Eating Your Own Dog Food

Balisage 2019

Ari Nordström | ari.nordstrom@gmail.com

Steven Seagal, Sir Tim Berners-Lee, and a Canadian guy walk into a conference...

eat (one's) own dog food (redirected from *Eat His Own Dog Food*)

Also found in: Financial.

eat (one's) own dog food

1. To use the products one's company produces or develops as a means of demonstrating or validating its quality, capabilities, or superiority to other brands. Used primarily in reference to software industries, the phrase is thought to have originated with advertisements for Alpo dog food in the 1950s, in which actor Lorne Greene promoted the product by pointing out that he fed it to his own dogs.

The company sent out a memo to all of its employees telling them to eat their own dog food to demonstrate their new operating system's speed and ease of use.

2. By extension, to use software one's company is developing—usually in its beta form—so as to test it for flaws and ensure its ease of use by end users before it is released.

We didn't have time to eat our own dog food before the new operating system's release, so I'm worried it may still have a lot of glitches that haven't been accounted for yet.

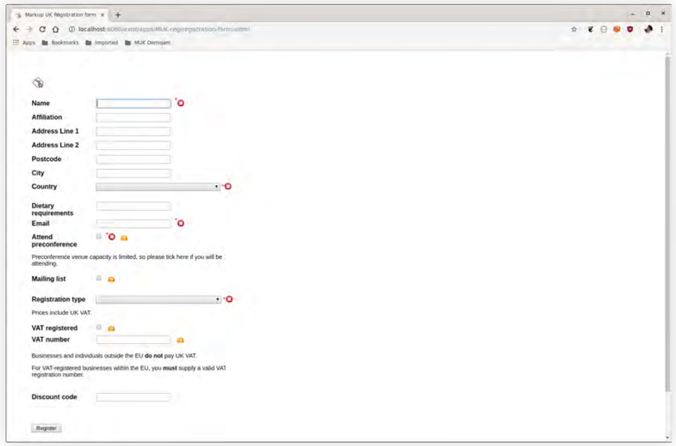
See also: dog, eat, food, own


Programming

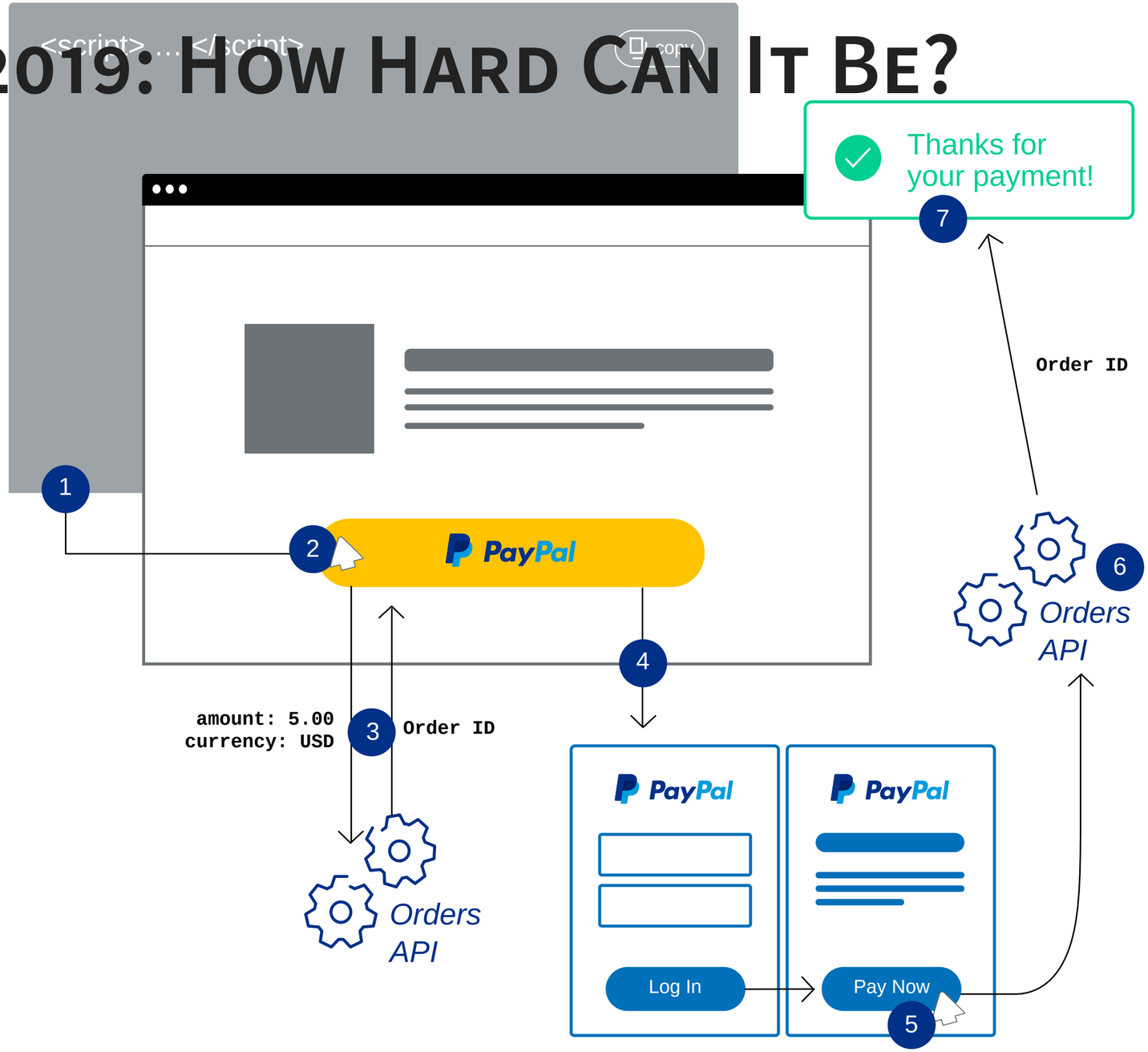
A Conference Is Born


Panic

2018: Partial Solution



2019: How Hard Can It Be?



