BlackBerry AtHoc

API Quick Start Guide

7.7

Contents

What is the BlackBerry AtHoc API?

BlackBerry AtHoc integrates with existing systems and investments to create a comprehensive end-to-end crisis communication system. A common integration use case is to sync all user contact details between an authoritative source and BlackBerry AtHoc. This integration is possible thanks to an extensible set of Web APIs. The APIs are designed to integrate a BlackBerry AtHoc system with other systems to make alerting more successful.

BlackBerry has incremented the Web-based API to include new REST endpoints. The new REST-based Web APIs are referred to as the BlackBerry AtHoc API V2.

This document describes how to get started using the BlackBerry AtHoc API V2. This document assumes that the reader is familiar with the BlackBerry AtHoc product, the end-user interaction, and the use of the management system. Familiarity with API V1 is helpful but not required.

This document also assumes that the reader has a customer relationship with BlackBerry or is working as a developer for a BlackBerry customer.

This document does not contain a full list of the available API endpoints. This list, including detailed definitions of each endpoint, is available in the interactive documentation installed with BlackBerry AtHoc and can be accessed at [*server-address*]/api/v2/docs.

Key differences between API Version 1 and API Version 2

The BlackBerry AtHoc API V2 makes establishing new integrations easier for developers. It follows the popular REST pattern with HTTP methods and JSON-formatted payloads. The authentication and authorization are OpenID Connect and OAuth 2.

The following table summarizes the differences between the API V1 and V2:

	API Version 1	API Version 2
Payload format	XML over HTTP	JSON over HTTP
Authorization	Inline Username and Password	OpenID Connect with OAuth2 JWT Access Tokens
Calling pattern	HTTP POST of Custom XML Payload Definitions	REST with HTTP methods GET, PUT, POST, DELETE
Scenarios covered	• User Sync • Distribution List • Sync Alert • Publishing Get Content	• User Sync • Distribution List Sync • Get Content • Accountability Officer
Unique identifier for users	MID (Mapping ID)	LOGIN_ID (Username)

System level guidelines

Throttling Limits

- There are throttling limits in place when calling the API. Try to optimize the workflow of calls to the API to achieve the maximum work within the number of allowed calls.
- Your calls may be blocked if they exceed the defined limits of your system or organization.

Dates and Times

- Dates and times will be in the organization time zone unless otherwise specified.

User Sync

- Use batches to update multiple users in one request instead of single-user updates in each call.
- Do not exceed more than 1000 users in each call as the SyncBy endpoints are in real-time. If you need to sync more than 1000 users in a call, use the background job CSV import endpoint.
- Don't store UserIDs inside your application. The identifier for the User is Username or Mapping ID.

Attributes

- The API GET method does not retrieve CommonName attributes when they contain the following special characters : + @ #
- Common Names may be optional in the user interface.

Set up your environment

The following topics are included in this section:

Create a user account with operator permissions

To use the BlackBerry AtHoc API, you must create a user account with operator permissions. The user must have the SDK User role and permissions to access the specific API module. For example, you must have the User Manager role to access the User Sync API .

Provision an application that can call the web API

To provision a new API integration with the BlackBerry AtHoc management system, you must have organization administrator, enterprise administrator, or system administrator permissions. You must have system administrator permissions to enable a provisioned application.

Note: The Client ID and Client Secret can only be used in the organization in which they are created. If the Client ID and Client Secret are created in the System Setup (3) organization, they can be used in any organization. If the Client ID and Client Secret are created in an Enterprise organization, they can be used in any of that Enterprise's sub organizations. If the Client ID provided does not follow these inheritance rules, a 400 (Bad request) error code is returned.

To provision a new API integration, complete the following steps:

1. Log in to the BlackBerry AtHoc management system as an organization administrator, enterprise administrator, or system administrator.
2. In the navigation bar, click the (**Settings**) icon.
3. In the System Setup section, click **API Applications**. The API Applications window opens.
4. Click **New**. The New API Application window opens.
5. Enter a name for the API integration.
6. (System administrators only) Next to Status, select the **Enabled** check box.
7. In the Authentication section, select a Grant Type. Password is the default. If you select Implicit, enter a redirect URI in the text box that appears.
8. Click **Save**. A Success message appears that includes the Client ID and Client Secret.
9. Take note of the displayed Client Secret. It is displayed only once and will need to be regenerated if lost.

