Livestock Mandatory Price Reporting (LMPR), Dairy Products Mandatory Reporting Program (DPMRP) & Federal Milk Marketing Orders (FMMOS) Application Programming Interface (API) User Guide, v2.3

U.S. Department of Agriculture
Agricultural Marketing Service
1400 Independence Avenue SW
Washington DC 20250

December 2020
# Table of Contents

1 Overview .............................................................................................................................. 4
  1.1 Purpose of LMPR & DPMRP API ....................................................................................... 4
  1.2 Overuse .............................................................................................................................. 4
  1.3 Record Limit ....................................................................................................................... 4
  1.4 “Report Date” vs. “Report End Date” ................................................................................ 4
  1.5 Document Audience .......................................................................................................... 5
  1.6 Definitions ......................................................................................................................... 5

2 Design .................................................................................................................................. 5
  2.1 Goals .................................................................................................................................. 5
  2.2 Implementation .................................................................................................................. 6
  2.3 Important Note ................................................................................................................... 6
  2.4 Corrections ......................................................................................................................... 6
  2.5 Using Excel for an API request .......................................................................................... 7
  2.6 Using Dynamic Parameters in Excel ................................................................................ 12

3 Examples ........................................................................................................................... 13
  3.1 Livestock Report Examples ............................................................................................... 14
  3.2 Important Note on Report Section Names ......................................................................... 15
  3.3 Dairy Examples ................................................................................................................ 26
    3.3.1 Dairy Product Mandatory Reporting Program (DPMRP) Examples ........................... 26
    3.3.2 Federal Milk Marketing Order Statistics (FMMOS) Examples .................................... 39
  3.4 Examples of Corrections ..................................................................................................... 53

4 Report Holidays .................................................................................................................. 58
## Change History

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>06 APR 20</td>
<td>Initial Draft</td>
<td>1.0</td>
</tr>
<tr>
<td>07 APR 20</td>
<td>Updated to include examples for “AllSections”</td>
<td>1.1</td>
</tr>
<tr>
<td>29 APR 20</td>
<td>Section added to help identify the correct Section names</td>
<td>1.2</td>
</tr>
<tr>
<td>06 MAY 20</td>
<td>Includes updated Dairy examples, brief discussion on parameters, and an explanation for using Excel</td>
<td>1.3</td>
</tr>
<tr>
<td>08 MAY 20</td>
<td>Added video link that explains using Excel Data Query “parameters” to make is easier to query dynamically. Add clarification on O365 in Section 2.4</td>
<td>1.4</td>
</tr>
<tr>
<td>01 JUN 20</td>
<td>Added Section 1.3 Record Limit clarification statements.</td>
<td>1.5</td>
</tr>
<tr>
<td>02 JUN 20</td>
<td>Added Section 1.4 offering clarification on “Report_Date” and “Report_End_Date”</td>
<td>1.6</td>
</tr>
<tr>
<td>10 JUN 20</td>
<td>Updates to API to support a between clause for “Published_Date”</td>
<td>1.7</td>
</tr>
<tr>
<td>25 JUN 20</td>
<td>Updates to API to support using multiple variables together such as Report Year and Report Month</td>
<td>1.8</td>
</tr>
<tr>
<td>27 JUL 20</td>
<td>Updates made to Section 2.4 regarding Microsoft Excel 2016, (32 bit) versions.</td>
<td>1.9</td>
</tr>
<tr>
<td>13 AUG 20</td>
<td>Updated to include examples for “Final Prices” for the National Dairy Products Sales Report</td>
<td>2.0</td>
</tr>
<tr>
<td>4 SEP 20</td>
<td>Updated National Dairy Products Sales Report examples</td>
<td>2.1</td>
</tr>
<tr>
<td>10 NOV 20</td>
<td>Added two new reports to the list of FMMOS reports in Section 3.3.2</td>
<td>2.2</td>
</tr>
<tr>
<td>21 DEC 20</td>
<td>Added new examples of enhanced API features related to “Corrections” under Section 2.4. Examples are provided under Section 3.4.</td>
<td>2.3</td>
</tr>
</tbody>
</table>
1 Overview

1.1 Purpose of LMPR & DPMRP API

The LMPR & DPMRP & FMMOS API allows public access to Livestock Mandatory Price Reporting (LMPR), Dairy Products Mandatory Reporting Program (DPMRP), and Federal Milk Marketing Orders (FMMOS) market report information.

The output of the LMPR API is JavaScript Object Notation (JSON). JSON is an open standard format and data interchange format. This file format uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format and easily consumable in various applications.

1.2 Overuse

Overloading the LMPR API with high frequency automated requests unnecessarily taxes computing resources. High frequency requests consume all the network bandwidth, create website performance issues often causing the website to crash, and reduce data availability to other customers attempting to reach the site. To mitigate overloading the systems, AMS will temporarily block IP addresses found taxing the systems with high frequency requests.

If you find DataMart unavailable, or notice performance issues on your LMPR API request, it is possible your IP address has been temporarily blocked. Often these high frequency requests occur due to simple coding errors. Please email Wash.LPGMN@ams.usda.gov for assistance restoring your access.

1.3 Record Limit

Both the LMR and MyMarketNews API’s limit data calls record results to 100,000 per request. This is done so as to not overwhelm the system and ensure that the systems remain operational, responsive and available to all parties.

1.4 “Report Date” vs. “Report End Date”

AMS would like to make users aware that certain reports that, since inception, have slightly different “Report Date” offerings. Certain Summary reports may not offer “Report Date”, but will offer “Report End Date”. Examples of this would be in, but not limited to, LM_CT106, LM_CT109, & LM_CT168. Pending future budgetary conditions, AMS may be able to work towards standardizing this, but it is not in the immediate future. AMS recommends checking the “Summary” section in DataMart, LMR Web Service XML or LMR API to determine if the report uses “Report Date” or “Report End Date”.
1.5 Document Audience

This document is technical in nature. This document was written to assist technical support staff in configuring LMPR API to pull data into their own environment or network. This document contains technical information and is not intended for non-technical audience(s).

1.6 Definitions

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMS</td>
<td>Agricultural Marketing Service</td>
</tr>
<tr>
<td>DPMRP</td>
<td>Dairy Product Mandatory Reporting Program</td>
</tr>
<tr>
<td>FMMOS</td>
<td>Federal Milk Marketing Order Statistics</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>LPGMN</td>
<td>Livestock, Poultry, and Grain Market News</td>
</tr>
<tr>
<td>LMPR</td>
<td>Livestock Mandatory Price Reporting</td>
</tr>
<tr>
<td>MN</td>
<td>Market News</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>USDA</td>
<td>United States Department of Agriculture</td>
</tr>
</tbody>
</table>

2 Design

2.1 Goals

The LMPR API is designed to meet these goals:

- Simple
- Lightweight
- Flexible
- Intuitive
- Extendable
- Integration available
- Minimal development effort required
- Structurally predictable
- Consumable

Every effort has been made to ensure the LMPR API works in the same fashion as the MyMarketNews API; however the LMPR data structure is different than MyMarketNews. LMPR has a slightly different underlying database structure than MyMarketNews. These differences may show themselves in LMPR API request that yield slightly different data set.
results.

2.2 Implementation

The LMPR API does not require a user key like the MyMarketNews API requires. User request are unrestricted with no limit on record counts or restrictions.