Off the PayPal Website...

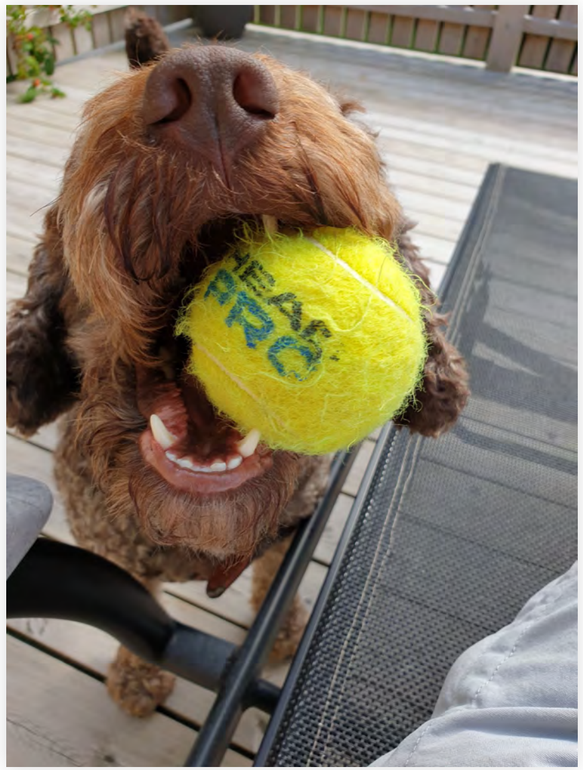
```
<script src="https://www.paypal.com/sdk/js?client-id=sb"/>
<script>paypal.Buttons().render('body');</script>
...
<form
    action="https://www.paypal.com/cgi-bin/webscr"
    method="post">
  <input
        type="hidden"
        name="cmd"
        value="_xclick"/>
</form>
```

```
<INPUT TYPE="hidden" NAME="currency_code" value="GBP"/>
```

