Notifications - Deep Links

Story: I want to send a list of vacancies or shifts to a candidate via notifications and the notification should contain a deep link that when they click on it it opens a web page to say that they have been added to the shortlist, update their progress status on the vacancy shortlist, add them to the shift shortlist and log a contact event.

There are a number of elements that need to be in place, expand the sections below for details on what needs to be set up.

Shortlist/Progress States

Suitable shortlist states need to be set up in Maintenance - Agency Setup - Progress States and if you are sending shift details Maintenance - Temps Setup - Shift Progress States.

Hub

A dedicated IQXHub for Deep Link Jobs must be active and the RunJob Schedule Group and RunJob UserID should be set in the ini, the UserId is used as the consultant linked to the Progress and Shift Progress records created. This example is based on a local hub.

Elements for the Notification

Boiler Plate Text - If you want to send individual lines for each shift then you will need a Boiler Plate Text item set up called SHIFTLINES with a type of Notification. It is HTML code and will refer to a Custom Merge Field. Only required for sending shift lines.

literal<tr><td>{SHF_DATE}</td><td>{SHF_FROM}</td> <td>{SHF_TO} </td><td> Accept Booking<u><<</u></td></tr>

This example should be amended so that 'localhost:54000' is changed to the address of the hub that will run the job and 'UpdateProgressAndShiftProgress.xml' is changed to the name of the job that is being run.

Custom Merge Field - again only required if you are sending individual shift information. The category should be Shift, our example is called DEEPLINKQUERYBOOK and the SQL should be 'SELECT string('vacancyid=',vacancyid,'&tempshiftplanid=', tempshiftplanid) from tempshiftplan where tempshiftplanID = #' tick for Notifications only.

Sample Notification

Two notifications are required one to create the web page and another that will be selected from the notifications button.

The notification that is used to draw a web page is called HTMLTEST in this example (this is the name specified in the custom merge field if you are using it and the subsequent notification template), type of Job. The contents of the web page can be customised to each agencies requirements.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="/docs/4.0/assets/img/favicons/favicon.ico">

    <title>Product example for Bootstrap</title>
    <link rel="canonical"
href="https://getbootstrap.com/docs/4.0/examples/product/">
    <!-- Bootstrap core CSS -->
    <link
href="https://getbootstrap.com/docs/4.0/dist/css/bootstrap.min.css"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link
href="https://getbootstrap.com/docs/4.0/examples/product/product.css"
rel="stylesheet">
  </head>

  <body>

    <nav class="site-header sticky-top py-1">
      <div class="container d-flex flex-column flex-md-row justify-content-
between">
        <a class="py-2" href="#">
          <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round" class="d-block mx-
auto"><circle cx="12" cy="12" r="10"></circle><line x1="14.31" y1="8"
x2="20.05" y2="17.94"></line><line x1="9.69" y1="8" x2="21.17"
y2="8"></line><line x1="7.38" y1="12" x2="13.12" y2="2.06"></line><line
x1="9.69" y1="16" x2="3.95" y2="6.06"></line><line x1="14.31" y1="16"
x2="2.83" y2="16"></line><line x1="16.62" y1="12" x2="10.88"
y2="21.94"></line></svg>
        </a>
        <a class="py-2 d-none d-md-inline-block" href="#">Tour</a>
```

```
        <a class="py-2 d-none d-md-inline-block" href="#">Product</a>
        <a class="py-2 d-none d-md-inline-block" href="#">Features</a>
        <a class="py-2 d-none d-md-inline-block" href="#">Enterprise</a>
        <a class="py-2 d-none d-md-inline-block" href="#">Support</a>
        <a class="py-2 d-none d-md-inline-block" href="#">Cart</a>
      </div>
    </nav>

    <div class="position-relative overflow-hidden p-3 p-md-5 m-md-3 text-
center bg-light">
      <div class="col-md-5 p-lg-5 mx-auto my-5">
        <h1 class="display-4 font-weight-normal">Punny headline</h1>
        <p class="lead font-weight-normal">And an even wittier subheading to
boot. Jumpstart your marketing efforts with this example based on Apple's
marketing pages.</p>
        <a class="btn btn-outline-secondary" href="#">Interest
Acknowledged</a>
      </div>
      <div class="product-device box-shadow d-none d-md-block"></div>
      <div class="product-device product-device-2 box-shadow d-none d-md-
block"></div>
    </div>