Note: After you provision your application in the BlackBerry AtHoc management system, contact BlackBerry AtHoc Customer Support to have the application reviewed and enabled.

Set up an organization code in the BlackBerry AtHoc system

Complete the following task to set up an organization code for your specific organization in the BlackBerry AtHoc management system. This organization code is not propagated to PSS, so if you already have an organization code in PSS, use that one to complete this task.

This task is not required if an organization code for your organization has already been provided to you.

To set up an organization code for your organization in the BlackBerry AtHoc management system, complete the following steps:

1. Log in to the BlackBerry AtHoc management system as a System Administrator.
2. Switch to the specific organization.
3. Go to **Settings** > **General Settings**.
4. In the Organization Details section, enter the organization code. Do not use spaces.

Authentication

The BlackBerry AtHoc API V2 uses OAuth2-compliant authentication and authorization. To call the API, the client must first obtain an access token. These access tokens are per organization. You will need to request an access token for every individual organization that you are calling against. The authentication step returns an access token which will be used when calling the APIs.

The access token is only useful if the user has the SDK User role along with any other role required for accessing the specific API module. For example, the User Manager role is required for User Sync.

The parameter acr_values should contain the organization code in a key value pair with the Key=tenant (for example, acr_values=tenant:<OrgCode>) where <OrgCode> is the organization code of the organization for which you want to access the API.

Scope should be a space-delimited string of the resources that you want to access. If you also need long-term access to the API, you can request a Refresh Token with the offline_access scope. For example, openid profile athoc.iws.web.api offline_access.

Depending on your application and security requirements, you can obtain an access token from any of the following supported grant types:

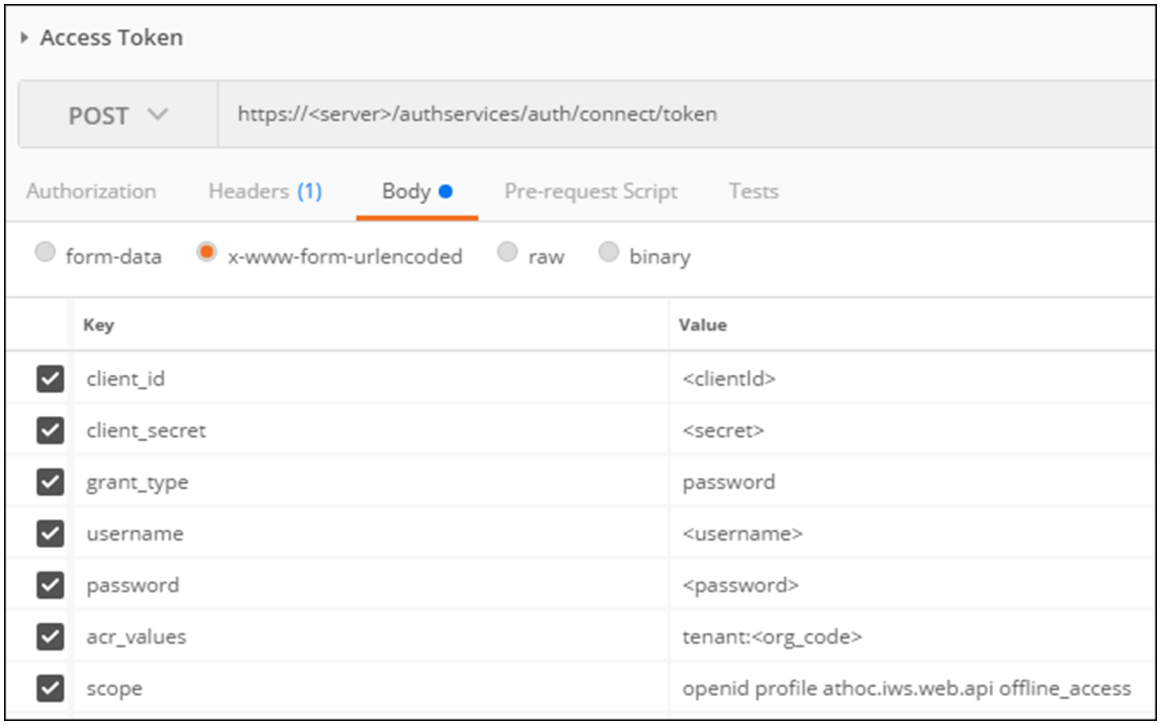
- Password Grant
- Authorization Code Grant
- Implicit Grant
- Change Org Grant
- Refresh Token Grant

