Connect using Power Query and Azure DevOps functions

13/02/2018 • 5 minutos para ler • Colaboradores tudo

Neste artigo

VSTS.Feed
VSTS.Contents
VSTS.AccountContents
Related articles

Azure DevOps Services | Azure DevOps Server 2019

The Data Connector for Azure DevOps includes Power Query M functions which can be used by query authors. These functions can handle Azure DevOps specific requirements, such as authentication for you. This article describes the arguments for the functions and how to use them to connect to the Analytics service.

The following functions are provided:

Function	Description
VSTS.Feed	Replacement for Power Query M function OData.feed. Allows users to easily execute OData queries against the Analytics service.
VSTS.Contents	Deprecated. Instead use VSTS.AccountContents.
VSTS.AccountContents	Replacement for Power Query M function Web.Contents. Intended for more advanced scenarios, VSTS.AccountContents returns the contents downloaded from the URL for the Analytics service as a binary value.

VSTS.Feed

Allows for users to easily execute OData queries against the Analytics service.

The `VSTS.Feed` function has the same arguments, options and return value format as `OData.Feed`. For more information, see Power Query (M) Formula Reference - OData.Feed.

If you are already using `OData.Feed` to access data from the Analytics service, you can replace it with `VSTS.Feed` to leverage Data Connector authentication.
This will also inform Power BI that these requests are referencing the same data source and you will be able to combine the data without violating the single data source constraints for refreshing datasets in the PowerBI.com.

'VSTS.Feed' provides a subset of the arguments and options available through 'OData.Feed'. The specific limitations are outlined in the following table:

Arguments for VSTS.Feed

Argument	Description

`url`	A URL to the OData endpoint of the Analytics service.
`options`	An options record to control the behavior of this function.

Options fields for VSTS.Feed

Field	Description
`MaxSize`	Controls the max size of the table the client is interested in. If request exceeds this limit then the server can fail the request immediately. Default value is zero, which tells the server to use its default value.
`Query`	Programmatically add query parameters to the URL without having to worry about escaping.
`ShowHidden`	Flag which indicates if all tables (including snapshot) should be shown.
`ShowOnlyReportingViews`	Flag which indicates if only `EntitySets` with `IsReportingView` annotation should be returned.
`Timeout`	Specifying this value as a duration will change the timeout for an HTTP request. The default value is 600 seconds.
`Version`	Version of the data model. This option is primary for diagnostics.

Examples for VSTS.Feed

Use `VSTS.Feed` function to count the number of work items in a project.

1. Create a new blank query and choose "Advanced Editor" in the Power BI Query Editor.
2. In the editor add the text below to load the feed for the `fabrikam-fiber-inc` organization and `Fabrikam-Fiber-Git` for the project name using full URL to OData endpoint.
3. Select relevant columns using "Choose Columns", in this case select `Count`.

Basic Query:

Copiar

```
let
    Source = VSTS.Feed("https://fabrikam-fiber-inc.analytics.visualstudio.com/Fabrikam-Fiber-
Git/_odata/v1.0-preview/"
        & "WorkItems?$apply=aggregate($count as Count)")
in
    Source
```

Query with Columns Selected:

Copiar

```
let
    Source = VSTS.Feed("https://fabrikam-fiber-inc.analytics.visualstudio.com/Fabrikam-Fiber-
Git/_odata/v1.0-preview/"
        & "WorkItems?$apply=aggregate($count as Count)"),
    #"Removed Other Columns" = Table.SelectColumns(Source,{"Count"})
```

```
in
    #"Removed Other Columns"
```

Use `VSTS.Feed` function to load a count of User Stories for each Iteration Path.

1. Create a new blank query and click on "Advanced Editor" in the Power BI Query Editor.
2. In the editor add the text below to load the feed for the `fabrikam-fiber-inc` organization and `Fabrikam-Fiber-Git` project using full URL to OData endpoint.
3. Select relevant columns using "Choose Columns", in this case expand 'Iteration' and select 'Iteration Path' then select `Count`.

