**Power BI Embedded 的工作流及相关REST API命令介绍**

Power BI Embedded服务使用HTTP请求实现对REST API的调用，完成创建Power BI工作区，将.pbix报表文件导入工作区，数据源连接，在网页中进行身份验证和托管（嵌入）报表以及将报表嵌入网页等一系列。这些工作均可在PowerBI-Cli工具中完成，本文则主要介绍Power BI Embedded工作流所用到的REST API命令。

本文所采用的工具为Postman V4.10.3的64位版本。本文中所介绍的工作流可参考[https: //www.azure.cn/documentation/articles/power-bi-embedded-iframe](https://www.azure.cn/documentation/articles/power-bi-embedded-iframe); Power BI Embedded相关的REST API命令介绍可参考[https: //msdn.microsoft.com/library/azure/mt711507.aspx](https://msdn.microsoft.com/library/azure/mt711507.aspx); PowerBI-Cli工具介绍可参考<https://github.com/Microsoft/PowerBI-cli>。

**前提**

1. 已创建Power BI Embedded的工作区集合(workspace collection);
2. 已使用Power BI Desktop创建.pbix报表文件;
3. 已安装Postman V4.10.3版本(旧的版本在执行有些功能时会报错)。

**参数**

文中所提到的一些Uri参数含义如下:

{collectionName}: 工作区集合名

{Key}: 工作区集合密钥

{workspaceID}: 工作区ID

{datasetDisplayName}: 数据集名

{importID}: 导入ID

{datasetID}: 数据集ID

{reportID}: 报表ID

{createDateTime}: 创建导入的时间戳，格式如“2016-04-25T14:09:53.803”。

{updatedDateTime}: 更新导入的时间戳，格式同上。

{datasourceID}: 数据源ID

{gatewayID}: 网关ID

{userName}: 数据源的用户名

{password}: 数据源的密码

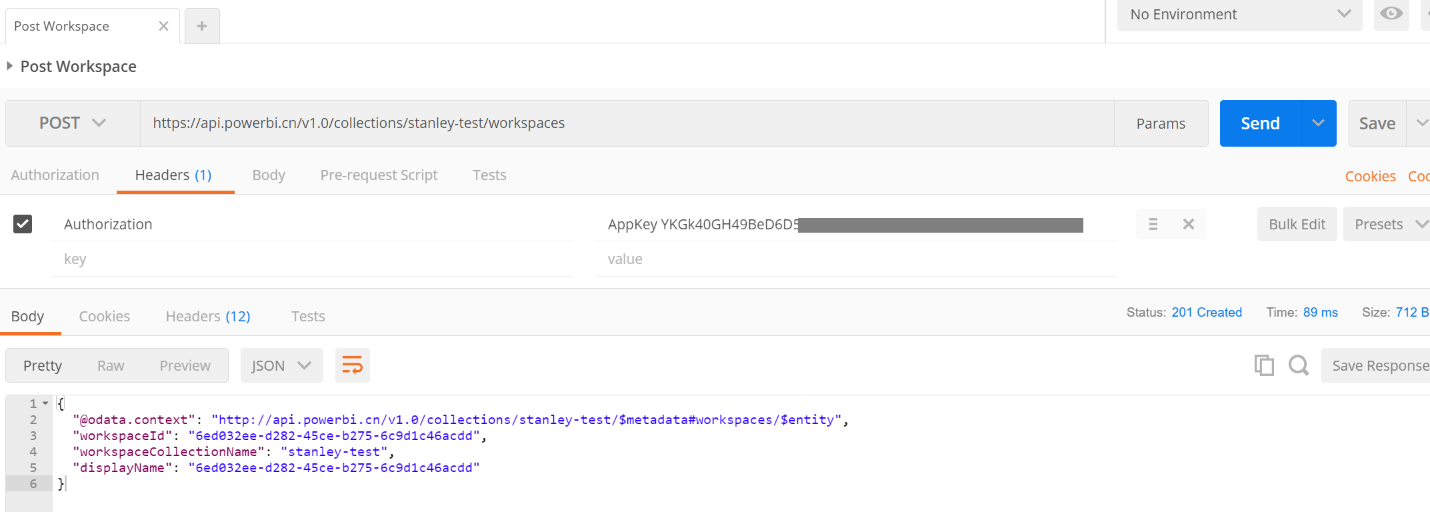
**工作流**

Power BI Embedded工作流的主要步骤如下:

1. 创建 Power BI 工作区集合并获取访问密钥（预配）
2. 使用 Power BI Desktop 创建 .pbix 文件
3. 创建 Power BI 工作区
4. 将 .pbix 文件导入工作区
5. 获取导入状态(GET Import)
6. 获取网关及数据源信息(Get BoundGatewayDatasources)
7. 更新数据源凭据(PATCH GatewayDatasource)
8. 获取嵌入URL和报表ID(GET Reports)
9. 创建令牌
10. 在Demo页面上展示

