MORNINGSTAR®

AccountView

Single Sign-On Guide

AccountView Version:	1.49
DataConnect Version:	4
Document Version:	7
Document Issue Date:	December 28, 2017

Technical Support:	(866) 856-4951
Telephone:	(781) 376-0801
Fax:	(781) 376-8040
Web:	byallaccounts.morningstar.com

Table of Contents

About this Guide

The guide explains how to manage configuration, user maintenance, and application integration for SSO, using a set of single sign-on (SSO) capabilities:

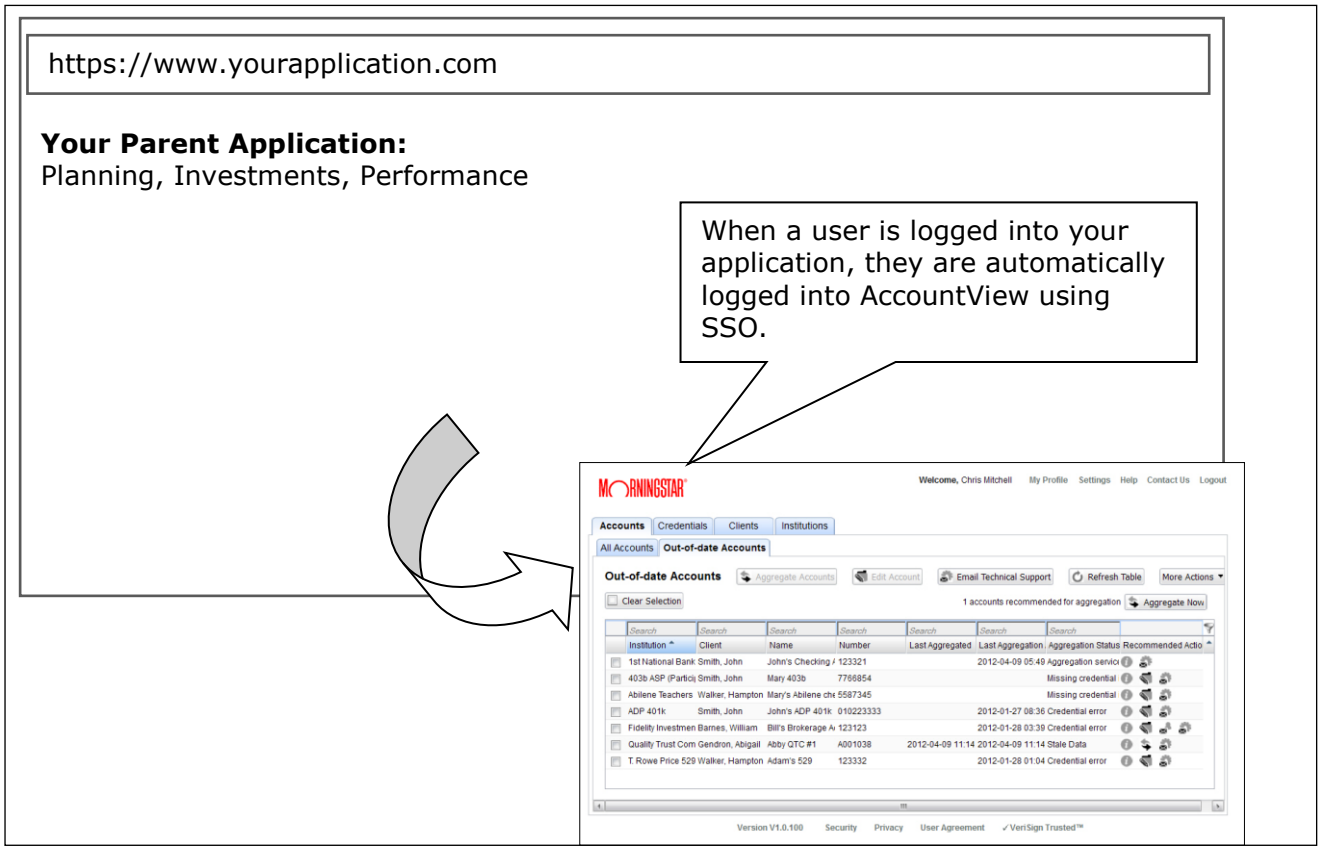
- Configuration – the SSO capability has a number of available configuration options, for example, a URL to redirect users to when they exit AccountView (AV).
- User Maintenance – the DataConnect application programming interface (API) provides requests to create, update, and delete users.
- Application Integration – this is the core SSO functionality that provides for authenticating, application invocation, termination, and other application lifecycle hooks such as "keep alive".
- Implementation Models – there are various ways to implement SSO with AccountView. ByAllAccounts can help you determine which model fits your needs.