For more information about these grants, see http://docs.identityserver.io/en/release/topics/grant_types.html.

Password grant

The resource owner password grant type allows requesting tokens on behalf of a user by sending the user's name and password to the token endpoint. This is "non-interactive" authentication and is generally not recommended. There may be instances in certain legacy or first-party integration scenarios where the password grant type is useful, but the general recommendation is to use an interactive flow like implicit or auth code for user authentication.

The following is a Postman request for an Access and a Refresh Token using the Password Grant:



HTTP URL: https://<server>/authservices/auth/connect/token

HTTP Verb: POST

Parameters: Form encoded body containing the following fields with values:

client_id, client_secret, grant_type, username, password, acr_values, scope

You should see the response with the Access and Refresh Tokens with an HTTP Status Code of 200 OK. You can now use the access_token for calling the API resources (and use the refresh token to retrieve a new access and refresh token without resubmitting the user credentials.)

Authorization code grant

The authorization code flow provides a way to retrieve tokens on a back-channel as opposed to the browser front-channel. The authorization code grant supports client authentication. The following is the recommended flow for native applications such as mobile apps and windows forms.

Step 1: An application requests an authorization code from the authentication server.

```
GET https://<server>/AuthServices/Auth/connect/authorize?
```

```
response_type=code
&client_id=<client_id>
&redirect_uri=<your_app_callback_url>
&scope=openid profile athoc.iws.web.api offline_access
&state=<guid>&acr_values=tenant:<org_code>
&code_challenge=<ClientGenerated_CodeChallenge>
&code_challenge_method=S256
```

state—An opaque value that the application adds to the initial request. During authentication, the application sends this parameter in the authorization request, and the authorization server returns this parameter unchanged in the response. This value must be used by the application to prevent cross-site request forgery (CSRF) attacks. This value can also be used by the application to restore the previous state of the application.

For more information about the state parameter, see:

https://auth0.com/docs/api-auth/tutorials/authorization-code-grant

https://auth0.com/docs/protocols/oauth2/oauth-state

code_challenge—The code_challenge is a Base64-URL-encoded string of the SHA256 hash of the code_verifier. Your application saves the code_verifier for later, and sends the code_challenge with the authorization request to your authorization server's authorization URL.

For more information about the code_challenge parameter, see

https://developer.okta.com/authentication-guide/implementing-authentication/auth-code-pkce

Step 2: The browser redirects the user to the login screen.

The browser redirects the user to the login screen. Upon entering login credentials, if the credentials are valid, the browser has the authentication code in the URL. If the credentials or organization code are invalid, the browser displays HTTP status code 400 "Invalid Request Error."

Step 3: The client requests the access_token based on the authentication code in Step 2.

```
POST https://<Server>/AuthServices/Auth/connect/token
{
     "grant_type":"authorization_code",
     "code":"<code>" //code returned in browser from 2nd Step
     "redirect_uri":"<your_app_callback_url>",
       "client_id":"<client_id>",
       "code_verifier":"<ClientGenerated_CodeVerifier>"
}
```

Step 4: The authentication server sends the access token response.

```
{
   "expires_in":3600,
   "token_type":"Bearer",
   "refresh_token":"ljiweoriwoer...",
   "access_token":"okljhgfdsighijuhdfgdkljhgdflkgjlkjdlfkgj..."
}
```

Implicit grant

The implicit grant type is optimized for browser-based applications. The implicit grant type is used for user authentication-only (both server-side and JavaScript applications), or for authentication and access token

requests (JavaScript applications). In the implicit flow, all tokens are transmitted through the browser. Advanced features such as refresh tokens are not allowed as the security of the tokens cannot be guaranteed.