其中1和2分别在Azure Portal上和Power BI Desktop中完成，9未包含在[https: //msdn.microsoft.com/library/azure/mt711507.aspx](https://msdn.microsoft.com/library/azure/mt711507.aspx)中，因此本文主要介绍其他步骤及相应的REST API命令。

**创建Power BI工作区(POST Workspace)**



请求: POST http: //api.powerbi.cn/v1.0/collections/{workspaceCollectionName}/workspaces

标头: Authorization: AppKey {Key}

正文: 无

返回内容:

{

"@odata.context": "http: //api.powerbi.cn/v1.0/collections/{workspaceCollectionName}/workspaces",

Workspace {

workspaceID: string

workspaceCollectionName: string

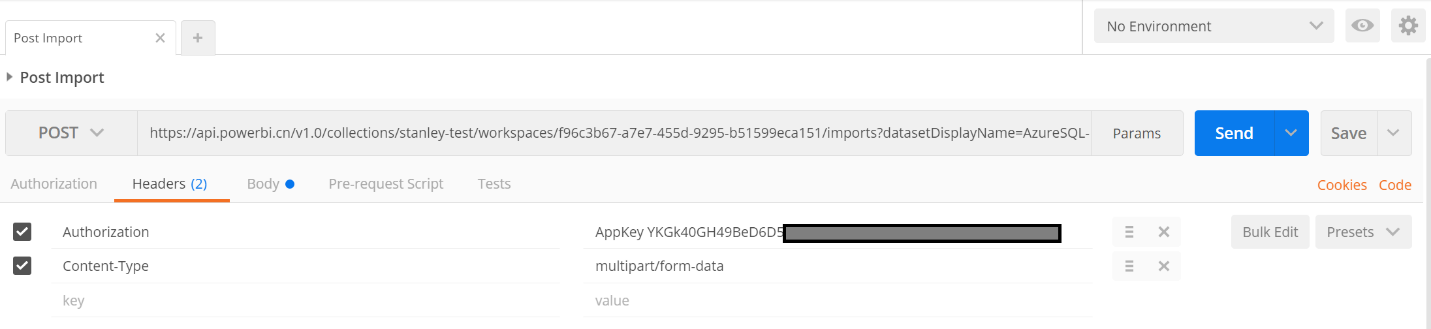
displayName: string

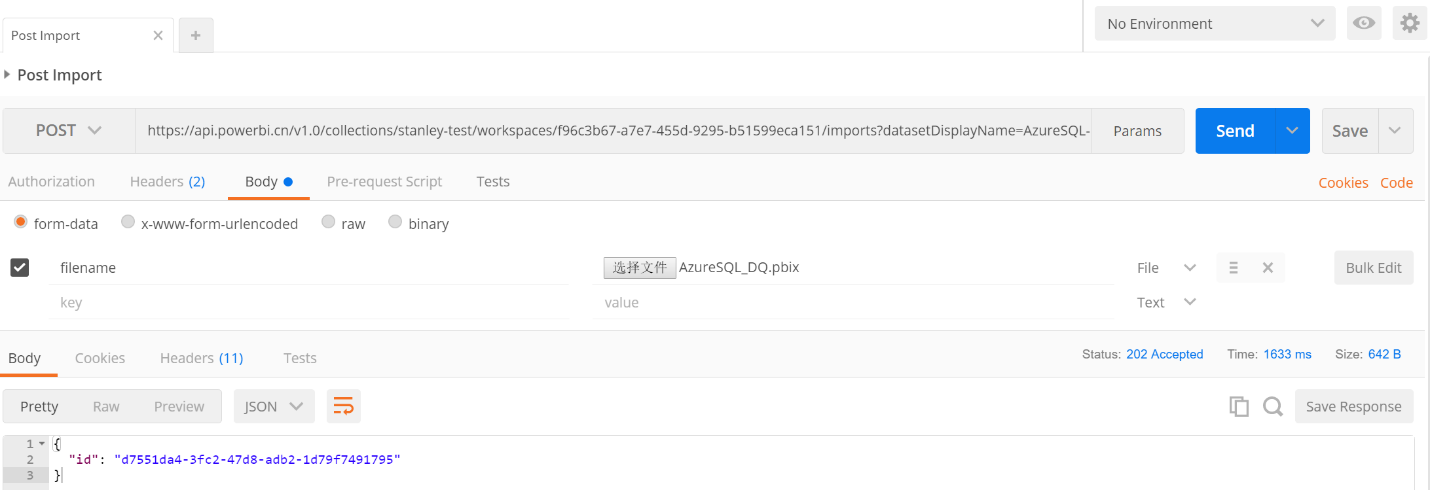
}

}

此命令在现有的工作区集合中创建新的工作区并返回所创建的工作区信息，等效于PowerBI-Cli工具中的powerbi create-workspace命令，其返回的json内容中的workspace ID将用于后续的API调用。