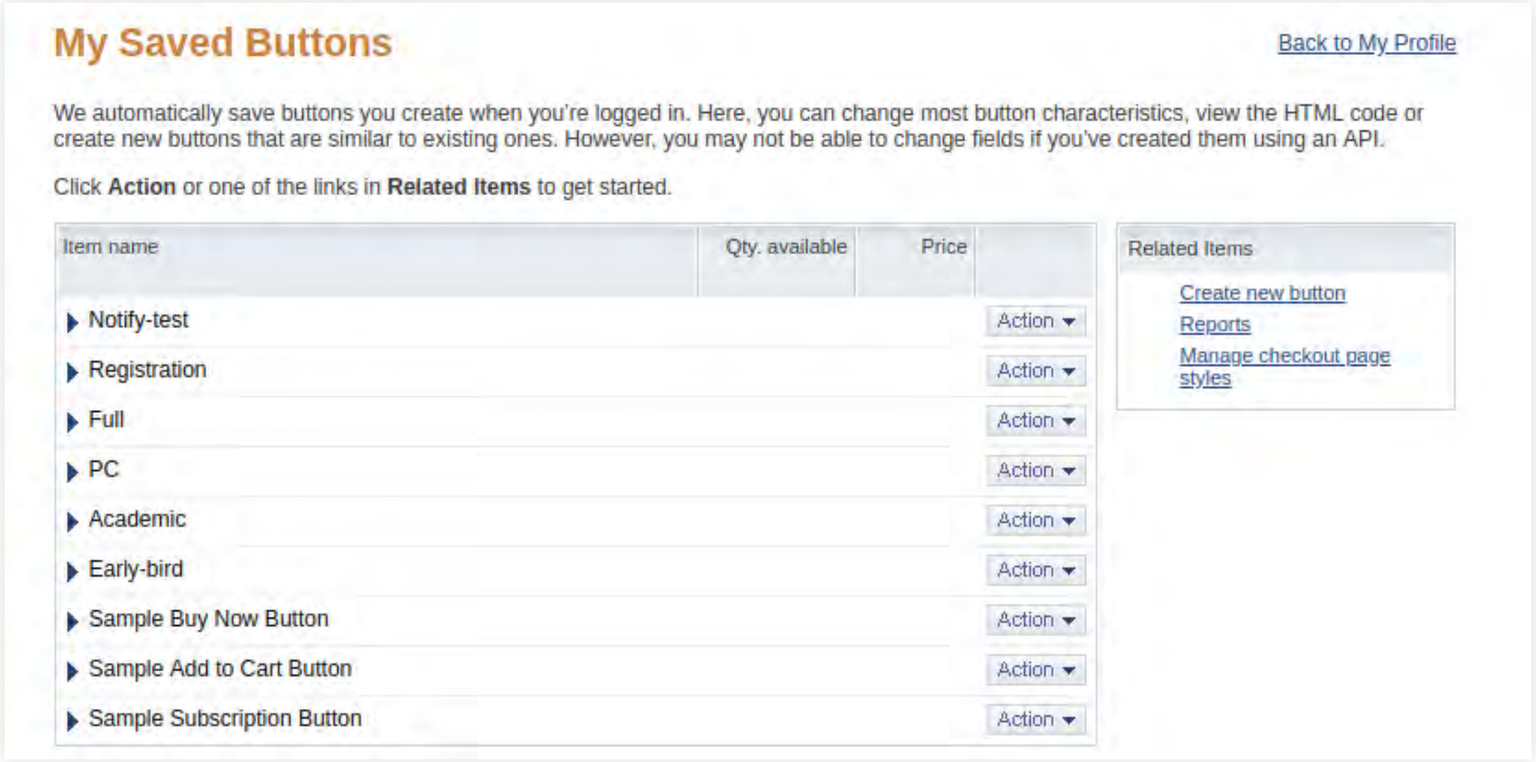

Cutting corners to meet arbitrary management deadlines
Essential
Copying and Pasting from Stack Overflow
O'REILLY
The Practical Developer
@ThePracticalDev

No Custom Amounts

EARLY-BIRD, ACADEMIC, FULL, DISCOUNTS...

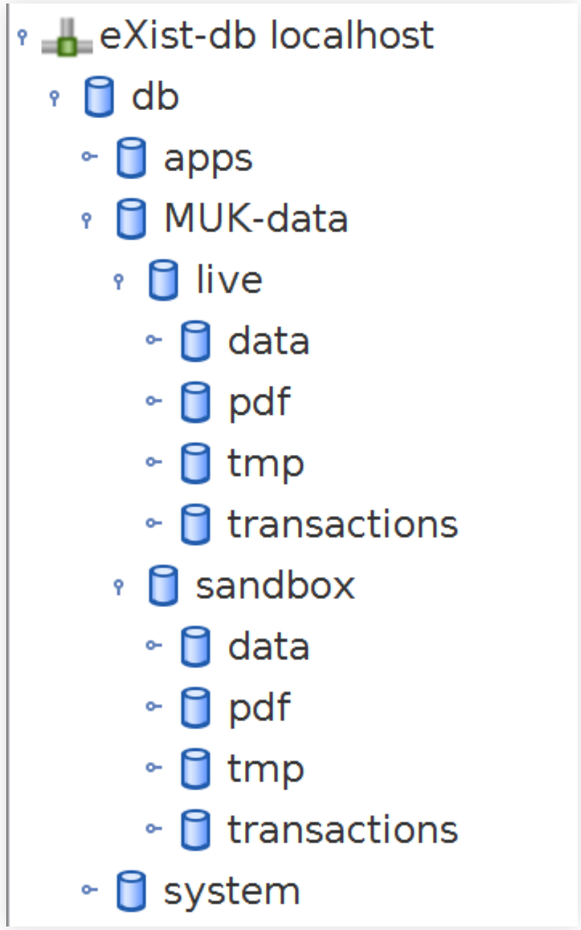


Generating Button Code



```
<button id="early-bird-sandbox">
    <form
        action="https://www.sandbox.paypal.com/cgi-bin/webscr"
        method="post"
        target="_top">
        <input
            type="hidden"
            name="cmd"
            value="_s-xclick"/>
        <input
            type="hidden"
            name="hosted_button_id"
            value="PM4DRL6BX3226"/>
        <input
            type="image"
            src="https://www.sandbox.paypal.com/en_US/GB/i/btn/btn_bu
            border="0"
            name="submit"
            alt="PayPal – The safer, easier way to pay online!"/>
        <img
            alt=""
            border="0"
```

Sandbox/Live



Lots of Buttons

```
<buttons>
    <button id="early-bird-sandbox">
        ...
    </button>

    <button id="academic-sandbox">
        ...
    </button>

    <button id="pc-sandbox">
        ...
    </button>

    <button id="full-sandbox">
        ...
    </button>

    ...

</buttons>
```


Thankfully I Found A Third Option

5. Capture the transaction

Next, implement the `onApprove` function, which is called after the buyer approves the transaction on paypal.com. This function:

- Optionally calls PayPal using `actions.order.get()` to get the transaction details and display them to the buyer.
- Calls PayPal using `actions.order.capture()` to capture the funds from the transaction.
- Shows a message to the buyer to let them know the transaction is successful.

Note: For the basic integration you'll do this step on the client. For [advanced use cases](), you can also call from your server to capture a transaction.

HTML

```
<script>
  paypal.Buttons({
    createOrder: function(data, actions) {
      return actions.order.create({
        purchase_units: [{
          amount: {
            value: '0.01'
          }
        }]
      });
    },
    onApprove: function(data, actions) {
      // Capture the funds from the transaction
      return actions.order.capture().then(function(details) {
        // Show a success message to your buyer
        alert('Transaction completed by ' + details.payer.name.given_name);
```

PayPal Developer Documentation

I Found This Example

```
<script>
  paypal.Buttons({
    createOrder: function(data, actions) {
      // Set up the transaction
      return actions.order.create({
        purchase_units: [{
          amount: {
            value: '0.01'
          }
        }]
      });
    }
  }).render('#paypal-button-container');
</script>
```

