# **Power BI Data Retention Service API Design Document**

## **✨ Overview**

Retention is a core functionality of this service, and this document is intended to clearly guide developers in implementing and integrating with the API.

The service is designed to execute and cache the results of Power BI DAX queries for a configurable period of time. This allows users to avoid re-running long and expensive queries and instead retrieve persisted results on demand, improving performance and reliability of Power BI reports.

The **Power BI Data Retention Service API** enables users to:

* Register and manage reusable DAX-based queries (Power BI Queries)
* Group queries into logical containers (QuerySpaces)
* Execute individual queries or entire QuerySpaces on demand or on a schedule
* Invalidate (purge) retained query data when no longer needed
* Reuse and share queries across multiple teams and use cases

The design promotes modularity, reusability, and clarity in managing recurring or long-running Power BI queries at scale.

## **Core Concepts**

### **QuerySpace**

A **QuerySpace** is a logical grouping of Power BI Queries. Think of it as a folder or workspace that collects queries tied to a specific business function, team, or report. QuerySpaces make it easier to:

* Organize DAX queries by domain or team
* Trigger batch executions
* Track execution history and status

**Attributes:**

* id: BIGINT
* name: string (unique)
* description: string
* created\_at: timestamp
* updated\_at: timestamp
* updated\_by: string
* last\_executed\_at: timestamp (nullable)
* last\_status: string (e.g., "Success", "Failed", "Pending")

### **Power BI Query**

A **Power BI Query** encapsulates the DAX logic or metadata used to extract, summarize, and cache data. Queries can be defined as raw DAX expressions or built from attribute selections.

**Attributes:**

* id: BIGINT
* name: string
* description: string
* query\_type: enum (DaxQuery, AttributeList)
* dax\_query: string (required if type is DaxQuery)
* attributes: text[] (required if type is AttributeList)
* execution\_frequency: enum (Manual, Daily, Weekly)
* retention\_period\_days: int
* created\_at: timestamp
* updated\_at: timestamp
* updated\_by: string
* last\_executed\_at: timestamp (nullable)
* last\_status: string (e.g., "Success", "Failed", "Pending")

### **Query Reuse**

Each Power BI Query may be:

* Added to one or many QuerySpaces
* Shared or reused across teams
* Independently managed while remaining linked to multiple use cases

This relationship is stored in the query\_space\_queries join table.

## **API Endpoints**

### **QuerySpaces Management**

#### **POST /powerbi/api/queryspaces**

Create a new QuerySpace.

**Request Payload:**

{  
 "name": "Marvel-Trend-Analysis",  
 "description": "Query space for Marvel trend metrics"  
}

#### **PUT /powerbi/api/queryspaces/{name}**

Update an existing QuerySpace.

**Request Payload:**

{  
 "description": "Updated description for this query space",  
 "updated\_by": "developer.username"  
}

#### **GET /powerbi/api/queryspaces**

List all QuerySpaces.

**Sample Response:**

[  
 {  
 "id": 1,  
 "name": "Marvel-Trend-Analysis",  
 "description": "Query space for Marvel trend metrics",  
 "created\_at": "2025-03-01T10:00:00Z",  
 "updated\_at": "2025-04-01T12:00:00Z",  
 "updated\_by": "julio.diaz",  
 "last\_executed\_at": "2025-04-06T18:00:00Z",  
 "last\_status": "Success"  
 }  
]

#### **GET /powerbi/api/queryspaces/{name}**

Return details about a specific QuerySpace and its linked queries.

#### **DELETE /powerbi/api/queryspaces/{name}**

Delete a QuerySpace and remove all associations.

#### **POST /powerbi/api/queryspaces/{name}/queries/{id}**

Link an existing Power BI Query to a QuerySpace.

#### **DELETE /powerbi/api/queryspaces/{name}/queries/{id}**

Unlink a Power BI Query from a QuerySpace.

### **Power BI Queries Management**

#### **POST /powerbi/api/queries**

Create a new query.

**Request Payload:**

{  
 "name": "DailyTrendQuery",  
 "description": "Executes daily DAX trend logic",  
 "query\_type": "DaxQuery",  
 "dax\_query": "EVALUATE SUMMARIZE(...)",  
 "execution\_frequency": "Daily",  
 "retention\_period\_days": 90  
}

#### **GET /powerbi/api/queries/{id}**

Retrieve full details for a specific Power BI Query.

