DigitalDelve

API Documentation

V1.2

Data Format

XML

Transmit Method

HTTPS post with the request XML contained in the body of the post

Request Endpoint URL

https://demo.digitaldelve.com/listeners/sso.cfm

General Integration Information

This document covers functionality for the following items:

1. AppScreen Requests (Both Fully Integrated and Single Sign-on initiated)
2. Credential Verification
3. Single Sign-on Redirect
4. Status Updates & Results

New in this Documentation

- New function called ViewReportByClientRef cal be used to single sign-on the user and automatically direct them to the report look up page at which the system will automatically search on the reference number provided in the XML and display the associated report.
- New function called RetrieveInvitations which can be used to retrieve a list of AppScreen invitations that can be used with this API.
- If you are using our Risk Assessment model the <Recommendation> element will now appear in the results transmitted back to your system.

FUNCTION: AppScreen

The AppScreen function allows you to send basic applicant data over to be used for the purpose of sending Applicant Invitations out. This feature is available full or partial integration.

The primary difference between full and partial integration is that the partial integration option will single sign-on the requesting user at which time they will manually send the invitation to the applicant. No matter whether you use the partial or full integration option the candidate provided in the request XML is saved and will not need to be re-entered by the candidate or the requesting user. With the full integration option, you will need to send all of the required applicant details to include the AppScreen invitation option you desire to use. Single Sign-on is not required.

Partial Integration Request Example: (Elements in RED are required)

```
<SSO>
      <Authentication>
        <UserName></UserName>
        <Password></Password>
      </Authentication>
       <Function>AppScreen</Function>
      <ResultPostURL> </ResultPostURL> - Required if results to be delivered via API
      <StatusPostURL> </StatusPostURL>
      <Applicant>
        <ReferenceNumber> </ReferenceNumber>
        <FirstName> </FirstName>
        <MiddleName></MiddleName>
        <LastName> </LastName>
        <DateOfBirth></DateOfBirth>
        <SocialSecurityNumber></SocialSecurityNumber>
        <EmailAddress> </EmailAddress>
```

```
        </Applicant>
        <Address>
            <StreetAddress> </StreetAddress>
            <City> </City>
            <State> </State>
            <ZipCode></ZipCode>
        </Address>
    </SSO>
```

If request is successful, the requesting user will be automatically logged into the system to complete the request.

Full Integration Request Example: (Elements in RED are required)

```
<SSO>
    <Authentication>
        <UserName></UserName>
        <Password> </Password>
    </Authentication>
    <Function>AppScreen</Function>
    <ResultPostURL> </ResultPostURL>
    <StatusPostURL> </StatusPostURL>
    <InvitationId> </InvitationId>
    <Applicant>
        <ReferenceNumber> </ReferenceNumber>
        <FirstName> </FirstName>
        <MiddleName></MiddleName>
        <LastName></LastName>
        <DateOfBirth></DateOfBirth>
        <SocialSecurityNumber></SocialSecurityNumber>
        <EmailAddress> </EmailAddress>
    </Applicant>
    <Address>
        <StreetAddress> </StreetAddress>
        <City> </City>
        <State> </State>
        <ZipCode></ZipCode>
    </Address>
</SSO>
```

If Successful, the following response will be generated:

```xml
<?xml version="1.0"?>
<SSO>
        <Status>Success</Status>
         <StatusMessage>Invitation Sent to Candidate</StatusMessage>
     </SSO>
```

Whether using the fully or partially integrated option, if there is a failure the following response will be generated:

```xml
<?xml version="1.0"?>
<SSO>
        <Status>Error</Status>
        <Errors>
                <Error> </Error>
                <Error> </Error>
        </Errors>
</SSO>
```

FUNCTION: AuthOnly

The AuthOnly function can be used to verify the user credentials you have are correct. The request XML is as follows: (All Elements are required)

```xml
<SSO>
        <Authentication>
          <UserName> </UserName>
          <Password> </Password>
        </Authentication>
        <Function>AuthOnly</Function>
   </SSO>
```

If the credentials are good the following response will be generated:
```xml
<?xml version="1.0"?>
<SSO>
        <Status>Success</Status>
</SSO>
```

If not, the following response is generated:
```xml
<?xml version="1.0"?>
<SSO>
        <Status>Error</Status>
```

```xml
        <Errors>
                <Error>Invalid UserName and/or Password</Error>
        </Errors>
</SSO>
```

FUNCTION: RetrieveInvitations

This function can be used by a remote system to pull a list of AppScreen Invitations that can be used via this integration.