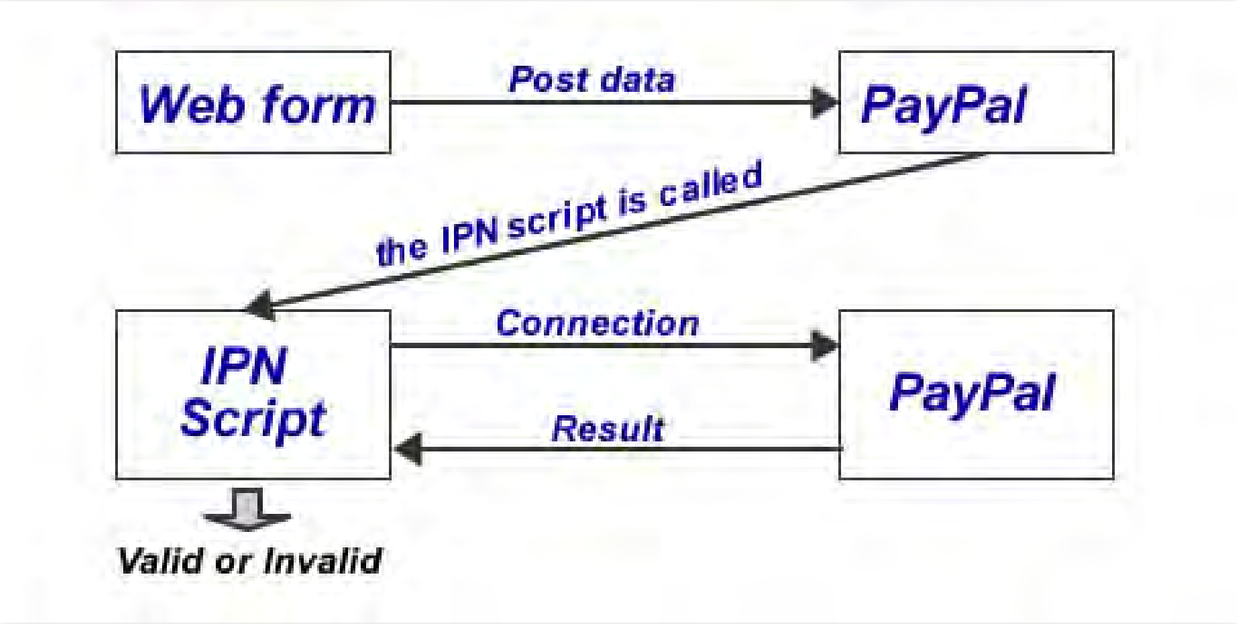
```
amount: {
    value: '{$amount}'
}
```

A Better Button

```
<script>
  paypal.Buttons({{
    createOrder: function(data, actions) {{...;}},

    onApprove: function(data, actions) {{...}},

    onCancel: function(data, actions) {{...}}

  }}).render('#paypal-button-container');

</script>
```

```
onApprove: function(data, actions) {{
  return actions.order.capture().then(function(details) {{

    actions.redirect(
        '{$muk:config//.../thanks.xquery?tmp-file={$registered}'
    );
    // Call your server to save the transaction
    return fetch(...);
  }});
}},

onCancel: function(data, actions) {{
    actions.redirect('{$muk:config//.../cancel.xquery');
}}
}}).render('#paypal-button-container');
```

But First, You Need to Verify



IPN Message

```
verify_sign=AtkOfCXbDm2hu0ZELryHFjY-Vb7PAUvS6nMXgysbElEn9v-1XcmSoGtf&
payer_email=gpmac_1231902590_per%40paypal.com&
txn_id=61E67681CH3238416&
payment_type=instant&
last_name=User&
address_state=CA&
receiver_email=gpmac_1231902686_biz%40paypal.com&
payment_fee=0.88&
receiver_id=S8XGHLYDW9T3S&
txn_type=express_checkout&
item_name=&
mc_currency=USD&
item_number=&
residence_country=US&
test_ipn=1&
handling_amount=0.00&
transaction_subject=&
payment_gross=19.95&
shipping=0.00
```

Um, what's a "listener?"

I Created a Test Form

```
<form
    target="_new"
    method="post"
    action="http://localhost:8080/exist/rest/db/test/test.xquery"
    enctype="multipart/form-data">

    <input type="hidden" name="payer_id" value="LPLWNMTBWMFAY"/>
    <input type="hidden" name="payment_status" value="Completed"/>

    <!-- code for other variables to be tested ... -->
    <input type="hidden" name="custom" value="my_info"/>

    <input type="submit"/>
</form>
```

Pointing Out This XQuery

```
xquery version "3.1";
declare namespace exist = "http://exist.sourceforge.net/NS/exist";
declare namespace xmldb="http://exist-db.org/xquery/xmldb";
declare namespace request="http://exist-db.org/xquery/request";
declare option exist:serialize "method=xml media-type=text/xml indent

let $post-data := request:get-data()

return
<post-data>
   {$post-data}
</post-data>
```

HEAR, HEAR!