Market News will monitor the usage of the LMPR API for abuse. If the system shows excessive taxation where other user request are being affected, LPGMN will move to limit, restrict, or block abusive user request.

The LMPR API is set to Central Standard Time (CST) time zone. Standard and daylight time rules apply.

2.3 Important Note

The LMPR API offers access to the same data set that is available at https://mpr.datamart.ams.usda.gov/ and the LMPR Web Service. If the report is not on the DataMart website or in the current Web Service, then it is not available via the LMPR API.

LPGMN uses both Postman and Microsoft Excel to show data sample in this User Guide. LPGMN does not endorse either product but references them to aid in articulating expected results.

Before creating API requests either in Postman or Excel, there are few filter parameter standards that a data user must know. First a common way to limit the data received from your API request is limiting it by a time threshold. The most common variables used to limit your data by time are report_date, report_year and report month. To add this to your request use the following syntax: ?q=time variable=value (e.g. ?q=report_year=2018 or ?q=report_date=09/15/2017)

If you want to limit your API request to a range of values use a colon in the value portion of the parameter (e.g. ?q=report_year=2012:2014).

There are two other useful parameters that are commonly used for API requests. The first is the &sort=variable (e.g. &sort=report_date). This sorts the results of your API request based on the variable you supply in the &sort parameter. The second is the &allSections=True parameter. Almost all the reports that use the LMPR API have different sections to the report. You can use your API request to pull a certain section of the report or use the &allSections=True parameter to pull all sections of the report at one time. There are helpful examples later in this documentation that illustrate how to use both the &sort= and the &allSections=True parameters.

2.4 Corrections

In late December 2020, the LMR API introduced support for identifying and consuming Report Corrections. Additionally, enhanced syntax was introduced that allows for consuming X amount of days of data and X amount of a particular report. Samples of that syntax is listed in Examples section titled “Examples of Corrections”
Users now have the ability to:

1. List reports that are “Correction”
2. List reports that are “Correction” since X amount of days.
3. List reports in the last X amount of days.
4. List X amount of a particular report.

2.5 Using Excel for an API request

Both Microsoft Excel 2016 (64 bit), and 2013 (with the optional Power Query Tab installed) support data calls to web based API. The Microsoft 2013 Power Query Tab can be downloaded here (https://www.microsoft.com/en-us/download/details.aspx?id=39379).

Note: Users who have Office 365 may have slightly different menus or slightly different steps than the ones shown below.

In Excel 2016, the Tab is called “Data”. By following the instructions included in this document, you will be creating linked data sources to the LMR API from an Excel file. Clicking the “Refresh” button automatically connects to the LMR API and pulls the latest publicly available data.

To start a LMR API connection, click “From Web” on the Data Tab. For Microsoft Excel 2016 (32 bit), users should select “Get Data” >> “From Other Sources” >> From Web.

Enter the URL for the report you would like to get. For this example we will pull the data for the Table of Contents (https://mpr.datamart.ams.usda.gov/services/v1.1/reports/). Click “Ok”. The screen will pause for a few seconds while the request is made to the LMR API. The screen will refresh.
The page will refresh, and the Query Editor will launch.
Place your mouse over the column heading titled “List”. Right click your mouse and select “Copy Entire List”

Click the “To Table” button

A menu will appear. Click “Ok”
The screen will refresh. There will be an icon to the right of “Column1”.

Click the “Double Arrow” icon. The screen will refresh. Click “Ok”.

The screen will refresh again and show

![Excel screenshot with queries](image)

Click the “Close and Load” button. This will load all data into a new Excel worksheet.

2.6 Using Dynamic Parameters in Excel

There are ways to dynamically pass parameters to the Data Query Editor on the fly to pull different data. A YouTube video outlining how to do that is located here:

[https://www.youtube.com/watch?v=sdR2B12e5Y8&feature=youtu.be](https://www.youtube.com/watch?v=sdR2B12e5Y8&feature=youtu.be)
3  Examples

The LMPR API offers a table of contents of all published reports accessible at https://mpr.datamart.ams.usda.gov/services/v1.1/reports/.

Legacy Slug-ID were added to the Report Title for ease of identification. All specific report drill down queries should be done using the new Slug-ID column denoted below in column A.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>slug</td>
<td>slug name</td>
<td>report title</td>
<td>published_date</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
3.1 Livestock Report Examples

To pull the Summary Section of the “5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)”, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466

Note the usage of the slug_id to access this particular report. Results by default show the most recent report first.

Expected results would be:
3.2 Important Note on Report Section Names

The sections of each report differs depending on the commodity. The recommended way to identify unique “Section” names is to query the Summary first. Examples are below.

Example 1:
https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2668/?q=report_date=03/09/2020&allSections=true

Example 2:
https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2511/?q=report_date=03/09/2020&allSections=true

Example 3:

Example 4:
To pull the Summary for this same report (“5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)”), but for only one report_date, the sample syntax would be:

```
https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466/Summary?q=report_date=08/05/2019
```

Expected results would be:
To access the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)” for the same report_date, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466/Detail?q=report_date=08/05/2019

Expected results would be:
To pull all sections of the “5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)” at one time, the sample syntax would be:


Expected results would be:

```

[{
    "reportSection": "Summary",
    "reportSections": [
        "Summary",
        "Detail"
    ],
    "status": {
        "totalRows": 1,
        "returnedRows": 1,
        "userAllowedRows": 99999
    },
    "results": [
        {
            "report_date": "03/30/2020",
            "previous_day_head_count": "5,557",
            "narrative": null,
            "slug_name": "LM_CT100",
            "slug_id": "2466",
            "report_title": "5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (LM_CT100)",
            "office_name": "St Joseph, MO",
            "office_code": "LS-57",
            "office_city": "St Joseph",
            "office_state": "MO",
            "market_location_name": "St Joseph, MO",
            "market_location_city": "St Joseph",
            "market_location_state": "MO",
            "market_type": "Direct Livestock - LMR Cattle",
            "market_type_category": "Direct Livestock - LMR Cattle",
            "published_date": "03/30/2020 10:49:13"
        }
    ]
}]
```

```
To access the Summary section of “5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)” with a published date of 03/25/2020, the sample syntax would be:


Special note. The published date query also accepts HH:MM:SS as shown below. The published date query accepts either, and on any section of a report


Expected results would be:
To access the Summary section of “National Daily Pork FOB Plant - Negotiated Sales - Afternoon (PDF) (LM_PK602)” with a published date between 05-01-2020 and 05-06-2020, the sample syntax would be:


Expected results would be:

```
```

```
[{
  "report_date": "05/06/2020",
  "narrative": null,
  "slug_name": "AMS_2498",
  "slug_uri": "2498",
  "report_title": "National Daily Pork FOB Plant - Negotiated Sales - Afternoon (PDF) (LM_PK602)",
  "office_name": "Des Moines, IA",
  "office_code": "LS-NU",
  "office_city": "Des Moines",
  "office_state": "IA",
  "market_location_name": "Des Moines, IA",
  "market_location_city": "Des Moines",
  "market_location_state": "IA",
  "market_type": "Direct Livestock - LMR Pork",
  "market_type_category": "Direct Livestock - LMR Pork",
  "published_date": "05/06/2020 14:56:19"
},
{
  "report_date": "05/05/2020",
  "narrative": null,
  "slug_name": "AMS_2498",
  "slug_uri": "2498",
  "report_title": "National Daily Pork FOB Plant - Negotiated Sales - Afternoon (PDF) (LM_PK602)",
  "office_name": "Des Moines, IA",
  "office_code": "LS-NU",
  "office_city": "Des Moines",
  "office_state": "IA",
  "market_location_name": "Des Moines, IA",
  "market_location_city": "Des Moines",
  "market_location_state": "IA",
  "market_type": "Direct Livestock - LMR Pork",
  "market_type_category": "Direct Livestock - LMR Pork",
  "published_date": "05/05/2020 14:54:11"
},
{
  "report_date": "05/04/2020",
  "narrative": null,
  "slug_name": "AMS_2498",
  "slug_uri": "2498",
  "report_title": "National Daily Pork FOB Plant - Negotiated Sales - Afternoon (PDF) (LM_PK602)",
  "office_name": "Des Moines, IA",
  "office_code": "LS-NU",
  "office_city": "Des Moines",
  "office_state": "IA"
}]
```
To access the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)” for the report_date range of 08/05/2019 to 08/06/2019, but add a Sort filter on previous_day_head_count field, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466/Detail?q=report_date=08/05/2019:08/06/2019&sort=previous_day_head_count

Expected results would be:

![Image of API request and response example]
To pull the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)” for the report_date range of 08/05/2019 to 08/06/2019, but only select class_description of STEER with a selling_basis of LIVE DELIVERED, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466/Detail?q=report_date=08/05/2019:08/06/2019;class_description=STEER;selling_basis_description=LIVE DELIVERED

Expected results would be:
To pull the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)” for the report_date range of 08/05/2019 to 08/06/2019, but only select class_description of either STEER or HEIFER, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466/Detail?q=report_date=08/05/2019:08/06/2019;class_description=STEER,HEIFER

Expected results would be:
To pull the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)” for the report_date range of 08/05/2019 to 08/10/2019, but only select class_description of STEER with a selling_basis of LIVE DELIVERED sorted with the oldest published_date first, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466/Detail?q=report_date=08/05/2019:08/10/2019;class_description=STEER;selling_basis_description=LIVE DELIVERED&sort=published_date

Expected results would be:
To pull the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle – Negotiated (LM_CT100)” for the report_date range of 08/05/2019 to 08/10/2019, but only select class_description of STEER with a selling_basis of LIVE DELIVERED sorted with more recent published_date first, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466/Detail?q=report_date=08/05/2019:08/10/2019;class_description=STEER;selling_basis_description=LIVE DELIVERED&sort=-published_date

Note the “-” before the published_date sort syntax. Expected results would be:
3.3 Dairy Examples

3.3.1 Dairy Product Mandatory Reporting Program (DPMRP) Examples

To pull the Summary Section of the National Dairy Products Sales Report, the sample syntax would be:
https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993

Denote the usage of the Slug_ID to access this particular report. Results by default show the most recent report first. In addition, the summary section for Dairy reports will not show any data.

Expected results in Postman:
Expected result in Excel:

<table>
<thead>
<tr>
<th>Column1:week_ending_date</th>
<th>Column1:created_date</th>
<th>Column1:narrative</th>
<th>Column1:lic噫</th>
<th>Column1:report_type</th>
<th>Column1:file_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/14/2020</td>
<td>03/16/2020</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>03/17/2020</td>
<td>03/19/2020</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>03/20/2020</td>
<td>03/22/2020</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>03/22/2020</td>
<td>03/24/2020</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>03/28/2020</td>
<td>03/30/2020</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>03/31/2020</td>
<td>04/02/2020</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>04/01/2020</td>
<td>04/04/2020</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>04/04/2020</td>
<td>04/07/2020</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>04/10/2019</td>
<td>04/12/2019</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>04/12/2019</td>
<td>04/14/2019</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>04/18/2019</td>
<td>04/20/2019</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>04/19/2019</td>
<td>04/21/2019</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>04/29/2019</td>
<td>05/01/2019</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>05/02/2019</td>
<td>05/04/2019</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>05/03/2019</td>
<td>05/05/2019</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>05/06/2019</td>
<td>05/08/2019</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
<tr>
<td>05/12/2019</td>
<td>05/14/2019</td>
<td>DYWDAIRYPRODUCTSALLES</td>
<td>2993</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
</tr>
</tbody>
</table>
The “National Dairy Products Sales Report” is comprised of six sections, including Summary (above), Butter, Cheddar 40s, Cheddar 500s, Dry Whey, and Nonfat Dry Milk Sections. To pull the each Section of the “National Dairy Products Sales Report”, the sample syntaxes would be:

- **Butter** - [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Butter Prices and Sales](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Butter Prices and Sales)
- **Cheddar 40s** - [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/40 Pound Block Cheddar Cheese Prices and Sales](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/40 Pound Block Cheddar Cheese Prices and Sales)
- **Cheddar 500s** - [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/500 Pound Barrel Cheddar Cheese Prices, Sales, and Moisture Content](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/500 Pound Barrel Cheddar Cheese Prices, Sales, and Moisture Content)
- **Dry Whey** - [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Dry Whey Prices and Sales](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Dry Whey Prices and Sales)
- **Nonfat Dry Milk** - [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Nonfat Dry Milk Prices and Sales](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Nonfat Dry Milk Prices and Sales)

Denote that there are six sections to this report. When setting up your workbook you will need to pull one section per tab. For these examples the URL is the same for Postman and Excel.

**Expected results in Excel for butter:**

```
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column 1</td>
<td>1.8273</td>
</tr>
<tr>
<td>Column 2</td>
<td>6,176,846</td>
</tr>
<tr>
<td>Column 3</td>
<td>National Dairy Products Sales Report (PDF)</td>
</tr>
</tbody>
</table>

Note: The Excel table above shows the expected results for butter prices and sales. The table includes columns for date, prices, and sales data, with the last column indicating whether it's a PDF report.
```
To pull the Butter section for this same report “National Dairy Products Sales Report” but for only one report date, the sample syntax for Postman and Excel would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Butter Prices and Sales?q=week_ending_date=03/14/2020

Expected results in Postman:

![Postman screenshot](image)

Expected results in Excel:

<table>
<thead>
<tr>
<th>Week Ending Date</th>
<th>Created Date</th>
<th>Reporting Date</th>
<th>Butter Price</th>
<th>Butter Sales</th>
<th>Report Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/14/2020</td>
<td>03/13/2020</td>
<td>03/14/2020</td>
<td>1.8270</td>
<td>3,985,572</td>
<td>National Dairy Products Sales Report [PDF]</td>
</tr>
</tbody>
</table>
To access the Butter section of “National Dairy Products Sales Report” for the date range of 02/22/2020 to 03/14/2020, but Sort with the oldest date first, the sample syntax for Postman and Excel would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Butter Prices and Sales?q=week_ending_date=02/22/2020:03/14/2020&sort=published_date

Expected results in Postman:

```
{
    "query_params": {
        "key": "week_ending_date",
        "value": "02/22/2020:03/14/2020",
        "sort": "published_date"
    }
}
```

Expected results in Excel:

![Excel screenshot](image-url)
To pull the all sections of the *National Dairy Products Sales Report*, but for only one report date the sample syntax for Postman and Excel would be:


Expected results:
The “National Dairy Products Sales Report” report allows revisions to the four weeks of data prior to the current reporting week. To pull final price and volume information that includes all revisions, the sample syntaxes would be:

- **Butter** – [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final Butter Prices and Sales](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final Butter Prices and Sales)
- **Cheddar 40s** - [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final 40 Pound Block Cheddar Cheese Prices and Sales](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final 40 Pound Block Cheddar Cheese Prices and Sales)
- **Cheddar 500s** - [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final 500 Pound Barrel Cheddar Cheese Prices, Sales, and Moisture Content](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final 500 Pound Barrel Cheddar Cheese Prices, Sales, and Moisture Content)
- **Dry Whey** - [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final Dry Whey Prices and Sales](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final Dry Whey Prices and Sales)
- **Nonfat Dry Milk** - [https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final Nonfat Dry Milk Prices and Sales](https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2993/Final Nonfat Dry Milk Prices and Sales)

Denote that for these examples the URL is the same for Postman and Excel.

Expected results in Excel for dry whey:

<table>
<thead>
<tr>
<th>Column1 Week Ending Date</th>
<th>Column2 Location Date</th>
<th>Column3 Narrative</th>
<th>Column4 Week Ending Date</th>
<th>Column5 whey Price</th>
<th>Column6 whey Sales</th>
<th>Column7 Report Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/01/2020</td>
<td>01/01/2020</td>
<td>01/01/2020</td>
<td>01/01/2020</td>
<td>01/01/2020</td>
<td>01/01/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/02/2020</td>
<td>01/02/2020</td>
<td>01/02/2020</td>
<td>01/02/2020</td>
<td>01/02/2020</td>
<td>01/02/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/03/2020</td>
<td>01/03/2020</td>
<td>01/03/2020</td>
<td>01/03/2020</td>
<td>01/03/2020</td>
<td>01/03/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/04/2020</td>
<td>01/04/2020</td>
<td>01/04/2020</td>
<td>01/04/2020</td>
<td>01/04/2020</td>
<td>01/04/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/05/2020</td>
<td>01/05/2020</td>
<td>01/05/2020</td>
<td>01/05/2020</td>
<td>01/05/2020</td>
<td>01/05/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/06/2020</td>
<td>01/06/2020</td>
<td>01/06/2020</td>
<td>01/06/2020</td>
<td>01/06/2020</td>
<td>01/06/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/07/2020</td>
<td>01/07/2020</td>
<td>01/07/2020</td>
<td>01/07/2020</td>
<td>01/07/2020</td>
<td>01/07/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/08/2020</td>
<td>01/08/2020</td>
<td>01/08/2020</td>
<td>01/08/2020</td>
<td>01/08/2020</td>
<td>01/08/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/10/2020</td>
<td>01/10/2020</td>
<td>01/10/2020</td>
<td>01/10/2020</td>
<td>01/10/2020</td>
<td>01/10/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/12/2020</td>
<td>01/12/2020</td>
<td>01/12/2020</td>
<td>01/12/2020</td>
<td>01/12/2020</td>
<td>01/12/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/13/2020</td>
<td>01/13/2020</td>
<td>01/13/2020</td>
<td>01/13/2020</td>
<td>01/13/2020</td>
<td>01/13/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/14/2020</td>
<td>01/14/2020</td>
<td>01/14/2020</td>
<td>01/14/2020</td>
<td>01/14/2020</td>
<td>01/14/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/15/2020</td>
<td>01/15/2020</td>
<td>01/15/2020</td>
<td>01/15/2020</td>
<td>01/15/2020</td>
<td>01/15/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
<tr>
<td>01/16/2020</td>
<td>01/16/2020</td>
<td>01/16/2020</td>
<td>01/16/2020</td>
<td>01/16/2020</td>
<td>01/16/2020</td>
<td>National Dairy Products Sales Report (PD)</td>
</tr>
</tbody>
</table>
To pull the Detail section for the “Announcement of Class and Component Prices”, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2991/detail

Denote that for this example the URL is the same for both Postman and Excel.

Expected results in Postman:

```
{
  "reportSection": "detail",
  "reportSections": [
    "summary",
    "detail"
  ],
  "stats": {
    "totalRows": 96,
    "returnRows": 96,
    "userQueryId": "999999"
  },
  "results": [
    {
      "week_ending_date": "02/29/2020",
      "created_date": "03/04/2020",
      "narrative": null,
      "class_2_price": "16.64",
      "class_2_butterfat_price": "1.8883",
      "advanced_white_class_2_price": "18.24",
      "class_3_price": "17.08",
      "class_3_butterfat_price": "10.43",
      "class_4_price": "18.26",
      "class_4_butterfat_price": "9.68",
      "butterfat_price": "1.9812",
      "nonfat_solids_price": "1.3667",
      "protein_price": "3.8389",
      "other_solids_price": "1.1758",
      "somatic_cell_adjustment_rate": "0.0000",
      "butter_monthly_avg_price": "1.8878",
      "nfds_monthly_avg_price": "1.2453",
      "cheese_monthly_avg_price": "1.7884",
      "whey_monthly_avg_price": "0.3998",
      "report_title": "Announcement of Class and Component Prices (PDF)",
      "slug_name": "GMMCLASSPRICES",
      "slug_id": "2991",
      "office_name": "Dairy MRR",
      "office_code": "0Y-WK",
      "office_city": "Washington",
      "office_state": "DC",
      "market_location_name": "Washington, DC",
      "market_location_city": "Washington",
    }
  ]
}```
Expected results in Excel:

<table>
<thead>
<tr>
<th>ColumnA</th>
<th>ColumnB</th>
<th>ColumnC</th>
<th>ColumnD</th>
<th>ColumnE</th>
<th>ColumnF</th>
<th>ColumnG</th>
<th>ColumnH</th>
<th>ColumnI</th>
<th>ColumnJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/29/2020</td>
<td>03/04/2020</td>
<td>16.84</td>
<td>1.9883</td>
<td>10.24</td>
<td>17.85</td>
<td>10.43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02/01/2020</td>
<td>03/05/2020</td>
<td>17.05</td>
<td>2.1887</td>
<td>9.98</td>
<td>17.05</td>
<td>10.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02/28/2019</td>
<td>03/02/2019</td>
<td>18.81</td>
<td>2.3322</td>
<td>9.43</td>
<td>15.87</td>
<td>12.11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/31/2019</td>
<td>04/04/2019</td>
<td>18.05</td>
<td>2.3285</td>
<td>9.62</td>
<td>15.42</td>
<td>12.70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/24/2019</td>
<td>05/03/2019</td>
<td>18.86</td>
<td>2.4091</td>
<td>8.24</td>
<td>18.72</td>
<td>10.85</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/24/2019</td>
<td>06/02/2019</td>
<td>16.93</td>
<td>2.5052</td>
<td>8.46</td>
<td>16.31</td>
<td>9.91</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06/21/2018</td>
<td>07/08/2018</td>
<td>17.80</td>
<td>2.8844</td>
<td>8.57</td>
<td>17.08</td>
<td>6.82</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/17/2019</td>
<td>08/03/2019</td>
<td>17.61</td>
<td>2.8928</td>
<td>8.46</td>
<td>17.03</td>
<td>8.43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08/24/2019</td>
<td>09/01/2019</td>
<td>17.30</td>
<td>2.9649</td>
<td>8.26</td>
<td>16.17</td>
<td>7.22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09/30/2019</td>
<td>10/01/2019</td>
<td>16.48</td>
<td>2.7088</td>
<td>7.72</td>
<td>16.38</td>
<td>7.63</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/17/2019</td>
<td>11/01/2019</td>
<td>16.38</td>
<td>2.5445</td>
<td>7.75</td>
<td>15.96</td>
<td>7.34</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/20/2019</td>
<td>12/04/2019</td>
<td>16.91</td>
<td>2.3311</td>
<td>7.95</td>
<td>15.84</td>
<td>6.33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/23/2019</td>
<td>01/03/2020</td>
<td>16.13</td>
<td>2.3425</td>
<td>7.09</td>
<td>13.89</td>
<td>5.20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01/14/2020</td>
<td>02/27/2020</td>
<td>15.76</td>
<td>2.3051</td>
<td>7.22</td>
<td>13.96</td>
<td>5.41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02/13/2020</td>
<td>02/26/2020</td>
<td>15.67</td>
<td>2.1309</td>
<td>7.12</td>
<td>13.78</td>
<td>5.38</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/18/2020</td>
<td>04/06/2020</td>
<td>15.03</td>
<td>2.5425</td>
<td>6.96</td>
<td>14.44</td>
<td>5.70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/17/2020</td>
<td>05/02/2020</td>
<td>15.54</td>
<td>3.5821</td>
<td>8.81</td>
<td>15.53</td>
<td>6.82</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/12/2020</td>
<td>06/01/2020</td>
<td>15.53</td>
<td>3.5312</td>
<td>8.43</td>
<td>16.09</td>
<td>7.43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06/20/2020</td>
<td>07/06/2020</td>
<td>15.07</td>
<td>2.8079</td>
<td>6.16</td>
<td>18.45</td>
<td>6.06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/14/2020</td>
<td>08/01/2020</td>
<td>15.30</td>
<td>2.9517</td>
<td>8.55</td>
<td>14.10</td>
<td>5.44</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08/14/2020</td>
<td>09/01/2020</td>
<td>15.40</td>
<td>2.6292</td>
<td>6.63</td>
<td>15.21</td>
<td>6.08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09/23/2020</td>
<td>10/01/2020</td>
<td>14.47</td>
<td>2.8309</td>
<td>5.45</td>
<td>15.10</td>
<td>6.21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/12/2020</td>
<td>11/01/2020</td>
<td>14.03</td>
<td>2.2933</td>
<td>5.41</td>
<td>14.47</td>
<td>5.89</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/13/2020</td>
<td>12/04/2020</td>
<td>13.38</td>
<td>2.4443</td>
<td>5.35</td>
<td>14.22</td>
<td>5.93</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/17/2020</td>
<td>01/02/2021</td>
<td>14.41</td>
<td>2.1990</td>
<td>5.30</td>
<td>15.40</td>
<td>5.37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01/18/2021</td>
<td>02/11/2021</td>
<td>14.51</td>
<td>2.4691</td>
<td>5.70</td>
<td>14.41</td>
<td>5.41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02/15/2021</td>
<td>03/04/2021</td>
<td>14.49</td>
<td>2.3221</td>
<td>5.94</td>
<td>15.44</td>
<td>6.53</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/17/2021</td>
<td>04/02/2021</td>
<td>15.72</td>
<td>2.2816</td>
<td>6.08</td>
<td>15.80</td>
<td>8.23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/24/2021</td>
<td>05/01/2021</td>
<td>15.05</td>
<td>2.8756</td>
<td>8.04</td>
<td>15.88</td>
<td>7.82</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/14/2021</td>
<td>06/04/2021</td>
<td>16.00</td>
<td>2.8629</td>
<td>7.03</td>
<td>16.36</td>
<td>6.59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06/24/2021</td>
<td>07/03/2021</td>
<td>17.56</td>
<td>3.0179</td>
<td>7.25</td>
<td>16.57</td>
<td>6.23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/17/2021</td>
<td>08/03/2021</td>
<td>17.48</td>
<td>2.9528</td>
<td>7.41</td>
<td>15.49</td>
<td>5.33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08/24/2021</td>
<td>09/02/2021</td>
<td>18.15</td>
<td>2.7396</td>
<td>6.89</td>
<td>15.84</td>
<td>7.22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09/21/2021</td>
<td>10/01/2021</td>
<td>14.84</td>
<td>2.4204</td>
<td>6.60</td>
<td>15.57</td>
<td>7.18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/19/2021</td>
<td>11/05/2021</td>
<td>14.41</td>
<td>2.9618</td>
<td>7.48</td>
<td>15.22</td>
<td>7.23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/16/2021</td>
<td>12/04/2021</td>
<td>15.21</td>
<td>2.3845</td>
<td>8.02</td>
<td>15.31</td>
<td>7.61</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/22/2021</td>
<td>01/01/2022</td>
<td>16.52</td>
<td>2.4444</td>
<td>8.29</td>
<td>16.54</td>
<td>8.39</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01/14/2022</td>
<td>02/01/2022</td>
<td>16.36</td>
<td>2.5323</td>
<td>7.77</td>
<td>15.77</td>
<td>6.22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02/12/2022</td>
<td>03/02/2022</td>
<td>15.26</td>
<td>2.4304</td>
<td>7.32</td>
<td>15.45</td>
<td>5.58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/24/2022</td>
<td>04/11/2022</td>
<td>14.60</td>
<td>2.1134</td>
<td>7.47</td>
<td>15.79</td>
<td>9.74</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/22/2022</td>
<td>05/02/2022</td>
<td>14.09</td>
<td>2.0563</td>
<td>7.14</td>
<td>14.83</td>
<td>7.92</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To pull the Detail section for the “Announcement of Advanced Prices and Pricing Factors”, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2989/detail

Denote that for this example the URL is the same for both Postman and Excel.

Expected results in Postman:

```
GET https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2989/detail

```

```
{}
```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```
Expected results in Excel:

<table>
<thead>
<tr>
<th>Column 1: week ending date</th>
<th>Column 2: created date</th>
<th>Column 3: Component 1</th>
<th>Column 4: Component 2</th>
<th>Column 5: Component 3</th>
<th>Column 6: Component 4</th>
<th>Column 7: Component 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/14/2020</td>
<td>01/14/2020</td>
<td>10.39</td>
<td>10.06</td>
<td>8.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01/21/2020</td>
<td>01/20/2020</td>
<td>10.02</td>
<td>10.47</td>
<td>5.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01/28/2020</td>
<td>01/28/2020</td>
<td>9.40</td>
<td>9.09</td>
<td>5.54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02/04/2020</td>
<td>02/04/2020</td>
<td>9.33</td>
<td>9.03</td>
<td>5.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02/11/2020</td>
<td>02/10/2020</td>
<td>8.83</td>
<td>8.27</td>
<td>7.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02/18/2020</td>
<td>02/17/2020</td>
<td>8.18</td>
<td>7.69</td>
<td>7.78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02/25/2020</td>
<td>02/24/2020</td>
<td>7.89</td>
<td>7.14</td>
<td>7.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/01/2020</td>
<td>7.50</td>
<td>6.85</td>
<td>7.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/08/2020</td>
<td>03/07/2020</td>
<td>7.25</td>
<td>6.97</td>
<td>7.23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/15/2020</td>
<td>03/14/2020</td>
<td>6.80</td>
<td>6.47</td>
<td>6.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/22/2020</td>
<td>03/21/2020</td>
<td>6.40</td>
<td>6.12</td>
<td>6.52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03/29/2020</td>
<td>03/28/2020</td>
<td>6.02</td>
<td>5.60</td>
<td>6.42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/05/2020</td>
<td>04/04/2020</td>
<td>5.63</td>
<td>5.19</td>
<td>6.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/12/2020</td>
<td>04/11/2020</td>
<td>5.21</td>
<td>5.12</td>
<td>6.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/19/2020</td>
<td>04/18/2020</td>
<td>4.86</td>
<td>4.59</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/26/2020</td>
<td>04/25/2020</td>
<td>4.50</td>
<td>4.25</td>
<td>5.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/03/2020</td>
<td>05/02/2020</td>
<td>4.20</td>
<td>4.00</td>
<td>5.63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/10/2020</td>
<td>05/09/2020</td>
<td>3.90</td>
<td>3.70</td>
<td>4.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/17/2020</td>
<td>05/16/2020</td>
<td>3.62</td>
<td>3.52</td>
<td>4.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05/24/2020</td>
<td>05/23/2020</td>
<td>3.32</td>
<td>3.28</td>
<td>4.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06/07/2020</td>
<td>06/06/2020</td>
<td>3.13</td>
<td>3.10</td>
<td>4.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06/14/2020</td>
<td>06/12/2020</td>
<td>2.96</td>
<td>2.93</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06/21/2020</td>
<td>06/20/2020</td>
<td>2.80</td>
<td>2.80</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06/28/2020</td>
<td>06/27/2020</td>
<td>2.67</td>
<td>2.67</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/05/2020</td>
<td>07/04/2020</td>
<td>2.55</td>
<td>2.55</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/12/2020</td>
<td>07/11/2020</td>
<td>2.45</td>
<td>2.45</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/19/2020</td>
<td>07/18/2020</td>
<td>2.35</td>
<td>2.35</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/26/2020</td>
<td>07/25/2020</td>
<td>2.25</td>
<td>2.25</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>08/02/2020</td>
<td>08/01/2020</td>
<td>2.15</td>
<td>2.15</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>08/09/2020</td>
<td>08/08/2020</td>
<td>2.05</td>
<td>2.05</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>08/16/2020</td>
<td>08/15/2020</td>
<td>1.95</td>
<td>1.95</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>08/23/2020</td>
<td>08/22/2020</td>
<td>1.85</td>
<td>1.85</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>08/30/2020</td>
<td>08/29/2020</td>
<td>1.75</td>
<td>1.75</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09/06/2020</td>
<td>09/05/2020</td>
<td>1.66</td>
<td>1.66</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09/13/2020</td>
<td>09/12/2020</td>
<td>1.56</td>
<td>1.56</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09/20/2020</td>
<td>09/19/2020</td>
<td>1.47</td>
<td>1.47</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>09/27/2020</td>
<td>09/26/2020</td>
<td>1.38</td>
<td>1.38</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/04/2020</td>
<td>10/03/2020</td>
<td>1.29</td>
<td>1.29</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/11/2020</td>
<td>10/10/2020</td>
<td>1.20</td>
<td>1.20</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/18/2020</td>
<td>10/17/2020</td>
<td>1.11</td>
<td>1.11</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/25/2020</td>
<td>10/24/2020</td>
<td>1.02</td>
<td>1.02</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/01/2020</td>
<td>11/01/2020</td>
<td>0.93</td>
<td>0.93</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/08/2020</td>
<td>11/07/2020</td>
<td>0.85</td>
<td>0.85</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/15/2020</td>
<td>11/14/2020</td>
<td>0.77</td>
<td>0.77</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/22/2020</td>
<td>11/21/2020</td>
<td>0.69</td>
<td>0.69</td>
<td>5.93</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To pull the Detail section for the “Announcement of Advanced Prices and Pricing Factors” for the entire year of 2020 but Sort with the oldest date first, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2989/Detail?q=week_ending_date=01/04/2020:01/02/2021&sort=published_date

Denote that the week_ending_date parameter is used in a range form of 01/04/2020 to 01/02/2021. This range encompasses all the data for year 2020. The URL is the same for both Postman and Excel.

Expected results in Postman:

![Postman Response](image)

Expected results in Excel:

<table>
<thead>
<tr>
<th>Week ending Date</th>
<th>Created Date</th>
<th>Base Class 1 Price</th>
<th>Base Skim Milk Class 1 Price</th>
<th>Advanced Base Skim Milk Class 1 Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/15/2020</td>
<td>09/17/2020</td>
<td>17.55</td>
<td>17.46</td>
<td>10.46</td>
</tr>
<tr>
<td>09/16/2020</td>
<td>09/18/2020</td>
<td>17.46</td>
<td>17.46</td>
<td>10.47</td>
</tr>
<tr>
<td>09/17/2020</td>
<td>09/19/2020</td>
<td>16.04</td>
<td>16.04</td>
<td>10.04</td>
</tr>
<tr>
<td>09/18/2020</td>
<td>09/20/2020</td>
<td>12.90</td>
<td>12.90</td>
<td>8.93</td>
</tr>
<tr>
<td>09/19/2020</td>
<td>09/21/2020</td>
<td>11.42</td>
<td>11.42</td>
<td>6.68</td>
</tr>
<tr>
<td>09/20/2020</td>
<td>09/22/2020</td>
<td>10.42</td>
<td>10.42</td>
<td>6.93</td>
</tr>
</tbody>
</table>
To pull the Detail section for the “Announcement of Advanced Prices and Pricing Factors” for only one report date, e.g. the May Advanced Prices and Pricing Factors report, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2989/Detail?q=week_ending_date=04/18/2020

Denote the URL is the same for both Postman and Excel.

Expected results in Postman:

Expected results in Excel:

| week_ending_date | created_date | base_class_1 | base_class_2 | advanced_class_1 | advanced_class_2 | advanced_class_3 | advance_class_4 | base_milk_class_1 | base_milk_class_2 | butter_butter_class_1 | class_2 | diesel_class_1 | diesel_class_2 | average_rate_factor | report_title advanced_mgmt_factor | slug_name | id
|------------------|---------------|---------------|--------------|-----------------|-----------------|-----------------|-----------------|------------------|------------------|---------------------|---------|--------------|--------------|----------------|------------------|---------------|)--
| 04/18/2020       | 04/22/2020    | 12.95         | 8.72         | 8.53            | 7.63            | 1.2548          | 7.73            |
3.3.2 Federal Milk Marketing Order Statistics (FMMOS) Examples

