REST API Application Development

SAP Mobile Platform 3.0

Contents

REST API Application Development

The SAP® Mobile Platform REST Services API enables standard HTTP client applications running in any platform to access SAP Mobile Platform REST services.

SAP Mobile Platform Server provides REST services so that applications can be built as any standard HTTP application to leverage SAP Mobile Platform Server for security and push features among others.

Build client applications using third-party (JavaScript framework and helper libraries) developer tools, native client libraries, or the libraries provided by SAP Mobile Platform OData SDK (Only iOS and Android platforms). SAP Mobile Platform provides management and monitoring of the applications, and support for native push notification such as Apple Push Notification service (APNS), BlackBerry Internet/Enterprise Service (BIS/BES), Google Cloud Messaging (GCM), Windows Notification Service (WNS), or Microsoft Push Notification Service (MPNS).

The client application should first register an application connection with device information, such as device type, password capability, and so on. After registering, the application can retrieve and update the application connection settings through the REST API. You can enable or disable the push notification only after registering.

Note: An application connection can be deleted through the REST API, as long as it is not in use. Any data that is stored by application in the custom string of the application connection properties is lost.

To initialize a client application, the application can download resources (such as metadata files, multimedia files, and so on.), using the resource bundles service.

After downloading resources, the application can access OData-compatible data sources through the proxy service, and receive native push notifications triggered by the gateway if push properties are configured and enabled.

This development approach supports:

- Registration (creating an application connection)
- Authentication
- Native push notification
- Configuration

For more information on supported REST API application development environment versions in SAP Mobile Platform 3.0, see *http://service.sap.com/pam*.

Configuring the Application in the Management Cockpit

Configure the application settings in the Management Cockpit. These settings enable you to monitor and manage your applications.

Prerequisites

- Make sure SAP Mobile Platform Server is installed.
- Make sure the server is started.
- Launch the Management Cockpit.

Task

Defining Applications

Create a new native, hybrid, or Agentry application definition. The definition enables you to manage the application using Management Cockpit.

1. In Management Cockpit, select **Applications**, and click **New**.
2. In the New Application window, enter:

Field	Value
ID	Unique identifier for the application in reverse domain notation. This is the application or bundle identifier that is defined or generated during application development. Reverse domain notation is the practice of reversing a registered domain name; for example, the reverse domain notation for an object in `sap.com` might be `MyApp.sap.com`). The application identifier: • Must be unique. • Must start with an alphabetic character. • Can contain only alphanumeric characters, underscores (_), and periods (.). • Must not include spaces. • Can be up to 64 characters long. Formatting guidelines: • SAP recommends that application IDs contain a minimum of two dots ("."). For example, this ID is valid: `com.sap.mobile.app1`. • Application IDs should not start with a dot ("."). For example, this ID is invalid: `.com.sap.mobile.app1`. • Application IDs should not include two consecutive dots ("."). For example, this ID is invalid: `com..sap.mobile.app1`.
Name	Application name. The name: • Can contain only alphanumeric characters, spaces, underscores (_), and periods (.). • Can be up to 80 characters long.
Vendor	(Optional) Vendor who developed the application. The vendor name: • Can contain only alphanumeric characters, spaces, underscores (_), and periods (.). • Can be up to 255 characters long.
Type	Application type. • Native – native iOS and Android applications. • Hybrid – container-based applications, such as Kapsel. • Agentry – metadata-driven application. **Note:** You can configure only one Agentry application per SAP Mobile Platform Server. Once configured, Agentry no longer appears as an option.

Field	Value
Description	(Optional) Short description of the application. The description: • Can contain alphanumeric characters. • Can contain most special characters, except for percent signs (%) and ampersands (&). • Can be up to 255 characters long.

3. Click **Save**. Application-related tabs appear, such as Back End, Authentication, Push, and so forth. You are ready to configure the application, based on the application type.

   Note: These tabs appear in Management Cockpit only after you define or select an application. The tabs used differ by application type.

Defining Back-end Connections for Native and Hybrid Apps

Define back-end connections for the selected native or hybrid application.

1. From Management Cockpit, select **Applications** > **Back End**, and enter values for the selected application.

Field	Value
Connection Name **Note:** Appears only when adding a connection under Back-End Connections.	Identifies the back-end connection by name. The connection name: • Must be unique. • Must start with an alphabetic character. • Can contain only alphanumeric characters, underscores (_), and periods (.). • Must not include spaces.

Endpoint	The back-end connection URL, or service document URL the application uses to access business data on the back-end system or service. The service document URL is the document destination you assigned to the service in Gateway Management Cockpit. Typical format:
	`http://`*`host`*`:`*`port`*`/gateway/odata/`*`namespace`*`/`*`Connection_or_ServiceName`*`...`
	
	Examples:
	`http://testapp:65908/help/abc/app1/opg/sdata/TESTFLIGHT`
	
	`http://vmw3815.wdf.sap.corp:50009/sap/opu/odata/rwfnd/RMTSAMPLE`

Field	Value
Use System Proxy	(Optional) Whether to use system proxy settings in the SAP Mobile Platform `props.ini` file to access the back-end system. This setting is typically disabled, because most back-end systems can be accessed on the intranet without a proxy. The setting should only be enabled in unusual cases, where proxy settings are needed to access a remote back-end system outside of the network. When enabled, this particular connection is routed via the settings in `props.ini` file.
Rewrite URL	(Optional) Whether to mask the back-end URL with the equivalent SAP Mobile Platform Server URL. This is necessary to ensure the client makes all requests via SAP Mobile Platform Server and directly to the back end. Rewriting the URL also ensures that client applications need not do any additional steps to make requests to the back end via SAP Mobile Platform Server. If enabled, the back-end URL is rewritten with the SAP Mobile Platform Server URL. By default, the property is enabled.
Allow anonymous connections	(Optional) Whether to enable anonymous access. This means the application user can access the application without entering a user name and password. However, the back-end system still requires log on credentials to access the data, whether it is a read-only user, or a back-end user with specific roles. If enabled, enter the log on credential values used to access the back-end system: • **User name** – supply the user name for the back-end system. • **Password** – supply the password for the back-end system. If disabled, you do not need to provide these credentials. By default, the property is disabled. **Note:** If you use Allow Anonymous Connections for a native OData application, do not assign the No Authentication Challenge security profile to the application, or the anonymous OData requests will not be sent (Status code: 401 is reported).
Maximum Connections	The number of back-end connections that are available for connection pooling for this application. The larger the pool, the larger the number of possible parallel connections to this specific connection. The default is 500 connections. Factors to consider when resetting this property: • The expected number of concurrent users of the application. • The load acceptable to the back-end system. • The load that the underlying hardware and network can handle. **Note:** The maximum connections can be increased only if SAP Mobile Platform Server hardware can support the additional parallel connections, and if the underlying hardware and network infrastructure can handle it.

Field	Value
Certificate Alias	The name under which the administrator has imported the certificate key-pair in the `smp_keystore` file. The alias must be set when the back-end URL is `https://`, and the back-end server requires mutual authentication. There are conditions when https is used but the server does not require a client certificate. This certificate alias is required when the back end requires mutual SSL connectivity. Use the alias of a certificate stored in the SAP Mobile Platform Server keystore. SAP recommends that the CN value of the generated certificate be the fully qualified domain name of SAP Mobile Platform Server.

2. (Optional) Under Back-end Connections, view additional connections, or add new connections.
   a) Click **New**, to add additional back-end connections in the server.
   b) Enter values for the new back-end connection, using the values shown above.
   c) Click **Save**. The new connection is added to the list.

      Administrator maintains a list of server-level back-end connections (it includes all the connections in the SAP Mobile Platform Server) and application specific back-end connections. Application specific back-end connections are the connections enabled for an application. Users registered to an application can access only these back-end connections. Request-response to a back end connection that is not enabled for an application is not allowed (throws 403, "Forbidden" error).

      By default, these additional back-end connections are enabled for an application.

      Back-end connection is displayed in the list.
3. Select the **Application-specific Connections** from the drop-down to show the back-end connections that are enabled for a particular application.

   You can view the **Server-level Connections** and enable the connections for this application using the checkbox.

   Note: Multiple back ends can be authenticated using various options of authenticating requests available in security configuration.

Defining Application Authentication

Assign a security profile to the selected application. The security profile defines parameters for how the server authenticates the user during onboarding, and request-response interactions.

Prerequisites

You must configure security profiles for application authentication before you can complete this step.

Task

1. From Management Cockpit, select **Applications** > **Authentication**.
2. Click **Existing Profile**.

   Note: You can also create a new profile.

3. In Name, select a security profile name from the list. The name appears under Security Profile Properties, and one or more security providers appear under Authentication Providers.
4. Under Security Profile Properties, enter values.

Field	Value
Name	A unique name for the application authentication profile.
Check Impersonation	(Optional) In token-based authentication, whether to allow authentication to succeed when the user name presented cannot be matched against any of the user names validated in the login modules. To prevent the user authentication from succeeding in this scenario, the property is enabled by default.

5. Under Authentication Providers, you can select a security profile URL to view its settings. To change its settings, you must modify it through **Settings** > **Security Profiles**.

Configuring Push on SAP Mobile Platform Server

You must explicitly register the application connection using the Management Cockpit.

1. Start the Management Cockpit.
2. Select **Applications**, and click **New**.
3. In the **New Application** window, enter values.

Field	Value
ID	Unique identifier for the application in reverse domain notation.
Name	Application name.
Vendor	(Optional) Vendor who developed the application.
Version	Application version. Currently, only version 1.0 is supported.
Type	Application type. • Native – native iOS and Android applications. • Hybrid – container-based applications, such as Kapsel. • Agentry – metadata-driven applications, such as Agentry. Application configuration options differ depending on your selection.