**Sample Response:**

{  
 "id": 101,  
 "name": "DailyTrendQuery",  
 "description": "Executes daily DAX trend logic",  
 "query\_type": "DaxQuery",  
 "dax\_query": "EVALUATE SUMMARIZE(...)",  
 "attributes": null,  
 "execution\_frequency": "Daily",  
 "retention\_period\_days": 90,  
 "created\_at": "2025-03-01T10:00:00Z",  
 "updated\_at": "2025-04-01T12:00:00Z",  
 "updated\_by": "julio.diaz",  
 "last\_executed\_at": "2025-04-06T18:00:00Z",  
 "last\_status": "Success"  
}

#### **PUT /powerbi/api/queries/{id}**

Update an existing Power BI Query.

#### **DELETE /powerbi/api/queries/{id}**

Delete a Power BI Query.

### **Execution Endpoints**

#### **POST /powerbi/api/queryspaces/{name}/execute**

Manually trigger execution of all linked queries in the QuerySpace.

#### **POST /powerbi/api/queries/{id}/execute**

Manually execute an individual Power BI Query.

**Optional Payload:**

{  
 "targetDate": "2025-04-07",  
 "triggeredBy": "manual",  
 "forceRefresh": true  
}

### **Data Invalidation Endpoints**

#### **DELETE /powerbi/api/queryspaces/{name}/data**

Permanently delete retained results for all queries in the QuerySpace.

#### **DELETE /powerbi/api/queries/{id}/data**

Invalidate retained results for a specific Power BI Query.

**Optional Payload:**

{  
 "reason": "Schema change - column removed",  
 "performedBy": "admin.user"  
}

## **DTO Classes (C# Style)**

public class QuerySpaceDto  
{  
 public long Id { get; set; }  
 public string Name { get; set; }  
 public string? Description { get; set; }  
 public string? UpdatedBy { get; set; }  
 public DateTime CreatedAt { get; set; }  
 public DateTime UpdatedAt { get; set; }  
 public DateTime? LastExecutedAt { get; set; }  
 public string? LastStatus { get; set; }  
 public List<PowerBIQueryDto> Queries { get; set; } = new();  
}  
  
public class PowerBIQueryDto  
{  
 public long Id { get; set; }  
 public string Name { get; set; }  
 public string? Description { get; set; }  
 public QueryType QueryType { get; set; }  
 public string? DaxQuery { get; set; }  
 public List<string>? Attributes { get; set; }  
 public ExecutionFrequency ExecutionFrequency { get; set; }  
 public int RetentionPeriodInDays { get; set; }  
 public string? UpdatedBy { get; set; }  
 public DateTime CreatedAt { get; set; }  
 public DateTime UpdatedAt { get; set; }  
 public DateTime? LastExecutedAt { get; set; }  
 public string? LastStatus { get; set; }  
}  
  
public enum QueryType { DaxQuery, AttributeList }  
  
public enum ExecutionFrequency { Manual, Daily, Weekly }  
  
public class ExecuteQueryOptions  
{  
 public DateTime? TargetDate { get; set; }  
 public string? TriggeredBy { get; set; }  
 public bool ForceRefresh { get; set; } = false;  
}  
  
public class InvalidationRequest  
{  
 public string? Reason { get; set; }  
 public string? PerformedBy { get; set; }  
}

## **Considerations**

* ✅ Secure all mutation endpoints with RBAC
* ✅ Log all executions, deletions, and updates with audit trails
* ✅ Support asynchronous job execution and background processing
* ✅ Add webhook or notification support for completion or failure
* ✅ Validate data structure compatibility before executing queries
* ✅ Use partition pruning in Databricks where possible for performance

## **Database Schema Summary**

### **Table: query\_spaces**

* id (BIGINT)
* name (VARCHAR(255))
* description (TEXT)
* created\_at (TIMESTAMP)
* updated\_at (TIMESTAMP)
* updated\_by (VARCHAR(255))
* last\_executed\_at (TIMESTAMP)
* last\_status (VARCHAR(255))

### **Table: power\_bi\_queries**

* id (BIGINT)
* name (VARCHAR(255))
* description (TEXT)
* query\_type (VARCHAR(255))
* dax\_query (TEXT)
* attributes (TEXT[])
* execution\_frequency (VARCHAR(255))
* retention\_period\_days (INT)
* created\_at (TIMESTAMP)
* updated\_at (TIMESTAMP)
* updated\_by (VARCHAR(255))
* last\_executed\_at (TIMESTAMP)
* last\_status (VARCHAR(255))

### **Table: query\_space\_queries**

* query\_space\_id (BIGINT)
* power\_bi\_query\_id (BIGINT)
* PRIMARY KEY (query\_space\_id, power\_bi\_query\_id)