# Table of Contents

1 Overview .............................................................................................................................. 5
   1.1 Purpose of LMPR & DPMRP API ................................................................. 5
   1.2 Overuse ........................................................................................................ 5
   1.3 Record Limit ................................................................................................. 5
   1.4 “Report Date” vs. “Report End Date” ......................................................... 6
   1.5 Document Audience ...................................................................................... 6
   1.6 Definitions ....................................................................................................... 6

2 Design .................................................................................................................................. 6
   2.1 Goals .............................................................................................................. 6
   2.2 Implementation ............................................................................................. 7
   2.3 Important Note ............................................................................................. 7
   2.4 Corrections .................................................................................................... 8
   2.5 Using Excel for an API request ..................................................................... 8
   2.6 Using Dynamic Parameters in Excel .......................................................... 13

3 Examples ........................................................................................................................... 14
   3.1 Livestock Report Examples ........................................................................ 15
   3.2 Important Note on Report Section Names ................................................ 16
   3.3 Dairy Examples ........................................................................................... 27
      3.3.1 Dairy Product Mandatory Reporting Program (DPMRP) Examples .... 27
      3.3.2 Federal Milk Marketing Order Statistics (FMMOS) Examples .......... 40
   3.4 Examples of Corrections ............................................................................. 54
   3.5 Examples of enhanced “All Sections” data pull ........................................ 58

4 Email API Feature .............................................................................................................. 59

5 Report Holidays .................................................................................................................. 60
## 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>
<tr>
<td>22 JUN 21</td>
<td>Added new example of enhanced API features related to “All Sections” under Section 3.5</td>
<td>2.4</td>
</tr>
<tr>
<td>11 AUG 21</td>
<td>Introduce new Email API feature in Section 4.</td>
<td>2.5</td>
</tr>
<tr>
<td>30 NOV 21</td>
<td>Add “is_correction” field to “Summary/Header section” of each report. More information is available under Section 4 of User Guide. New section 1.2 added titled “Notification of Changes to LMR API”.</td>
<td>2.6</td>
</tr>
<tr>
<td>Date</td>
<td>Changes</td>
<td>Page</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>09 SEP 22</td>
<td>Added clarification remarks to the limits on date range when using the Email Zip feature in Section 4.0</td>
<td>2.7</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 Notification of Changes to LMR API

Occasionally there will be updates to the LMR API and LMR Web Service. The updates are a reflection of report enhancements or changes, which are necessary to ensure AMS is providing data and reports that reflect the current marketplace. Any changes to the LMR API and LMR Web Service XML files are essential to ensure data users are receiving all components and communication regarding the desired report. If you would like to receive an advanced notice of any API and XML changes, please send an email to mailto:mymarketnews@usda.gov and request to be added to the distribution list.

1.3 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. Email mailto:mymarketnews@usda.gov for assistance in restoring your access.

1.4 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.
1.5 “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, but not limited to, are 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”.

1.6 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).

1.7 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

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 publically
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.

![Screenshot of From Web dialog box with URL set to https://mpr.datamart.ams.usda.gov/services/v1.1/reports/]

The page will refresh, and the Query Editor will launch.
The image contains a screenshot of a Power Query Editor window with the title "Query1 - Power Query Editor." The window is in the "List Tools" tab and shows a list of queries, each labeled as "Record" from 1 to 26. The window includes various tools and options such as "Remove Duplicates," "Reverse Items," and "Statistics."
Place your mouse of 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”.

![Diagram of the interface with a red arrow pointing to the icon next to "Column1". The interface shows a list of records with options for selection. There is also a warning message stating that the list may be incomplete.](image-url)
The screen will refresh again and show

![Excel screenshot](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=sdR2Bl2e5Y8&feature=youtu.be
3 Examples

The LMPR API offers a table of contents of all published reports accessible @ 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>Slug-ID</th>
<th>Slug Name</th>
<th>Report Title</th>
<th>Published Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>329319</td>
<td>298039</td>
<td>Announcement of Class and Component Prices (PDF)</td>
<td>03/30/2020 13:20:20</td>
</tr>
<tr>
<td>329504</td>
<td>298039</td>
<td>Announcement of Advanced Prices and Pricing Factors (PDF)</td>
<td>03/30/2020 13:20:20</td>
</tr>
<tr>
<td>2463</td>
<td>298039</td>
<td>S Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_C110)</td>
<td>03/11/2020 10:25:45</td>
</tr>
<tr>
<td>2847</td>
<td>298039</td>
<td>Daily Direct Cow and Bull Negotiated Report - Summary (PDF) (LM_C130)</td>
<td>03/11/2020 14:19:22</td>
</tr>
<tr>
<td>2636</td>
<td>298039</td>
<td>National Direct Slaughter Cattle Report - Committed and Delivered Cattle - Summary (PDF) (LM_C140)</td>
<td>03/11/2020 10:20:21</td>
</tr>
<tr>
<td>2653</td>
<td>298039</td>
<td>TX/OH/NM Daily Direct Slaughter Cattle - Negotiated Purchases - Afternoon (PDF) (LM_C137)</td>
<td>03/11/2020 15:06:06</td>
</tr>
<tr>
<td>2664</td>
<td>298039</td>
<td>TX/OH/NM Daily Direct Slaughter Cattle - Negotiated Purchases - Summary (PDF) (LM_C138)</td>
<td>03/11/2020 10:27:46</td>
</tr>
<tr>
<td>2665</td>
<td>298039</td>
<td>Kansas Daily Direct Slaughter Cattle - Negotiated Purchases - Afternoon (PDF) (LM_C126)</td>
<td>03/11/2020 15:06:15</td>
</tr>
<tr>
<td>2666</td>
<td>298039</td>
<td>Kansas Daily Direct Slaughter Cattle - Negotiated Purchases - Summary (PDF) (LM_C127)</td>
<td>03/11/2020 15:06:16</td>
</tr>
<tr>
<td>2667</td>
<td>298039</td>
<td>Nebraska Daily Direct Slaughter Cattle - Negotiated Purchases - Afternoon (PDF) (LM_C122)</td>
<td>03/11/2020 15:06:33</td>
</tr>
<tr>
<td>2668</td>
<td>298039</td>
<td>Nebraska Daily Direct Slaughter Cattle - Negotiated Purchases - Summary (PDF) (LM_C124)</td>
<td>03/11/2020 15:08:28</td>
</tr>
<tr>
<td>2669</td>
<td>298039</td>
<td>CO Daily Direct Slaughter Cattle - Negotiated Purchases - Afternoon (PDF) (LM_C131)</td>
<td>10/23/2019 15:00:13</td>
</tr>
<tr>
<td>2670</td>
<td>298039</td>
<td>CO Daily Direct Slaughter Cattle - Negotiated Purchases - Summary (PDF) (LM_C144)</td>
<td>10/23/2019 15:00:59</td>
</tr>
<tr>
<td>2671</td>
<td>298039</td>
<td>IA-MN Daily Direct Slaughter Cattle - Negotiated Purchases - Afternoon (PDF) (LM_C136)</td>
<td>03/11/2020 15:07:07</td>
</tr>
<tr>
<td>2672</td>
<td>298039</td>
<td>IA-MN Daily Direct Slaughter Cattle - Negotiated Purchases - Summary (PDF) (LM_C137)</td>
<td>10/23/2019 15:00:57</td>
</tr>
<tr>
<td>2674</td>
<td>298039</td>
<td>IA-MN Weekly Direct Slaughter Cattle - Formula, Grid and Contract Purchases (PDF) (LM_C139)</td>
<td>05/09/2020 10:52:26</td>
</tr>
<tr>
<td>2675</td>
<td>298039</td>
<td>Nebraska Weekly Direct Slaughter Cattle - Formula, Grid and Contract Purchases (PDF) (LM_C141)</td>
<td>05/09/2020 10:53:06</td>
</tr>
<tr>
<td>2676</td>
<td>298039</td>
<td>Nebraska Weekly Direct Slaughter Cattle - Formula, Grid and Contract Purchases (PDF) (LM_C142)</td>
<td>05/09/2020 10:53:21</td>
</tr>
<tr>
<td>2678</td>
<td>298039</td>
<td>S Area Weekly Direct Slaughter Cattle - Formula, Grid, and Contract Purchases (PDF) (LM_C146)</td>
<td>05/09/2020 10:53:38</td>
</tr>
<tr>
<td>2679</td>
<td>298039</td>
<td>S Area Weekly Weighted Average Direct Slaughter Cattle (PDF) (LM_C120)</td>
<td>05/09/2020 10:54:19</td>
</tr>
<tr>
<td>2684</td>
<td>298039</td>
<td>National Weekly Direct Slaughter Cattle - Premiums and Discounts (PDF) (LM_C151)</td>
<td>05/09/2020 09:29:49</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.amr.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:

```
GET https://mpr.datamart.amr.usda.gov/services/v1.1/reports/2466

Params Authorization Headers Body Pre-request Script Tests

<table>
<thead>
<tr>
<th>KEY</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Value</td>
</tr>
</tbody>
</table>

Body Cookies Headers (0) Test Results

Pretty Raw Preview JSON 