Field	Value
Description	(Optional) Short description of the application.

4. Click the **Backend** tab and configure the endpoint information.
5. Click the **Push** tab to configure the push settings.

   For Android GCM, see *Android Push Notifications*.

   For Apple APNS, see *Apple Push Notifications*
6. In the settings for your device, enable the app to receive push notifications.

Android Push Notifications

Configure Android push notifications for the selected application, to enable client applications to receive Google Cloud Messaging (GCM) notifications.

1. From Management Cockpit, select **Applications** > **Push**.
2. Under Android, enter the access key for API key. This is the access key you obtained for your Google API project (*http://developer.android.com/google/gcm/gs.html*).
3. Enter a value for Sender ID. This is the project identifier.
4. (Optional) Configure push notifications for each device type supported.

Apple Push Notifications

Configure Apple Push Notifications for the selected application, to enable client applications to receive APNS notifications.

1. From Management Cockpit, select **Applications** > **Push**.
2. Under Apple, select **APNS endpoint**. "None" is the default endpoint value for all the applications.
3. Select **Sandbox** to configure APNS in a development and testing environment, or **Production** to configure APNS in a production environment.
   a) Click **Browse** to navigate to the certificate file.
   b) Select the file, and click **Open**.
   c) Enter a valid password.

   Note: The default URL is for a production environment; for a development and testing environment, change the URL to gateway.sandbox.push.apple.com.
4. (Optional) Configure push notifications for each device type supported.

Developing the Application

Develop your HTTP client application to use the REST Services API to access SAP Mobile Platform REST services.

Authentication Requests

For all requests that require authentication, send the authentication information to SAP Mobile Platform. The credentials depend on the type of security configuration. Credentials should be provided in the header.

- Basic authentication
  The user name and password should be valid for the specified authentication URL.
  - HTTP Header Name: Authorization
  - HTTP Header Value: Basic *<base64 encoded form of username:password>*
- SAP SSO authentication
  The user name and password should be valid for the specified ticket-issuing system (TIS) URL.
  - HTTP Header Name: Authorization
  - HTTP Header Value: Basic *<base64 encoded form of username:password>*
- SiteMinder SSO authentication (Client acquires SSO Token)
  - HTTP Header Name: *<value provided for 'Client HTTP Values To Send' in the security configuration>*
  - HTTP Header Value: actual `SMSESSION` token
- SiteMinder SSO authentication (SAP Mobile Platform acquires SSO token)
  The user name and password should be valid for the specified ticket-issuing system (SiteMinder server) URL.
  - HTTP Header Name: Authorization
  - HTTP Header Value: Basic *<base64 encoded form of username:password>*
- Certificate authentication
  Prepare a client certificate and get it signed by certification authority (CA) certificate of the server. The client certificate should be trusted by SAP gateway or any other EIS. The certificate then be used to register the client and perform the request-response with the server.

Creating an Application Connection

You must explicitly register the application connection using SAP Mobile Platform.

You can specify customized application properties for the client with the request. Provide the application connection ID, `X-SMP-APPCID`, using an explicit request header or a cookie. If the value is missing, SAP Mobile Platform generates a universally unique ID (UUID), which is communicated to the device through the response cookie `X-SMP-APPCID`.

Create an anonymous or authenticated application connection by issuing a POST request to this URL, including the application connection properties:

```
http[s]://<host:port>/[public/]/odata/applications/{latest|v1}/
{appid}/Connections
```

The URL contains these components:

- **host –** the host is defined by host name and should match with the domain registered with SAP Mobile Platform. If the requested domain name does not match, default domain is used..
- **port –** the port for listening to OData-based requests. By default the port number is 8080.
- **public –** if included, an anonymous connection is allowed.
- **odata/applications/ –** refers to the OData services associated with the application resources.
- **{latest|v1} –** version of the service document.
- **appid –** name of the application.
- **Connections –** name of the OData collection.

Application connection properties are optional. You can create an application connection without including any application properties.

`DeviceType` is an application connection property that you may set. Valid values for `DeviceType` are:

- **Android –** Android devices.
- **iPhone –** Apple iPhone.
- **iPad –** Apple iPad.
- **iPod –** Apple iPod.
- **iOS –** iOS devices.
- **Blackberry –** Blackberry devices.
- **Windows –** includes desktop or servers with Windows OS, such as Windows XP, Windows Vista, Windows 7, and Windows Server series OS.
- **WinPhone8 –** includes Windows mobile.
- **Windows8 –** includes Windows desktop version.

Specifying any other value for `DeviceType` returns a value "Unknown" in the `DeviceType` column.

Example of creating an application connection

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
    <category term="applications.Connection" scheme="http://
schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
    <content type="application/xml">
        <m:properties>
           <d:AndroidGcmRegistrationId>398123745023</d
AndroidGcmRegistrationId>
        </m:properties>
    </content>
</entry>
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xml:base="https://<smp base URL>/odata/applications/latest/
e2eTest/">

<id>https://<application URL>/odata/applications/latest/e2eTest/
Connections('4891dd0f-0735-47cc-a599-76bf8a16d457')</id>
<title type="text" />
<updated>2012-10-19T09:05:25Z</updated>
<author>
<name />
</author>
<link rel="edit" title="Connection"
href="Connections('4891dd0f-0735-47cc-a599-76bf8a16d457')" />
<category term="applications.Connection" scheme="http://
schemas.microsoft.com/ado/2007/08/dataservices/scheme" />
<content type="application/xml">
 <m:properties>
    <d:ETag>2012-10-19 14:35:24.0</d:ETag>
    <d:ApplicationConnectionId>4891dd0f-0735-47cc-
a599-76bf8a16d457</d:ApplicationConnectionId>
    <d:AndroidGcmPushEnabled m:type="Edm.Boolean">false</
d:AndroidGcmPushEnabled>
    <d:AndroidGcmRegistrationId>398123745023</d
AndroidGcmRegistrationId>
    <d:AndroidGcmSenderId />
    <d:ApnsPushEnable m:type="Edm.Boolean">false</d:ApnsPushEnable>
    <d:ApnsDeviceToken />
    <d:ApplicationVersion>1.0</d:ApplicationVersion>
    <d:BlackberryPushEnabled m:type="Edm.Boolean">false</
d:BlackberryPushEnabled>
    <d:BlackberryDevicePin m:null="true" />
    <d:BlackberryBESListenerPort m:type="Edm.Int32">0</
d:BlackberryBESListenerPort>
    <d:BlackberryPushAppID m:null="true" />
    <d:BlackberryPushBaseURL m:null="true" />
    <d:BlackberryPushListenerPort m:type="Edm.Int32">0</
d:BlackberryPushListenerPort>
    <d:BlackberryListenerType m:type="Edm.Int32">0</
d:BlackberryListenerType>
    <d:CustomizationBundleId />
    <d:CustomCustom1 />
    <d:CustomCustom2 />
    <d:CustomCustom3 />
    <d:CustomCustom4 />
    <d:DeviceModel m:null="true" />
    <d:DeviceType>Unknown</d:DeviceType>
    <d:DeviceSubType m:null="true" />
    <d:DevicePhoneNumber m:null="true" />
    <d:DeviceIMSI m:null="true" />
    <d:PasswordPolicyEnabled m:type="Edm.Boolean">false</
d:PasswordPolicyEnabled>
```

```
    <d:PasswordPolicyDefaultPasswordAllowed
m:type="Edm.Boolean">false</d:PasswordPolicyDefaultPasswordAllowed>
    <d:PasswordPolicyMinLength m:type="Edm.Int32">0</
d:PasswordPolicyMinLength>
    <d:PasswordPolicyDigitRequired m:type="Edm.Boolean">false</
d:PasswordPolicyDigitRequired>
    <d:PasswordPolicyUpperRequired m:type="Edm.Boolean">false</
d:PasswordPolicyUpperRequired>
    <d:PasswordPolicyLowerRequired m:type="Edm.Boolean">false</
d:PasswordPolicyLowerRequired>
    <d:PasswordPolicySpecialRequired m:type="Edm.Boolean">false</
d:PasswordPolicySpecialRequired>
    <d:PasswordPolicyExpiresInNDays m:type="Edm.Int32">0</
d:PasswordPolicyExpiresInNDays>
    <d:PasswordPolicyMinUniqueChars m:type="Edm.Int32">0</
d:PasswordPolicyMinUniqueChars>
    <d:PasswordPolicyLockTimeout m:type="Edm.Int32">0</
d:PasswordPolicyLockTimeout>
    <d:PasswordPolicyRetryLimit m:type="Edm.Int32">0</
d:PasswordPolicyRetryLimit>
    <d:ProxyApplicationEndpoint>http://<backend URL></
d:ProxyApplicationEndpoint>
    <d:ProxyPushEndpoint>http[s]://<host:port>/Push</
d:ProxyPushEndpoint>
    <d:MpnsChannelURI m:null="true" />
    <d:WnsChannelURI m:null="true" />
</m:properties>
</content>
</entry>
```

CORS Support

Cross-domain HTTP requests are requests for resources from a different domain than the domain of the resource making the request. Cross-Origin Resource Sharing (CORS) mechanism provides a way for web servers to support cross-site access controls, which enable secure cross-site data transfers.

Managing Application Settings

Application settings describe the application connection details such as application ID, security configuration, and customization resource.

Getting Application Settings

You can retrieve application connection settings for the device application instance by issuing the GET method.

You can retrieve application settings by either explicitly specifying the application connection ID, or by having the application connection ID determined from the call context (that is, from either the X-SMP-APPCID cookie or X-SMP-APPCID HTTP header, if specified). On the first call, you can simplify your client application code by having the application connection ID determined from the call context, since you have not yet received an application connection ID.

If you supply an application connection ID, perform an HTTP GET request at:

