320 E. Big Beaver Road
Suite 500
Troy, MI 48083
Phone 1 (248) 680-6200
Toll Free 1 (888) 274-5372
jdpower.com/business

J.D. POWER

Valuation Services REST Valuation API

Endpoint and Interface Documentation

Last updated: June 12, 2025

Table of Contents

1. Contents

1. Introduction

J.D. Power Valuation Services provides a REST-based web API to allow outside developers to integrate our vehicle selection and valuation processes into their own applications: either internal applications or VAR-delivered applications. The API is REST-based and utilizes J.D. Power's cloud infrastructure to provide better scaling and performance than our legacy data-center server-based SOAP web service.

1.1. Legacy Conversion notes

The REST-based API has been developed to provide as close to a 1:1 functionality mapping to the legacy/SOAP API as possible. Each endpoint or other specific topic in the documentation below will have any conversion notes indicated as below:

> *Legacy conversion:* This replaces …

Overall, there are the following changes from the SOAP API to the REST API:

- Interface changed from SOAP-based with complex XML data objects to REST-based with JSON data objects (XML response optional)
- SecureLogin service's getToken() method removed; access control is via api-key in header
- Input Period field changed from integer representing year/month to a string representing a complete date (as YYYY-MM-DD format.)
- Source data/content changed from legacy/book content to more closely align with the expanded content in use in the NADA Values Online web application
    - The Vehicle ID in use in this new data system is completely different from the legacy system. There is no direct correlation between the SOAP API's UID and the REST API's UcgVehicleId as there are often 1-to-many relationships between the legacy vehicle list and the expanded/master vehicle list.
- Adjusted Values Business Rules data added to most valuation endpoints/methods, removing need for the Floor Values method
- Mutual Exclusive and Package Inclusive option retrieval methods combined into single call.
    - This data has also now been moved into the consolidated Accessory Data endpoints, completely removing the need for a separate Accessory Logic API call.
- Several methods which provided an optional VIN input in the SOAP system have been split into two methods; one using vehicle ID only, the other using vehicle ID and VIN.
- An additional change in the VIN decoding refers to the use of a partial VIN; our legacy systems will decode based on the first 10 characters of a VIN (VIN10); our new system requires the 11th digit to receive a match, with the exceptions of the multiple Build endpoints which require a full 17-character VIN. Further, the new API requires a full 17character VIN input.

There is a list of SOAP Method to REST endpoints near the end of the documentation.

Developer note: Code examples are included for each endpoint using the common cURL command.

2. Change Control

For older update notes, please see Change Control History at the end of this document.

2025-07-08

API updates: Optional update to the **`statusByVehicleId`** and **`statusByVin`** endpoints to treat vehicles in our Early Release Values status to appear as "valued" rather than "unvalued" for the purpose of **`FirstValued`** and **`TargetPeriodStatus`** fields.

Documentation updates: Corrected documentation section name for statusByVehicleId endpoint; prior header used statusByUcgVehicleId.

2025-06-12

Documentation updates: Updated the various Accessory endpoints documentation with notes on handling `Includes` and `Excludes` when an accessory is flagged as either `isadded`:1 or `isincluded`:1.
Also added a Definition for Body Inclusive accessories.

2025-03-07

API updates: VIN Precision+ and VIN Precision+ Content Adjustment update: Added `VinSourceDetail` optional parameter to indicate the source of the build data.
Documentation updates: Updated the VIN Precision+ Content Adjustment section with the VinSourceDetail enhancement information. Updated all VIN Precision+ and VIN Precision+ Content Adjustment endpoints for new optional vinsourcedetail parameter.
Added missing VinSource parameter to VIN Precision+ endpoint references.
Added the missing MSRP parameter and corrected the base URL in the **msrpVehicleAndValuesByVin** endpoint documentation.
Updated **Deprecation** note for **accessoryLogicByVehicleId** endpoint – this endpoint is scheduled for removal from the API in Q3 or Q4 of 2025.
Added note regarding use of **"error" field** as a warning rather than an actual error (typically in empty responses) and a corresponding note in the Exceptions section.
Added note that you can **pass the VIN in valuation by ID calls** for logging/ reporting purposes.

2025-01-30

Documentation updates: Updated the **VIN Decode definitions: Build results** definition section to include references to Trim Reduction (2024) and VIN Precision+ Content Adjustment (2024-2025) enhancements.
Also updated the **vinsource field definition** for its behavior in VP+ Content Adjustment vs. Trim Reduction.

2025-01-03

API updates: VIN Precision+ Content Adjustment update: UAT deployment enhancement to remaining legacy VIN Valuation endpoints and accessoryDataByVinAndVehicleId endpoint.
Documentation updates: Added a note regarding the VID field location in the weeklyBuildVehicleAndValuesByVin endpoint's result array.