Overview of Single Sign-On (SSO) with AccountView

Single Sign-on (SSO) provides a mechanism for partners to integrate AccountView into their own web applications rather than having users log themselves into AccountView. Using SSO, users who log into a partner's application can seamlessly access AccountView without having to log in again.

When the AccountView application is configured for SSO, the unified authentication framework for SSO passes authentication along with other security tokens between the applications. The session can be established using an identifier or a combination of login and password.

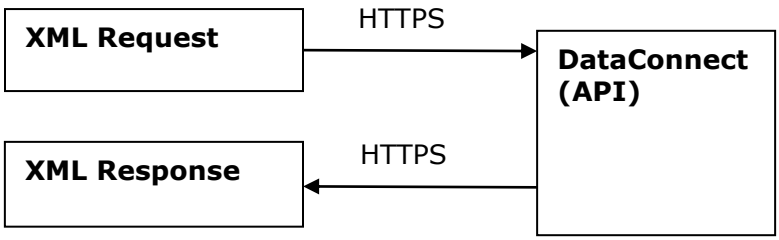
If logins and passwords are established for users of the parent application, they cannot be viewed or changed within AccountView. All maintenance of user logins and passwords must be performed programmatically by the parent application and passed to AccountView using DataConnect.



COMMUNICATION THROUGH DATACONNECT

Your application and AccountView can communicate using DataConnect, which is an application programming interface (API) developed by ByAllAccounts that enables AccountView and your application to exchange XML 1.0 messages over the internet using Hypertext Transfer Protocol Secure (HTTPS) protocol. The messages are data requests and data responses. Using HTTPS ensures that unauthorized individuals are not able to obtain access to sensitive data.

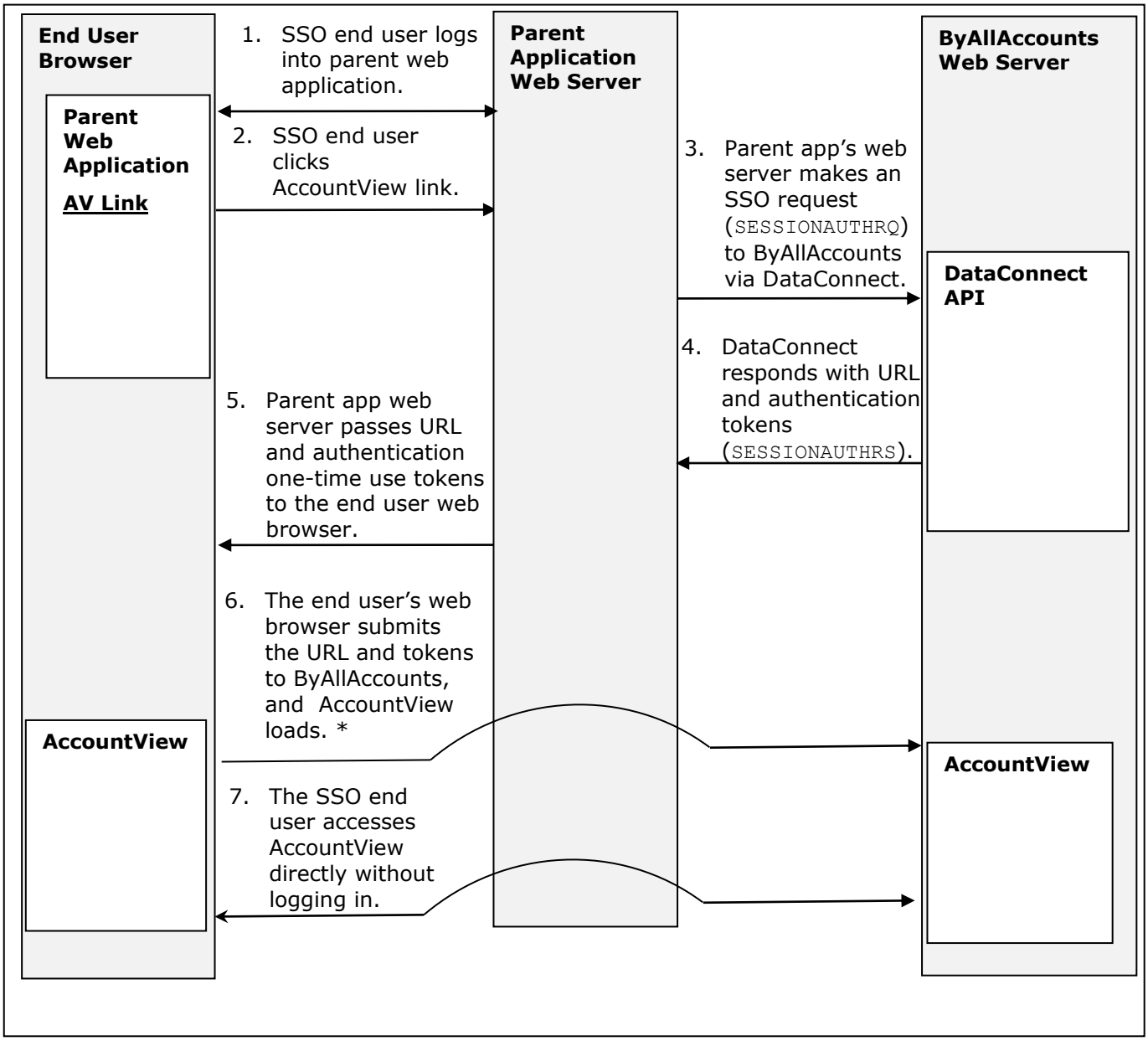
The XML messages that are passed back and forth between the applications through DataConnect comply with a simple proprietary format developed by ByAllAccounts.