The implicit grant flow has the following steps:

1. Your application directs the browser to the authentication server sign in page, where the user authenticates.
2. The authentication server redirects the browser to the specified redirect URI, and includes the access and ID tokens as a hash fragment in the URI.
3. Your application extracts the tokens from the URI.
4. Your application can now use these tokens to call the resource server (for example, an API) on behalf of the user.

Starting this flow is very similar to the authorization code flow except that the `response_type` is `token` or `id_token` instead of `code`.

Step 1: Your browser makes a request to authorize the endpoint of the authorization server.

```
GET https://<server>/AuthServices/Auth/connect/authorize?
response_type=token
&client_id=<client_id>
&redirect_uri=<your_app_callback_url>
&scope=openid profile athoc.iws.web.api offline_access
&state=<guid>
&acr_values=tenant:<org_code>
```

Step 2: The user logs in.

If the user does not have an existing session, this will open the authentication server sign-in page. After authenticating, or if the user has an existing session, the user arrives at the specified `redirect_uri` with a token as a hash fragment.

Step 3: The authentication server sends a redirect response.

```
https://localhost:8080/#
access_token=eyJhkjughfs...
&token_type=Bearer&expires_in=3600
&scope=openid
&state=<state>
```

Your application must now extract the tokens from the URI and store them.

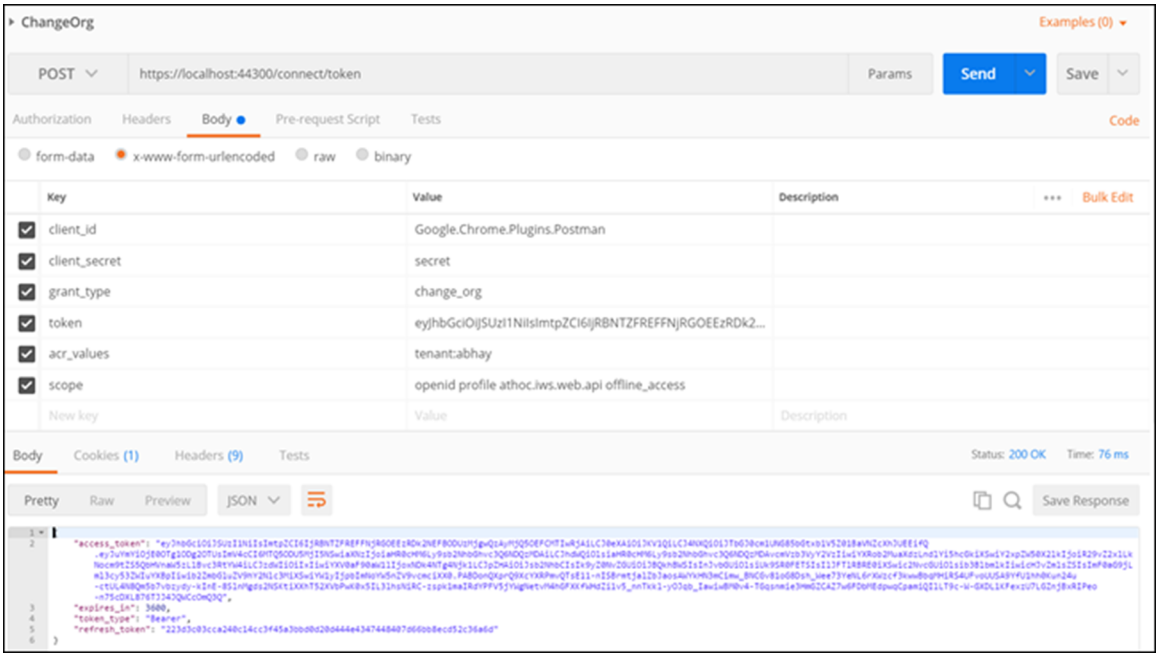
Change organization grant

The change organization grant has been specifically designed for external applications that allow their users to switch between multiple organizations.