The request XML is as follows:
```xml
<?xml version="1.0"?>
<SSO>
        <Authentication>
                <UserName>bryce</UserName>
                <Password>annyong9956!</Password>
        </Authentication>
        <Function>RetrieveInvitations</Function>
</SSO>
```

If Successful, the following response is generated:
```xml
<SSO>
        <Status>Success</Status>
        <Invitations>
                <Invitation>
                        <InvitationId></InvitationId>
                        <InvitationName></InvitationName>
                        <IncludedPackage></IncludedPackage>
                </Invitation>
        </Invitations>
</SSO>
```

If not, the following response is generated:
```xml
<?xml version="1.0"?>
<SSO>
        <Status>Error</Status>
        <Errors>
                <Error> </Error>
        </Errors>
</SSO>
```

Note that the <Invitation> section in a successful response will be repeated for each AppScreen Invitation configuration.

FUNCTION: Dashboard, PendingReports, CompletedReports, Adjudication, ReportLookup, AppScreenMgr, ViewReport, ViewReportByClientRef

The above function names when used will single sign-on the requestor and send them immediately to the page tied to the function name.

Function Name	Redirected To
Dashboard	User sent to client dashboard
PendingReports	User sent to Pending Reports Page
CompletedReports	User sent to Completed Reports Page
Adjudication	User sent to Adjudication Manager
ReportLookup	User sent to Report Lookup Page
AppScreenMgr	User sent to AppScreen Management Interface
ViewReport	User sent to Report Lookup with desired report displayed
ViewReportByClientRef	User sent to Report Lookup with desired report displayed. It is looked up using the reference number provided in the request XML.

The request XML is as follows: (All Elements are Required)
```
<SSO>
    <Authentication>
      <UserName></UserName>
      <Password></Password>
    </Authentication>
    <Function>Plug Function Name in here</Function>
    <ReportId>Plug Report Id in here</ReportId> This is required for ViewReport and
ViewReportByClientRef Function.  Don't be confused by the element name.  If you are trying to
track down a report using your internal reference number you will put that value in this
element rather than our report Id.
   </SSO>
```

If successful, the user will be automatically logged in and directed to the desired page.

If not, the following response is generated:
```
<?xml version="1.0"?>
<SSO>
        <Status>Error</Status>
```

```
        <Errors>
                <Error> </Error>
        </Errors>
</SSO>
```

Status Updates & Results

Once you or your applicant have completed the data entry to order the background check the following status update XML will be posted back to the status update URL provided with the initial order:

```
<?xml version="1.0"?>
<SSO>
  <Authentication>
    <UserName></UserName>
    <Password></Password>
  </Authentication>
  <Status>In Progress</Status>
  <Applicant>
    <ReferenceNumber></ReferenceNumber>
  </Applicant>
</SSO>
```

When the background check has been completed the following XML will sent to the result post URL provided at time of order:

```
<?xml version="1.0"?>
<SSO>
  <Authentication>
    <UserName></UserName>
    <Password></Password>
  </Authentication>
  <Status>Complete</Status>
<Recommendation></Recommendation>
  <ReportURL></ReportURL>
```

```
   <ReportId></ReportId>
```
You should store this value in your system. You will need it to pull the result of the background check.
```
   <Applicant>
     <ReferenceNumber></ReferenceNumber>
   </Applicant>
</SSO>
```

Note that any response but a 200 ok will be seen as a failure and the system will make 4 more attempts to send the results back to your system. We do use the same credentials for the result post as were used when the order was sent to our system. You can get access to the full report result through the URL provided in the <ReportURL> element.