**将.pbix文件导入工作区(POST Import)**





请求: POST https: //api.powerbi.cn/v1.0/collections/{collectionName}/workspaces/{workspaceID}/imports?datasetDisplayName={datasetDisplayName}

标头: Authorization: AppKey {Key}

Content-Type: multipart/form-data

正文: Content-Disposition: form-data;

name="filename"; filename="xxx.pbix"

返回内容: 202 Accepted

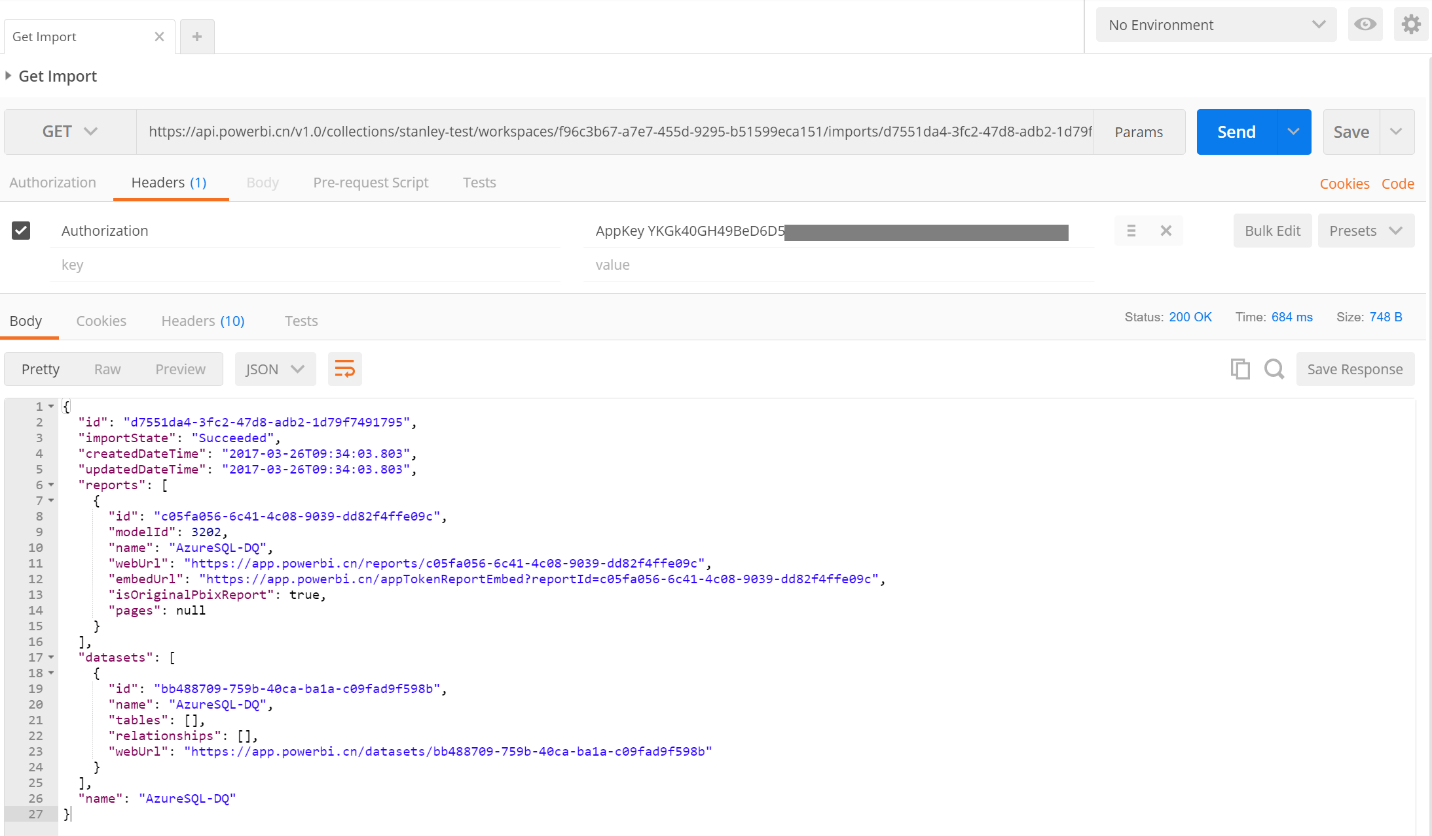
{

"id": "{importID}"