Once the application has received an access token based on user credentials, the same access token can be used as the user's identity to get new access tokens for organizations that the user has access to.

The response of this call is a new Access (and Refresh) token based on the user's permissions within the new organization. If the user is not authorized in this organization, an error is returned.

The is a Postman request and response for the change_org grant:



Note: The change organization grant type must be requested from BlackBerry AtHoc Customer Support for the provisioned application. The change organization grant is an add-on grant that can be added to any provisioned application using the implicit, authentication code, and password grants.

URL—https://<server>/AuthServices/Auth/connect/token

HTTP Verb—POST

Parameters:

- **client_id**—<client_id>
- **client_secret**—<secret>
- **grant_type**—change_org
- **scope**—openid profile athoc.iws.web.api
- **acr_values**—tenant:<org_code>
- **token**—<current valid access token>

API Error Response—If the user is not authorized for the given tenant (organization), the following error code is returned:

```
401: Unauthorized
```

Refresh tokens

Refresh tokens enable granting long-term access to APIs. You should keep the lifetime of access tokens as short as possible. However, you want to avoid forcing the user to perform repeated front-channel round-trips to the authentication server to request new access tokens.

Refresh tokens allow new access tokens to be requested without user interaction. Every time the client refreshes a token, it needs to make an authenticated back-channel call to the authentication server. This call allows verifying if the refresh token is still valid, or has been revoked.

Refresh tokens are supported in the authorization code and resource owner password flows. To request a refresh token, the client must include the offline_access scope in the token request and must be authorized for that scope.

To obtain a new access token from a refresh token, send the following information:

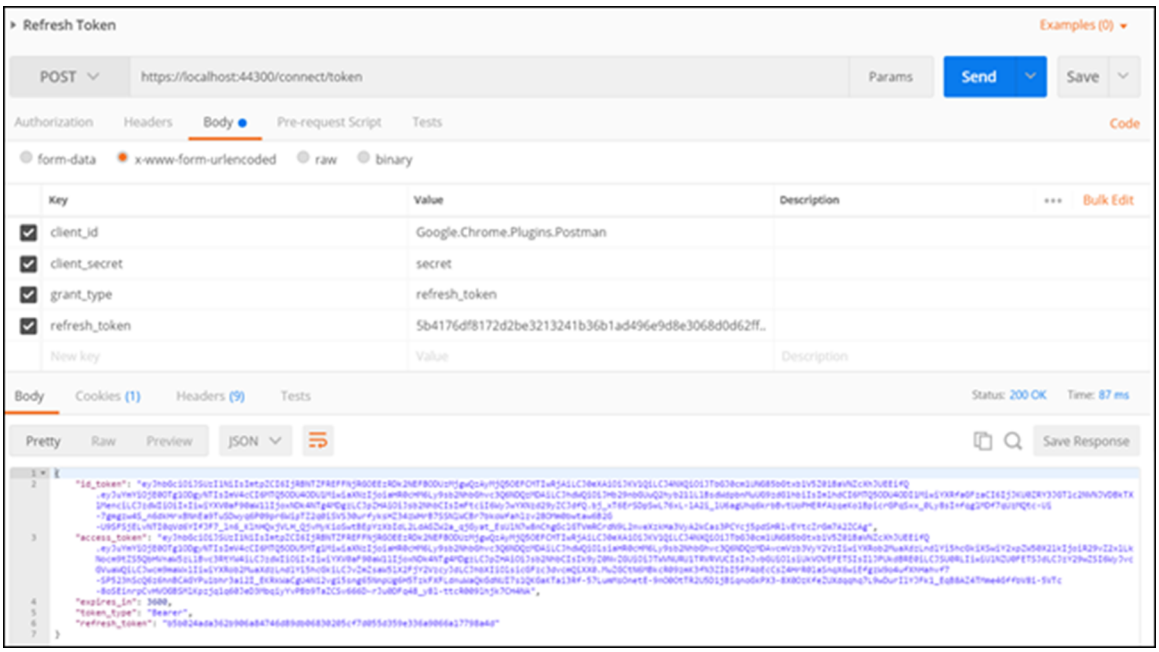
URL—https://<server>/AuthServices/Auth/connect/token