2024-11-21

API updates: **VIN Precision+ Content Adjustment** update:
Reverted the default behavior on the legacy VIN Value endpoints to legacy behavior, now requiring optional parameter to enable the "as built" enhancement.

Documentation updates: Removed reference to an optional `vinsource` parameter in the VP+ Content Adjustment endpoints. The "opt-in" `vpp` parameter controls this behavior.

2024-11-06

API updates: **VIN Precision+ Content Adjustment** update:
Monthly "legacy" VIN Valuation endpoints will now return VIN Precision+ results when exact VIN data is available, otherwise their prior behavior. Includes optional/hidden parameter to override the enhancement, and a new field in the response to indicate the type of result returned (VIN Precision+ or Standard.)
Note: The Weekly Used and Combined Used Build Value endpoints will be updated soon.

2024-07-15

API updates: Added Period validation checks for /build/* endpoints.
Documentation updates: Removed outdated note regarding availability of period=0 special parameter in /build/ endpoints.

2024-07-30

Documentation updates: Corrected parameter list for `weeklyBuildVehicleAndValuesByVin` endpoint

2024-07-16

API updates: Added Chrome Verified Engineered sources to the remaining VIN Precision+ endpoints on UAT.

- buildAccessoriesByVin *note: originally listed incorrectly on 7/16 update below*
- weeklyBuildVehicleAndValuesByVin
- buildVehicleAndUsedValuesByVin

2024-07-16

API updates: Added Chrome Verified Engineered sources to some of the VIN Precision+ endpoints on UAT. The remaining VIN Precision+ endpoints should receive the same enhancement later in July. Expect Production Deployment in mid-August.

- buildDataAvailableByVin
- buildVehicleAndValuesByVin
- buildAccessoryDataByVin *note: originally listed incorrectly as buildAccessoriesByVin*

The same update applies to the recent Trim Reduction enhancement to the 'standard' VIN Decode endpoints from April.

- vehiclesByVin
- vehiclesWithTrimByVin

Older Change Control notes have been moved to the Change Control History section at end of this document (in reverse chronological order, newer items first). Re-ordered the main Change Control section to newest at top order as well.

3. Online service documentation

Our REST API includes an online documentation page where you can see the available endpoints and their inputs and try them out and see the result formats.

You can open a web browser and go to the below URLs; please note that there is a different URL for the UAT/test deployment vs. the Production deployment, and you must use the corresponding (test vs. production) API Key value to try the methods out. You do not need an API Key to view the documentation page and see the endpoints and input parameters.

UAT URL:

https://cloud.jdpower.ai/data-api/UAT/valuationservices/api-docs/static/index.html#/

Production URL:

https://cloud.jdpower.ai/data-api/valuationservices/api-docs/static/index.html#/

3.1. Swagger files

An OpenApi-compatible Swagger.json file is available from the documentation links above; it is available via a hyperlink near the top of the documentation web page. When you click on the link, a new page will be loaded. Save the loaded page/text to get a copy of the swagger.json file.

Please note that the UAT and PROD deployments will each have their own unique swagger file.

Developer note: ***Because our API responses are in many cases dynamic, the swagger.json may not include the response data schema for all endpoints. Review the example responses in each endpoint in the documentation or see the list below for endpoints that have the response schema defined currently.***

The response schema is currently available on for the below endpoints as of the dates for each phase noted below. Endpoints in grey text have their response schema defined on the UAT API only as of the updated date of this document.

Phase	Endpoint
1	/valuation/accessoriesByVehicleId
1a	/valuation/accessoriesByVinAndVehicleId
1	/valuation/accessoryDataByVehicleId
1a	/valuation/accessoryDataByVehicleId
1b	/valuation/accessoryLogicByVehicleId
4	/valuation/apiUsageReport
4	/valuation/batchDefaultVehicleAndValuesByVin (GET informational message only)
1a	/valuation/bodies
1	/valuation/defaultVehicleAndValuesByVin
1a	/valuation/highVehicleAndValuesByVin
1a	/valuation/lowVehicleAndValuesByVin
1a	/valuation/makes
2	/valuation/masterToLegacyVehicleMapping
2	/valuation/masterVehiclesByLegacyId
2	/valuation/masterVehiclesByLegacyIdAndVin (GET and POST)
1a	/valuation/mileageByVehicleId
1a	/valuation/models