```
http[s]://<host:port>/[public/]odata/applications/{latest|v1}/
{appid}/Connections('{appcid}')
```

Response

```
<?xml version='1.0' encoding='utf-8'?>
<entry xmlns="http://www.w3.org/2005/Atom"
       xmlns:m="http://schemas.microsoft.com/ado/2007/08/
dataservices/metadata"
       xmlns:d="http://schemas.microsoft.com/ado/2007/08/
dataservices"
       xml:base="https://<smp base URL>/odata/applications/v1/
e2eTest/">
  <id>http://https://mobilesmpdev.netweaver.ondemand.com/smp/odata/
applications/v1/e2eTest/
Connections('c9d8a9da-9f36-4ae5-9da5-37d6d90483b5')</id>
  <title type="text" />
  <updated>2012-06-28T09:55:48Z</updated>
  <author><name /></author>
  <link rel="edit" title="Connections"
href="Connections('c9d8a9da-9f36-4ae5-9da5-37d6d90483b5')" />
  <category term="applications.Connection" scheme="http://
schemas.microsoft.com/ado/2007/08/dataservices/scheme" />
  <content type="application/xml">
    <m:properties>
      <d:ETag m:type="Edm.DateTime">2012-06-28T17:55:47.685</d:ETag>

<d:ApplicationConnectionId>c9d8a9da-9f36-4ae5-9da5-37d6d90483b5</
d:ApplicationConnectionId>
      <d:AndroidGcmPushEnabled m:type="Edm.Boolean">false</
d:AndroidGcmPushEnabled>
      <d:AndroidGcmRegistrationId m:null="true" />
      <d:AndroidGcmSenderId m:null="true" />
      <d:ApnsPushEnable m:type="Edm.Boolean">true</d:ApnsPushEnable>
      <d:ApnsDeviceToken m:null="true" />
      <d:ApplicationVersion m:null="true" />
      <d:BlackberryPushEnabled m:type="Edm.Boolean">true</
d:BlackberryPushEnabled>
      <d:BlackberryDevicePin>00000000</d:BlackberryDevicePin>
      <d:BlackberryBESListenerPort m:type="Edm.Int32">5011</
d:BlackberryBESListenerPort>
      <d:BlackberryPushAppID m:null="true" />
      <d:BlackberryPushBaseURL m:null="true" />
      <d:BlackberryPushListenerPort m:type="Edm.Int32">0</
d:BlackberryPushListenerPort>
      <d:BlackberryListenerType m:type="Edm.Int32">0</
d:BlackberryListenerType>
      <d:CustomCustom1>custom1</d:CustomCustom1>
      <d:CustomCustom2 m:null="true" />
      <d:CustomCustom3 m:null="true" />
      <d:CustomCustom4 m:null="true" />
      <d:DeviceModel m:null="true" />
      <d:DeviceType>Unknown</d:DeviceType>
      <d:DeviceSubType m:null="true" />
      <d:DevicePhoneNumber>12345678901</d:DevicePhoneNumber>
      <d:DeviceImsi m:null="true" />
      <d:PasswordPolicyEnabled m:type="Edm.Boolean">true</
d:PasswordPolicyEnabled>
      <d:PasswordPolicyDefaultPasswordAllowed
m:type="Edm.Boolean">false</d:PasswordPolicyDefaultPasswordAllowed>
      <d:PasswordPolicyMinLength m:type="Edm.Int32">8</
d:PasswordPolicyMinLength>
      <d:PasswordPolicyDigitRequired m:type="Edm.Boolean">false</
d:PasswordPolicyDigitRequired>
      <d:PasswordPolicyUpperRequired m:type="Edm.Boolean">false</
d:PasswordPolicyUpperRequired>
      <d:PasswordPolicyLowerRequired m:type="Edm.Boolean">false</
d:PasswordPolicyLowerRequired>
      <d:PasswordPolicySpecialRequired m:type="Edm.Boolean">false</
d:PasswordPolicySpecialRequired>
      <d:PasswordPolicyExpiresInNDays m:type="Edm.Int32">0</
d:PasswordPolicyExpiresInNDays>
      <d:PasswordPolicyMinUniqueChars m:type="Edm.Int32">0</
d:PasswordPolicyMinUniqueChars>
      <d:PasswordPolicyLockTimeout m:type="Edm.Int32">0</
d:PasswordPolicyLockTimeout>
      <d:PasswordPolicyRetryLimit m:type="Edm.Int32">20</
d:PasswordPolicyRetryLimit>
      <d:ProxyApplicationEndpointm:null="true" />
      <d:ProxyPushEndpoint>http://xxue-desktop:8080/GWC/
SMPNotification</d:ProxyPushEndpoint>
      <d:WnsChannelURI m:null="true" />
      <d:MpnsChannelURI m:null="true" />
      <d:WnsPushEnable m:type="Edm.Boolean">false</d:WnsPushEnable>
      <d:MpnsPushEnable m:type="Edm.Boolean">true</d:MpnsPushEnable>
      </m:properties>
  </content>
</entry>
```

You can also retrieve a property value by appending the property name in the URL. For example, to retrieve the `ClientLogLevel` property value, enter:

```
http[s]://<host:port>/[public/]odata/applications/{v1|latest}/
{appid}/Connections(‘{appcid}’)/ClientLogLevel
```

Retrieving Changed Settings and Connections Metadata

You can conditionally retrieve only the changed settings and connections metadata.

To retrieve the changed application settings information:

```
http[s]://<host:port>/[public/]odata/applications/{latest|v1}/
{appID}/Connections(‘{appcid}’)?If-None-Match=”${ETag}”
```

The `${ETag}` part of the URL is a version identifier included in the response of the GET method. If the ETag value of the current application settings is the same as the value included in the request, a status code 304 without a response body is returned to the client to indicate that there are no application setting changes.

Note: The query function is not currently supported.

To retrieve the metadata document of the SAP Mobile Platform Server:

```
http[s]://<host:port>/[public/]odata/applications/{latest|v1}/
{appID}/$metadata
```

To retrieve the service document of the SAP Mobile Platform Server:

```
http[s]://<host:port>/[public/]odata/applications/{latest|v1}/
{appID}
```

Setting or Updating Application Settings

Set or update application settings by issuing a PUT request.

Use the PUT method to update the application settings with all the properties in the request.

```
PUT http[s]://<host:port>/[public/]odata/applications/{latest|v1}/
{appid}/Connections('{appcid}')
```

Use the PUT method to update specified properties. Any properties you do not specify retain their current values.

```
http[s]://<host:port>/[public/]odata/applications/{latest|v1}/
{appid}/Connections('{appcid}')
```

The URL contains these components:

- **host:port –** Hostname and port number of SAP Mobile Platform.
- **public –** If included, an anonymous security configuration is used.
- **odata/applications/ –** Refers to the OData services associated to the application resources.
- **{latest|v1} –** The version of the service document.
- **appid –** The name of the application.
- **Connections –** The name of the OData collection.
- **appcid –** The application connection ID of the application instance that is interacting with the service.

Native Push Notification for a Back End

The SAP Mobile Platform supports native notification for device platforms.

The SAP Mobile Platform uses the native notification mechanisms provided by individual device platforms such as APNS, GCM, BIS/BES, WNS, and MPNS to send notifications. Back end systems use the 'Push' REST service to notify the SAP Mobile Platform about any notification message it has to send to the devices.

Delivery address URL format is:

```
http[s]://<host:port>/Notification/<registration id>
```

where:

- *host:port/Notification* can be received from the proxy push endpoint.

- *registration ID* is sent to the device when the user registers and connects to the application from the device.

The notification data can also be sent using URL arguments.

Notification Data Sent Through HTTP Headers

Notification data can be sent by the back end as a generic HTTP headers or as device platform-specific HTTP headers.

The notification URL is:

```
http[s]://<host:port/Notification>/<registration id>
```

Note: Applications built in SAP Mobile Platform 3.0 and later should adopt the header format `X-SMP-XXX`. To maintain backward compatibility, applications built in earlier versions can continue to use the header format `X-SUP-XXX`. However, `X-SUP-XXX`headers will be removed future releases.

- **Generic header**

  The generic HTTP header is used in the HTTP request to send any notification type such as APNS, GCM, Blackberry, or WNS.

  Header format for notification data in SAP Mobile Platform 3.x and later:

  ```
  <X-SMP-DATA>
  ```
- **APNS-specific headers**

  Use these APNS-specific HTTP headers to send APNS notifications via SAP Mobile Platform:

Header Structure (SAP Mobile Platform and later)	Consists of
`<X-SMP-APNS-ALERT>`	A JSON document. You can use this header or other individual headers listed in this table.
`<X-SMP-APNS-ALERT-BODY>`	Text of the alert message.
`<X-SMP-APNS-ALERT-ACTION-LOC-KEY>`	If a string is specified, this header shows an alert with two buttons: **Close** and **View**. iOS uses the string as a key to get a localized string for the correct button title instead of **View**. If the value is null, the system shows an alert. Clicking **OK** dismisses the alert.
`<X-SMP-APNS-ALERT-LOC-KEY>`	Key to an alert-message string in a `Localizable.strings` file for the current localization.
`<X-SMP-APNS-ALERT-LOC-ARGS>`	Variable string values to appear in place of the format specifiers in `loc-key`.

Header Structure (SAP Mobile Platform and later)	Consists of
`<X-SMP-APNS-ALERT-LAUNCH-IMAGE>`	File name of an image file in the application bundle. It may include the extension. Used as the launch image when you tap the action button or move the action slider. If this property is not specified, the system uses on of the following: • The previous snapshot • The image identified by the `UILaunchImageFile` key in the `Info.plist` file of the application • The `Default.png`.
`<X-SMP-APNS-BADGE>`	Number that appears as the badge on the application icon.
`<X-SMP-APNS-SOUND>`	Name of the sound file in the application bundle.
`<X-SMP-APNS-DATA>`	Custom payload data values. These values must use the JSON-structured and primitive types, such as dictionary (object), array, string, number, and Boolean.

For additional information about APNS headers, see the Apple Web site: *http://developer.apple.com/library/mac/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/ApplePushService/ApplePushService.html*.

- **GCM-specific headers**
  Use these GCM-specific HTTP headers to send GCM notifications:

Header Structure (SAP Mobile Platform and later)	Consists of
`<X-SMP-GCM-COLLAPSEKEY >`	An arbitrary string (such as "Updates Available") that collapses a group of like messages when the device is offline, so that only the last message is sent to the client. **Note:** If you do not include this header, the default value "Updates Available, is used
`<X-SMP-GCM-DATA>`	Payload data, expressed as parameters prefixed with data and suffixed as the key.
`<X-SMP-GCM-DELAYWHILEIDLE>`	(Optional) Represented as 1 or true for true, any other value for false, which is the default value.

Header Structure (SAP Mobile Platform and later)	Consists of
`<X-SMP-GCM-TIMETOLIVE>`	Time (in seconds) that the message remains available on GCM storage if the device is offline.

For additional information about GCM headers, see the Android Web site: *http://developer.android.com/guide/google/gcm/gcm.html#send-msg*.

- **BES/BIS-specific header**

  Use the BlackBerry-specific HTTP header to send BES/BIS notifications:

  ```
  <x-sup-rim-data> or <X-SMP-RIM-DATA>
  ```
- **WNS specific header**

  Use these HTTP headers to send Windows 8 desktop and tablet application notifications:

Header Structure (SAP Mobile Platform and later)	Consists of
`<X-SMP-WNS-DATA>`	Send payload data to the device as raw notification. Payload data may also be a binary data encoded as a Base64-encoded string. Size should not exceed 5KB.
`<X-SMP-WNS-ALERT>`	Text string of the notification, as Tile and Toast notifications.
`<X-SMP-WNS-BADGE>`	Number that appears as the badge on the application icon.

- MPNS (Notification for Windows Phone)

  Use these Windows Phone-specific HTTP headers to send MPNS notifications:

Request Header Structure	Consists of
`<X-SMP-MPNS-DATA>`	Send payload data to device as raw notification. Payload data may also be a binary data encoded as a Base64-encoded string. String length should not exceed more than 2900 characters.
`<X-SMP-MPNS-ALERT>`	Text string of the notification, as Tile and Toast notifications.
`<X-SMP-MPNS-BADGE>`	Number that appears as the badge on the application icon.

SAP Gateway Notification Support

There are no specific handling requirements for sending notifications on the SAP gateway side. SAP Mobile Platform sends notifications using gateway-specific headers.

The SAP Mobile Platform identifies the device type, based on the device type converts the gateway notification headers into the third-party notification context data for APNS, GCM or BES/BIS,WNS, and MPNS.

Note: Non-SAP gateway back ends also use the headers listed below to send generic notifications. In this scenario, the backend is unaware of the device platform.

SAP gateway-specific headers that are handled by SAP Mobile Platform for sending notifications:

Structure Header	Consists of
`<x-sap-poke-title>`	Text of the alert message
`<x-sap-poke-entriesofinter-est>`	Number that appears as the badge on the application icon
`<x-sap-poke-data>`	Custom payload data values. These values must use the JSON structured and primitive types such as dictionary (object), array, string, number, and boolean

1. **APNS**
   The SAP Mobile Platform converts the gateway notification headers into APNS notifications:

Structure Header	Consists of
`<x-sap-poke-title>`	Text of the alert message
`<x-sap-poke-entriesofinter-est>`	Number that appears as the badge on the application icon
`<x-sap-poke-data>`	Custom payload data values. These values must use the JSON structured and primitive types such as dictionary (object), array, string, number, and boolean.

2. **GCM**
   The SAP Mobile Platform converts the gateway notification headers into GCM notifications:

Header Structure	Consists of
`<x-sap-poke-title>`	An arbitrary string (such as "Updates Available") collapses a group of like messages when the device is offline, so that only the last message is sent to the client
`<x-sap-poke-data>`	Payload data. Size should not exceed 4KB

3. **BIS/BES**

   The SAP Mobile Platform converts the gateway notification headers into BIS/BES notifications:

Structure Header	Consists of
`<x-sap-poke-data>`	BES/BIS notification data

4. **WNS**

   The SAP Mobile Platform converts the gateway notification headers into WNS notifications:

Structure Header	Consists of
`<x-sap-poke-title>`	Text of the alert message to be shown on the Tile and Toast notifications
`<x-sap-poke-entriesofinterest>`	Number that appears as the badge on the application icon
`<x-sap-poke-data>`	Custom payload data to be sent to the device as a raw notification

Notification Sent in URL Format

The notification data can also be sent using URL arguments as part of the SAP Mobile Platform push endpoint, or as the delivery address URL.

All URL arguments (zero to many) are optional. The arguments are converted into APNS/GCM/BES/BIS/WNS/MPNS notifications as explained:

```
http[s]://<host:port/Notification>/<application connection ID>?
alert=<alert>&badge=<badge>&sound=<sound>&data=<data in text
format>
```

- **APNS**

Parameters	Description
`alert`	Text of the alert message.
`badge`	Number that appears as the badge on the application icon.

Parameters	Description
`sound`	Name of the sound file in application bundle.
`data`	Custom payload data values. These values must use the JSON-structured and primitive types, such as dictionary (object), array, string, number, and boolean.

- **GCM**

Parameters	Description
`alert`	An arbitrary string (such as "Updates Available") that collapses a group of like messages when the device is offline, so that only the last message is sent to the client.
`data`	Payload data, expressed as parameters prefixed with data and suffixed as the key.

- **BIS/BES**

Parameters	Description
`data`	Notification data
`alert`	Text of the alert message
`badge`	Number that appears as the badge on the application icon

- **WNS**

Parameters	Description
`alert`	The text of the alert message to be sent as a Tile notification
`badge`	Number that appears as the badge on the application icon
`data`	Payload data to be sent

- **MPNS (Notification for Windows Phone)**

Parameters	Description
`alert`	The text of the alert message to be sent as a Tile notification

Parameters	Description
`badge`	Number that appears as the badge on the application icon
`data`	Payload data to be sent

Based on the data send either in headers or in the URL, corresponding notification is sent to the device:

Header|Notification	Tile Notification	Toast Notification	Raw Notification
Alert	Yes	Yes	No
Badge	Yes	No	No
Data	No	No	Yes

Registering the Client for Native Push Notification

Enable native push notifications and register your application to receive push notifications.

Prerequisites

- Configure the registration ID before registering the client for native notifications.
- Configure the application to send push notifications.

Registering the Android Client

Register and enable your Android device clients to receive push notifications.

Prerequisites

- (Administrator) Configure the application for push notification in Management Cockpit by specifying the sender ID and API key.
- Specify the device type should be specified during application connection and registration.
- HTTP header must include X-SMP-APPCID and Authorization headers.

Task

1. If `AndroidGcmPushEnabled` is enabled, sender ID is sent in the response. On successful onboarding of client, the response indicates the GCM push is enabled.
2. If GCM is enabled and the sender ID is available, client uses that sender ID to register itself with GCM and get its unique GCM registration ID.
3. Use the PUT method in the URL, along with the registration ID:

```
http://<host:port>/odata/applications/{latest|v1/}{appid}/
Connections('{appcid}')
```

```
Method : PUT
HTTP Headers "Content-Type" = "application/atom+xml" and "X-HTTP-
METHOD" = "MERGE"
Body:
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
  <content type="application/xml">
    <m:properties>
      <d:AndroidGcmRegistrationId>{GCM registration ID}</
d:AndroidGcmRegistrationId>
    </m:properties>
  </content>
</entry>
```

Registering the BlackBerry Client

Register and enable your BlackBerry device clients to receive push notifications.

1. Configure push notification in Management Cockpit. Specify the device type during application connection and registration, and ensure that the HTTP header also includes the X-SMP-APPCID and Authorization headers.

   Note: To configure push notifications for BIS, you must import BIS certificate into the smp_keystore.jks and keystore files in the configuration folder of the SAP Mobile Platform Server.

2. Enable push notifications in the application:
   a) Update the application connection settings with the BES/BIS registration ID.
   a) Implement the `BlackberryPushListenerPort` and `BlackberryDevicePin` properties in your application.

```
Http payload to update the blackberry (BES) device PIN and push
port:
<?xml version="1.0" encoding="utf-8"?>
  <entry xmlns="http://www.w3.org/2005/Atom" xmlns:d="http://
schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/
dataservices/metadata">
    <m:properties>
      <d:BlackberryDevicePin> </d:BlackberryDevicePin>
      <d:BlackberryBESListenerPort><XXXX></
d:BlackberryBESListenerPort>
     </m:properties>
</content>
</entry>

Http payload to update the blackberry (BIS) device PIN and push
port:
<?xml version="1.0" encoding="utf-8"?>
  <entry xmlns="http://www.w3.org/2005/Atom" xmlns:d="http://
schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/
dataservices/metadata">
```

```
    <m:properties>
      <d:BlackberryPushEnabled>true</d:BlackberryPushEnabled>
      <d:BlackberryDevicePin> </d:BlackberryDevicePin>
      <d:BlackberryPushAppID></d:BlackberryPushAppID>
      <d:BlackberryPushBaseURL> </d:BlackberryPushBaseURL>
      <d:BlackberryPushListenerPort><XXXX></
d:BlackberryPushListenerPort>
  </m:properties>
</content>
</entry>
```

Registering the iOS Client

Register and enable your iOS device clients to receive push notifications.

1. Configure push notification in Management Cockpit. Specify the device type during application connection and registration.
2. Enable push notifications in the application:
   a) To receive the device token, implement the `application:didRegisterForRemoteNotificationsWithDeviceToken:` method in your application delegate.
   b) Update the `ApnsDeviceToken` and `DeviceType` properties via a PUT request. The HTTP header must also include X-SMP-APPCID and Authorization headers.