A Guide to Training a Deaf Dog

By Barry Eaton

Help Me (You're My Only Hope)

```
1   xquery version "3.1";
2
3   module namespace ipnl = "http://markupuk.org/registration/ipn-listener";
4
5   declare namespace err = "http://www.w3.org/2005/xqt-errors";
6   declare namespace rest = "http://exquery.org/ns/restxq";
7   declare namespace output = "http://www.w3.org/2010/xslt-xquery-serialization";
8   declare namespace http = "http://expath.org/ns/http-client";
9
10  (: Indicates if the sandbox endpoint is used :)
11  declare variable $ipnl:use-sandbox := true();
12
13  (: Production Postback URL :)
14  declare variable $ipnl:verify-uri := "https://ipnpb.paypal.com/cgi-bin/webscr";
15
16  (: Sandbox Postback URL :)
17  declare variable $ipnl:sandbox_verify_uri := "https://ipnpb.sandbox.paypal.com/cgi-bin/webscr";
18
19  (: Response from PayPal indicating validation was successful :)
20  declare variable $ipnl:valid := "VERIFIED";
21
22  (: Response from PayPal indicating validation failed :)
23  declare variable $ipnl:invalid := "INVALID";
24
25  declare
26      %rest:POST("{$body}")
27      %rest:path("/registration/ipnl")
28  function ipnl:listener($body) {…};
52
53
54  declare
55      %private
56  function ipnl:verify-ipn($body) as xs:boolean {…};
120
121
122 (:~
123  : Parses an IPN from a HTTP Request Body into an XDM array
124  :
125  : Each entry in the array is a sequence of two values, the first is a key, the second is the value
126  :
127  :)
128 declare
129     %private
130 function ipnl:parse-ipn-from-body($body) as array(*) {…};
158
159 declare
160     %private
161 function ipnl:serialize-ipn-for-body($my-post as array(*)) as xs:string? {…};
178
179 declare
180     %private
181 function ipnl:get-paypal-uri() {…};
188
189 declare
190     %private
191 function ipnl:substr-count($str as xs:string, $pattern as xs:string) as xs:integer {…};
194
```

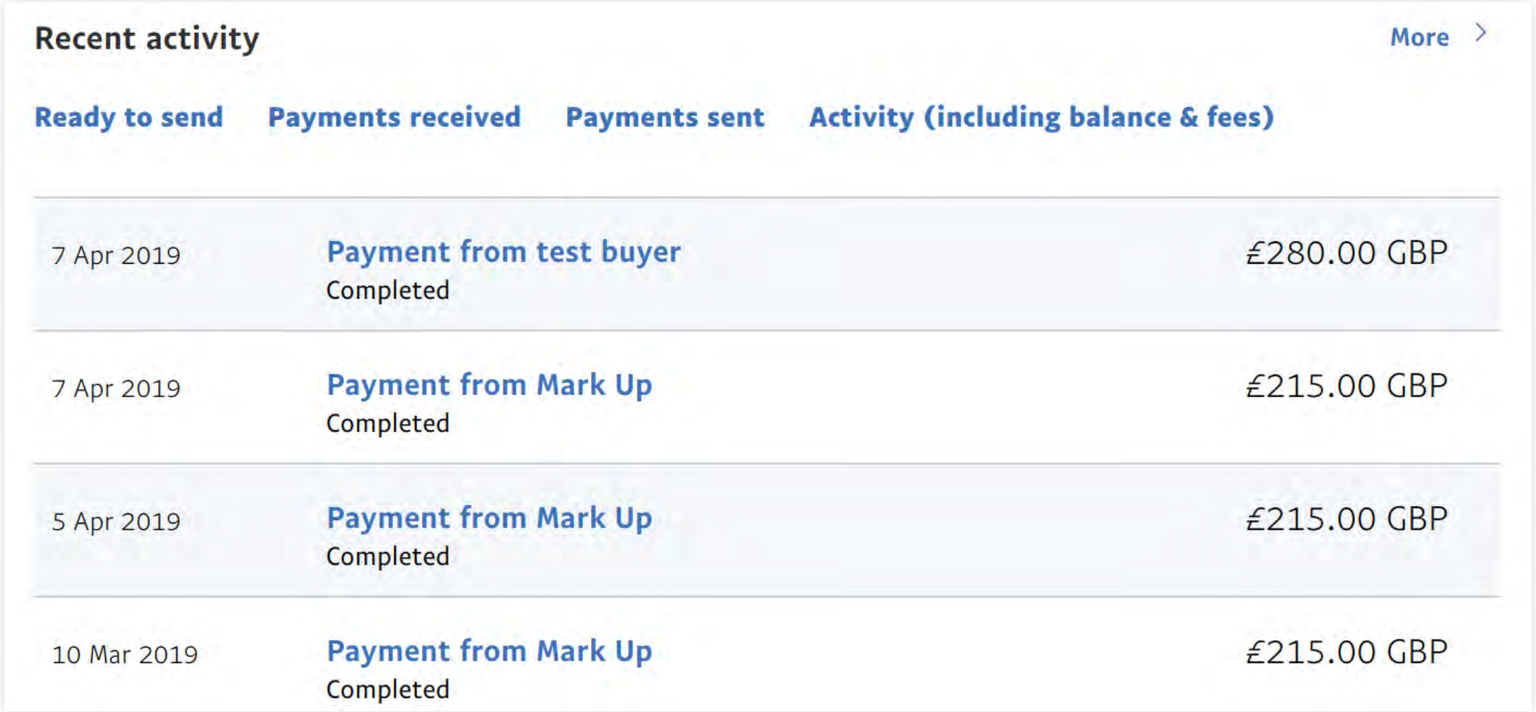
My Simple "Listener"