}

此命令将.pbix文件导入工作区并返回一个导入ID，等效于PowerBI-Cli工具中的powerbi import命令。执行此命令后的HTTP返回状态为”202 Accepted”，表示请求得到接受但未说明是否被执行，可根据返回的导入ID用Get Import命令获知导入状态。而当用PowerBI-Cli工具进行文件导入时，导入状态将直接在powerbi import命令执行时返回。

**获取导入状态(GET Import)**



请求: GET https://api.powerbi.cn/v1.0/collections/{collectionName}/workspaces/{workspaceID}/imports/{importId}

标头: Authorization: AppKey {Key}

正文: 无

返回内容: 200 OK

{

"id":"{importID}",

"importState":"Succeeded",

"createdDateTime":"{createDateTime}",

"updatedDateTime":"{updatedDateTime}",

"reports":[

{

"id":"{reportID}",

"name":"{datasetDisplayName} ",

"webUrl":"https://app.powerbi.cn/reports/{reportID}",

"embedUrl":"https://app.powerbi.cn/appTokenReportEmbed?reportId= {reportID}"

}

],

"datasets":[{

"id":"{datasetID}",

"name":{datasetDisplayName},

"tables":[],

"relationships":[],

"webUrl":"https://embedded.powerbi.cn/datasets/{datasetID}"}],

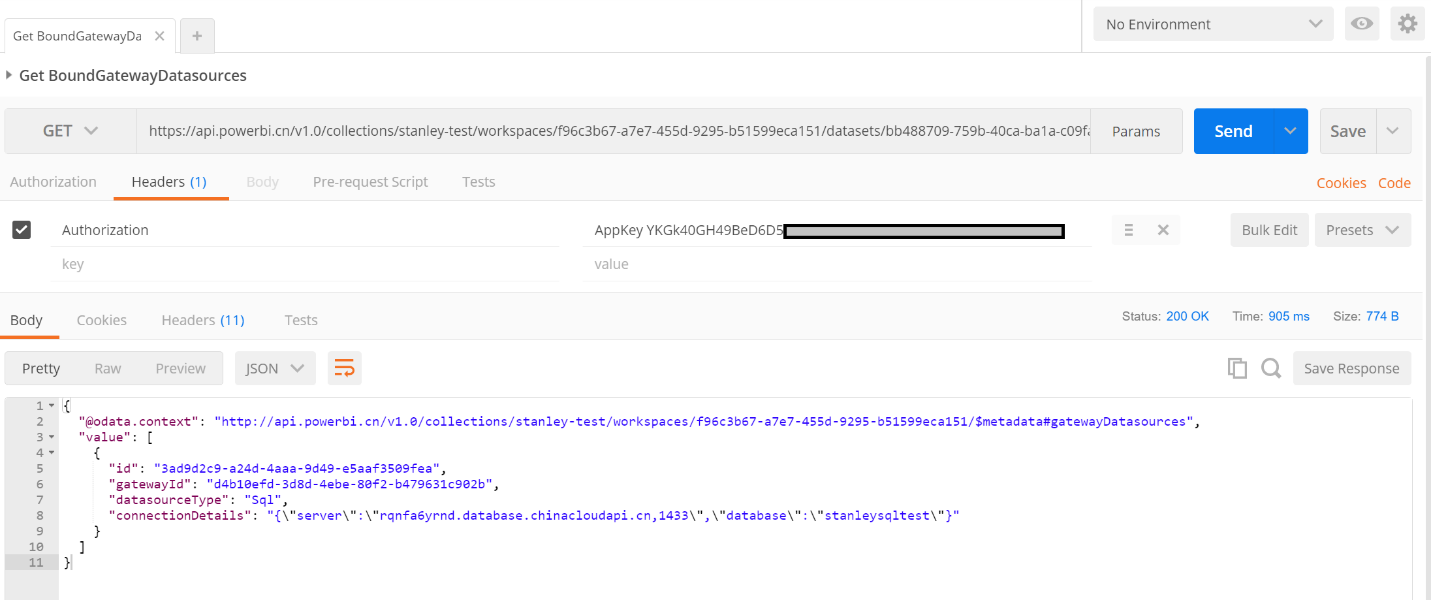
"name":{datasetDisplayName}

}

此命令用于根据importID查询导入情况，导入成功的话将返回以上的json内容，否则importState将为不同的状态(如”Publishing”或”Failed”)。

导入的.pbix文件中不包含数据源的凭据。当导入的.pbix文件是由DirectQuery方式创建时(即文件中没有数据集，只有针对数据源的查询部分)，其数据源是通过数据源ID和网关ID来进行唯一标识的，因此需要先用[GET BoundGatewayDatasources](https://msdn.microsoft.com/zh-cn/library/azure/mt711500.aspx)命令获取.pbix文件中的数据源ID及网关ID，然后再用PATCH GatewayDatasource命令对数据源凭据进行更新。连续执行这两个命令等效于PowerBI-Cli工具中的powerbi update-connection命令。