    <footer class="container py-5">
      <div class="row">
        <div class="col-12 col-md">
          <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round" class="d-block mb-2"><circle
cx="12" cy="12" r="10"></circle><line x1="14.31" y1="8" x2="20.05"
y2="17.94"></line><line x1="9.69" y1="8" x2="21.17" y2="8"></line><line
x1="7.38" y1="12" x2="13.12" y2="2.06"></line><line x1="9.69" y1="16"
x2="3.95" y2="6.06"></line><line x1="14.31" y1="16" x2="2.83"
y2="16"></line><line x1="16.62" y1="12" x2="10.88" y2="21.94"></line></svg>
          <small class="d-block mb-3 text-muted">&copy; 2017-2018</small>
        </div>
        <div class="col-6 col-md">
          <h5>Features</h5>
          <ul class="list-unstyled text-small">
            <li><a class="text-muted" href="#">Cool stuff</a></li>
            <li><a class="text-muted" href="#">Random feature</a></li>
            <li><a class="text-muted" href="#">Team feature</a></li>
            <li><a class="text-muted" href="#">Stuff for developers</a></li>
          </ul>
        </div>
      </div>
    </footer>
```

```
    <!-- Bootstrap core JavaScript
    ================================================== -->
    <!-- Placed at the end of the document so the pages load faster -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script>window.jQuery || document.write('<script
src="https://getbootstrap.com/docs/4.0/assets/js/vendor/jquery-slim.min.js">
<\/script>')</script>
    <script
src="https://getbootstrap.com/docs/4.0/assets/js/vendor/popper.min.js"></scr
ipt>
    <script
src="https://getbootstrap.com/docs/4.0/dist/js/bootstrap.min.js"></script>
    <script
src="https://getbootstrap.com/docs/4.0/assets/js/vendor/holder.min.js"></scr
ipt>
    <script>
      Holder.addTheme('thumb', {
        bg: '#55595c',
        fg: '#eceeef',
        text: 'Thumbnail'
      });
    </script>
  </body>
</html>
```

The second notification is the one that will be selected by the user, the type will depend on where you want to send the list from, if during the fill shifts process then the type should be QueryBook or if it is from the progress view on the person selector the type should be APP_ProgressContact. You also need to set the contact event type, updated progress state and if sending shift lines the Shift Lines in the top right section of the notification form.

QueryBook Example

```
<html>
{{app_name}
<table >
{{SHN_BEGIN}
<tr><td>Company</td> <td>{{SHN_vac_sitename}</td><tr>
<tr><td>Contact</td> <td>{{SHN_vac_sitecontact}</td><tr>
<tr><td>reference</td> <td>{{SHN_vac_refcode}</td><tr>
<tr><td>position</td> <td>{{SHN_vac_position}</td><tr>
<tr><td></td> <td>Shifts</td><tr>
<tr ><td ></td> <td><table border=1>{{SHF_SHIFTLINES}</table></td><tr>
<tr ><td > </td> <td>-------------- </td><tr/>
```

```
{{SHN_END}
</table>
<html>
```

App_ProgressContact Example - as this one doesn't send shift lines it doesn't include the Boiler Plate text instead the link is contained within the notification itself

```
<HTML>
<table>
{{PRG_BEGIN}
<TR><TD>{{vac_refcode}</TD>
<TD>{{com_companyname}</TD>
<TD>{{com_addr1}</TD>
<TD>{{VAC_POSITION}</TD>

<td> <a
href="http://localhost:54000/iqxjob/public_test1?personid={{APP_PERSONID}&va
cancyid={{VAC_VACANCYID}&templateid={{LET_TEMPLATEID}&displaytemplateid=HTML
TEST"><font color="#FF4D4D">Accept Booking</font></a><u><font
color="#FF4D4D">&lt;&lt;</font></u></td>
{{PRG_END}
</Table>
</HTML>
```

Job

The job will need to be added to the jobs folder on the hub server and is triggered when the link is clicked. The job can be customised to your agency requirements, the example below updates the progress state, adds to the shift progress and logs a contact event.