```
https://<host:port>/odata/applications/{latest|v1}/{appid}/
Connections/('{appcid}')
Method : PUT
HTTP Headers : "Content-Type" = "application/atom+xml"

Body:
<?xml version='1.0' encoding='utf-8'?>
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/
dataservices" xml:base="http://host:port/odata/applications/
v1/com.example.IOS/">
  <id>https://{application URL}/odata/applications/{latest|
v1}/e2eTest/Connections('32552613-470f-45e0-8acc-
b7d73d501682')</id>
    <content type="application/xml">
    <m:properties>
      <d:ApnsDeviceToken>{APNS device token received by the
application from APNS}</d:ApnsDeviceToken>
      <d:DeviceType>iPhone</d:DeviceType>
    </m:properties>
  </content>
</entry>
```

Registering the Windows 8 (Desktop and Tablet) Client

Register and enable your Windows 8 (desktop and tablet) devices to receive push notifications.

Prerequisites

- Configure Windows push notification in Management Cockpit.
- The HTTP header must also include X-SMP-APPCID and Authorization headers.

Task

1. To obtain the channel URI, register the application with WNS. See *Push notification overview (Windows Store apps)* on the Windows Dev Center Web site.
2. Check the `WnsPushEnable` value returned from the registration, and continue with the rest of the WNS or notification registration processing only if the value is true. Set the `WnsChannelURI` value received from the application.
3. Update the application connection settings with the registration ID:

```
https://host:port/odata/applications/{latest|v1/}{appid}/
Connections/('{appcid}')
Method : PUT
HTTP Headers "Content-Type" = "application/atom+xml" and "X-HTTP-
METHOD" = "PUT"
Body:
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
  <content type="application/xml">
    <m:properties>
      <d:WnsChannelURI>{WNS Channel URI}</d:WnsChannelURI>
    </m:properties>
  </content>
</entry>
```

Registering the MPNS Client

Register and enable your Windows phone 8 to receive push notifications.

Prerequisites

- Administrator configures the application for push notification in Management Cockpit.
- The HTTP header must also include X-SMP-APPCID and Authorization headers.

Task

1. To obtain the channel URI, register the application with MPNS. See *Push notifications for Windows Phone* on the Windows Phone Dev center Web site.

2. Check the `MpnsPushEnable` value returned from the registration, and continue with the rest of the MPNS or notification registration processing only if the value is true. Set the `MpnsChannelURI` value received from the application.
3. Using the ApplicationConnection ID ({appcid}) returned from the SAP Mobile Platform registration call (in either the X-SMP-APPCID HTTP header or the `ApplicationConnectionId` property), the application should update the `MpnsChannelURI` property for the application connection for your application by using the Channel URI returned by the application:

```
https://host:port/odata/applications/{latest|v1/}{appid}/
Connections/('{appcid}')
Method : PUT
HTTP Headers "Content-Type" = "application/atom+xml" and "X-HTTP-
METHOD" = "PUT"
Body:
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
  <content type="application/xml">
    <m:properties>
                    <d:MpnsChannelURI>{MPNS Channel URI}</
d:MpnsChannelURI>
    </m:properties>
  </content>
</entry>
```

Retrieving a Customization Resource Bundle

Retrieve a customization resource bundle by issuing a GET method.

Application developers can customize and retrieve the resource bundles during development.

Issue the GET method to the following URL:

```
http[s]://<host:port>/[public/] bundles/{appid}/
[{resourceBundlename:resourceBundleVersion}]
```

If the `{resourceBundlename:resourceBundleVersion}` is specified in the URL, the specified resource bundle is returned in the response body as a stream, otherwise, the resource bundle that is bound to the application is returned. Extension of the resource bundle is provided in the response header X-BUNDLE-EXTENSION.

If the resource bundle is not found in SAP Mobile Platform, error code 404 is returned. You cannot issue other HTTP methods (PUT/POST/DELETE) at the above URL.

Accessing a Service Through a Proxy URL

Access your backend or internet-based service through a proxy URL. The URL supports read (HTTP GET), create (HTTP POST), update (HTTP PUT), and delete (HTTP DELETE).

You can specify the customized application properties for the client with the request. Provide the application connection ID, X-SMP-APPCID by using an explicit request header or a cookie.

Perform an HTTP request at the following URL:

```
http[s]://<host:port>/[public/]/ApplicationName/{connectionName}
```

`connectionName` - connection name of the whitelisted URL.

Note:

- Ensure that all the URLs to be proxied are whitelisted.
- If the session is not handled by the client implicitly, you need to handle it explicitly in the code.

- **HTTP GET –** Invoke GET request to retrieve the data from the back end through SAP Mobile Platform.

  ```
  http[s]://<host:port>/[public/]/ApplicationName/[Collection]
  ```

  Example:

  Request header

  ```
  X-SMP-APPCID : <Application connection Id recieved in the response
  of the onboarding xml>
  Authorization  :  <Base64 format of username/password>
  Content-type  : <application/atom+xml>
  X-CSRF-Token: <"Fetch">
  ```

  Response header

  ```
  Response code 200 ok
  //Response header contains requested data from back end with the
  below mentioned response headers:
  x-csrf-token: "value"
  content-encoding
  Content-Type
  Content-Length
  ```
- **HTTP POST –** Use POST to request the server to accept the data included in the request message body.

  ```
  http[s]://<host:port>/[public/]/ ApplicationName/[Collection]
  ```

  Example:

  Request header

  ```
  X-SMP-APPCID : <Application connection Id recieved in the response
  of the onboarding xml>
  ```

```
Content-Type : application/atom+xml
X-Requested-With  : XMLHttpRequest
X-CSRF-TOKEN : <value of x-csrf-token is recieved as part of GET
which needs to be passed here>
Authorisation : <Base 64 encoded value of Authorisation>
```

Response header

```
201 //on successful creation.
```

- **HTTP PUT –** Use PUT to update the entry in the back end.

```
http[s]://<host:port>/[public/]/ ApplicationName/[Collection]/
<'EntryID'>
```

Example:

Request header

```
X-SMP-APPCID : <Application connection Id recieved in the response
of the onboarding xml>
Content-Type : application/atom+xml
X-CSRF-TOKEN : <value of x-csrf-token is recieved as part of GET
which needs to be passed here>
Authorisation : < Base 64 encoded value of Authorisation>
X-Requested-With  : XMLHttpRequest
```

Response header

```
204 //on successful update.
```

- **HTTP DELETE –** Use DELETE to the delete an entry in back end.

```
http[s]://<host:port>/[public/]/ApplicationName/[Collection]
(EntryID)
```

Example:

Request header

```
X-SMP-APPCID : <Application connection Id recieved in the response
of the onboarding xml>
Authorization  :  <Base64 format of username/password>
Content-type  : <application/atom+xml>
X-CSRF-TOKEN : <value of x-csrf-token is recieved as part of GET
which needs to be passed here>
X-Requested-With  : XMLHttpRequest
```

Response header