**获取网关及数据源信息(Get BoundGatewayDatasources)**



请求: GET https://api.powerbi.cn/v1.0/collections/{collectionName}/workspaces/{workspaceID}/datasets/{datasetID}/Default.GetBoundGatewayDatasources

标头: Authorization: AppKey {Key}

正文: 无

返回内容: 200 OK

{

"@odata.context": "http://api.powerbi.cn/v1.0/collections/{workspaceName}/workspaces/{workspaceID}/datasets/{datasetID}/$metadata#datasources",

"value":[

{

“id”:“{datasourceID}”

“gatewayId”:“{gatewayID}”

“datasourceType”: string

“connectionDetails”: string

“credentialType”: string

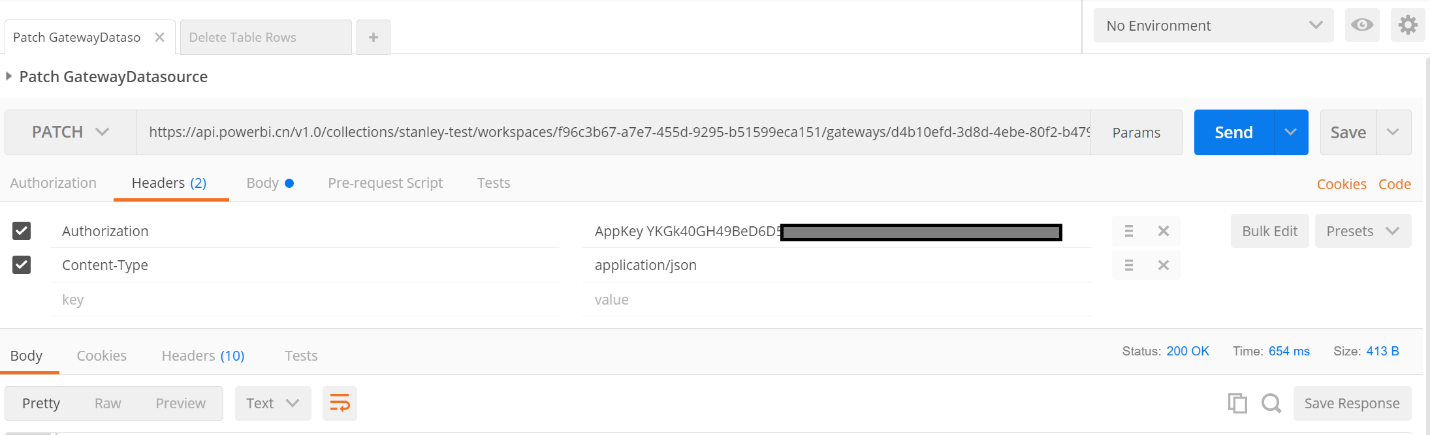
}

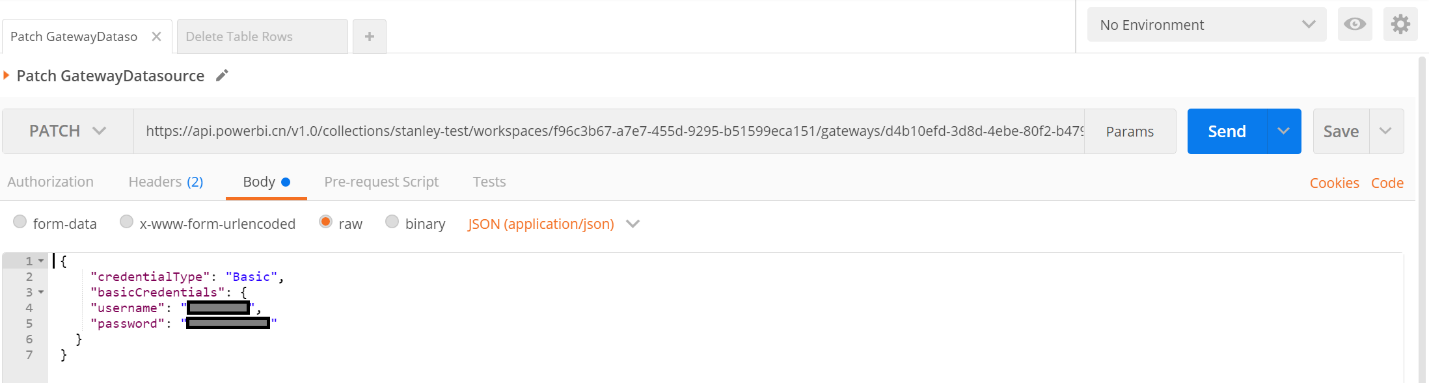
]