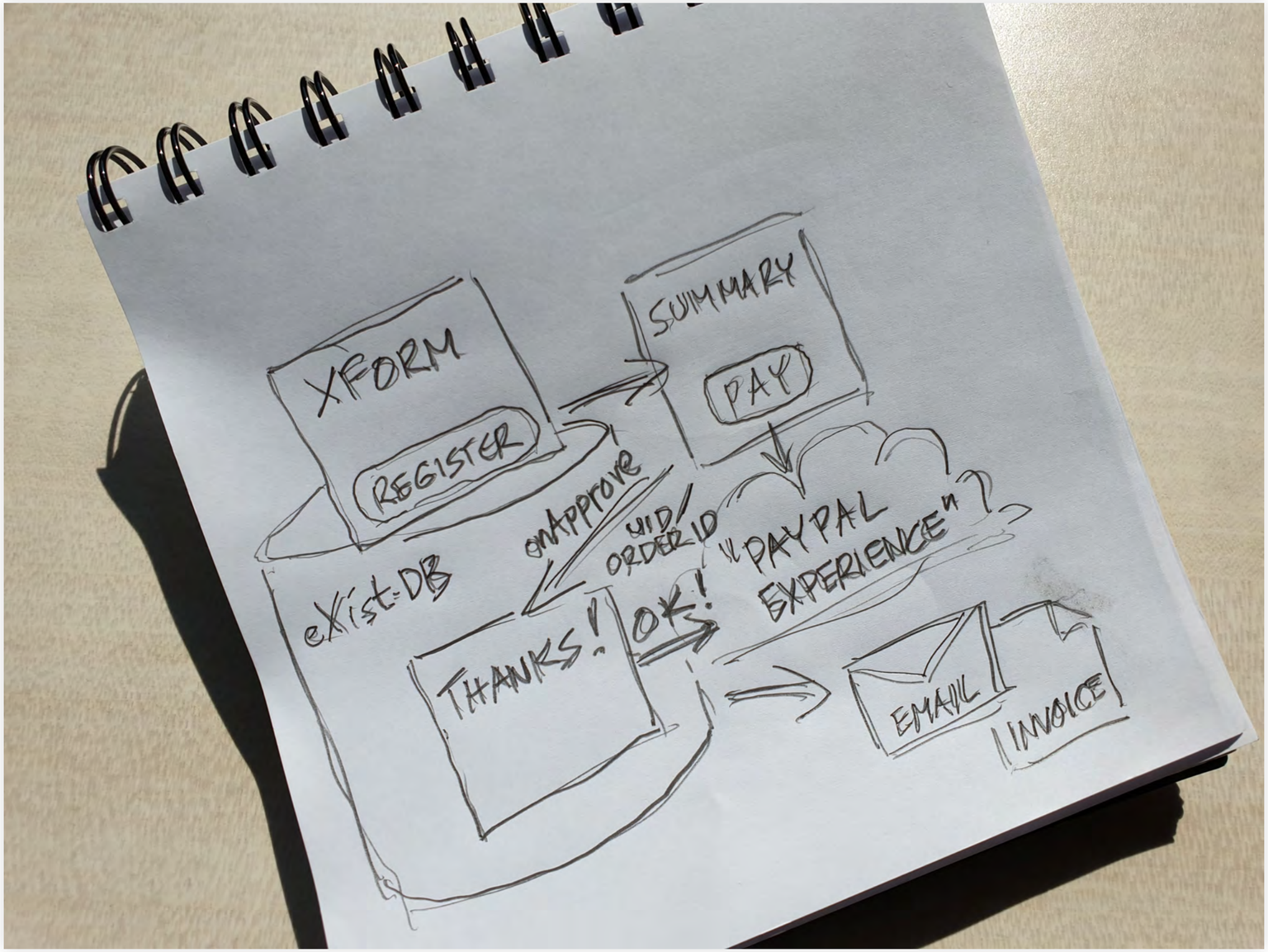
```
declare
    %rest:POST("{$body}")
    %rest:path("/paypal-transaction-complete")
function pay:listener($body) {
    let $json := fn:parse-json(util:base64-decode($body))

    return
    xmldb:store($muk:transactions-collection-uri,
        concat('uid-',$json?custom,'.xml'),
        <transaction tstamp='{current-dateTime()}'
            orderID='{$json?orderID}'
            userID='{$json?custom}'/>),
    pay:all-good()
};

declare function pay:all-good(){
    (: Reply with an empty 200 response to indicate to paypal the IPN was receiv
        (
            <rest:response>
                 <http:response status="200" reason="OK"/>
            </rest:response>,
            ""
        )
};
```

Adding The Endpoint

```
onApprove: function(data, actions)
    ...

    // Call your server to save the transaction
    return fetch(
        '{$muk:config//domain}/exist/restxq/paypal-transaction-complete', {{
      method: 'post',
      body: JSON.stringify({{
        orderID: data.orderID,
        custom: {string($uid)}
      }})
    }});

    ...

onCancel: function(data, actions) {{
    ...
```