```
```
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:

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

<table>
<thead>
<tr>
<th>KEY</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>report_date=08/05/2019</td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
</tr>
</tbody>
</table>

```

```
```
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:

```json
{
  "reportSection": "Detail",
  "reportSections": [{
    "summary": {
      "Detail": {
        "stats": {
          "totalRows": 64,
          "returnedRows": 64,
          "userAllowedRows": 9999
        }
      }
    }
  },
  "results": {
    "report_date": "08/05/2019",
    "previous_day_head_count": "24,519",
    "carrcative": "null",
    "class_description": "ALL BEEF TYPE",
    "selling_basis_description": "DRESSED DELIVERED",
    "grade_description": "Total All grades",
    "head_count": "3,095",
    "weight_range_low": "729",
    "weight_range_high": "945",
    "weight_range_avg": "867",
    "price_range_low": "120.00",
    "price_range_high": "180.00",
    "price_range_avg": "140.46",
    "report_title": "5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)",
    "slug_name": "LM_CT100",
    "slug": "2466",
    "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"
  }
}
```
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:

```json

1. 
2. { 
3.   "reportSection": "Summary", 
4.   "reportSections": [ 
5.     "Summary", 
6.     "Detail" 
7.   }, 
8.   "stats": { 
9.     "totalRows": 1, 
10.    "returnedRows": 1, 
11.    "userAllowedRows": 99999 
12. }, 
13.   "results": [ 
14.     { 
15.       "report_date": "03/30/2020", 
16.       "previous_day_head_count": "3,557", 
17.       "narrative": null, 
18.       "slug_name": "AVS_2466", 
19.       "slug_id": "2466", 
20.       "report_title": "5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (LM_CT100)", 
21.       "office_name": "St Joseph, MO", 
22.       "office_code": "LS-5", 
23.       "office_city": "Saint Joseph", 
24.       "office_state": "MO", 
25.       "market_location_name": "St Joseph, MO", 
26.       "market_location_city": "St Joseph", 
27.       "market_location_state": "MO", 
28.       "market_type": "Direct Livestock - LMR Cattle", 
29.       "market_type_category": "Direct Livestock - LMR Cattle", 
30.       "published_date": "03/30/2020 10:48:13" 
31.  } }, 
32. }, 
33. "reportSection": "Detail", 
34. "reportSections": [ 
35.   "Summary", 
36.   "Detail" 
37. ], 
38. "stats": { 
39.   "totalRows": 64, 
40.   "returnedRows": 64, 
41.   "userAllowedRows": 99999 
42. }, 
43. "results": [ 
44.   { 
45.     "report_date": "03/30/2020", 
46.     "previous_day_head_count": "3,557", 
47.     "narrative": null, 
48.     "class_description": "ALL BEEF TYPE", 
49.     "selling_basis_description": "DRESSED DELIVERED", 
50.     "grade_description": "Total All Grades"
51.  }
52. ]
```
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:

```json
{  
  "reportSections": {  
    "Summary": {  
      "stats": {  
        "totalRows": 1,  
        "returnedRows": 1,  
        "userAllowedRows": 99999  
      },  
      "results": [  
        {  
          "report_date": "03/25/2020",  
          "previous_day_head_count": "914",  
          "slug_name": "AMS-2466",  
          "slug_id": "2466",  
          "report_title": "5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)",  
          "office_name": "St. Joseph, MO",  
          "office_code": "LS-SL",  
          "office_location": "St. Joseph",  
          "market_location_name": "St. Joseph, MO",  
          "market_location_city": "St. Joseph",  
          "market_location_state": "MO",  
          "market_type": "Direct Livestock - LPR Cattle",  
          "market_type_category": "Direct Livestock - LPR Cattle",  
          "published_date": "03/25/2020 10:45:06"  
        }  
      ]  
    }  
  }  
} 
```
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:
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:

```
GET 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