}

此命令用于根据datasetID进行查询，获取其数据源的信息，其中包含数据源ID和网关ID。当得到了这二者的值之后，可通过PATCH GatewayDatasource对数据源凭据进行更新。

**更新数据源凭据(PATCH GatewayDatasource)**





请求: PATCH https://api.powerbi.cn/v1.0/collections/{collectionName}/workspaces/{workspaceID}/gateways/{gatewayID}/datasources/{datasourceID}

标头: Authorization: AppKey {Key}

Content-Type: application/json

正文:

{

"credentialType": "Basic",

"basicCredentials": {

"username": "{username}",

"password": "{password}"

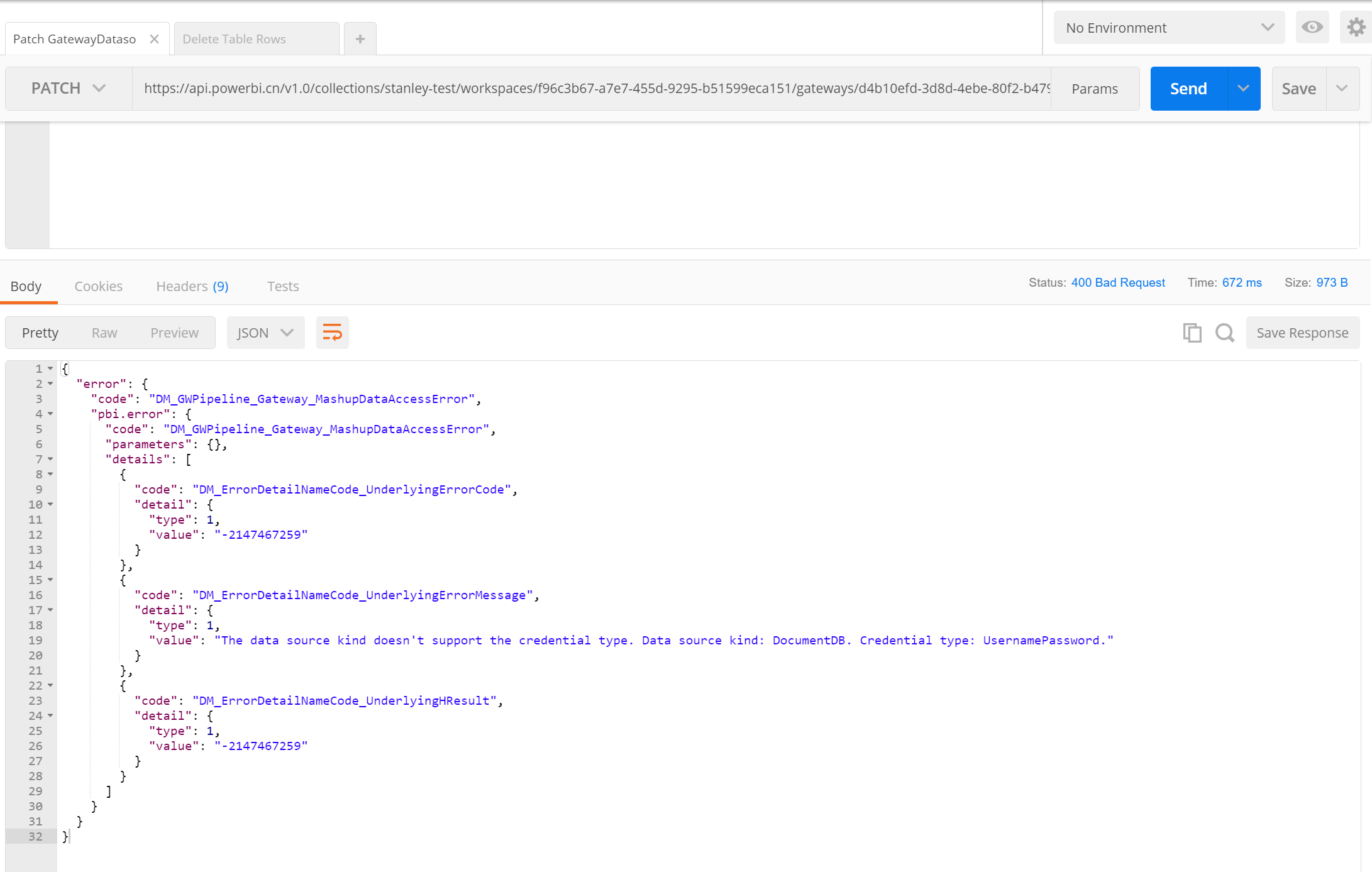
}

}

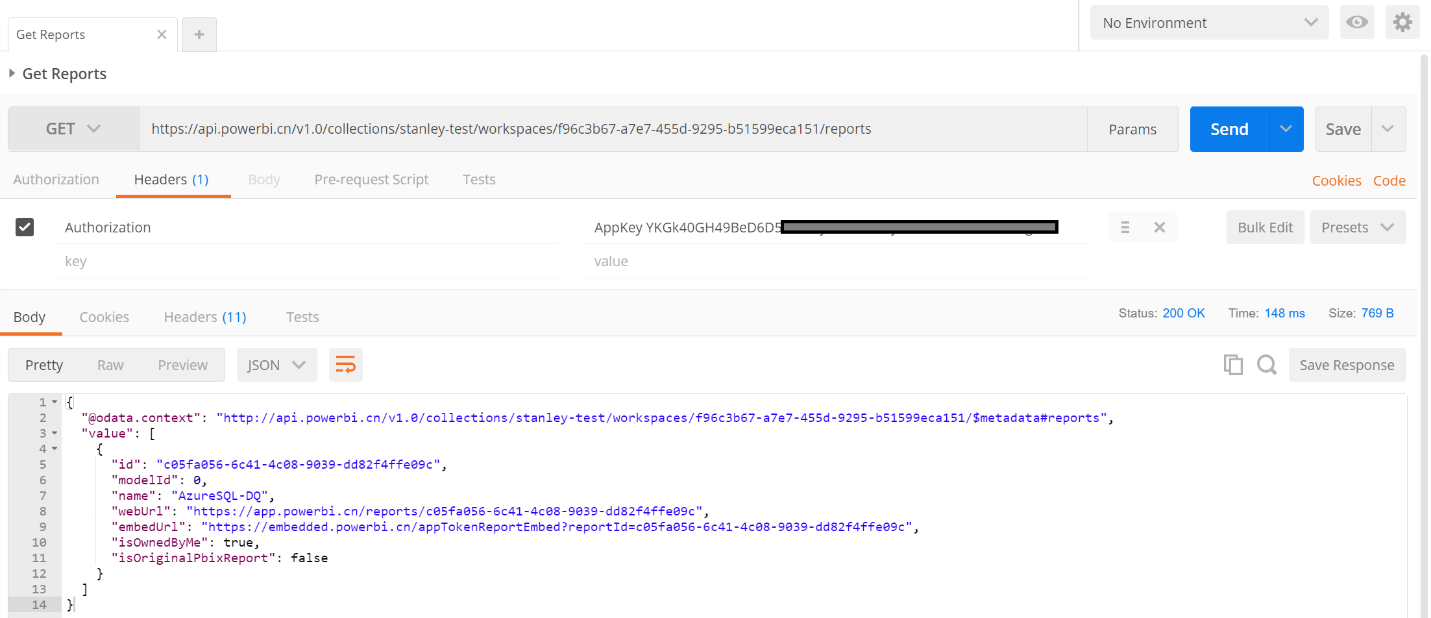
返回内容: 200 OK

此命令用于更新数据源的凭据信息，当返回状态为200 OK时，表示凭据更新成功，上传的用DirectQuery方式创建的.pbix文件可在demo页面上显示。

注意:如果用的是Import模式而非DirectQuery模式创建的.pbix文件，执行GET BoundGatewayDatasource命令也会返回gatewayID和datasourceID，但执行PATCH GatewayDatasource命令会返回400 Bad Request的状态，如下所示:



**获取嵌入URL和报表ID(GET Reports)**



请求: GET https://api.powerbi.cn/v1.0/collections/{collectionName}/workspaces/{workspaceID}/reports

标头: Authorization: AppKey {Key}

正文: 无

返回内容: 200 OK

{

"@odata.context":"https://api.powerbi.com/v1.0/collections/{WorkspaceName}/workspaces/{WorkspaceID}/$metadata#reports",

"value":[

{

"id":"{reportID}",

"name":"{datasetDisplayName}",

"webUrl":"https://app.powerbi.cn/reports/{reportID}",

"embedUrl":"https://embedded.powerbi.cn/appTokenReportEmbed?reportId={reportID}"

}

]