Listed below are some easy ways to pull FMMOS data by section of a particular report using the example syntax:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/nnnn/ssssssss

nnnn = Slug_ID of the desired report.
ssssssss = section name

On the following page, please refer to a table of the FMMOS reports, Slug_ID’s and section names.
<table>
<thead>
<tr>
<th>Slug_ID</th>
<th>Report</th>
<th>Report Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>3345</td>
<td>Class I Prices</td>
<td>Summary, Milk, Butterfat, Skim</td>
</tr>
<tr>
<td>3346</td>
<td>Class I Utilization</td>
<td>Summary, Milk, Utilization, Butterfat, NFS</td>
</tr>
<tr>
<td>3347</td>
<td>Class II Utilization</td>
<td>Summary, Milk, Utilization, Butterfat, NFS</td>
</tr>
<tr>
<td>3348</td>
<td>Class III Utilization</td>
<td>Summary, Milk, Utilization, Butterfat, Protein, Other Solids</td>
</tr>
<tr>
<td>3349</td>
<td>Class IV Utilization</td>
<td>Summary, Milk, Utilization, Butterfat, NFS</td>
</tr>
<tr>
<td>3350</td>
<td>Total Receipts of Producer Milk</td>
<td>Summary, Producers, Receipts, Avg Daily, Butterfat, NFS, Protein, Other Solids, SomCell</td>
</tr>
<tr>
<td>3461</td>
<td>Producer Receipts</td>
<td>Summary, Producers, Receipts, Avg Daily</td>
</tr>
<tr>
<td>3462</td>
<td>Producer Milk Components</td>
<td>Summary, Receipts, Butterfat, NFS, Protein</td>
</tr>
<tr>
<td>3351</td>
<td>Uniform Milk Prices</td>
<td>Summary, Milk, Butterfat, Skim, PPD</td>
</tr>
<tr>
<td>3352</td>
<td>Price and Pool – Monthly</td>
<td>Summary, Price and Pool Monthly</td>
</tr>
<tr>
<td>3353</td>
<td>Price and Pool – Annual</td>
<td>Summary, Price and Pool Annual</td>
</tr>
<tr>
<td>3354</td>
<td>Advanced Prices by Order</td>
<td>Summary, Advanced Class Prices by Order</td>
</tr>
<tr>
<td>3355</td>
<td>Class Prices by Order</td>
<td>Summary, Final Class Prices by Order</td>
</tr>
<tr>
<td>3356</td>
<td>Retail Prices</td>
<td>Summary</td>
</tr>
<tr>
<td>------</td>
<td>---------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3357</td>
<td>Mailbox Milk Prices</td>
<td>Summary</td>
</tr>
<tr>
<td>3358</td>
<td>Estimated Fluid Milk Sales</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3359</td>
<td>Regulated Pool Plant Lists</td>
<td>Summary</td>
</tr>
</tbody>
</table>

To pull the Milk Section of the “Class I Utilization”, the sample syntax for Postman would be: https://mpr.datamart.ams.usda.gov/services/v1.1/reports/3346/Milk

Denote the usage of the Slug_ID to access this particular report.

Expected results in Postman:
GET https://mpr.datamart.ams.usda.gov/services/v1.1/reports/3345/Milk

```
{
  "reportSection": "Milk",
  "reportSections": [
    "Summary",
    "Milk",
    "Utilization",
    "Butterfat",
    "NFS"
  ],
  "stats": {
    "totalRows": 239,
    "returnedRows": 239,
    "userAllowedRows": 99999
  },
  "results": {
    "report_month": "Dec",
    "report_year": 2000,
    "narrative": null,
    "Pool_Order_Name": "All Markets Combined",
    "Pool_Order_No": null,
    "jan": 3.965,
    "feb": 3.745,
    "mar": 4.636,
    "apr": 3.611,
    "may": 3.839,
    "jun": 3.824,
    "jul": 3.549,
    "aug": 3.895,
    "sep": 3.875,
    "oct": 3.946,
    "nov": 3.956
  }
}
```
To use the API request in Excel, start by following the first few steps on page 7 above. You will then see the following:

Next, click on the Into Table button on the top left part of the screen. Next you will see:
At this point, right click on the List cell in the second column. Click on Drill Down on the menu that pops up. Next you will see a list of records:
Again, you will click on the To Table button in the upper left part of the window. Click Ok when prompted. Your list will now look like this:

![Power Query Editor with sample data](image)

Click on the double arrows icon that is circled in red above. Click OK when prompted. This will all the variables associated with the records from your API request as shown below.

![Expanded data view](image)
At this point, you may sort your data as you choose, and you can remove any columns of data that are not needed for your analysis purposes. I have sorted and reduced the number of variables for the Expected results in Excel snaps you see on the next number of pages.

**Expected results in Excel:**
To pull the 2018 Butterfat Section of the “Class I Prices”, the sample syntax for Postman would be:


Denote the usage of the Slug_ID and the report_year variables to access this particular report.

Expected results in Postman:

![Postman results](image)

Expected results in Excel:

![Excel results](image)
To pull the Summary Section for years 2015-2018 of the “Uniform Milk Prices”, the sample syntax for Postman would be:


Denote the usage of the Slug_ID and report_year range to access this particular report. NOTE: This pull would only give you confirmation of the years selected but not any actual data from this report.

Expected results in Postman:

![Postman screenshot](image-url)

Expected results in Excel:

![Excel screenshot](image-url)
To pull all the report sections for 2017 of the “Total Receipts of Producer Milk”, the sample syntax for Postman would be:


Denote the usage of the Slug_ID, report_year, and allSections variables to access this particular report.

Expected results in Postman:

![Screenshot of Postman results](https://example.com/postman-screenshot.png)
Expected results in Excel:

<table>
<thead>
<tr>
<th>Summary</th>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
<th>Column 6</th>
<th>Column 7</th>
<th>Column 8</th>
<th>Column 9</th>
<th>Column 10</th>
<th>Column 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Producers</td>
<td>All Markets Combined</td>
<td>30,296</td>
<td>32,801</td>
<td>33,390</td>
<td>35,679</td>
<td>32,042</td>
<td>32,668</td>
<td>35,038</td>
<td>34,114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Appalachia</td>
<td>5,783</td>
<td>5,477</td>
<td>5,277</td>
<td>2,657</td>
<td>3,131</td>
<td>2,708</td>
<td>4,358</td>
<td>4,056</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Arizona</td>
<td>313,196</td>
<td>382</td>
<td>382</td>
<td>382</td>
<td>382</td>
<td>1,382</td>
<td>382</td>
<td>382</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Central</td>
<td>12,689</td>
<td>2,378</td>
<td>3,721</td>
<td>2,761</td>
<td>2,452</td>
<td>2,689</td>
<td>2,488</td>
<td>3,511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Florida</td>
<td>6,543</td>
<td>148</td>
<td>343</td>
<td>143</td>
<td>143</td>
<td>143</td>
<td>143</td>
<td>143</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Midwest</td>
<td>10,150</td>
<td>5,649</td>
<td>5,879</td>
<td>5,105</td>
<td>5,634</td>
<td>6,961</td>
<td>6,925</td>
<td>6,934</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Northwest</td>
<td>15,141</td>
<td>11,322</td>
<td>13,170</td>
<td>11,617</td>
<td>13,023</td>
<td>16,947</td>
<td>11,386</td>
<td>11,389</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Pacific Northwest</td>
<td>124,446</td>
<td>436</td>
<td>436</td>
<td>436</td>
<td>436</td>
<td>436</td>
<td>436</td>
<td>436</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Southeast</td>
<td>7,572</td>
<td>1,672</td>
<td>2,872</td>
<td>1,672</td>
<td>2,872</td>
<td>1,672</td>
<td>2,872</td>
<td>1,672</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Southwest</td>
<td>126,468</td>
<td>595</td>
<td>595</td>
<td>595</td>
<td>595</td>
<td>595</td>
<td>595</td>
<td>595</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers</td>
<td>Upper Midwest</td>
<td>30,130</td>
<td>865</td>
<td>9,469</td>
<td>9,590</td>
<td>9,752</td>
<td>9,589</td>
<td>10,040</td>
<td>10,541</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>All Markets Combined</td>
<td>52,537</td>
<td>15,195</td>
<td>31,222</td>
<td>13,779</td>
<td>11,358</td>
<td>13,779</td>
<td>11,358</td>
<td>11,358</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Appalachia</td>
<td>5,493</td>
<td>412</td>
<td>352</td>
<td>482</td>
<td>482</td>
<td>482</td>
<td>482</td>
<td>482</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Arizona</td>
<td>132,489</td>
<td>192</td>
<td>418</td>
<td>441</td>
<td>441</td>
<td>441</td>
<td>441</td>
<td>441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Central</td>
<td>12,726</td>
<td>1,236</td>
<td>1,456</td>
<td>1,661</td>
<td>1,867</td>
<td>1,867</td>
<td>1,867</td>
<td>1,867</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Florida</td>
<td>6,312</td>
<td>119</td>
<td>216</td>
<td>229</td>
<td>229</td>
<td>229</td>
<td>229</td>
<td>229</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Midwest</td>
<td>19,172</td>
<td>1,610</td>
<td>2,876</td>
<td>3,143</td>
<td>3,714</td>
<td>1,777</td>
<td>3,714</td>
<td>3,714</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Northeast</td>
<td>1,205</td>
<td>2,449</td>
<td>2,836</td>
<td>3,035</td>
<td>2,676</td>
<td>2,730</td>
<td>2,836</td>
<td>2,836</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Pacific Northwest</td>
<td>128,446</td>
<td>152</td>
<td>823</td>
<td>833</td>
<td>833</td>
<td>833</td>
<td>833</td>
<td>833</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Southeast</td>
<td>7,493</td>
<td>493</td>
<td>513</td>
<td>542</td>
<td>497</td>
<td>497</td>
<td>497</td>
<td>497</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Southwest</td>
<td>126,379</td>
<td>854</td>
<td>2,219</td>
<td>2,219</td>
<td>2,219</td>
<td>2,219</td>
<td>2,219</td>
<td>2,219</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts</td>
<td>Upper Midwest</td>
<td>30,140</td>
<td>2,251</td>
<td>2,818</td>
<td>3,099</td>
<td>2,476</td>
<td>2,740</td>
<td>3,099</td>
<td>3,099</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>All Markets Combined</td>
<td>5,168</td>
<td>7,004</td>
<td>8,628</td>
<td>7,884</td>
<td>7,477</td>
<td>7,004</td>
<td>7,928</td>
<td>7,928</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>Appalachia</td>
<td>5,726</td>
<td>7,904</td>
<td>8,888</td>
<td>7,884</td>
<td>7,477</td>
<td>7,004</td>
<td>7,928</td>
<td>7,928</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>Arizona</td>
<td>10,495</td>
<td>13,398</td>
<td>18,730</td>
<td>16,147</td>
<td>13,399</td>
<td>16,303</td>
<td>14,359</td>
<td>15,863</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>Central</td>
<td>10,644</td>
<td>14,283</td>
<td>17,790</td>
<td>16,698</td>
<td>17,687</td>
<td>18,544</td>
<td>16,698</td>
<td>17,687</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>Florida</td>
<td>6,762,393</td>
<td>73,200</td>
<td>53,129</td>
<td>52,249</td>
<td>49,225</td>
<td>47,629</td>
<td>47,225</td>
<td>47,225</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>Midwest</td>
<td>30,672</td>
<td>11,299</td>
<td>13,297</td>
<td>12,691</td>
<td>13,448</td>
<td>13,876</td>
<td>13,133</td>
<td>13,544</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>Northeast</td>
<td>3,604</td>
<td>6,731</td>
<td>8,877</td>
<td>7,532</td>
<td>7,335</td>
<td>6,913</td>
<td>6,913</td>
<td>6,913</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>Southeast</td>
<td>7,735</td>
<td>9,630</td>
<td>9,992</td>
<td>10,009</td>
<td>9,501</td>
<td>9,991</td>
<td>8,148</td>
<td>8,148</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>Southwest</td>
<td>126,727</td>
<td>68,980</td>
<td>76,227</td>
<td>77,227</td>
<td>69,095</td>
<td>87,314</td>
<td>77,176</td>
<td>77,176</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily</td>
<td>Upper Midwest</td>
<td>30,677</td>
<td>9,679</td>
<td>10,025</td>
<td>10,114</td>
<td>9,858</td>
<td>9,729</td>
<td>9,729</td>
<td>9,729</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
At certain times a data user may want to pull a report for just one month of a particular year. In that situation you would use both the `report_year` and `report_month` parameters.

To pull the Price and Pool Monthly section for August 2018 of the “Price and Pool - Monthly”, the sample syntax for Postman would be:

```
```

Denote the usage of the Slug_ID, `report_year`, and `report_month` (in three characters, e.g. `report_month=feb`) variables to access this particular report.

Expected results in Postman:

![Postman screenshot](image)

Expected results in Excel:
Please note that the requesting of just one specific month for a particular year will only work on the following reports that have a unique report for each month:

- 3352 – Price and Pool - Monthly
- 3354 – Advanced Prices by Order
- 3355 – Class Prices by Order
- 3358 – Estimated Fluid Milk Sales

The remainder of the FMMOS reports simply build a year-to-date table that eventually has all monthly data included. For the remainder of the tables, any request can only use report_month=dec.
3.4 Examples of Corrections

To pull Corrections only of the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)” the sample syntax would be:


Expected results would be:

```json
{
  "reportSection": "Detail",
  "reportSections": [
    "Summary",
    "Detail"
  ],
  "stats": {
    "totalRows": 13871,
    "returnedRows": 13871,
    "userAllowedRows": 99999
  },
  "results": {
    "report_date": "12/14/2020",
    "previous_day_head_count": "1,310",
    "head_count": "948",
    "weight_range_low": "$99",
    "weight_range_high": "$63",
    "weight_range_avg": "$508",
    "price_range_low": "$164.00",
    "price_range_high": "$185.00",
    "weighted_avg_price": "$165.40",
    "report_title": "5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)",
    "slug_name": "LM_CT100",
    "slug_id": "2466",
    "office_name": "St Joseph, MO",
    "office_code": "LS-S1",
    "office_city": "St Joseph",
    "office_state": "MO",
    "market_location_name": "St Joseph, MO",
    "market_location_city": "St Joseph",
    "market_location_state": "MO",
    "market_type": "Direct Livestock - LHR Cattle",
    "head_count": "948",
    "weight_range_low": "$99",
    "weight_range_high": "$63",
    "weight_range_avg": "$508",
    "price_range_low": "$164.00",
    "price_range_high": "$185.00",
    "weighted_avg_price": "$165.40"
  }
}
```
To pull Corrections only of the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)” since 04 July 2020, the sample syntax would be:


Expected results would be:
To pull the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)” for the last one hundred days, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466/Detail?lastDays=100

Expected results would be:
To pull last five reports of the Detail section of “5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)”, the sample syntax would be:


Expected results would be:
4 Report Holidays

There are six national holidays that are usually observed when reports are not issued. Reports resume following these holidays. The observed dates do not follow actual dates for the holiday, but are a subset of observed holidays derived from the U.S. OPM Federal Holiday schedule. The six holidays normally selected from this schedule are as follows:

1. New Year’s Day
2. Memorial Day
3. Independence Day
4. Labor Day
5. Thanksgiving Day
6. Christmas Day

Besides the holidays above, extenuating circumstances may also impact the dates when reports are issued.