

# Integrating with Cloudant by using an adapter

## What is Cloudant?

Cloudant is a NoSQL Database based on CouchDB that is included with the product as a component called IBM MobileFirst Cloudant Local Data Layer Edition. Cloudant is also available as a standalone installed product and as a Database as a Service (DBaaS) on IBM Bluemix and cloudant.com.

The Cloudant API is documented at <https://docs.cloudant.com/index.html>  
(<https://docs.cloudant.com/index.html>)

## Cloudant adapter

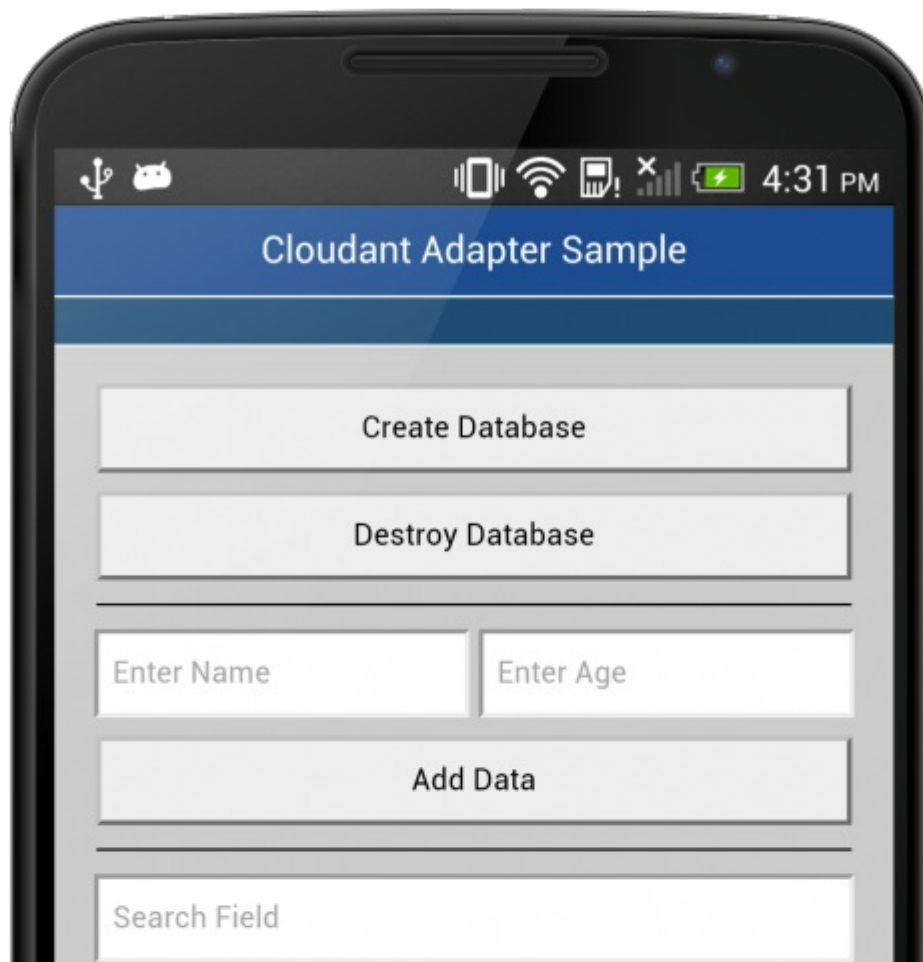
You can create and use a MobileFirst adapter to communicate with Cloudant.

The adapter within the sample that is provided with this module is an HTTP adapter.

- This adapter implements a subset of the Cloudant API.
- You can extend it to implement more Cloudant API

## Sample Application

The sample module contains an application that





demonstrates the use of the adapter to store, list, and query a set of names and ages in a Cloudant-hosted database.

## Configuring the sample application

First, download and import the sample module.

Then, open the `CloudantAdapterDemo/adapters/CloudantAdapter/CloudantAdapter.xml` file.