```
204 //on successful deletion.
```

Note: No information is returned in case of DELETE request.

Reference

Describes REST API resources.

Note: The examples in the reference section use basic authentication, which you can see in the HTTP header, for example Authorization: Basic Base64-encoded string.

HTTP Headers and Cookies

Use HTTP headers and cookies to retrieve application connection information.

Note: In SAP Mobile Platform 3.0, applications should adopt the header format X-SMP-XXX. To maintain backward compatibility, the applications built in earlier version continue to use the header format X-SUP-XXX. However, X-SUP-XXX headers will be removed from future releases.

Cookies are returned by servers in the HTTP response header (`Set-Cookie` header) and included by the HTTP client (for example, a browser) in the subsequent HTTP request header (cookie header). Supported headers are:

- **X-SMP-APPCID –** communicates the application connection ID as that is generated by the application from which the request originates, or as issued by the server using the X-SMP-APPCID cookie. The X-SMP-APPCID value has a maximum length of 128 characters. If an onboarding request does not include X-SMP-APPCID, the server automatically generates a value. The client or application should send this value for all subsequent requests. If the X-SMP-APPCID value is not sent from the client, all subsequent request-response interactions result in an `403 Forbidden` error.

  X-SMP-APPCID is received as a cookie, but the client returns it to the server either as a cookie or as a header.
- **X-SMP-BACKEND-URL –** application can provide the back-end request URL via an X-SMP-BACKEND-URL header while sending the request to SAP Mobile Platform Server. The server forwards the request to the URL provided in the X-SMP-BACKEND-URL.

  If the X-SMP-BACKEND-URL header is present in the request, URL rewrite is disabled.

  Note: To avoid errors, ensure that the URL is whitelisted in the server.
- **X-SMP-REQUESTID –** communicates the request ID as generated by the application from which the request originates, or as issued by the server that is using the X-SMP-REQUESTID cookie.

Application Connection Properties

Describes application connection properties, and indicates whether the properties are read-only or Nullable from the HTTP client.

Note: If you attempt to modify a read-only property, the client application throws the following exception: `HTTP 403 - The property "XXX" cannot be updated by a client application`.

Table 1. Application Connection Properties: Uncategorized

Property Name	Type	Read-only?	Is Nullable?
ETag	String	Yes	No
ApplicationConnectionId	String	Yes	No

Table 2. Application Connection Properties: Android Push

Property Name	Type	Read-only?	Is Nullable?
AndroidGcmPushEnabled	Boolean	No	No
AndroidGcmRegistrationId	String	No	Yes
AndroidGcmSenderId	String	Yes	Yes

Table 3. Application Connection Properties: Apple Push

Property Name	Type	Read-only?	Is Nullable?
ApnsPushEnable	Boolean	No	No
ApnsDeviceToken	String	No	Yes

Table 4. Application Connection Properties: Application Settings

Property Name	Type	Read-only?	Is Nullable?
CustomizationBundleId	String	Yes	Yes
ApplicationVersion	String	No	Yes
ClientSdkVersion	String	No	Yes

Table 5. Application Connection Properties: BlackBerry Push

Property Name	Type	Read-only?	Is Nullable?
BlackberryPushEnabled	Boolean	No	No
BlackberryDevicePin	String	No	Yes
BlackberryBESListenerPort	Int32	No	No

Table 6. Application Connection Properties: Windows Push

Property Name	Type	Read-only?	Is Nullable?
WnsChannelURI	String	No	Yes
WnsPushEnable	Boolean	No	No

Table 7. Application Connection Properties: MPNS Push

Property Name	Type	Read-only?	Is Nullable?
MpnsChannelURI	String	No	Yes
MpnsPushEnable	Boolean	No	No

Table 8. Application Connection Properties: Capabilities

Property Name	Type	Read-only?	Is Nullable?
CapabilitiesPasswordPolicy	Boolean	No	No

Table 9. Application Connection Properties: Custom Settings

Property Name	Type	Read-only?	Is Nullable?
CustomCustom1	String	No	Yes
CustomCustom2	String	No	Yes
CustomCustom3	String	No	Yes
CustomCustom4	String	No	Yes

Table 10. Application Connection Properties: Device Information

Property Name	Type	Read-only?	Is Nullable?
DeviceModel	String	No	Yes
DeviceType	String	No	Yes
DeviceSubType	String	No	Yes
DevicePhoneNumber	String	No	Yes
DeviceIMSI	String	No	Yes

Table 11. Application Connection Properties: Password Policy

Property Name	Type	Read-only?	Is Nullable?
PasswordPolicyEnabled	Boolean	Yes	No
PasswordPolicyDefaultPasswordAllowed	Boolean	Yes	No
PasswordPolicyMinLength	Int32	Yes	No
PasswordPolicyDigitRequired	Boolean	Yes	No
PasswordPolicyUpperRequired	Boolean	Yes	No
PasswordPolicyLowerRequired	Boolean	Yes	No
PasswordPolicySpecialRequired	Boolean	Yes	No
PasswordPolicyExpiresInNDays	Int32	Yes	No
PasswordPolicyMinUniqueChars	Int32	Yes	No
PasswordPolicyLockTimeout	Int32	Yes	No
PasswordPolicyRetryLimit	Int32	Yes	No

Table 12. Application Connection Properties: Proxy

Property Name	Type	Read-only?	Is Nullable?
ProxyApplicationEnd-point	String	Yes	Yes
ProxyPushEndpoint	String	Yes	Yes

Proxy Responses

Proxy responses include all the cookies and headers from the proxied backend.

Application Connections

Methods for creating, updating, or reading application connections.

Note: Application connection service is implemented as an OData service and therefore follows OData standards.

Service Document

Get the service document for the application connection. Retrieving the service document allows the client to discover the capabilities and locations of the available collections.

Syntax

Perform an HTTP GET request at the following URL:

```
http[s]://<host:port>/[public/]odata/applications/{v1|latest}/
{appid}
```

Parameters

- **appid –** The application ID that uniquely identifies the application.

Returns

Returns a 200 OK status code if successful, and a service document in the response body.

Examples

- **Service document –** HTTP request header:

  ```
  GET /odata/applications/v1/com.sap.myapp HTTP/1.1
  Host: smpserver:8080
  Connection: Keep-Alive
  User-Agent: Apache-HttpClient/4.1.3 (java 1.5)
  Authorization: Basic REVWMDAwMTppbml0aWFs
  ```

  HTTP response header:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/xml;charset=utf-8
  ```

```
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: X-SMP-SESSID=97ts80gwhxkc;Path=/
Set-Cookie: X-SMP-APPCID=b6d50e93-bcaa-439d-9741-660a3cb56771
DataServiceVersion: 1.0
Date: Tue, 14 Aug 2012 21:28:55 GMT
Transfer-Encoding: chunked
```

The HTTP response body is a service document that includes two collections named "Connections" and "Endpoints".

```
<?xml version='1.0' encoding='utf-8'?>
<service xmlns="http://www.w3.org/2007/app"
         xml:base="http://host:port/appSettings/odata/v1/appid/"
         xmlns:atom="http://www.w3.org/2005/Atom"
         xmlns:app="http://www.w3.org/2007/app">
    <workspace>
        <atom:title>Default</atom:title>
        <collection href="Connections">
            <atom:title>Connections</atom:title>
        <collection href="Endpoints">
            <atom:title>Endpoints</atom:title>
        </collection>
    </workspace>
 </service>
```

Metadata

Get the metadata document that includes the metadata for the application connection settings and proxy endpoints.

Metadata documents are based on the OData standard and required for implementing application connection services.

Syntax

Perform an HTTP GET request at the following URL:

```
http[s]://<host:port>/[public/]odata/applications/{v1|latest}/
{appid}/$metadata
```

Parameters

- **appid –** The application ID that uniquely identifies the application.

Returns

If successful, returns a 200 OK status code and a metadata document in the response body.

Examples

- **Get Metadata –** HTTP request header:

```
GET /odata/applications/v1/com.sap.myapp/$metadata HTTP/1.1
Host: smpserver:8080
Connection: Keep-Alive
```

```
User-Agent: Apache-HttpClient/4.1.3 (java 1.5)
Authorization: Basic REVWMDAwMTppbml0aWFs
```

HTTP response header:

```
HTTP/1.1 200 OK
Content-Type: application/xml;charset=utf-8
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: X-SMP-SESSID=97ts80gwhxkc;Path=/
Set-Cookie: X-SMP-APPCID=b6d50e93-bcaa-439d-9741-660a3cb56771
DataServiceVersion: 1.0
Date: Tue, 14 Aug 2012 21:32:34 GMT
Transfer-Encoding: chunked
```

HTTP response body (metadata):

```
<?xml version="1.0" encoding="utf-8"?>
<edmx:Edmx Version="1.0" xmlns:edmx="http://
schemas.microsoft.com/ado/2007/06/edmx"
xmlns:smp="http://www.sap.com/smp/odata"><edmx:DataServices
m:DataServiceVersion="2.0" xmlns:m="http://schemas.microsoft.com/
ado/2007/08/dataservices/metadata"><Schema
Namespace="applications" xmlns="http://schemas.microsoft.com/ado/
2008/09/edm">
<EntityType Name="Connection">
    <Key>
    <PropertyRef Name="ApplicationConnectionId"></PropertyRef></
Key>
    <Property Name="ETag" Type="Edm.String" Nullable="false"
sup:ReadOnly="true"></Property>
    <Property Name="ApplicationConnectionId" Type="Edm.String"
Nullable="false" sup:ReadOnly="true"></Property>
    <Property Name="AndroidGcmPushEnabled" Type="Edm.Boolean"
Nullable="false" sup:ReadOnly="false"></Property>
    <Property Name="AndroidGcmRegistrationId" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="AndroidGcmSenderId" Type="Edm.String"
Nullable="true" sup:ReadOnly="true"></Property>
    <Property Name="ApnsPushEnable" Type="Edm.Boolean"
Nullable="false" sup:ReadOnly="false"></Property>
    <Property Name="ApnsDeviceToken" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="ApplicationVersion" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="BlackberryPushEnabled" Type="Edm.Boolean"
Nullable="false" sup:ReadOnly="false"></Property>
    <Property Name="BlackberryDevicePin" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="BlackberryBESListenerPort" Type="Edm.Int32"
Nullable="false" sup:ReadOnly="false"></Property>
    <Property Name="BlackberryPushAppID" Type="Edm.String"
Nullable="true" sup:ReadOnly="true"></Property>
    <Property Name="BlackberryPushBaseURL" Type="Edm.String"
Nullable="true" sup:ReadOnly="true"></Property>
    <Property Name="BlackberryPushListenerPort" Type="Edm.Int32"
Nullable="false" sup:ReadOnly="true"></Property>
    <Property Name="BlackberryListenerType" Type="Edm.Int32"
```

```
Nullable="false" sup:ReadOnly="true"></Property>
    <Property Name="ConnectionLogLevel" Type="Edm.String"
Nullable="true" sup:ReadOnly="true"></Property>
    <Property Name="CustomizationBundleId" Type="Edm.String"
Nullable="true" sup:ReadOnly="true"></Property>
    <Property Name="CustomCustom1" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="CustomCustom2" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="CustomCustom3" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="CustomCustom4" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="DeviceModel" Type="Edm.String" Nullable="true"
sup:ReadOnly="false"></Property>
    <Property Name="DeviceType" Type="Edm.String" Nullable="true"
sup:ReadOnly="false"></Property>
    <Property Name="DeviceSubType" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="DevicePhoneNumber" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="DeviceIMSI" Type="Edm.String" Nullable="true"
sup:ReadOnly="false"></Property>
    <Property Name="MpnsChannelURI" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="MpnsPushEnable" Type="Edm.Boolean"
Nullable="false" sup:ReadOnly="false"></Property>
    <Property Name="PasswordPolicyEnabled" Type="Edm.Boolean"
Nullable="false" sup:ReadOnly="true"></Property>
    <Property Name="PasswordPolicyDefaultPasswordAllowed"
Type="Edm.Boolean" Nullable="false" sup:ReadOnly="true"></
Property>
    <Property Name="PasswordPolicyMinLength" Type="Edm.Int32"
Nullable="false" sup:ReadOnly="true"></Property>
    <Property Name="PasswordPolicyDigitRequired"
Type="Edm.Boolean" Nullable="false" sup:ReadOnly="true"></
Property>
    <Property Name="PasswordPolicyUpperRequired"
Type="Edm.Boolean" Nullable="false" sup:ReadOnly="true"></
Property>
    <Property Name="PasswordPolicyLowerRequired"
Type="Edm.Boolean" Nullable="false" sup:ReadOnly="true"></
Property>
    <Property Name="PasswordPolicySpecialRequired"
Type="Edm.Boolean" Nullable="false" sup:ReadOnly="true"></
Property>
    <Property Name="PasswordPolicyExpiresInNDays" Type="Edm.Int32"
Nullable="false" sup:ReadOnly="true"></Property>
    <Property Name="PasswordPolicyMinUniqueChars" Type="Edm.Int32"
Nullable="false" sup:ReadOnly="true"></Property>
    <Property Name="PasswordPolicyLockTimeout" Type="Edm.Int32"
Nullable="false" sup:ReadOnly="true"></Property>
    <Property Name="PasswordPolicyRetryLimit" Type="Edm.Int32"
Nullable="false" sup:ReadOnly="true"></Property>
    <Property Name="ProxyApplicationEndpoint" Type="Edm.String"
Nullable="true" sup:ReadOnly="true"></Property>
```

```
    <Property Name="ProxyPushEndpoint" Type="Edm.String"
Nullable="true" sup:ReadOnly="true"></Property>
   <Property Name="UploadLogs" Type="Edm.String" Nullable="true"
sup:ReadOnly="true"></Property>
    <Property Name="WnsChannelURI" Type="Edm.String"
Nullable="true" sup:ReadOnly="false"></Property>
    <Property Name="WnsPushEnable" Type="Edm.Boolean"
Nullable="false" sup:ReadOnly="false"></Property>
    </EntityType>
<EntityContainer Name="Container"
m:IsDefaultEntityContainer="true">
<EntitySet Name="Connections"
EntityType="applications.Connection">
</EntitySet>
</EntityContainer>
</Schema>
</edmx:DataServices>
</edmx:Edmx>
```