The required formats for AccountView single sign-on operations session authenticate request (SESSIONAUTHRQ) and session expire (SESSIONEXPIRERQ) are described in http://www.byallaccounts.net/Manuals/DataConnect/DataConnect_V4_Ultra_User_Guide.PDF. Some code examples are provided later in this guide.

Overview of Application Communication

The following diagram illustrates how the communication process works after user logins have been created for the parent application, and programmatically created for AccountView (AV).



*You can implement AccountView to open in a new window, new tab, or in a frame.

SINGLE SIGN-ON (SSO)

Configuration for SSO

There are a number of SSO configuration options to control the application behavior. Some of these options are chosen by the Firm, and set by ByAllAccounts.

Default SSO Settings Based on User Type

The Firm chooses whether they want to the default access for each user type to be SSO or Non-SSO. Default setting for SSO can be enabled by user type:

- Advisor
- Assistant
- Investor (called Client in AccountView)

These options allow for different default combinations of SSO use by user type. For example, by default Advisors could have SSO access to AccountView while Investors by default could log in directly to AccountView.

SSO Enabled for Individual Users

When a new user is created, the default SSO setting is based on their user type. However, administrator users can give or remove SSO access for individual users.

SSO Navigation Options set for the Firm

The following options can also be configured for SSO in order to manage navigation for different scenarios. ByAllAccounts can set these for your firm.

CONFIGURATION OPTION	DESCRIPTION	DEFAULT
Exit URL *	Redirect to this URL on application exit. This field is optional.	A default exit URL is provided
Timeout URL *	Redirect to this URL on session timeout.	Exit URL
Declined user agreement URL *	Redirect to this URL when a user declines the user agreement.	Exit URL
Keep parent alive URL *	URL to maintain parent session.	No default
Firm product URL	The login URL for the parent product. It is appended to application-generated emails that refer the user back to the application. For non-SSO users the AccountView URL is used. For SSO users, if this URL is set it will be used. If no URL is set, the option to include a product link in the email will not be provided.	No default
Firm product name	The name of the firm's product. It is used only to refer to the parent application when constructing email messages within AccountView and when "Firm product URL" is set.	No default

***** Dynamic URLs can be used to override URLs configured for the firm. See Dynamic URLs, page 5.

Dynamic URLs

Any or all of the dynamic URLs listed in the table below can be specified as input parameters when POSTing to the application. When a dynamic URL is specified, it overrides the corresponding setting described in SSO Navigation Options set for the Firm, page 4.

