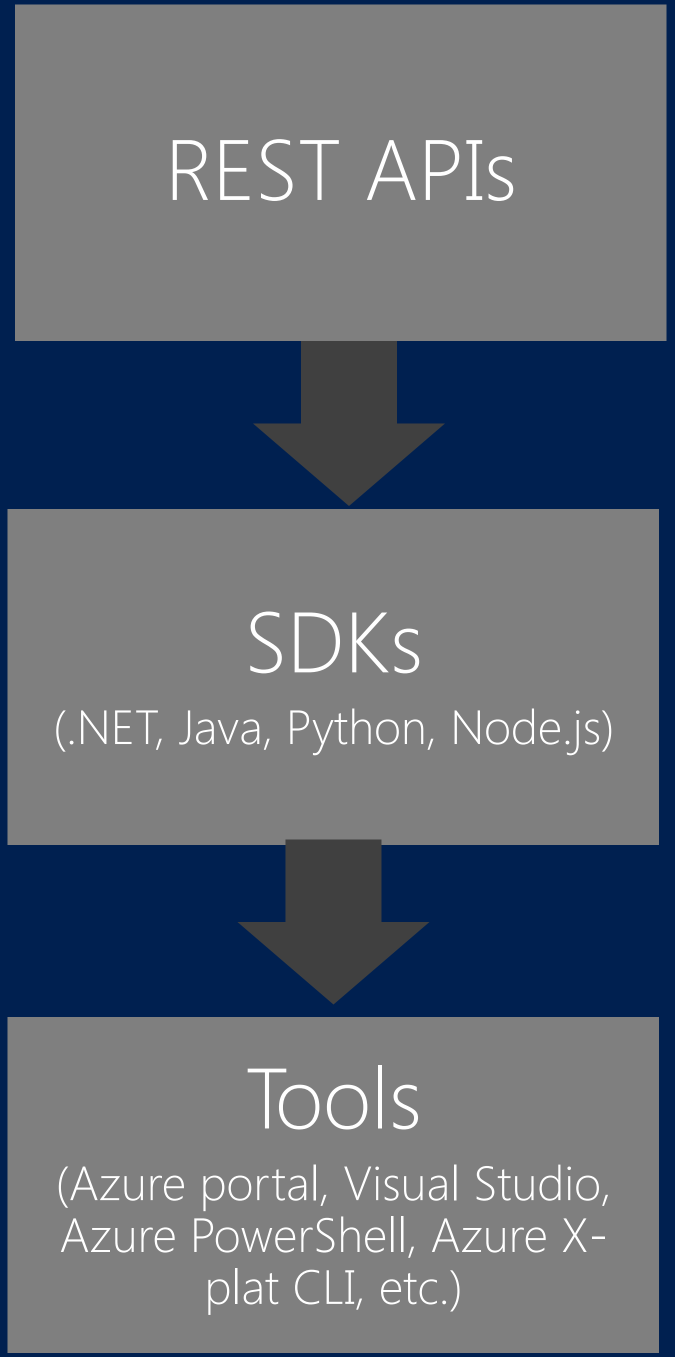
Azure Data Lake Store & Analytics REST APIs, SDKs, & CLIs

Mithun Prasad, PhD

Senior Program Manager @ Microsoft

Overview

- All Data Lake service features exposed via REST APIs
  - Multiple REST APIs per service
  - REST APIs use OAuth 2.0 for authentication
  - Fully documents
- All DataLake SDKs use the REST APIs
  - SDKs for auto-generated from REST APIs + Some custom SDK code (for some languages)
  - Enables development with multiple languages on multiple platforms
  - C#, Python, Java
- All DataLake Tools use the SDKs or the REST APIs
- Building an Application?
  - We recommend using the SDKs first – they make app development much simpler.



REST APIs

SDKs

(.NET, Java, Python, Node.js)

Tools

(Azure portal, Visual Studio, Azure PowerShell, Azure X-plat CLI, etc.)

Microsoft

REST API Endpoints

Data Lake Analytics

Management
Create and manage ADLA accounts

Jobs
Submit and manage jobs

Catalog
Explore catalog items

Data Lake Store

Management
Create and manage ADLS accounts

File System (WebHDFS)
Upload, download, list, delete, rename, append

Microsoft

Data Lake Analytics API Feature Set

Accounts

- Create
- List
- Update properties
- Delete

Data Sources

- Add
- List
- Update
- Delete

Jobs

- Get Details
- List
- Submit
- Cancel

Catalog Items (Tables, Views, etc.)