Create Application Connection

Create an application connection and initially set the application connection settings. Because all application connection settings are optional, the minimal body contains no properties at all. SAP Mobile Platform populates default values as needed.

Syntax

Perform an HTTP POST request at the following URL:

```
http[s]://<host:port>/[public/]/odata/applications/{v1|latest}/
{appid}/Connections
```

Parameters

- **appid –** The application ID that uniquely identifies the application.

  Note: If an application is configured for anonymous access using Management Cockpit, the registration is successful irrespective of wrong or no credentials present in the authorization header.

Returns

If successful, a 201 Created status code is returned, and the new application connection settings are included in the response body.

Examples

- **Create application connection –** Example HTTP request header:

```
POST /odata/applications/v1/com.sap.myapp/Connections HTTP/1.1
Content-Length: 4704
Content-Type: application/atom+xml; charset=UTF-8
Host: smpserver:8080
```

```
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.3 (java 1.5)
Authorization: Basic <XXXX>
```

Example HTTP response header:

```
HTTP/1.1 201 Created
Content-Type: application/atom+xml;charset=utf-8
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: X-SMP-SESSID=<XXXX>;Path=/
Set-Cookie: X-SMP-APPCID=<XXXX>
Location: http:/smpserver:8080/applications/
Connections('b6d50e93-bcaa-439d-9741-660a3cb56771')
DataServiceVersion: 1.0
Date: Mon, 13 Aug 2012 22:40:50 GMT
Transfer-Encoding: chunked
```

Response body:

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xml:base="http://localhost:8080/public/odata/applications/
v1/com.sap.myapp/"
xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
  <content type="application/xml">
    <m:properties>
    </m:properties>
  </content>
</entry>
```

Get Application Settings

Get the application settings.

Syntax

Perform an HTTP GET request at the following URL:

```
http[s]://<host:port>/[public/]odata/applications/{v1|latest}/
{appid}/Connections('{appcid}')
```

Parameters

- **appid –** The application ID that uniquely identifies the application.
- **appcid –** The application connection ID of the application instance interacting with the service.

Returns

If successful, returns a 200 OK status code and an HTTP response body with the application settings.

Examples

- **Get application settings –** HTTP request header:

```
GET /odata/applications/v1/com.sap.myapp/Connections('b6d50e93-
bcaa-439d-9741-660a3cb56771') HTTP/1.1
Cookie: X-SMP-APPCID=<XXXX>; X-SMP-SESSID=<XXXX>
Host: smpserver:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.3 (java 1.5)
Authorization: Basic <XXXX>
```

HTTP response header:

```
HTTP/1.1 200 OK
Content-Type: application/atom+xml;charset=utf-8
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: X-SMP-APPCID=<XXXX>
DataServiceVersion: 1.0
Date: Mon, 13 Aug 2012 22:56:50 GMT
Transfer-Encoding: chunked
```

HTTP response body:

```
<?xml version="1.0" encoding="UTF-8"?>
-<entry xml:base="http://10.53.138.170:8080/odata/applications/
latest/G3T/"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/
metadata" xmlns="http://www.w3.org/2005/Atom">
<id>http://10.53.138.170:8080/odata/applications/latest/G3T/
Connections<XXXX></id>
<title type="text"/>
<updated>2013-11-07T09:15:17Z</updated>
<author><name/></author>
<link title="Connection" href="Connections('xxxx')" rel="edit"/>
<category scheme="http://schemas.microsoft.com/ado/2007/08/
dataservices/scheme" term="applications.Connection"/>
<content type="application/xml">
<m:properties>
    <d:ETag>2013-11-07 14:44:43.0</
d:ETag><d:ApplicationConnectionId>xxxx</
d:ApplicationConnectionId>
    <d:AndroidGcmPushEnabled m:type="Edm.Boolean">false</
d:AndroidGcmPushEnabled>
    <d:AndroidGcmRegistrationId m:null="true"/
><d:AndroidGcmSenderId/>
    <d:ApnsPushEnable m:type="Edm.Boolean">true</
d:ApnsPushEnable>
    <d:ApnsDeviceToken m:null="true"/>
    <d:ApplicationVersion>1.0</d:ApplicationVersion>
    <d:BlackberryPushEnabled m:type="Edm.Boolean">false</
d:BlackberryPushEnabled>
    <d:BlackberryDevicePin m:null="true"/>
    <d:BlackberryBESListenerPort m:type="Edm.Int32">0</
d:BlackberryBESListenerPort>
    <d:BlackberryPushAppID m:null="true"/>
    <d:BlackberryPushBaseURL m:null="true"/>
    <d:BlackberryPushListenerPort m:type="Edm.Int32">0</
d:BlackberryPushListenerPort>
    <d:BlackberryListenerType m:type="Edm.Int32">0</
d:BlackberryListenerType>
    <d:ConnectionLogLevel>NONE</d:ConnectionLogLevel>
    <d:CustomizationBundleId m:null="true"/><d:CustomCustom1/
><d:CustomCustom2/><d:CustomCustom3/><d:CustomCustom4/>
    <d:DeviceModel m:null="true"/>
    <d:DeviceType>Unknown</d:DeviceType>
    <d:DeviceSubType m:null="true"/>
    <d:DevicePhoneNumber m:null="true"/>
    <d:DeviceIMSI m:null="true"/>
    <d:MpnsChannelURI m:null="true"/>
    <d:MpnsPushEnable m:type="Edm.Boolean">true</
d:MpnsPushEnable>
    <d:PasswordPolicyEnabled m:type="Edm.Boolean">false</
d:PasswordPolicyEnabled>
    <d:PasswordPolicyDefaultPasswordAllowed
m:type="Edm.Boolean">false</
d:PasswordPolicyDefaultPasswordAllowed>
    <d:PasswordPolicyMinLength m:type="Edm.Int32">8</
d:PasswordPolicyMinLength>
    <d:PasswordPolicyDigitRequired m:type="Edm.Boolean">false</
d:PasswordPolicyDigitRequired>
    <d:PasswordPolicyUpperRequired m:type="Edm.Boolean">false</
d:PasswordPolicyUpperRequired>
    <d:PasswordPolicyLowerRequired m:type="Edm.Boolean">false</
d:PasswordPolicyLowerRequired>
    <d:PasswordPolicySpecialRequired m:type="Edm.Boolean">false</
d:PasswordPolicySpecialRequired>
    <d:PasswordPolicyExpiresInNDays m:type="Edm.Int32">0</
d:PasswordPolicyExpiresInNDays>
    <d:PasswordPolicyMinUniqueChars m:type="Edm.Int32">0</
d:PasswordPolicyMinUniqueChars>
    <d:PasswordPolicyLockTimeout m:type="Edm.Int32">0</
d:PasswordPolicyLockTimeout>
    <d:PasswordPolicyRetryLimit     m:type="Edm.Int32">20</
d:PasswordPolicyRetryLimit>
<d:ProxyApplicationEndpoint>http://vmw3815.wdf.sap.corp:50009/
sap/opu/sdata/iwfnd/RMTSAMPLEFLIGHT/</d:ProxyApplicationEndpoint>
<d:ProxyPushEndpoint>http://INLC50802847A:8080/Notification</
d:ProxyPushEndpoint>
<d:UploadLogs>false</d:UploadLogs>
<d:WnsChannelURI m:null="true"/>
<d:WnsPushEnable m:type="Edm.Boolean">false</d:WnsPushEnable>
</m:properties>
</content>
</entry>
```

Get Application Settings (Property)

Get the specific property value for a property from the application settings.

Syntax

Perform an HTTP GET request at the following URL:

```
http://<host:port>/[public/]odata/applications/{v1|latest}/{appid}/
Connections(‘{registrationID}’)/<property-name>
```

Parameters

- **appid –** The application ID that uniquely identifies the application.
- **registrationID –** The registration ID of the application instance that is interacting with the service.
- **property-name –** The property-name can be appended to the URL to retrieve the value of a specific property.

Returns

If successful, returns a 200 OK status code and an HTTP response body with the application settings.

Examples

- **Get the DeviceType property –** HTTP request:

  ```
  GET /odata/applications/v1/com.sap.myapp/Connections('b6d50e93-
  bcaa-439d-9741-660a3cb56771')/DeviceType HTTP/1.1
  Cookie: X-SMP-APPCID=b6d50e93-bcaa-439d-9741-660a3cb56771; X-SMP-
  SESSID=97ts80gwhxkc
  Host: smpserver:8080
  Connection: Keep-Alive
  User-Agent: Apache-HttpClient/4.1.3 (java 1.5)
  Authorization: Basic REVWMDAwMTppbml0aWFs
  ```

  HTTP response header:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/xml;charset=utf-8
  Expires: Thu, 01 Jan 1970 00:00:00 GMT
  Set-Cookie: X-SMP-APPCID=<XXXX>
  DataServiceVersion: 1.0
  Date: Mon, 13 Aug 2012 23:00:27 GMT
  Transfer-Encoding: chunked
  ```

  HTTP response body:

  ```
  <?xml version='1.0' encoding='utf-8'?>
  <d:DeviceType>iPhone</d:DeviceType>
  ```

Update Application Settings (PUT)

Update the application settings with all the properties in the request.

Any properties that you do not specify (and that can be changed) are set to the default value.

Syntax

```
PUT http[s]://<host:port>/[public/]odata/applications/{v1|latest}/
{appid}/Connections(‘{registrationID}’)
```

Parameters

- **appid –** The application ID that uniquely identifies the application.
- **registrationID –** The registrationID of the application instance that is interacting with the service.

Returns

When processing a PUT request, SAP Mobile Platform returns a 200 status code to indicate success, and there is no response body.

If you have not explicitly registered the client, PUT request returns a 404 status code.

Examples

- **Update application settings (PUT) –** HTTP request header:

  ```
  PUT /odata/applications/v1/com.sap.myapp/Connections('<XXXX>')
  HTTP/1.1
  Cookie: X-SMP-APPCID=<XXXX>; X-SMP-SESSID=<XXXX>
  Content-Length: 4744
  Content-Type: application/atom+xml; charset=UTF-8
  Host: smpserver:8080
  Connection: Keep-Alive
  User-Agent: Apache-HttpClient/4.1.3 (java 1.5)
  Authorization: Basic <XXXX>
  ```

  [PUT HTTP request message body]

  HTTP response header:

  ```
  HTTP/1.1 200 OK
  Expires: Thu, 01 Jan 1970 00:00:00 GMT
  Set-Cookie: X-SMP-APPCID=<XXXX>
  DataServiceVersion: 1.0
  Date: Mon, 13 Aug 2012 23:07:51 GMT
  Content-Length: 0
  ```

Delete Application Connection

Delete an application connection.

Syntax

Perform an HTTP DELETE request at the following URL:

```
http[s]://<host:port>/[public/]odata/applications/{v1|latest}/
{appid}/Connections(‘{appcid}’)
```

Parameters

- **appid –** The application ID that uniquely identifies the application.
- **appcid –** The application connection ID of the application instance interacting with the service.

Returns

If successful, returns a 200 OK status code.

If you never explicity registered the client, returns a 404 status code.

Examples

- **Delete application connection –** HTTP request:

```
DELETE /odata/applications/v1/com.sap.myapp/
Connections('b6d50e93-bcaa-439d-9741-660a3cb56771') HTTP/1.1
Cookie: X-SMP-APPCID=<XXXX>; X-SMP-SESSID=<XXXX>
Host: smpserver:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.3 (java 1.5)
Authorization: Basic <XXXX>
```

  HTTP response:

```
HTTP/1.1 200 OK
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: X-SMP-APPCID=<XXXX>
DataServiceVersion: 1.0
Date: Mon, 13 Aug 2012 23:19:42 GMT
Content-Length: 0
```

Error Code and Message Format

The server returns different formats for error codes and messages according to different "Accept" values in request headers.

Table 13. Accept Header and Data Format

Type and Format	Accept Header Values	Sample Response Body
XML	application/xml, application/xhtml+xml, application/atom+xml	<html><head><title>"message string"</title</head><body><h1>"error code" - "error string".</h1>"<p><b>message string</b> <u>error string</u></p><p><b>description</b> <u>error message</u></p><h3>"text string"</h3></body></html>

Type and Format	Accept Header Values	Sample Response Body
JSON	application/json, text/json	{ "error": {"code": "403", "message": {"lang": "en-US", "value": "some specific error text string" } } }
TEXT	text/html, text/plain	"some specific error text string"

Note: If the Accept header does not include any of these data types, the response body is null.

Download Application Customization Resource Bundle

Download the customization resource bundle.

Note: If you add multiple resource bundles and no resource is specified in the URL, the one indicated as default considered as the default resource bundle.

Syntax

Perform an HTTP GET request at the following URL:

```
http[s]://<host:port>/[public/] bundles/{appid}/
[{resourceBundleName:Version}]
```

Parameters

- **resourceBundleName –** Optionally specifies a resource bundle to be returned.
- **Version –** Version of the resource bundle.

Returns

If you specify the `{resourceBundleName}` in the URL, the specified resource bundle is returned in the response body as a stream. Otherwise, the resource bundle that is bound to the application is returned.

If the resource bundle is not found, error code 404 is returned. You cannot issue other HTTP methods (PUT/POST/DELETE) at the URL labels parameters shown in syntax.

Examples

- **Download –** HTTP request:

  ```
  GET /bundles/com.sap.myapp/MyApp:1.0 HTTP/1.1
  Cookie: X-SMP-APPCID=<XXXX>; X-SMP-SESSID=<XXXX>
  Host: smpserver:8080
  Connection: Keep-Alive
  User-Agent: Apache-HttpClient/4.1.3 (java 1.5)
  Authorization: Basic <XXXX>
  ```

  HTTP response:

```
HTTP/1.1 200 OK
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: X-SMP-APPCID=<XXXX>
Transfer-Encoding: chunked

[resource bundle content]
```

Proxy Connections

Methods for accessing proxy connections.

Access External Service

SAP Mobile Platform Server can also be used to access OData services that are provided by any external OData provider. The proxy URL supports read (HTTP GET), create (HTTP POST), update (HTTP PUT), and delete (HTTP DELETE).

Syntax

Perform an HTTP request at the following URL:

```
http[s]://<host:port>/[public/]/[connectionName]
```

Parameters

- **connectionName –** connection name of the whitelisted URL

Note: If an application is configured for anonymous access, the request-response is made using the same user credentials provided in "Allow anonymous connections" field for defining the backend connection in Management Cockpit.

Returns

If successful, returns a response from the back end and a response body.

Examples

- **Access external service –** HTTP request header:

```
GET /com.sap.myapp HTTP/1.1
Cookie: X-SMP-APPCID=<XXXX>; X-SMP-SESSID=<XXXX>
Host: smpserver:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.3 (java 1.5)
Authorization: Basic <XXXX>
```

HTTP response header:

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: X-SMP-APPCID=<XXXX>
server: SAP NetWeaver Application Server / ABAP 731
dataserviceversion: 2.0
```

```
set-cookie: SAP_SESSIONID_DG1_001=<Some-Token>; path=/
set-cookie: MYSAPSSO2=<SSO-Token>; path=/; domain=.sap.com
Content-Length: 2651
```

CORS Enabled Browser-Based Applications

Cross-Origin Resource Sharing (CORS) is a specification that allows scripts from one domain to make requests to another domain.

When a browser-based application, such as a JavaScript or jQuery application, sends a request to a domain other than the one it is hosted in, it is called a cross domain request. All valid CORS requests are accompanied by an origin header, which is added automatically by the browser. SAP Mobile Platform can handle all the received CORS requests.

The response headers that are added to all valid CORS requests received by SAP Mobile Platform include:

- **Access-Control-Allow-Origin –** origin value as specified in the request header.
- **Access-Control-Allow-Credentials –** when set to true, enables credential requests and requests accompanied with cookies.
- **Access-Control-Expose-Headers –** to include any custom header in the application, the list of exposed headers must be set as the value in this header.

Note: If the backend does not explicitly specify **Access-Control-Expose-Headers**, SAP Mobile Platform adds all response headers (except for simple HTTP headers and set-cookie header) to this header value, allowing the browser application to access all the headers from the backend response.

In case of a pre-flight (HTTP OPTIONS) request, SAP Mobile Platform adds these headers to the response:

- **Access-Control-Allow-Methods –** Based on the request URL:
  - POST for onboarding URL
  - GET, PUT, DELETE for application settings URL
  - GET for resource bundle URL
  - Value of Access-Control-Request-Method header for all other URLs
- **Access-Control-Allow-Headers –** Access-Control-Request-Headers header value as specified in request.
- **Access-Control-Max-Age –** the time period for which the browser caches the results of the pre-flight request is by default, 3600 seconds.

Browser Restrictions with CORS

- Internet Explorer versions 9 and earlier do not support CORS-enabled browser-based applications.
- Safari supports Document Object Model (DOM) parsing with restrictions.

Index