```

```
{  "reportsection": "Detail",  "reportsections": [    {      "summary": {        "detail": {  "stats": {    "totalRows": 120,    "returnedRows": 120,    "userAllowedRows": 99999 },  "results": [    {      "report_date": "08/05/2019",      "previous_day_head_count": "88",      "class_description": "ALL BEEF TYPE",      "grade_description": "Total all grades",      "head_count": "62",      "weight_range_low": "806",      "weight_range_high": "889",      "price_range_low": "181.88",      "price_range_high": "182.38",      "report_title": "5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (LM_CT100)",      "slug_name": "LM_CT100",      "slug_id": "2066",      "office_name": "St. Joseph, MD",      "office_code": "15-55",      "office_city": "Saint Joseph",      "office_state": "MO",      "market_location_name": "St. Joseph, MO",      "market_location_city": "St. Joseph",      "market_location_state": "MO",    }  ]  }  ] }
```
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:

```
GET 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
```

```

```
```
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:

```
GET 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

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>report_date=08/05/2019:08/06/2019;class_description=STEER,HEIFER</td>
</tr>
</tbody>
</table>

```

```json
{   "reportSection": "Detail",   "reportSections": [   {   "Summary": null,   "Detail": null   }],   "state": {   "totalRows": 89,   "returnedRows": 89,   "userAllowedRows": 0 },   "results": [   {   "report_date": "08/05/2019",   "previous_day_head_count": 88,   "sells_to": "1",   "class_description": "STEER",   "sells_to_shears_description": "DRESSED DELIVERED",   "grade_description": "A - 35% Choice",   "head_count": null,   "weight_range_low": null,   "weight_range_high": null,   "weight_range_avg": null,   "price_range_low": null,   "price_range_high": null,   "weight_avg_price": null,   "report_title": "5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)",   "slug_name": "5A_2466",   "slug_id": "2466",   "office_name": "55 Joseph, ND",   "office_code": "55-21",   "state_abbrev": "ND"   }]
```
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:

```
GET 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

Params
Value
Authorization, Headers, Body, Pre-request Script, Tests

<table>
<thead>
<tr>
<th>KEY</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>report_date=08/05/2019:08/10/2019;class_description=STEER;selling_basis_description=LIVE DELIVERED&amp;sort=published_date</td>
<td></td>
</tr>
<tr>
<td>sort</td>
<td>published_date</td>
<td></td>
</tr>
</tbody>
</table>

Body
Cookies, Headers (8), Test Results

Pretty, Raw, Preview, JSON

```

```json

```
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:service</th>
<th>Column1:Log name</th>
<th>Column1:Log type</th>
<th>Column1:Report Role</th>
<th>Column1:Service name</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/14/2020</td>
<td>03/16/2020</td>
<td>DYNDAIRYPRODUCTSALLIES</td>
<td>2913</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
<td></td>
</tr>
<tr>
<td>03/21/2020</td>
<td>03/16/2020</td>
<td>DYNDAIRYPRODUCTSALLIES</td>
<td>2913</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
<td></td>
</tr>
<tr>
<td>03/22/2020</td>
<td>03/16/2020</td>
<td>DYNDAIRYPRODUCTSALLIES</td>
<td>2913</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
<td></td>
</tr>
<tr>
<td>03/26/2020</td>
<td>03/16/2020</td>
<td>DYNDAIRYPRODUCTSALLIES</td>
<td>2913</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
<td></td>
</tr>
<tr>
<td>03/28/2020</td>
<td>03/16/2020</td>
<td>DYNDAIRYPRODUCTSALLIES</td>
<td>2913</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
<td></td>
</tr>
<tr>
<td>01/15/2019</td>
<td>01/15/2019</td>
<td>DYNDAIRYPRODUCTSALLIES</td>
<td>2913</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
<td></td>
</tr>
<tr>
<td>01/16/2019</td>
<td>01/15/2019</td>
<td>DYNDAIRYPRODUCTSALLIES</td>
<td>2913</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
<td></td>
</tr>
<tr>
<td>12/19/2019</td>
<td>12/16/2019</td>
<td>DYNDAIRYPRODUCTSALLIES</td>
<td>2913</td>
<td>National Dairy Products Sales Report (PDF)</td>
<td>Dairy MMR</td>
<td></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 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>Date</th>
<th>Date</th>
<th>Date</th>
<th>Date</th>
<th>Date</th>
<th>Date</th>
<th>Date</th>
<th>Date</th>
<th>Date</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/14/2020</td>
<td>09/14/2020</td>
<td>09/14/2020</td>
<td>09/14/2020</td>
<td>09/14/2020</td>
<td>09/14/2020</td>
<td>09/14/2020</td>
<td>09/14/2020</td>
<td>09/14/2020</td>
<td>09/14/2020</td>
</tr>
<tr>
<td>09/16/2020</td>
<td>09/16/2020</td>
<td>09/16/2020</td>
<td>09/16/2020</td>
<td>09/16/2020</td>
<td>09/16/2020</td>
<td>09/16/2020</td>
<td>09/16/2020</td>
<td>09/16/2020</td>
<td>09/16/2020</td>
</tr>
<tr>
<td>09/18/2020</td>
<td>09/18/2020</td>
<td>09/18/2020</td>
<td>09/18/2020</td>
<td>09/18/2020</td>
<td>09/18/2020</td>
<td>09/18/2020</td>
<td>09/18/2020</td>
<td>09/18/2020</td>
<td>09/18/2020</td>
</tr>
<tr>
<td>09/19/2020</td>
<td>09/19/2020</td>
<td>09/19/2020</td>
<td>09/19/2020</td>
<td>09/19/2020</td>
<td>09/19/2020</td>
<td>09/19/2020</td>
<td>09/19/2020</td>
<td>09/19/2020</td>
<td>09/19/2020</td>
</tr>
<tr>
<td>09/20/2020</td>
<td>09/20/2020</td>
<td>09/20/2020</td>
<td>09/20/2020</td>
<td>09/20/2020</td>
<td>09/20/2020</td>
<td>09/20/2020</td>
<td>09/20/2020</td>
<td>09/20/2020</td>
<td>09/20/2020</td>
</tr>
<tr>
<td>09/30/2020</td>
<td>09/30/2020</td>
<td>09/30/2020</td>
<td>09/30/2020</td>
<td>09/30/2020</td>
<td>09/30/2020</td>
<td>09/30/2020</td>
<td>09/30/2020</td>
<td>09/30/2020</td>
<td>09/30/2020</td>
</tr>
</tbody>
</table>

National Dairy Products Sales Report (PDF)
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:

```
1 - {
  2  "reportSection": "Butter Prices and Sales",
  3  "reportSections": [
  4    "Summary",
  5    "Butter Prices and Sales",
  6    "40 Pound Block Cheddar Cheese Prices and Sales",
  7    "500 Pound Barrel Cheddar Cheese Prices, Sales, and Moisture Content",
  8    "Dry Whey Prices and Sales",
  9    "Nonfat Dry Milk Prices and Sales"
  10  ],
  11  "stats": {
  12    "totalRows": 1,
  13    "returnedRows": 1,
  14    "userAllowedRows": 99999
  15  },
  16  "results": [
  17    {
  18      "week_ending_date": "03/14/2020",
  19      "create_date": "03/13/2020",
  20      " narration": "null",
  21      "week Ending Date": "03/14/2020",
  22      "Butter_Price": "1.8279",
  23      "Butter_Sales": "3,985,572",
  24      "report_title": "National Dairy Products Sales Report (PDF)",
  25      "slug_name": "USDA/DairyProductsSales",
  26      "slug_id": "3559",
  27      "office_name": "Dairy PBE",
  28      "office_code": "D/4-H",
  29      "office_city": "Washington",
  30      "office_state": "DC",
  31      "market_location_name": "Washington, DC",
  32      "market_location_city": "Washington",
  33      "market_location_state": "DC",
  34      "market_type": "LUR Dairy",
  35      "market_type_category": "LUR Dairy",
  36      "published_date": "03/18/2020 10:18:13"
  37    }
  38  ],
```

Expected results in Excel:
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:

![Postman API Request Example](image_url)

Expected results in Excel:

![Excel Data Table](image_url)
To pull 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:

```json
{
  "reportSection": "Summary",
  "reportSections": [
    "Summary",
    "Butter Prices and Sales",
    "40 Pound Block Cheddar Cheese Prices and Sales",
    "500 Pound Barrel Cheddar Cheese Prices, Sales, and Moisture Content",
    "Dry Whey Prices and Sales",
    "Nonfat Dry Milk Prices and Sales"
  ],
  "stats": {
    "totalRows": 1,
    "returnedRows": 1,
    "userAllowedRows": 99999
  },
  "results": {
    "week_ending_date": "03/28/2020",
    "created_date": "04/01/2020",
    "narrative": null,
    "slug_name": "DNYDAIRYPRODUCTSSALES",
    "slug_id": "2993",
    "report_title": "National Dairy Products Sales Report (PDF)",
    "office_name": "Dairy MRR",
    "office_code": "DY-WA",
    "office_city": "Washington",
    "office_state": "DC",
    "market_location_name": "Washington, DC",
    "market_location_city": "Washington",
    "market_location_state": "DC",
    "market_type": "URR Dairy",
    "market_type_category": "URR Dairy",
    "published_date": "04/03/2020 13:04:43"
  }
}
```
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
- Cheddar 40s - 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
- Dry Whey - 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

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>Column 1: ending_date</th>
<th>Column 2: Location</th>
<th>Column 3: Narrative</th>
<th>Column 4: Week Ending Date</th>
<th>Column 5: whey Price</th>
<th>Column 6: whey Sales</th>
<th>Column 7: Report Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/01/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.2406</td>
<td>5,045,680</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/02/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.2427</td>
<td>4,930,476</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/03/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.2470</td>
<td>5,721,767</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/04/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.340</td>
<td>5,568,121</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/05/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.346</td>
<td>4,953,473</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/06/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.353</td>
<td>6,972,451</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/07/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.353</td>
<td>6,972,451</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/08/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.353</td>
<td>6,972,451</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/09/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.353</td>
<td>6,972,451</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/10/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.353</td>
<td>6,972,451</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
<tr>
<td>09/11/2020</td>
<td></td>
<td></td>
<td></td>
<td>0.353</td>
<td>6,972,451</td>
<td>National Dairy Products Sales Report (Dry Whey)</td>
</tr>
</tbody>
</table>

Expected results in Excel for dry whey:
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:**

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

```

![Postman Request](image)

```
{
  "reportSection": "detail",
  "reportSections": [
    "Summary",
    "Detail"
  ],
  "stats": {
    "totalRows": 96,
    "returnedRows": 96,
    "userLimitHonor": 99999
  },
  "results": [
    {
      "week_ending_date": "02/29/2020",
      "created_date": "03/04/2020",
      "narrative": null,
      "class_2_price": "16.63",
      "class_2_butterfat_price": "1.9833",
      "advanced_skim_milk_class_2_price": "18.24",
      "class_3_price": "17.08",
      "class_3 skim milk_price": "10.43",
      "class_a_price": "16.28",
      "class_a skim milk_price": "9.68",
      "butterfat_price": "1.9813",
      "nonfat_solid_price": "1.0867",
      "protein_price": "1.0399",
      "other_solids_price": "0.1750",
      "somatic cell adjustment_rate": "0.00899",
      "butter_monthly_avg_price": "1.8078",
      "nonfat_monthly_avg_price": "1.2453",
      "cheese_monthly_avg_price": "1.7884",
      " whey_monthly_avg_price": "0.3698",
      "report_title": "Announcement of Class and Component Prices (PDF)",
      "slug name": "OPENCLASSPRICES",
      "slug_id": "2991",
      "office_name": "Dairy MRR",
      "office_code": "OW-MR",
      "office_city": "Washington",
      "office_state": "DC",
      "market_location_name": "Washington, DC",
      "market_location_city": "Washington",
    }
  ]
}
```
Expected results in Excel:

<table>
<thead>
<tr>
<th>Column1</th>
<th>Column2</th>
<th>Column3</th>
<th>Column4</th>
<th>Column5</th>
<th>Column6</th>
<th>Column7</th>
<th>Column8</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/29/2020</td>
<td>03/04/2020</td>
<td>16.04</td>
<td>1.9983</td>
<td>10.24</td>
<td>17.36</td>
<td>10.41</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/05/2020</td>
<td>17.05</td>
<td>2.1887</td>
<td>9.98</td>
<td>17.09</td>
<td>10.31</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/06/2020</td>
<td>16.81</td>
<td>2.3222</td>
<td>9.43</td>
<td>15.97</td>
<td>12.11</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/07/2020</td>
<td>16.85</td>
<td>2.3565</td>
<td>9.62</td>
<td>15.67</td>
<td>12.70</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/08/2020</td>
<td>16.88</td>
<td>2.4821</td>
<td>8.24</td>
<td>16.72</td>
<td>10.85</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/09/2020</td>
<td>16.93</td>
<td>2.5352</td>
<td>8.46</td>
<td>16.31</td>
<td>9.91</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/10/2020</td>
<td>17.00</td>
<td>2.6644</td>
<td>8.77</td>
<td>17.80</td>
<td>8.83</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/11/2020</td>
<td>17.81</td>
<td>2.8626</td>
<td>8.46</td>
<td>17.39</td>
<td>8.43</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/12/2020</td>
<td>17.90</td>
<td>2.8549</td>
<td>8.26</td>
<td>16.72</td>
<td>7.22</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/13/2020</td>
<td>16.46</td>
<td>2.3788</td>
<td>7.72</td>
<td>16.30</td>
<td>7.63</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/14/2020</td>
<td>16.38</td>
<td>2.4455</td>
<td>7.75</td>
<td>15.96</td>
<td>7.34</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/15/2020</td>
<td>16.51</td>
<td>2.5511</td>
<td>7.95</td>
<td>15.84</td>
<td>6.35</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/16/2020</td>
<td>16.13</td>
<td>2.3453</td>
<td>7.09</td>
<td>15.89</td>
<td>5.20</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/17/2020</td>
<td>15.76</td>
<td>2.0531</td>
<td>7.22</td>
<td>15.96</td>
<td>5.41</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/18/2020</td>
<td>15.67</td>
<td>2.0190</td>
<td>7.32</td>
<td>15.76</td>
<td>5.35</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/19/2020</td>
<td>15.63</td>
<td>2.0425</td>
<td>6.96</td>
<td>14.44</td>
<td>5.76</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/20/2020</td>
<td>15.54</td>
<td>2.0821</td>
<td>6.81</td>
<td>15.53</td>
<td>6.83</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/21/2020</td>
<td>15.53</td>
<td>2.3522</td>
<td>8.43</td>
<td>15.09</td>
<td>7.43</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/22/2020</td>
<td>15.07</td>
<td>2.8079</td>
<td>6.16</td>
<td>14.93</td>
<td>6.06</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/23/2020</td>
<td>15.20</td>
<td>2.9357</td>
<td>6.55</td>
<td>14.10</td>
<td>5.44</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/24/2020</td>
<td>15.48</td>
<td>2.9292</td>
<td>6.63</td>
<td>15.71</td>
<td>6.68</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/25/2020</td>
<td>14.47</td>
<td>2.8309</td>
<td>5.40</td>
<td>15.13</td>
<td>6.21</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/26/2020</td>
<td>14.03</td>
<td>2.1383</td>
<td>5.41</td>
<td>14.47</td>
<td>5.83</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/27/2020</td>
<td>13.88</td>
<td>2.4343</td>
<td>5.35</td>
<td>14.22</td>
<td>5.93</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/28/2020</td>
<td>13.44</td>
<td>2.1900</td>
<td>5.30</td>
<td>15.40</td>
<td>5.37</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/29/2020</td>
<td>14.11</td>
<td>2.4801</td>
<td>5.70</td>
<td>14.95</td>
<td>5.61</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/30/2020</td>
<td>14.49</td>
<td>2.5221</td>
<td>5.94</td>
<td>15.44</td>
<td>6.53</td>
<td></td>
</tr>
<tr>
<td>02/29/2020</td>
<td>03/31/2020</td>
<td>15.72</td>
<td>2.1016</td>
<td>6.08</td>
<td>15.85</td>
<td>6.23</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/02/2020</td>
<td>15.05</td>
<td>2.8726</td>
<td>6.04</td>
<td>14.89</td>
<td>7.62</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/03/2020</td>
<td>16.00</td>
<td>2.8529</td>
<td>7.03</td>
<td>16.36</td>
<td>6.59</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/04/2020</td>
<td>17.56</td>
<td>3.0779</td>
<td>7.25</td>
<td>16.72</td>
<td>6.25</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/05/2020</td>
<td>17.48</td>
<td>2.9528</td>
<td>7.41</td>
<td>15.49</td>
<td>5.35</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/06/2020</td>
<td>18.15</td>
<td>2.7316</td>
<td>6.89</td>
<td>15.44</td>
<td>7.22</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/07/2020</td>
<td>14.84</td>
<td>2.4204</td>
<td>6.60</td>
<td>15.57</td>
<td>7.18</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/08/2020</td>
<td>14.81</td>
<td>2.9318</td>
<td>6.70</td>
<td>15.22</td>
<td>7.23</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/09/2020</td>
<td>16.21</td>
<td>2.4345</td>
<td>6.60</td>
<td>15.25</td>
<td>7.64</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/10/2020</td>
<td>16.52</td>
<td>2.4904</td>
<td>6.29</td>
<td>15.88</td>
<td>8.89</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/11/2020</td>
<td>16.36</td>
<td>2.5323</td>
<td>7.77</td>
<td>15.72</td>
<td>8.22</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/12/2020</td>
<td>15.28</td>
<td>2.3404</td>
<td>7.32</td>
<td>15.40</td>
<td>5.53</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/13/2020</td>
<td>16.60</td>
<td>2.1114</td>
<td>7.47</td>
<td>15.76</td>
<td>9.74</td>
<td></td>
</tr>
<tr>
<td>03/01/2020</td>
<td>03/14/2020</td>
<td>15.09</td>
<td>2.0983</td>
<td>7.14</td>
<td>14.82</td>
<td>7.93</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:
Expected results in Excel:

<table>
<thead>
<tr>
<th>Column编号 Ending date</th>
<th>Column编号 created date</th>
<th>Column编号 class_1 Polish</th>
<th>Column编号 skin with class_2 Polish</th>
<th>Column编号 advanced skin with class_3 Factor</th>
<th>Column编号 advanced skin with class_4 Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/14/2020</td>
<td>09/18/2020</td>
<td>16.44</td>
<td>10.23</td>
<td>10.06</td>
<td>8.85</td>
</tr>
<tr>
<td>09/15/2020</td>
<td>09/19/2020</td>
<td>17.46</td>
<td>15.82</td>
<td>16.47</td>
<td>8.86</td>
</tr>
<tr>
<td>09/16/2020</td>
<td>09/20/2020</td>
<td>17.45</td>
<td>10.46</td>
<td>9.40</td>
<td>5.74</td>
</tr>
<tr>
<td>09/17/2019</td>
<td>09/21/2019</td>
<td>19.01</td>
<td>16.71</td>
<td>12.96</td>
<td>5.88</td>
</tr>
<tr>
<td>09/18/2019</td>
<td>09/22/2019</td>
<td>19.53</td>
<td>12.65</td>
<td>13.05</td>
<td>5.78</td>
</tr>
<tr>
<td>09/25/2019</td>
<td>09/29/2019</td>
<td>18.14</td>
<td>10.11</td>
<td>10.42</td>
<td>0.32</td>
</tr>
<tr>
<td>09/26/2019</td>
<td>09/30/2019</td>
<td>17.46</td>
<td>8.26</td>
<td>9.26</td>
<td>7.84</td>
</tr>
<tr>
<td>09/17/2019</td>
<td>09/21/2019</td>
<td>17.45</td>
<td>8.87</td>
<td>7.49</td>
<td>7.76</td>
</tr>
<tr>
<td>09/17/2019</td>
<td>09/21/2019</td>
<td>17.45</td>
<td>8.87</td>
<td>7.49</td>
<td>7.76</td>
</tr>
<tr>
<td>09/19/2019</td>
<td>09/23/2019</td>
<td>17.18</td>
<td>8.18</td>
<td>7.09</td>
<td>7.76</td>
</tr>
<tr>
<td>09/18/2019</td>
<td>09/22/2019</td>
<td>17.07</td>
<td>8.39</td>
<td>7.74</td>
<td>7.67</td>
</tr>
<tr>
<td>09/21/2019</td>
<td>09/25/2019</td>
<td>18.42</td>
<td>7.82</td>
<td>7.16</td>
<td>7.02</td>
</tr>
<tr>
<td>09/26/2019</td>
<td>09/30/2019</td>
<td>17.78</td>
<td>7.05</td>
<td>7.16</td>
<td>7.05</td>
</tr>
<tr>
<td>09/26/2020</td>
<td>09/30/2020</td>
<td>17.90</td>
<td>7.25</td>
<td>4.97</td>
<td>7.25</td>
</tr>
<tr>
<td>09/27/2020</td>
<td>09/31/2020</td>
<td>15.30</td>
<td>6.00</td>
<td>3.47</td>
<td>0.86</td>
</tr>
<tr>
<td>10/01/2020</td>
<td>10/05/2020</td>
<td>18.15</td>
<td>9.88</td>
<td>5.62</td>
<td>6.52</td>
</tr>
<tr>
<td>10/02/2020</td>
<td>10/06/2020</td>
<td>15.12</td>
<td>5.22</td>
<td>3.16</td>
<td>6.52</td>
</tr>
<tr>
<td>10/03/2020</td>
<td>10/07/2020</td>
<td>18.42</td>
<td>8.42</td>
<td>3.90</td>
<td>6.42</td>
</tr>
<tr>
<td>10/04/2020</td>
<td>10/08/2020</td>
<td>16.33</td>
<td>7.71</td>
<td>7.11</td>
<td>6.11</td>
</tr>
<tr>
<td>10/05/2020</td>
<td>10/09/2020</td>
<td>16.05</td>
<td>3.92</td>
<td>3.92</td>
<td>7.75</td>
</tr>
<tr>
<td>10/05/2020</td>
<td>10/09/2020</td>
<td>16.05</td>
<td>3.92</td>
<td>3.92</td>
<td>7.75</td>
</tr>
<tr>
<td>10/06/2020</td>
<td>10/10/2020</td>
<td>15.90</td>
<td>4.25</td>
<td>4.25</td>
<td>5.85</td>
</tr>
<tr>
<td>10/07/2020</td>
<td>10/11/2020</td>
<td>15.23</td>
<td>8.20</td>
<td>8.30</td>
<td>5.83</td>
</tr>
<tr>
<td>10/08/2020</td>
<td>10/12/2020</td>
<td>14.46</td>
<td>5.90</td>
<td>5.90</td>
<td>4.75</td>
</tr>
<tr>
<td>10/10/2020</td>
<td>10/14/2020</td>
<td>13.36</td>
<td>5.38</td>
<td>5.38</td>
<td>4.85</td>
</tr>
<tr>
<td>10/11/2020</td>
<td>10/15/2020</td>
<td>14.25</td>
<td>5.78</td>
<td>5.78</td>
<td>4.85</td>
</tr>
<tr>
<td>10/12/2020</td>
<td>10/16/2020</td>
<td>15.44</td>
<td>8.96</td>
<td>8.96</td>
<td>5.80</td>
</tr>
<tr>
<td>10/13/2020</td>
<td>10/17/2020</td>
<td>16.88</td>
<td>8.10</td>
<td>8.10</td>
<td>5.34</td>
</tr>
<tr>
<td>10/14/2020</td>
<td>10/18/2020</td>
<td>14.41</td>
<td>7.26</td>
<td>7.26</td>
<td>5.86</td>
</tr>
<tr>
<td>10/15/2020</td>
<td>10/19/2020</td>
<td>18.44</td>
<td>8.67</td>
<td>8.67</td>
<td>5.14</td>
</tr>
<tr>
<td>10/16/2020</td>
<td>10/20/2020</td>
<td>17.71</td>
<td>8.35</td>
<td>8.35</td>
<td>5.14</td>
</tr>
<tr>
<td>10/17/2020</td>
<td>10/21/2020</td>
<td>18.72</td>
<td>8.97</td>
<td>8.97</td>
<td>5.36</td>
</tr>
<tr>
<td>10/18/2020</td>
<td>10/22/2020</td>
<td>16.59</td>
<td>7.32</td>
<td>7.32</td>
<td>5.75</td>
</tr>
<tr>
<td>10/19/2020</td>
<td>10/23/2020</td>
<td>11.31</td>
<td>7.06</td>
<td>7.06</td>
<td>6.20</td>
</tr>
<tr>
<td>10/20/2020</td>
<td>10/24/2020</td>
<td>16.20</td>
<td>7.14</td>
<td>7.14</td>
<td>5.93</td>
</tr>
<tr>
<td>10/21/2020</td>
<td>10/25/2020</td>
<td>16.05</td>
<td>7.75</td>
<td>7.75</td>
<td>6.06</td>
</tr>
<tr>
<td>10/22/2020</td>
<td>10/26/2020</td>
<td>16.90</td>
<td>8.60</td>
<td>8.60</td>
<td>7.33</td>
</tr>
<tr>
<td>10/23/2020</td>
<td>10/27/2020</td>
<td>16.73</td>
<td>8.20</td>
<td>8.20</td>
<td>7.59</td>
</tr>
<tr>
<td>10/24/2020</td>
<td>10/28/2020</td>
<td>17.45</td>
<td>9.65</td>
<td>9.65</td>
<td>7.87</td>
</tr>
<tr>
<td>10/25/2020</td>
<td>10/29/2020</td>
<td>18.05</td>
<td>9.84</td>
<td>9.84</td>
<td>6.82</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:

```
GET * 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

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_class_1_Price</th>
<th>published_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/18/2020</td>
<td>01/18/2020</td>
<td>17.55</td>
<td>17.55</td>
<td>9.90</td>
<td>01/18/2020</td>
</tr>
<tr>
<td>02/15/2020</td>
<td>01/26/2020</td>
<td>17.46</td>
<td>10.82</td>
<td>10.47</td>
<td>02/15/2020</td>
</tr>
<tr>
<td>03/14/2020</td>
<td>01/18/2020</td>
<td>16.04</td>
<td>10.19</td>
<td>10.04</td>
<td>03/14/2020</td>
</tr>
<tr>
<td>04/18/2020</td>
<td>04/22/2020</td>
<td>12.90</td>
<td>8.72</td>
<td>8.93</td>
<td>04/18/2020</td>
</tr>
<tr>
<td>05/16/2020</td>
<td>05/20/2020</td>
<td>11.42</td>
<td>7.08</td>
<td>6.68</td>
<td>05/16/2020</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:
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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Milk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Butterfat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Skim</td>
</tr>
<tr>
<td>3346</td>
<td>Class I Utilization</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Milk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Butterfat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NFS</td>
</tr>
<tr>
<td>3347</td>
<td>Class II Utilization</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Milk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Butterfat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NFS</td>
</tr>
<tr>
<td>3348</td>
<td>Class III Utilization</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Milk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Butterfat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protein</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other Solids</td>
</tr>
<tr>
<td>3349</td>
<td>Class IV Utilization</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Milk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Butterfat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NFS</td>
</tr>
<tr>
<td>3350</td>
<td>Total Receipts of Producer Milk</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td>(Pool Data Prior to October 2020)</td>
<td>Producers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Receipts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avg Daily</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Butterfat</td>
</tr>
<tr>
<td>3461</td>
<td>Producer Receipts</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td>(Pool Data starting with October 2020)</td>
<td>Producers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Receipts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avg Daily</td>
</tr>
<tr>
<td>3462</td>
<td>Producer Milk Components</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td>(Pool Data starting with October 2020)</td>
<td>Receipts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Butterfat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NFS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protein</td>
</tr>
<tr>
<td>3351</td>
<td>Uniform Milk Prices</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Milk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Butterfat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Skim</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PPD</td>
</tr>
<tr>
<td>3352</td>
<td>Price and Pool – Monthly</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Price and Pool Monthly</td>
</tr>
<tr>
<td>3353</td>
<td>Price and Pool – Annual</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Price and Pool Annual</td>
</tr>
<tr>
<td>3354</td>
<td>Advanced Prices by Order</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Advanced Class Prices by Order</td>
</tr>
<tr>
<td>3355</td>
<td>Class Prices by Order</td>
<td>Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Final Class Prices by Order</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/9346/Milk

Query Params

<table>
<thead>
<tr>
<th>KEY</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Value</td>
</tr>
</tbody>
</table>

Body