The adapter uses basic authentication to connect to the Cloudant database. Enter your database location and username and password into the `CloudantAdapter.xml` file. You can obtain this information from Cloudant.

To avoid exposing the user credentials in `CloudantAdapter.xml`, you can leverage Cloudant support for API keys and passwords.

```

<?xml version="1.0" encoding="UTF-8"?>
<wl:adapter name="CloudantAdapter" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wl
="http://www.ibm.com/mfp/integration" xmlns:http="http://www.ibm.com/mfp/integration/http">
  <displayName>CloudantAdapter</displayName>
  <description>CloudantAdapter</description>
  <connectivity>
    <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
      <protocol>https</protocol>
      <domain>mylogin.cloudant.com</domain>
      <port>443</port>
      <connectionTimeoutInMilliseconds>30000</connectionTimeoutInMilliseconds>
      <socketTimeoutInMilliseconds>30000</socketTimeoutInMilliseconds>
      <authentication>
        <basic/>
        <serverIdentity>
          <username>mylogin</username>
          <password>mypassword</password>
        </serverIdentity>
      </authentication>
      <maxConcurrentConnectionsPerNode>4</maxConcurrentConnectionsPerNode>
      <!-- Following properties used by adapter's key manager for choosing specific certificate from key
store
      <sslCertificateAlias></sslCertificateAlias>
      <sslCertificatePassword></sslCertificatePassword>
      -->
    </connectionPolicy>
  </connectivity>
  <procedure name="createDB"/>
  <procedure name="deleteDB"/>
  <procedure name="getAllDocs"/>
  <procedure name="updateDocs"/>
  <procedure name="createDoc"/>
  <procedure name="getDoc"/>
  <procedure name="updateDoc"/>
  <procedure name="deleteDoc"/>
  <procedure name="createDesignDoc"/>
  <procedure name="search" />
  <procedure name="getView" />
</wl:adapter>

```

## Starting the sample application and adapter

Right-click adapters/CloudantAdapter and select **Run As > Deploy MobileFirst Adapter**.

Right-click apps/CloudantSampleApp and select **Run As > Run on MobileFirst Development Server**.

From the MobileFirst Platform Operations Console, select **Preview as Common Resources** (or view the app in the environment of your choice).

## Cloudant adapter methods

- `createDB (name)`  
Returns JSON result statement
- `deleteDB (name)`

Returns JSON result statement

- getAllDocs (name, limit, include\_docs)

Arguments:

- name - database name
- limit - limit on the number of docs
- include\_docs - whether or not to include the full body of the docs in the return

Returns JSON object with document information

- updateDocs (name, updates)

Arguments:

- name - database name
- updates - list of objects to create or update

Returns JSON result statement

- createDoc (name, doc)

Arguments:

- name - database name
- doc - JSON document to create

Returns JSON result statement

- getDoc (name, docID)

Arguments:

- name - database name
- docID - document ID

Returns JSON document

- updateDoc (name, doc)

Arguments:

- name - database name
- docID - document to update

Returns JSON result statement

- createDesignDoc (name, docName, designDoc)

Arguments:

- name - database name
- docName - design doc name
- designDoc - JSON design document to create

Returns JSON result statement

- search (name, designDocName, viewName, limit, include\_docs, query)

Arguments:

- name - database name
- designDocName - design doc name
- viewName - view name
- limit - limit on number of results
- include\_docs - whether or not to include the full body of the docs in the return

- query - the query

Returns JSON document with results

- getView (name, designDocName, viewName, group)

Arguments:

- name - database name
- designDocName - design doc name
- viewName - view name
- group - if results must be grouped

Returns JSON documents from the view

## Using the Cloudant adapter in your app

To use the adapter in your app, copy the `CloudantAdapter` folder to your MobileFirst app.

Extend the adapter as needed with the functionality for your app.

## Sample application

Click to download

(<http://public.dhe.ibm.com/software/products/en/MobileFirstPlatform/docs/v700/CloudantAdapterDemoProject.zip>)  
the Studio project.