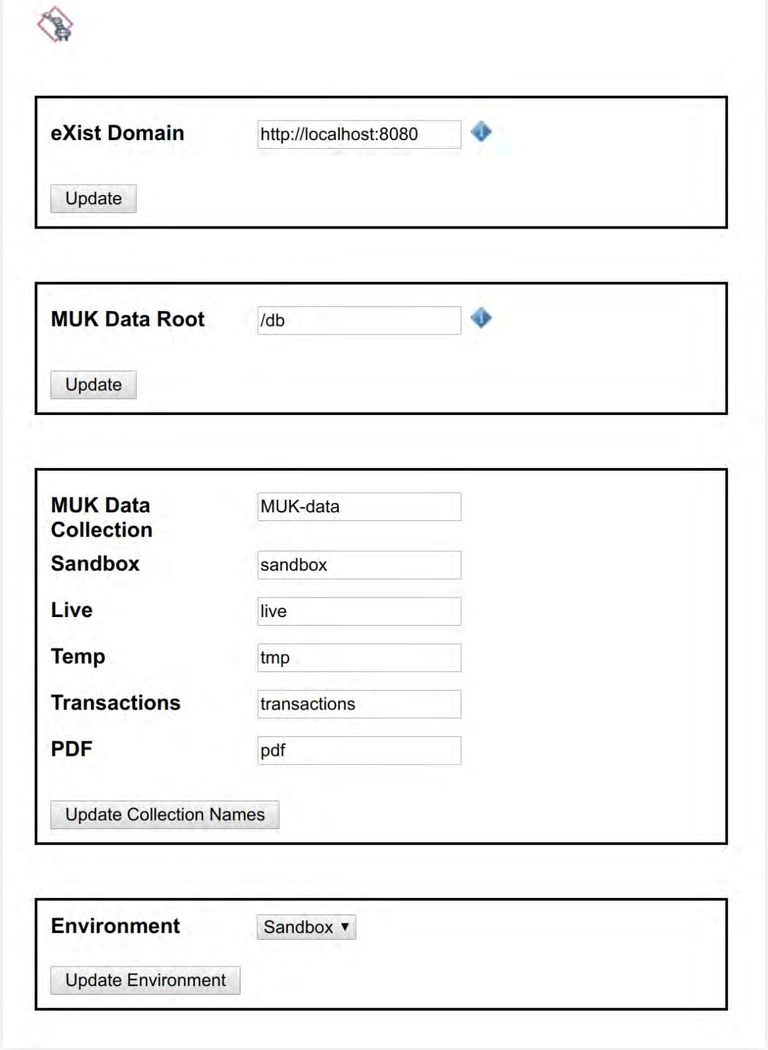
And It All Started Working



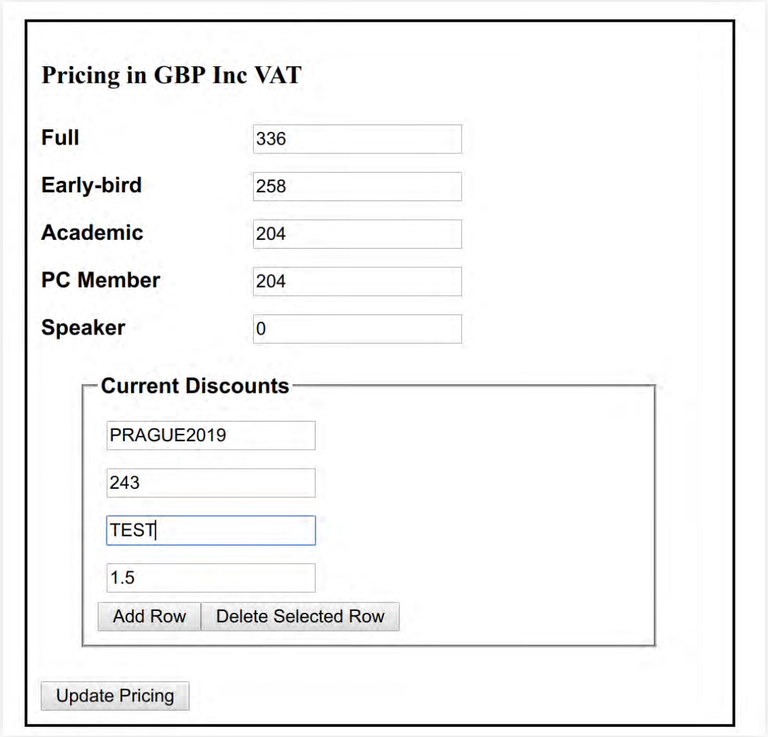

XFORM
REGISTER
SUMMARY
PAY
eXist-DB
onApprove
VID/
ORDERID
"PAYPAL
EXPERIENCE"
THANKS!
OK!
EMAIL
INVOICE


Now That I'm A Programmer...

