging in like a lumberjack in no time.

## You build up a cart

You peruse your Boundless catalog, say things like "ooooh" and "ahhhh" and "Does that come in purple?" and load a few things into your cart.

## You submit your cart to your procurement system

When it comes time to check out, your cart will say, "Submit order," and that is shorthand for "Send a Punchout Order Message to my PROCUREMENT SYSTEM breaking this order up into consumable cXML."

Now on your side, your tester will check to see if all the items loaded up into a requisition correctly and spot check quantities and pricing.

### Your cart gets approved and you send us a PO

Someone (possibly the tester or a super-tester) will say, "YES! That order is APPROVED!" And they click a button that sends the cart + PO back to Portal so that your account management team knows that they can go ahead and start to process of fulfilling your order on our end.

## Maybe we invoice you

If you have chosen to receive cXML invoices, we'll run a quick test and submit invoices into your system. Your tester will see if your accounting system can see the invoice and they may receive against the invoice.

## Usual testing cycles

Here is where we normally spend the most time during testing

- Punchout Setup Requests – can your buyers log in? Do users get created dynamically if that feature is enabled?
- Purchase Order Requests – when your buyers modify the cart in the procurement system and then send it back to us, do we see the information we expect to see? How does the pricing look?
- Invoicing – there are a lot of field mappings that happen here. Are we sending you back the info you need to populate the accounting system? Is your tolerance set to a high enough value (the original cart doesn't contain freight or tax, so the invoice is sometimes higher than the PO total)? How do Credit Memos look, where is the credit calculated, as a discount or as a negative total against the order header?

We've had Punchouts enabled and tested in under 2 days (once the Portal Store is built), and we've had some that have taken weeks. When it can take a bit of time, it's usually because the buying organization can't spare a person for dedicated testing and they can only do one test a week or so. If you have a dedicated tester, your Boundless Very Smart Person, and your I/T person, can hack it all out pretty quickly. If you don't have a dedicated tester, you'll still get a VSP, but they may be doing other work in between your testing.

## And then what?

Once all of the tests have passed to the satisfaction of both sides, we then enable your store on production (the "real-world" server) for go-live. This only takes a couple of hours because we know what configuration works based on all the testing we did on the preview server.

Usually, your I/T department will open up a very limited Access Control List (ACL) for users to start trying out the catalog production for a soft go-live.

These primary users may place an order and send over a PO or two (or a few). When both sides are satisfied that everything looks ok, optionally these POs are cancelled on both sides. Then, the ACL on your side usually gets expanded to include all of the users your organization wishes to have access to the Boundless Portal.

As soon as a go-live date is set, your Boundless VSP makes sure the test orders go through appropriately. They also make sure the first few general (real) orders go through as planned and that invoicing, if enabled, flows as well.

Then we're cooking with gas – and that means your Punchout integration is complete! So, pat yourself on the back – take all the praise from your colleagues – you made it through the process and look like a rock star! Congratulations! That's exactly what we're here to help you do.

If you need anything further, please feel free to reach out to us. We're happy to discuss – or even set you up to conference/screenshare with your very own Boundless Very Smart Person to answer any of the questions you have!