HTTP Verb—POST

Parameters:

- **client_id**—<client_id>
- **client_secret**—<secret>
- **grant_type**—refresh_token
- **refresh_token**—<current valid refresh token>

The following is a Postman request and response for the refresh_token grant:



Authentication errors

This topic describes the error codes you may see when authentication of an API client fails. When authentication fails because the client is disabled or not present, a 400 error code is displayed. The following table explains the errors:

Error code	Cause	Action to correct
invalid client	The client name does not exist or is incorrect, or the client secret is invalid.	Check that the client is provisioned in the API application page and that it is in the Enabled state. Reset the client secret and use the new one.

Error code	Cause	Action to correct
unsupported_grant_type	The grant type is invalid.	The Grant type cannot be empty. Check that the Grant type is populated with one of the following supported grant type values: Implicit, authorization_code, Password, Change_org.
invalid_grant	The username or password is invalid, or the tenant code is invalid.	Make sure that the user credentials are valid and the correct organization code is passed.
invalid_scope	The scope is invalid.	The Scope cannot be empty. The mandatory Scope value is **openid profile athoc.iws.web.api. offline_access**. The offline_access scope value is an optional value that is required only when requesting a refresh token.

If you received an error, verify the following items:

1. Your client is properly provisioned and your client_id and secret are valid.
2. Your client has the password grant configured and allowed.
3. Your username and password fields are correct.
4. The user exists in the organization defined in the acr_values tenant:<org_code>.
5. The operator account is not locked.

Reset the client secret

If you need to reset the client secret for your API integration, complete the following steps:

1. Log in to the BlackBerry AtHoc management system.
2. In the navigation bar, click the (**Settings**) icon.
3. In the System Setup section, click **API Applications**. The API Applications window opens.
4. (Optional) Enter a name in the search box to filter the list of applications.
5. (Optional) Select **Enabled Applications** or **Disabled Applications** from the All Applications list to filter the list of applications.
6. Click the application you want to modify.
7. Click **Reset Client Secret**. A confirmation window opens.

Note: Any existing calls to the selected API with the existing client secret will be blocked when you reset the client secret.Any existing calls to the selected API with the existing client secret will be blocked when you reset the client secret.

8. Click **Continue**. You are returned to the API application window. The new client secret is displayed.
9. Take note of the displayed client secret.

Call the API

You can call the BlackBerry AtHoc web API through a URL in your browser.

To access the BlackBerry AtHoc API, complete the following steps:

1. Log in to the BlackBerry AtHoc management system as an SDK User.
2. In the address bar of your browser, replace "athoc-iws" with "api/v2/docs" . The web API page opens.
3. Enter your organization ID in the Authorize field.
4. Click **Authorize**. The Available authorizations window opens.
5. Select the scope option.
6. Click **Authorize**. You are directed to a login page.
7. Enter your username and password.
8. Click **Log In**.

Resolve response codes

The following table lists response codes and how to resolve them:

Response code	Description	Steps to resolve
400	Bad request	Check that the payload and its format are correct. Check for validation errors and take necessary actions to correct the payload.
401	Unauthorized	Make sure that the access token is present, correct, and not expired.
403	Forbidden	• The operator does not have sufficient operator permissions to execute the request. Log in to the BlackBerry AtHoc management system to modify the roles for the operator. • The password used is changed or expired. Generate a new authentication token.
404	Not Found	The resources you are trying to find are not present in the system. Pass valid parameters.
429	Too Many Requests	There is a restriction on the number of API calls allowed. Try the service again. If you continue to see this response code, contact your system administrator and request that the API throttling limit be increased to allow the client to perform more requests.
500	Internal Server Error	Report this error to BlackBerry AtHoc customer support at athocsupport@blackberry.com.

Response code	Description	Steps to resolve
503	Service Unavailable	The server is currently unable to handle the request due to a temporary overload or scheduled maintenance. Wait and try again.