CONFIGURATION OPTION	CODE
Exit URL	<INPUT name='EXIT_URL' type=hidden value='http://mywebsite.com'>
Timeout URL *	<INPUT name='TIMEOUT_URL' type=hidden value='http://mywebsite.com'>
Declined user agreement URL *	<INPUT name='DECLINED_USER_AGREEMENT_URL' type=hidden value='http://mywebsite.com'>
Keep parent alive URL	<INPUT name='KEEP_PARENT_ALIVE_URL' type=hidden value='http://mywebsite.com'>

* If this URL is not specified but a dynamic Exit URL is, then the value for the Exit URL will be used.

Options and Settings

This SESSIONAUTHRQ operation used with SSO is only valid for users belonging to a firm for which AccountView has been licensed. Note that:

- A configuration option is available at the Firm level to set whether the user whose credentials are provided in LOGINRQ must be an Administrator. When the option is set to YES, the LOGINRQ must be made by an administrator. When it is set to NO, the LOGINRQ must be made by an investor, advisor, or assistant on their own behalf.
- SESSIONAUTHRQ identifies the user for whom the AccountView application session will be established. The target user can be identified using one of the following options:
    - USER_IDENT aggregate, which enables identification of a user by PERSON_ID, PERSON_FIRM_TAG1, or PERSON_LOGIN_NAME.

    or
    - LOGIN_NAME and LOGIN_PASSWORD for the target user.

For more details about the SESSIONAUTHRQ operation, refer to http://www.byallaccounts.net/Manuals/DataConnect/DataConnect_V4_Ultra_User_Guide.PDF for the details about this operation.

Additional AccountView Configuration of Interest to SSO

There are some additional configuration options for AccountView that are not specific to SSO but are of special interest when using SSO:

- **Require User Agreement** – by default, users logging into AccountView for the first time are presented with a user agreement that they must accept before they can enter the application proper. The user can accept the agreement and enter the application, decline the agreement and not enter the application, or cancel the agreement (defer accept/decline) and not enter the application. There is a customization option to disable this requirement. If it is disabled, there is no user agreement step. Note that the user agreement can be customized for your firm.
- **Use custom logo** – you can provide a custom logo (including an empty or blank logo) that AccountView will display instead of the ByAllAccounts logo. A custom logo must be Portable Network Graphics (PNG) format, and must fit in a space of 180w x 55h.

User Maintenance

The user types (roles) relevant to SSO are Advisor, Assistant, and Investor. (In AccountView, the Investor user is called a Client.) Default SSO access for each user type is set at the firm level, and administrators can make explicit settings per each user.

DataConnect API provides three operations to maintain users in the AccountView framework: Add User, Modify User, and Unsubscribe User. Refer to http://www.byallaccounts.net/Manuals/DataConnect/DataConnect_V4_Ultra_User_Guide.PDF for the details of these operations.

A sample "Add User" request from DataConnect is shown here.

```
<DATACONNECTRQ>
   <VERSION>VERSION4.0</VERSION>
   <LOGINRQ>…</LOGINRQ>
   <USERADDRQ>
      <PERSON>
         <ROLE>INVESTOR</ROLE>
         <FIRST_NAME>John</FIRST_NAME>
         <LAST_NAME>Smith</LAST_NAME>
         <EMAIL_ADDRESS>jsmith@aol.com</EMAIL_ADDRESS>
         <IS_SSO>1</IS_SSO>
      </PERSON>
      <LOGIN>
         <LOGIN_NAME>jsmith</LOGIN_NAME>
         <LOGIN_PW>asd98uvv3</LOGIN_PW>
      </LOGIN>
   </USERADDRQ>
</DATACONNECTRQ>
```

NOTE: If the Firm is set to require an Admin login for SESSIONAUTHRQ and the target user is enabled using USER_IDENT, then the LOGIN definition of LOGIN_NAME and LOGIN_PW are not required in the USERADDRQ. If the firm is set to *not* require an admin login, then the LOGIN may be defined here.

You can create Investors and Assistants in the AccountView application and designate their level of access to the application but AccountView will not allow either user type to see or modify a login or password for users that are configured to be SSO. Also, only administrators can set or change SSO access for users.