}

此命令用于获取导入的reportID和embedURI，等效于PowerBI-Cli工具中的powerbi get-reports命令。

**创建令牌**

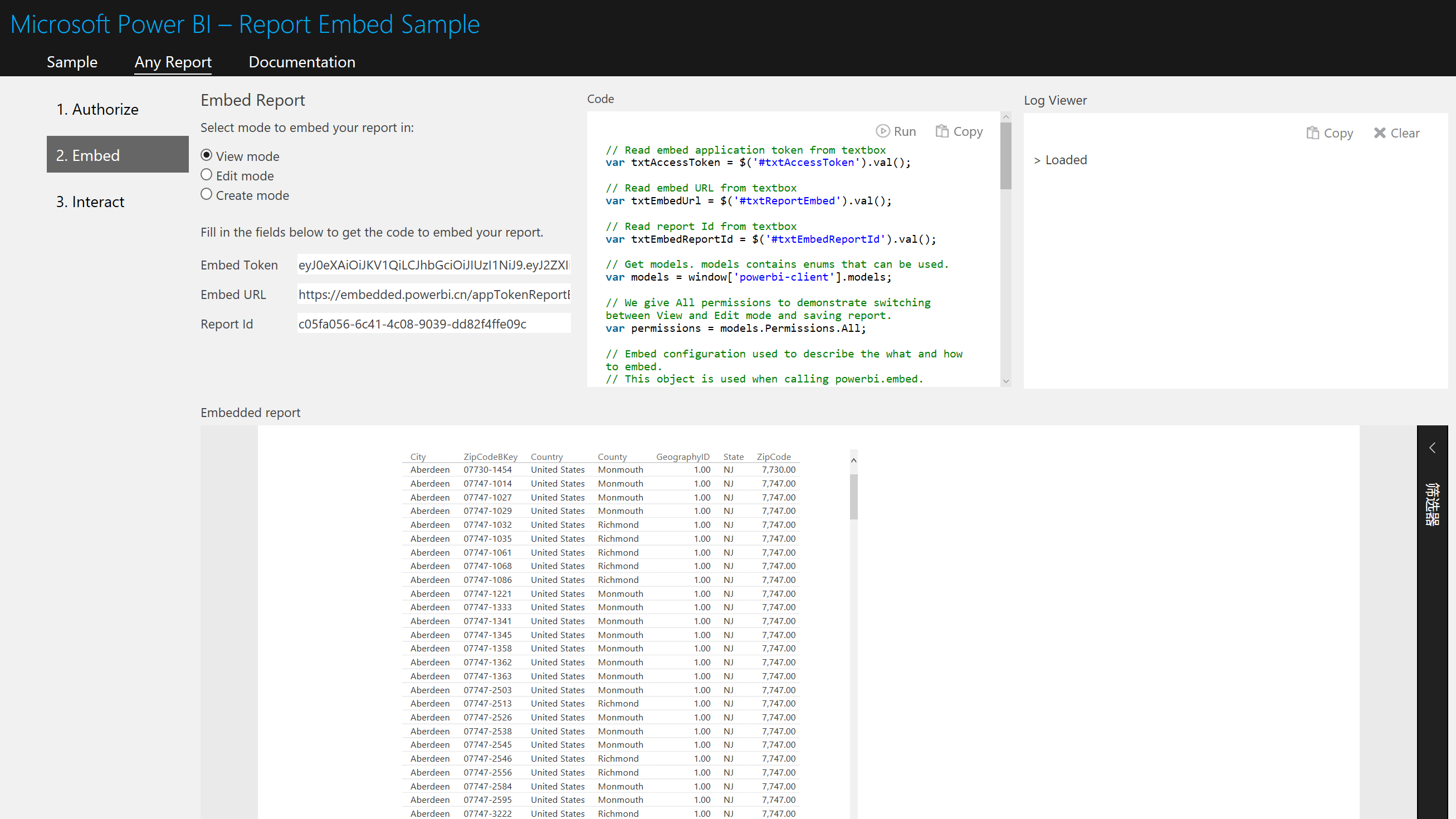
创建令牌等效于PowerBI-Cli工具中的powerbi create-embed-token命令，由于其机制相对复杂，且不在<https://msdn.microsoft.com/library/azure/mt711507.aspx>中，故本文不予详述，具体内容可参考以下几个链接:

令牌的创建: <https://www.azure.cn/documentation/articles/power-bi-embedded-iframe>

令牌的工作流: <https://www.azure.cn/documentation/articles/power-bi-embedded-app-token-flow/>

**在Demo页面上展示**

当reportID，embedURI和令牌均已获得后，便可在进行报表的内嵌展示，以下是本文介绍的AzureSQL\_DQ.pbix在官网demo页面<https://microsoft.github.io/PowerBI-JavaScript/demo/code-demo/index.html#>上的展示效果:



**其他的REST API命令**

除了以上介绍的部分外，Power BI Embedded所支持的其他REST API命令如下所示:

数据集和表格相关:

Delete Dataset

Delete Table Rows

Get Datasets

Get Dataset Schema

Get Tables

Post Datasets

Put Table Schema

数据源相关:

Get Datasources

Post SetAllConnections

导入相关:

Get Imports

工作区相关:

Get Workspaces

具体内容参见<https://msdn.microsoft.com/library/azure/mt711507.aspx>