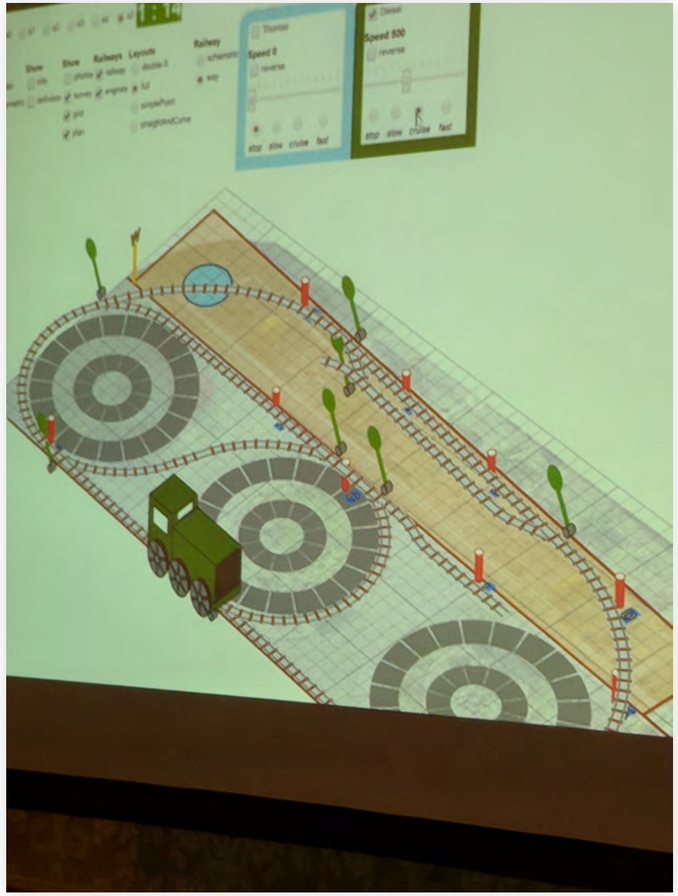
Admin Page


eXist Domain
http://localhost:8080
Update
MUK Data Root
/db
Update
MUK Data Collection
MUK-data
Sandbox
sandbox
Live
live
Temp
tmp
Transactions
transactions
PDF
pdf
Update Collection Names
Environment
Sandbox
Update Environment

Set Amounts and Discounts



Demojam Voting App

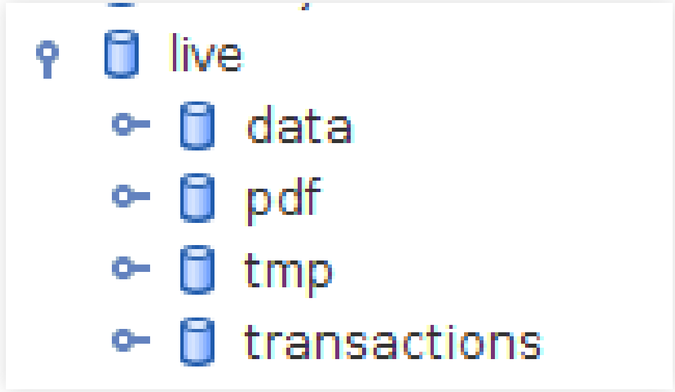


Presentations

- Edgar Wallace - Beer Is My Life
- John Smythe - Why I Am
- Adrian Hubert Rollerblade - The Joy of Money
- Charlotte Always - Thinking Fish
- Mycroft James - Study in Pink
- William Superman - Why My Life Is Ordinary
- Antonia Bucket - My Life
- Winona Kiefer - Interesting Newspaper Clippings
- Steven Seagal - Romantic Notions
- JFK - How I Bit the Bullet
- Lyndon B Johnson - Good Times
- Mia War - Final Call
- A B Letter - Books
- Test Again - For Show
- test - test
- Final - Call

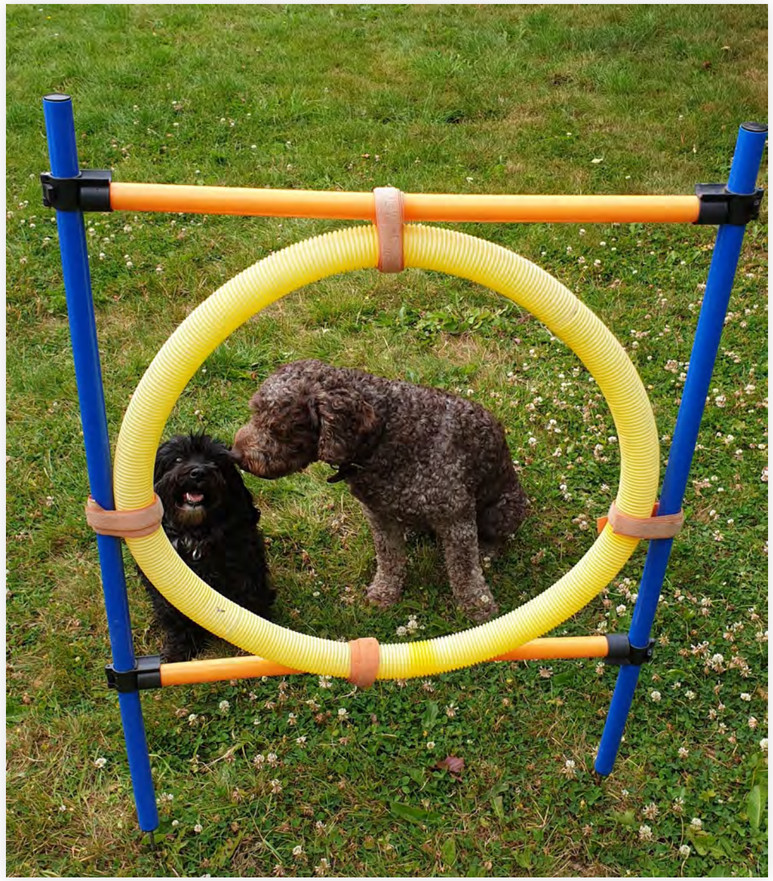
Vote

Reports



UID 46 Sheila T	full		2019-05-19T12:06:14.429-04:00
UID 47 Stephen	full	s	2019-05-20T16:14:55.34-04:00
UID 48 M	full		2019-05-22T10:14:02.226-04:00
UID 49 Nic Gi	full		2019-05-22T17:18:38.062-04:00
UID 50 Nic G	full		2019-05-23T05:18:57.337-04:00
UID 51 Andor	academic	a	2019-05-23T10:28:53.594-04:00
UID 52 Jenny She	speaker	J	2019-05-23T11:14:45.796-04:00

2020: Enterprise-ready?



Thank You!