Access for SSO users with logins can be disabled two ways: explicitly through the DataConnect API or indirectly through multiple failed login attempts. If a user does not have a login, then the application launch access to that user has via SSO can be disabled by deleting them. For assistants and investors (not advisors), the user access can be disabled by setting their role to "no access".

The email addresses of SSO enabled user types will be editable within AccountView but the system will prevent them from being removed if a login exists for the user.

Note: Depending on how you intend to implement AccountView, you may also want to define the access level a user has (such as read only), and their relationship to an advisor (if they are an Investor or Assistant). Refer to http://www.byallaccounts.net/Manuals/DataConnect/DataConnect_V4_Ultra_User_Guide.PDF for the details of these operations.

Application Integration

The main events in the application life cycle for SSO AccountView are as follows:

1. Session Authentication Request – Parent application uses a request to authenticate and obtain session tokens. Session tokens are a Session ID and a Cross-Site Request Forgery (CSRF) Token.
2. Invoke AccountView using URL and session tokens provided by step 1 (SSL is required).
3. AccountView Accept/Decline User Agreement – This step is optional, and is handled by AccountView. For more information, refer to Additional AccountView Configuration of Interest to SSO, page 6.
4. AccountView application is presented in browser/frame.
5. If the "keep alive" URL is specified as a dynamic URL or configured for the firm, then that URL is invoked in iFrame in AccountView when there is activity in the session.
6. Application termination events to URLs that are specified as dynamic URLs or configured for the firm:
    a. Session expiration causes redirect to the Session Timeout URL.
    b. User invoked "Exit" causes redirect to the Exit URL.

Authenticate, Launch AccountView, and Session Timeout

The overall control flow for when SSO is used to access AccountView embedded in another application is as follows.

1. Parent application sends a DataConnect API "Session Authentication Request" (<SESSIONAUTHRQ>) request and receives back: 1) application URL, 2) session ID, and 3) CSRF token as shown in the following sample request and response.

    Sample SESSIONAUTHRQ providing USER_IDENT

```
<DATACONNECTRQ>
    <VERSION>VERSION4.0</VERSION>
    <LOGINRQ>
```

```
      <LOGIN_NAME>AdminUser</LOGIN_NAME>
      <LOGIN_PW>adminpassword123</LOGIN_PW>
   </LOGINRQ>
   <SESSIONAUTHRQ>
      <USER_IDENT>
         <PERSON_ID>62469</PERSON_ID>
      </USER_IDENT>
   </SESSIONAUTHRQ>
</DATACONNECTRQ>
```

Sample SESSIONAUTHRQ providing user LOGIN_NAME and LOGIN_PW:

```
<DATACONNECTRQ>
   <VERSION>VERSION4.0</VERSION>
   <LOGINRQ>
      <LOGIN_NAME>jsmith</LOGIN_NAME>
      <LOGIN_PW>mypassword123</LOGIN_PW>
   </LOGINRQ>
   <SESSIONAUTHRQ>
      <LOGIN_NAME>jsmith</LOGIN_NAME>
      <LOGIN_PW>mypassword123</LOGIN_PW>
   </SESSIONAUTHRQ>
</DATACONNECTRQ>
```

Sample SESSIONAUTHRS Response:

```
<DATACONNECTRS>
   <VERSION>VERSION4.0</VERSION>
   <LOGINRS>
      <STATUS>
         <ERRCODE>0</ERRCODE>
         <ERRMSG>Success</ERRMSG>
         </STATUS>
   </LOGINRS>
   <SESSIONAUTHRS>
      <STATUS>
         <ERRCODE>0</ERRCODE>
         <ERRMSG>Success</ERRMSG>
      </STATUS>

   <APPLICATION_URL>https://www.byallaccounts.net...</APPLICATION_URL>
         <SESSION_ID>B6B0948EF74DA6238B32E77669F6C9ED.s1a</SESSION_ID>
   <CSRF_TOKEN>BC53B58BC4C0E9FA6E2799D13183A68421EAC68E0503E41</CSRF_TO
KEN>
   </SESSIONAUTHRS>
</DATACONNECTRS>
```

SSO users may not log into the AccountView application through any AccountView login page. If a user attempts to log into AccountView directly, they will get an error message "User not configured for this login mode (user must log in through single sign-on).".