Basic Query:

Copiar

```
let
    #"Source" = VSTS.Feed("https://fabrikam-fiber-inc.analytics.visualstudio.com/Fabrikam-Fiber-
Git/_odata/v1.0-preview/"
        & "WorkItems?$apply=groupby((Iteration/IterationPath), aggregate($count as Count))")
in
    #"Source"
```

Query with Columns Selected:

Copiar

```
let
    #"Source" = VSTS.Feed("https://fabrikam-fiber-inc.analytics.visualstudio.com/Fabrikam-Fiber-
Git/_odata/v1.0-preview/"
        & "WorkItems?$apply=groupby((Iteration/IterationPath), aggregate($count as Count))"),
    #"Expanded Iteration" = Table.ExpandRecordColumn(Source, "Iteration", {"IterationPath"},
{"Iteration.IterationPath"}),
    #"Removed Other Columns" = Table.SelectColumns(#"Expanded Iteration",{"Count", "Itera-
tion.IterationPath"})
in
    #"Removed Other Columns"
```

Use VSTS.Feed function to load detailed information about bugs.

1. Create a new blank query and click on "Advanced Editor" in the Power BI Query Editor
2. In the editor, add the text below to load the feed for the `fabrikam-fiber-inc` organization and `Fabrikam-Fiber-Git` project using full URL to OData endpoint.
3. Select relevant columns using "Choose Columns", in this case select `WorkItemID` and `State`.

Basic Query:

Copiar

```
let
    #"Source" = VSTS.Feed("https://fabrikam-fiber-inc.analytics.visualstudio.com/Fabrikam-Fiber-
Git/_odata/v1.0-preview/"
        & "WorkItems?$select=WorkItemId,State&$filter=WorkItemType eq 'Bug'")
in
    #"Source"
```

Query with Columns Selected:

Copiar

```
let
    #"Source" = VSTS.Feed("https://fabrikam-fiber-inc.analytics.visualstudio.com/Fabrikam-Fiber-
Git/_odata/v1.0-preview/"
        & "WorkItems?$select=WorkItemId,State&$filter=WorkItemType eq 'Bug'"),
    #"Removed Other Columns" = Table.SelectColumns(Source,{"WorkItemId", "State"})
in
    #"Removed Other Columns"
```

VSTS.Contents

VSTS.Contents is deprecated and will be removed in an upcoming release. Instead, use VSTS.AccountContents.

VSTS.AccountContents

Advanced function which returns the contents downloaded from the URL for the Analytics service as a binary value.

The `VSTS.AccountContents` function has the same arguments, options and return value format as `Web.Concents`. For more information please refer to: Power Query (M) Formula Reference - Web.Contents.

If you are already using `Web.Contents` to access data from the Analytics service (REST API or OData), you can replace it with `VSTS.AccountContents` to leverage Data Connector authentication. This will also inform Power BI that these requests are referencing the same data source and you'll be able to combine the data without violating the single data source constraints in Power BI Service.

'VSTS.AccountContents' provides a subset of the Arguments and Options available through 'OData.Contents'. The specific limitations are outlined in the table below:

Arguments for VSTS.Contents

Argument	Description
`url`	URL to one of the VSTS service endpoints.
`options`	An options record to control the behavior of this function.

Options fields for VSTS.Contents

Field	Description
`IsRetry`	Specifying this logical value as true will ignore any existing response in the cache when fetching data.
`ManualStatusHandling`	Specifying this value as a list will prevent any builtin handling for HTTP requests whose response has one of these status codes.
`MaxSize`	Controls the max size of the table the client is interested in. If request exceeds this limit then server can fail the request immediately. Default value is zero, which tells the

	servers server to use its default value.
`Query`	Programmatically add query parameters to the URL.
`RelativePath`	Specifying this value as text appends it to the base URL before making the request.
`Timeout`	Specifying this value as a duration will change the timeout for an HTTP request. The default value is 600 seconds.
`Version`	Version of the data model. This option is primary for diagnostics.

Related articles

- Power Query (M) Formula Reference
- Power Query (M) Formula Reference - Accessing data functions