```
<Job  title="Respond to Notification deep link" autoclose="Yes">

<Library>
    <WriteWebPage>

        <Setvariable name="dtemplate" sql="select  TemplateHTML  from
NotificationTemplate where NotificationTemplateid = :displaytemplateid"/>

        <if x1="{dtemplate}" x2="" comparison="!=">
            <Write>{dtemplate}</Write>
        </if>
    </WriteWebPage>
</Library>

   <Parameters>
```

```
        <Parameter name="personid"/>
        <Parameter name="vacancyid"/>
        <Parameter name="tempshiftplanid"/>
        <Parameter name="templateid"/>
        <Parameter name="displaytemplateid"/>
   </Parameters>

      <SetVariable name="progid" value=""/>

    <Block>
                        <BeginTransaction/>

                              <SQLQuery>
                                <SQLSelect>
                                    select  PERSON.personid,person.name as
recipients,  isnull((select top 1 progressid from progress where progressid
is not null and progress.personid = person.personid and vacancyid =
:vacancyid),'') as pg,
                                          (select progressstatus from
notificationtemplate where notificationtemplateid = :templateid) as pstatus,
                                          (select classcode from
notificationtemplate where notificationtemplateid = :templateid) as ccode,
                                          (select Classdescrip from
contactclass where classcode = ccode) as cdesc
                                       from person  where person.personid =
:personid and
                                         exists (select * from vacancy where
vacancyid = :vacancyid)
                                </SQLSelect>
                                <IfAnyRows>
                                <SQLExec>INSERT INTO tempshiftprogress (
TempShiftProgressID,TempShiftPlanID,Status,StaffID,PersonID,WhenEntered )
    VALUES ( Uniquekey(''), :tempshiftplanid, 'S', userstaffid, :personid,
current timestamp );</SQLExec>
                                  <if x1="{pstatus}" x2="" comparison="!=">
                                       <SetVariable name="progid"
value="{uniquekey}"/>
                                       <if x1="{pg}" x2="">
                                           <SQLExec>Insert into
progress(progressid, personid, vacancyid, status, actiondate, staffid, Note)
values (
                                           :progid
,:personid,:vacancyid,:pstatus,current date, userstaffid,'Deep Link')
                                           </SQLExec>

                                       </if>
                                       <else>
                                           <SQLExec>update progress set
```

```
actiontime=current time, actiondate =current date, status= :pstatus
                                                    where personid = :personid
and vacancyid = :vacancyid</SQLExec>
                                          </else>
                                          <SetVariable name="progid"
sql="select progressid from progress where personid = :personid and
vacancyid = :vacancyid"/>
                                    </if>
                                    <if x1="{ccode}" x2="" comparison="!=">
                                          <SetVariable name="ceid"
value="{uniquekey}"/>

                                          <SQLExec><![CDATA[
                                          insert into contactevent
(contacteventid,staffid, personid,vacancyid, progressid, description,  who,
Contactdate, Contacttime, classcode)
                                          values (:ceid, userstaffid,
:personid, :vacancyid, nullif(:progid, '') , :cdesc,  'A', current date,
current time, :ccode);
                                          ]]></SQLExec>

                                          <SQLExec>call
GenerateContactEventEmailLog (:ceid, current timestamp, :recipients, '',
'IN', 'LINK')</SQLExec>
                                    </if>
                                    <Call>WriteWebPage</Call>
                                 </IfAnyRows>
                                 <IfNoRows>
                                 </IfNoRows>
                              </SQLQuery>
                               <Commit/>
    </Block>
    <Catch>

       <Rollback/>
    </Catch>

</Job>
```

From:
https://iqxusers.co.uk/iqxhelp/ - **IQX**

Permanent link:
https://iqxusers.co.uk/iqxhelp/doku.php?id=notifications_deeplink&rev=1681905951

Last update: **2023/04/19 12:05**