For more details about the session authenticate request (SESSIONAUTHRQ) and session expire (SESSIONEXPIRERQ) refer to http://www.byallaccounts.net/Manuals/DataConnect/DataConnect_V4_Ultra_User_Guide.PDF.

2. Parent application invokes AccountView by performing https POST to the application URL providing the one-time use Session ID and CSRF token (SESSION_ID and CSRF_TOKEN from step 1). This step could be performed by generating HTML like the following example. (This example is just a sample for illustration; any means of triggering the proper POST is sufficient.)

   Sample invoking AccountView:

```
<HTML>
<HEAD>
…
</HEAD>
<BODY onLoad='theForm.submit()'>
<FORM name='theForm' method='POST'
action='https://www.byallaccounts.net…'>
<INPUT type='hidden' name='SESSION_ID'
value='B6B0948EF74DA6238B32E77669F6C9ED.s1a'/>
<INPUT type='hidden' name='CSRF_TOKEN'
value='BC53B58BC4C0E9FA6E2799D13183A68421EAC68E0503E41'/>
</FORM>
</BODY>
</HTML>
```

Optionally, deep linking can be used to route users to a specific area of the application upon arrival via SSO within the Consumer UI version of AccountView. Specifically, deep linking makes it possible for SSO users to be launched directly into adding an account or editing a credential. It is also possible to exit back to the calling application when the credential connection succeeds or the user exits out of adding accounts.

Note that deep linking is *only* supported when launching AccountView as an Investor and 'Consumer UI' in enabled for the Investor's firm.

Deep linking can be invoked in the following ways and code samples are shown below.

- To launch into the Add Account wizard dialog (for "Read-Write" investors only):
  Use ON_LAUNCH_OPERATION with the "AddAccount" parameter. If the value of the ON_LAUNCH_OPERATION is invalid (for example, mistyped) then SSO will still proceed but the user will arrive at the AccountView landing page (they will not be routed).

- To launch into the Edit Credential dialog (for "Read-Write" and "Credential-Write" investors):
  Use `ON_LAUNCH_OPERATION` with the "EditCredential" parameter and use `ON_LAUNCH_OPERATION_PARAM1` with the numeric Account Credential ID value of an Account Credential within the user's scope. If the value in either of these parameters is invalid, or if one is submitted without the other, then SSO will still proceed but the user will arrive at the AccountView landing page (they will not be routed).

Note: The following examples are similar to the one above, but with the deep linking elements added inside the forms. Both examples use the optional EXIT_ON_COMPLETE, which is only available for deep linking. When the EXIT_ON_COMPLETE parameter is used the user exits from AccountView and back to the specified dynamic exit URL when the launched operation (EditCredential or AddAccount) is complete. If the dynamic exit URL were not provided, the user would exit back to the URL configured for the firm, or the default URL if none is configured.

Sample invoking AccountView with deep Linking for EditCredential, using the optional EXIT_ON_COMPLETE.

```
<HTML>
<HEAD>
…
</HEAD>
<BODY onLoad='theForm.submit()'>
<FORM name='theForm' method='POST'
action='https://www.byallaccounts.net…'>
<INPUT type='hidden' name='SESSION_ID'
value='B6B0948EF74DA6238B32E77669F6C9ED.s1a'/>
<INPUT type='hidden' name='CSRF_TOKEN'
value='BC53B58BC4C0E9FA6E2799D13183A68421EAC68E0503E41'/>
<INPUT type='hidden' name='ON_LAUNCH_OPERATION
value='EditCredential'/>
<INPUT type='hidden' name='ON_LAUNCH_OPERATION_PARAM1'
value='598204'/>
<INPUT type='hidden' name='EXIT_ON_COMPLETE' value=''/>
<INPUT name='EXIT_URL' type=hidden value='http://mywebsite.com'/>
</FORM>
</BODY>
</HTML>
```