Access the V1 and V2 web APIs in parallel

The BlackBerry AtHoc Web API V1 uses an XML payload over HTTP. The user sends login credentials as part of the authentication request to obtain a sessionId which can then be used for all other V1 API calls.

In the following example, authentication is performed, and then the sessionId is used for a publishing scenario.

Authenticating in the Web API V1

Request:

```
<AtHocSdk>
    <client>vpsID</client>
    <validation>
        <username>OperatorUserName</username>
        <password>OperatorPassword</password>
          <udid>OperatorDeviceIdentifier</udid>
    </validation>
 <payload type="LOGIN"/>
</AtHocSdk>
```

Response:

```
<AtHocSdkResponse>
 <payload type="LOGIN" trackId="">
 <ok>
     <systemDate>2012-09-26T20:56:24.0355987Z</systemDate>
     <responsePayload>
       <sessionId>6727aea1-b862-4370-a5df-9ea98e90d00c</sessionId>
     </responsePayload>
     <warnings />
 </ok>
 </payload>
</AtHocSdkResponse>
```

Publish scenario using SessionId

Request:

```
<AtHocSdk>
 <client>vpsID</client>
   <validation>
     <accessToken>6727aea1-b862-4370-a5df-9ea98e90d00c</accessToken>
   </validation>
  <payload type="INFOCASTING">
        <infocasting type='ALERTSCENARIO'>
          <alertData>
          <scenario ID="8391">
             <readyForPublish>Y</readyForPublish>
          </scenario>
         </alertData>
       </infocasting>
  </payload>
</AtHocSdk>
```

Authenticating in the Web API V2

All successful authentication requests respond with an access_token. The same access token can also be used instead of sessionId for all web API V1 requests. This enables you avoid requiring web API V2 users to authenticate again for publishing scenarios.

Publish Scenario using Access Token

Request:

```
<AtHocSdk>
 <client>vpsID</client>
   <validation>
<accessToken>eyJhbGciOiJSUzI1NiIsImtpZCI6IjRBNTZFREFFNjRGOEEzRDk2NEFBODUzMjgwQzAyMjQ5OEFCMTI
KHHPpc0BwnOQANSc36hS9J9zsorVhIlGBFm5Vj_DX9OV_Sn8m8ERuT9rjIOOdnQ76omFAWprDr9CUuuPKchCbYShDOYv
iaGpX19B31cM7msg1g5Ui5wEdpgYEgLKZ87qBLfAd1IB-
uMeCh14236BVfZGYH1IacqwCyob_MPVqSBzm8Ygcg-
yImeumxKL_JU1EwQi7ze1BHXEteASs0WRMVKMQgE5yZaxqXdOuJAhPYqyoaROw4nF3DoOUwt0qaPmqmsNtTQ</
accessToken>
   </validation>
  <payload type="INFOCASTING">
        <infocasting type='ALERTSCENARIO'>
         <alertData>
           <scenario id="8391">
              <readyForPublish>Y</readyForPublish>
           </scenario>
         </alertData>
        </infocasting>
  </payload>
</AtHocSdk>
```

Code samples

BlackBerry AtHoc has created a set of code written in C# that can be used as a template to call the APIs. This code handles authentication and allows you to use the .Net methods and functions instead of calling the REST endpoints directly.

Contact your implementation engineer or BlackBerry AtHoc Customer Support for the .zip file that contains these code files. You can also access code samples by clicking the **API Development Kit** link at [*server-address*]/api/v2/docs/apiguide.html.

BlackBerry AtHoc customer portal

BlackBerry AtHoc customers can obtain more information about BlackBerry AtHoc products or get answers to questions about their BlackBerry AtHoc systems through the Customer Portal:

https://support.athoc.com/customer-support-portal.html

The BlackBerry AtHoc Customer Portal also provides support via computer-based training, Operator checklists, best practice resources, reference manuals, and users guides.

Legal notices

Trademarks

Patents

BlackBerry Solution License Agreement

Contact Information

BlackBerry AtHoc

311 Fairchild Drive

Mountain View, CA 94043

Tel: 1-650-685-3000

Email: athocsupport@blackberry.com

Web: http://www.athoc.com