- List
- Update item

Catalog Credentials

- Create
- List
- Delete

Data Lake Store Feature Set

Accounts

- Create
- List
- Update properties
- Delete

Transferring Data

- Upload into store from local disk
- Download from store to local disk

Files and Folders

- List
- Create
- Move
- Delete
- Exists

Security

- Get ACLs
- Update ACLs
- Get Owner
- Set Owner

File Content

- Set content
- Append content
- Get content
- Merge files

.NET SDKs

Analytics

NuGet (nuget.org)
- Microsoft.Azure.Management.DataLake.Analytics

NPM (npmjs.com)
- azure-arm-datalake-analytics

Maven (maven.org)
- azure-mgmt-datalake-analytics
- azure-mgmt-datalake-store

Store

NuGet (nuget.org)
- Microsoft.Azure.Management.DataLake.Store
- Microsoft.Azure.Management.DataLake.StoreUploader

NPM (npmjs.com)
- azure-arm-datalake-store

Maven (maven.org)
- azure-mgmt-datalake-store

High-Performance, Multipart Uploader

- Hand-written uploader – Uploads multiple parts of the same file in parallel using multiple threads
- MUCH Faster than serial upload
- Used by:
  - ADL Tools for Visual Studio
  - ADL PowerShell cmdlets
- Available for .NET SDK and the Python SDK

Futures

- Java SDK Support coming

Microsoft

Using The SDKs

Workflow

1. Authenticate - Get an OAuth token from Azure Active Directory (Azure AD, AAD). ADL Uses OAuth V2.
2. Setup - Create a service client object
3. Do work - Call methods on the client object

Getting That OAuth Token

- You need to provide
  - Subscription name or ID
  - Tenant name or ID
  - Client ID

Authenticating With AAD (C#)

(Getting the OAuth tokens)

```
var tenantId = "common"; //Replace this if you know your Tenant ID.

// Replace these if you have a registered Azure AD Application.
var appClientId = "fbc174f6-bfd2-423a-b72a-e9393e789c21";
var appRedirectUri = new Uri("https://localhost");

// Authenticate
var authContext = new AuthenticationContext("https://login.microsoftonline.com/" + tenantId);
var tokenAuthResult = authContext.AcquireToken("https://management.core.windows.net/",
         appClientId, appRedirectUri, PromptBehavior.Auto, UserIdentifier.AnyUser);
var tokenCreds = new TokenCredentials(tokenAuthResult.AccessToken);
```

Instantiating The Service Client Objects (C#)

```
string subscriptionId = "83daeeec-f16c-47f8-9dc4-6ff1ebf9feb3";

var adlaClient = new DataLakeAnalyticsAccountManagementClient(tokenCreds);
adlaClient.SubscriptionId = subscriptionId;
```

Using The Service Client Object (C#)

```
var response = adlaJobClient.Job.List(adlaAccountName);
var jobs = new List<JobInformation>(response);

while (response.NextPageLink != null)
{
    response = adlaJobClient.Job.ListNext(response.NextPageLink);
    jobs.AddRange(response);
}
```

Sample Usage Of The SDKs Here

- .NET
  - https://github.com/Azure-Samples/data-lake-dotnet-client
- Python
  - https://github.com/Azure-Samples/data-lake-python-client

Demo & Lab:
Using Azure .NET SDK for ADL

Microsoft

Azure PowerShell Cmdlets

After installing Azure PowerShell, list our cmdlets:

```
    Get-Command *Azure*DataLake*
```

See how to use any of our cmdlets:

```
    help New-AzureRmDataLakeStoreItem
```

Azure CLI Commands

- After installing Azure CLI, list our commands:

```
azure datalake -h
```

- See how to use any of our commands:

```
azure datalake store filesystem create -h
```

Microsoft

PowerShell Tips

- Locating the Folder the running script is in

```
$script_folder = Split-Path $myinvocation.mycommand.path –Parent
```

- Location of mydocs

```
$mydocs = [environment]::getfolderpath("mydocuments")
```

- Useful when programming (place at top of scripts)

```
Set-StrictMode -Version 2
$ErrorActionPreference = "Stop"
```

Saving And Loading Your AzureRm Profile

- Saves (Subscription, Tenant)

```
Save-AzureRmProfile -Path “myprofile.json”
```

- Load it back

```
Select-AzureRmProfile -Path “myprofile.json”
```

Demo & Lab:
Using Azure PowerShell With Azure Data Lake

Microsoft

http://aka.ms/AzureDataLake

Microsoft