Sample invoking AccountView with deep Linking for AddAccount, using the optional EXIT_ON_COMPLETE:

```
<HTML>
<HEAD>
…
</HEAD>
<BODY onLoad='theForm.submit()'>
```

```
   <FORM name='theForm' method='POST'
   action='https://www.byallaccounts.net…'>
   <INPUT type='hidden' name='SESSION_ID'
   value='B6B0948EF74DA6238B32E77669F6C9ED.s1a'/>
   <INPUT type='hidden' name='CSRF_TOKEN'
   value='BC53B58BC4C0E9FA6E2799D13183A68421EAC68E0503E41'/>
   <INPUT type='hidden' name='ON_LAUNCH_OPERATION value='AddAccount'/>
   <INPUT type='hidden' name='EXIT_ON_COMPLETE' value=''/>
   <INPUT name='EXIT_URL' type=hidden value='http://mywebsite.com'/>
   <INPUT name='DECLINED_USER_AGREEMENT_URL' type=hidden
value='http://mywebsiteother.com'/>
   </FORM>
   </BODY>
   </HTML>
```

3. Typically the session is eventually terminated either by AccountView session timeout, a user clicking the exit link in AccountView, or explicitly by the parent application performing a DataConnect API "Session Expiration Request" (<SESSIONEXPIRERQ>) requesting that session be terminated. This request is invoked using the LOGINRQ and providing the session ID and CSRF token that was obtained in the original Session Authentication Request.

   Sample Session Expire Request when user USER_IDENT was used:

```
<DATACONNECTRQ>
<VERSION>VERSION4.0</VERSION>
    <LOGINRQ>
      <LOGIN_NAME>AdminUser</LOGIN_NAME>
      <LOGIN_PW>adminpassword123</LOGIN_PW>
    </LOGINRQ>
    <SESSIONEXPIRERQ>
      <SESSION_ID>B6B0948EF74DA6238B32E77669F6C9ED.s1a</SESSION_ID>
      <CSRF_TOKEN>BC53B58BC4C0E9FA6E2799D13183A68421EAC68E0503E41
      </CSRF_TOKEN>
    </SESSIONEXPIRERQ>
</DATACONNECTRQ>
```

   Sample Session Expire Request when user LOGIN_NAME and LOGIN_PW were used:

```
<DATACONNECTRQ>
   <VERSION>VERSION4.0</VERSION>
   <LOGINRQ>
      <LOGIN_NAME>jsmith</LOGIN_NAME>
      <LOGIN_PW>mypassword123</LOGIN_PW>
   </LOGINRQ>
   <SESSIONEXPIRERQ>
      <LOGIN_NAME>jsmith</LOGIN_NAME>
      <SESSION_ID>B6B0948EF74DA6238B32E77669F6C9ED.s1a</SESSION_ID>


<CSRF_TOKEN>BC53B58BC4C0E9FA6E2799D13183A68421EAC68E0503E41</CSRF_TOKEN
>
   </SESSIONEXPIRERQ>
</DATACONNECTRQ>
```

Sample Session Expire Response:

```
<DATACONNECTRS>
   <VERSION>VERSION4.0</VERSION>
   <LOGINRS>
      <STATUS>
      <ERRCODE>0</ERRCODE>
      <ERRMSG>Success</ERRMSG>
    </STATUS>
   </LOGINRS>
   <SESSIONEXPIRERS>
      <STATUS>
         <ERRCODE>0</ERRCODE>
         <ERRMSG>Success</ERRMSG>
      </STATUS>
   </SESSIONEXPIRERS>
</DATACONNECTRS>
```

The exception to those termination methods is when EXIT_ON_COMPLETE is used with deep linking SSO in AccountView.

- When used with EditCredential, it exits to the application it launched from when the user selects Cancel or the credential is successful.
- When used with AddAccount, it exits to the application it launched from when the use selects X.

Parent Session Keep-Alive

Keep Parent Alive URL provides a URL (configuration or code) to be used to maintain the parent application's session. AccountView monitors application activity (server requests) every 60 seconds and if recent activity has occurred in the session then the "Keep Parent Alive URL" is invoked. AccountView implements this utilizing a hidden iFrame inside the AccountView frame.

VARIOUS IMPLEMENTATION MODELS

There are many implementation models. For example:

- Advisors access AccountView using SSO. Clients (investors) are not SSO, and can log into AccountView directly.
- There are only Client (Investor) users, and they access AccountView using SSO.
- There are only Advisor users, and they access AccountView using SSO.
- There are Advisors with SSO access, and Clients defined in AccountView who do not access AccountView at all.
- Advisors and Clients access AccountView using SSO.
- Some users of a given user type log in directly, and some use SSO.

ByAllAccounts can help you determine which model best suits your needs and how to implement it.