```json
"reportSections": "Milk",
"reportSections": [
  "Summary",
  "Milk",
  "Utilization",
  "Butterfat",
  "NPS"
],
"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": "3,565",
  "feb": "3,745",
  "mar": "4,639",
  "apr": "3,611",
  "may": "3,830",
  "jun": "3,634",
  "jul": "3,549",
  "aug": "3,898",
  "sep": "3,875",
  "oct": "3,546",
  "nov": "3,958"
}
```
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:

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.
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:

```plaintext
```

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

**Expected results in Postman:**

![Postman API Response Example](image)

**Expected results in Excel:**

| Column | Report Month | Report Year | Slug_ID | Report_Name | Report_Order_Name | Report_Order_No | Column_1 | Column_2 | Column_3 | Column_4 | Column_5 | Column_6 | Column_7 | Column_8 | Column_9 | Column_10 |
|--------|--------------|-------------|---------|-------------|------------------|----------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Dec    | 2018         | All Markets Combined | 2.5145 | 2.3203 | 2.4341 | 2.5087 | 2.6385 | 2.8982 | 2.5463 |
| Dec    | 2018         | Appalachia | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | Arizona | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | California | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | Central | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | Florida | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | Midwest | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | Northeast | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | Pacific Northwest | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | Southeast | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | Southwest | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
| Dec    | 2018         | Upper Midwest | 1.3315 | 1.5104 | 1.4764 | 1.5677 | 1.6416 | 1.7681 | 1.5930 |
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:

```

Params

Query Params

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>report_year=2015:2018</td>
</tr>
</tbody>
</table>

Body

```

Expected results in Excel:

```
<table>
<thead>
<tr>
<th>Column1</th>
<th>Column2</th>
<th>Column3</th>
<th>Column4</th>
<th>Column5</th>
<th>Column6</th>
<th>Column7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep</td>
<td>2015</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>Washington, DC</td>
</tr>
<tr>
<td>Sep</td>
<td>2016</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>Washington, DC</td>
</tr>
<tr>
<td>Sep</td>
<td>2017</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>Washington, DC</td>
</tr>
<tr>
<td>Sep</td>
<td>2018</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>Washington, DC</td>
</tr>
</tbody>
</table>
```

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:
## 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>
<th>Column 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Producers: All Markets Combined</td>
<td>38,295</td>
<td>32,801</td>
<td>33,380</td>
<td>36,079</td>
<td>32,042</td>
<td>32,688</td>
<td>33,688</td>
<td>34,114</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Appalachia</td>
<td>5,483</td>
<td>4,797</td>
<td>4,877</td>
<td>4,877</td>
<td>4,707</td>
<td>4,707</td>
<td>4,707</td>
<td>4,707</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Arizona</td>
<td>931,190</td>
<td>8,810</td>
<td>8,810</td>
<td>8,810</td>
<td>8,810</td>
<td>8,810</td>
<td>8,810</td>
<td>8,810</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Central</td>
<td>31,687</td>
<td>2,995</td>
<td>2,995</td>
<td>2,995</td>
<td>2,995</td>
<td>2,995</td>
<td>2,995</td>
<td>2,995</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Florida</td>
<td>6,340</td>
<td>5,460</td>
<td>5,460</td>
<td>5,460</td>
<td>5,460</td>
<td>5,460</td>
<td>5,460</td>
<td>5,460</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Midwest</td>
<td>10,510</td>
<td>9,089</td>
<td>9,089</td>
<td>9,089</td>
<td>9,089</td>
<td>9,089</td>
<td>9,089</td>
<td>9,089</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Northwest</td>
<td>411,433</td>
<td>31,322</td>
<td>31,322</td>
<td>31,322</td>
<td>31,322</td>
<td>31,322</td>
<td>31,322</td>
<td>31,322</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Pacific Northwest</td>
<td>234,446</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Southeast</td>
<td>15,672</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Southwest</td>
<td>26,466</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Producers: Upper Midwest</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: All Markets Combined</td>
<td>32,507</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: Appalachian</td>
<td>5,483</td>
<td>4,797</td>
<td>4,877</td>
<td>4,877</td>
<td>4,707</td>
<td>4,707</td>
<td>4,707</td>
<td>4,707</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: Arizona</td>
<td>331,480</td>
<td>292,142</td>
<td>292,142</td>
<td>292,142</td>
<td>292,142</td>
<td>292,142</td>
<td>292,142</td>
<td>292,142</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: Central</td>
<td>13,270</td>
<td>11,266</td>
<td>11,266</td>
<td>11,266</td>
<td>11,266</td>
<td>11,266</td>
<td>11,266</td>
<td>11,266</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: Florida</td>
<td>6,030</td>
<td>5,310</td>
<td>5,310</td>
<td>5,310</td>
<td>5,310</td>
<td>5,310</td>
<td>5,310</td>
<td>5,310</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: Midwest</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: Northwest</td>
<td>234,446</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td>188,467</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: Southeast</td>
<td>15,672</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td>14,792</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: Southwest</td>
<td>26,466</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td>22,312</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipts: Upper Midwest</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily: All Markets Combined</td>
<td>31,585</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily: Appalachian</td>
<td>5,758</td>
<td>7,152</td>
<td>5,758</td>
<td>5,758</td>
<td>5,758</td>
<td>5,758</td>
<td>5,758</td>
<td>5,758</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily: Arizona</td>
<td>822,849</td>
<td>793,598</td>
<td>793,598</td>
<td>793,598</td>
<td>793,598</td>
<td>793,598</td>
<td>793,598</td>
<td>793,598</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily: Florida</td>
<td>9,744</td>
<td>9,149</td>
<td>9,149</td>
<td>9,149</td>
<td>9,149</td>
<td>9,149</td>
<td>9,149</td>
<td>9,149</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily: North Central</td>
<td>4,036</td>
<td>4,036</td>
<td>4,036</td>
<td>4,036</td>
<td>4,036</td>
<td>4,036</td>
<td>4,036</td>
<td>4,036</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily: Pacific Northwest</td>
<td>124,600</td>
<td>11,600</td>
<td>11,600</td>
<td>11,600</td>
<td>11,600</td>
<td>11,600</td>
<td>11,600</td>
<td>11,600</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily: Southwest</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg Daily: Upper Midwest</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td>30,156</td>
<td></td>
<td></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:

```

<table>
<thead>
<tr>
<th>Query Params</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>report_year=2018;report_month=aug</td>
<td></td>
</tr>
</tbody>
</table>

Body

```

```
"reportSection": "Price and Pool Monthly",
"reportSections": [
  "Summary",
  "Price and Pool Monthly"
],
"state": {
  "totalRows": 11,
  "returnedRows": 11,
  "userAllowedRows": 99999
},
"results": []
```

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:

```


<table>
<thead>
<tr>
<th>Params</th>
<th>Authorization</th>
<th>Headers</th>
<th>Body</th>
<th>Pre-request Script</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>correctionsOnly</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Body

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```
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:
3.5 Examples of enhanced “All Sections” data pull

Users are now able to pull “All Sections” for an “X” amount of the most recent days using the “lastReports” attribute. To pull “All Sections” of the last ten reports of the “5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)”, the sample syntax would be:

https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2466?lastReports=10&allSections=true

Expected results would be:

```json
{
  "reportSections": 
  "Summary",
  "reportSections": [
    "Summary",
    "Detail",
  ],
  "stats": {
    "totalRows": 10,
    "returnedRows": 10,
    "userAllowedRows": 99999
  },
  "results": [
    {
      "report_date": "09/21/2023",
      "previous_day_head_count": "2,323",
      "narrative": null,
      "report_title": "5 Area Daily Weighted Average Direct Slaughter Cattle - Negotiated (PDF) (LM_CT100)",
      "slug_name": "AMS_2466",
      "slug_id": "2466",
      "office_name": "St Joseph, MO",
      "office_code": "LS-83",
      "office_city": "Saint Joseph",
      "office_state": "MO",
      "market_location_name": "St. Joseph, MO",
      "market_location_city": "St. Joseph",
      "market_location_state": "MO",
```
4 Email API Feature

The LMR API now supports “sendEmail” and “Email=” attributes. Any standard API request can now be sent to the LMR API with these attributes and an email will be sent to that address with a link to download the zip file of your data request. The zip file will contain one or more CSV files. The URL is only good for 24hrs. The email address can be any email address.

For API requests where “allsections” is used. The zip file will contain separate CSV files for each section of the report.

Important Note: When using this feature, 30 days is the maximum date range that can be passed to the API for processing. Date ranges over 30 days will receive a message that the range needs to be reduced. This is done for performance reasons.

Examples are below.

1. The following URL pulls a date range of date for all sections and emails to provided address.

   https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2451?q=report_date=04/07/2020:05/07/2020&allSections=true&sendEmail=true&email=fakeemail@gmail.com

2. The following URL pulls the “Current Volume” section of the report for a date range.

   https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2451/CurrentVolume?q=report_date=04/07/2020:05/07/2020&sendEmail=true&email=fakeemail@usda.com

3. The following URL pulls the “Summary” section for one day.

   https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2451?q=report_date=04/07/2020&sendEmail=true&email=fakeemail@usda.com

4. The following URL pulls “All Sections” but only Corrections for a date range.

   https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2451?q=report_date=04/07/2020:05/07/2020&allSections=true&correctionsOnly=true&sendEmail=true&email=fakeemail@usda.com

5. The following URL pulls the “Current Volume” for a date range, but only Corrections.

   https://mpr.datamart.ams.usda.gov/services/v1.1/reports/2451/CurrentVolume?q=report_date=04/07/2020:05/07/2020&correctionsOnly=true&sendEmail=true&email=fakeemail@usda.com

4.1 Addition of “is_corrections” field to Summary/Header Section.

AMS has added an additional field to the Summary/Header section of all reports on LMR that will allow users to know if a particular report has been issued a correction. Users can filter on this attribute and only pull corrections.
5 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.