1a	/valuation/msrpVehicleAndValuesByVin
1	/valuation/newestAvailablePeriod
1	/valuation/regionIdByStateCode
1b	/valuation/regions
2	/valuation/statusByVehicleId
2	/valuation/statusByVin
1a	/valuation/valueByVehicleId
1	/valuation/vehicleAndValueByVehicleId
1a	/valuation/vehicleInformationByVehicleId
1	/valuation/vehicleList
1a	/valuation/vehiclesByVin
1	/valuation/weeklyAuctionValuesByVehicleId
1	/valuation/weeklyAuctionValuesByVinAndVehicleId
1	/valuation/weeklyDefaultVehicleAndUsedValuesByVin
1	/valuation/weeklyVehicleAndValueByVehicleId
1b	/valuation/weeklyDefaultVehicleAndUsedValuesByVin
1b	/valuation/weeklyUsedValuesByVehicleId
1b	/valuation/weeklyUsedValuesByLegacyId
1.	/valuation/years
4	/build/buildDataAvailableByVin
4	/build/buildVehicleAndValuesByVin
4	/build/buildAccessoriesByVin
4	/build/weeklyBuildVehicleAndValuesByVin

Deployment schedule for swagger.json updates *(response schemas manually defined)*

Phase 1	(UAT) 2021-04-25	
Phase 1a	(UAT) 2021-05-10	
Phase 1b	(UAT) 2021-05-11	
Phase 1 – 1b	(PROD) 2021-05-27	
Phase 2	(UAT) 2021-06-09	
	(PROD) 2021-06-25	
Phase 3*	(UAT) 2021-08-06	* optional Error field to appropriate response schemas
Phase 4	(UAT) 2021-08-17	

3.1.1.Swagger parameter description updates (February 2024)

These endpoints had their parameter descriptions updated in February 2024. Unless noted, the only change was to the vehicletype parameter description. No endpoint functionality changes were made.

- accessoriesByVehicleId
- accessoriesByVinAndVehicleId
- accessoryLogicByVehicleId
- bodies
- defaultVehicleAndValuesByVin
- highVehicleAndValuesByVin
- lowVehicleAndValuesByVin
- makes
- models
- msrpVehicleAndValuesByVin
- valueByVehicleId
- vehicleAndValueByVehicleId
- vehicleInformationByVehicleId
- vehiclesByVin
- years
- vehicleList
- vehiclesWithTrimByVin

The below endpoints had all their parameter descriptions added in the Swagger docs:

- weeklyDefaultVehicleAndUsedValuesByVin
- weeklyVehicleAndValueByVehicleId
- weeklyUsedValuesByVehicleId
- weeklyUsedValuesByLegacyId
- weeklyAuctionMileageByVehicleId

Others:

- defaultVehicleAndUsedValuesByVin
    - Updated vehicletype and userinfo parameter descriptions only

4. Authentication

4.1. Authentication

4.1.1.api-key

Each request will include a user API Key in the Request **header** section (as api-key) as below:

```
api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63
```

(this is not a valid key)

The API Key will be provided to the developer by UCG staff to uniquely identify the external user and is used to track all activity in the service, including those that incur a license charge.

Each API Key is specific for a company and to an API root and all methods in the root path and will grant access specifically to a Production or UAT version's set of endpoints.

Legacy conversion: This replaces the username/password → token approach used in the previous SOAP-based API (SecureLogin service's getToken() method).

Developer note:

The sample requests below are referencing the cUrl command line tool, where headers are added as:

```
-H "api-key: your_api_key_here"
```

C# developers for instance would instead use something as below:

```
request.Headers.Add("api-key", "your_api_key_here");
```

5. Common Service Elements and Design Notes

5.1. GET vs POST

All methods are GET unless otherwise indicated. All "batch" capable endpoints use POST.

Legacy conversion: This replaces the HTTP POST interface of the SOAP service.

5.2. XML or JSON

Data is returned as JSON-formatted objects, unless you over-ride by adding the following HTTP header:

```
Accepts: text/xml
```

Developer note: As of October 1, 2019, requesting an XML reply will add a <response> parent tag to the XML payload you receive. This was added to allow automatic parsing/injection of the XML response, as the older responses did not have a parent tag and so failed automatic XML parsing.

5.3. Test (QA) vs. Production deployments

The J.D. Power AI Cloud API platform supports two different deployments for each API; one is for testing (QA) and one is the production deployment. Access to each deployment requires a different API Key to be supplied to the developer by UCG staff, like how our legacy SOAP API used to have different accounts for test vs. production.

Developer note: The QA deployment endpoint path (URL) is generated by adding "/UAT" immediately before the API name in the URL, i.e.

Prod endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/years

Test endpoint: https://cloud.jdpower.ai/data-api/UAT/valuationservices/valuation/years

Developer note: ***We require HTTPS for all endpoints. As of June 2021 we support TLS 1.0 through TLS 1.3 only; SSL v3 and older HTTPS protocols are not supported.***

5.4. License Decrement

The endpoint documentation provides a note as to whether each endpoint counts as a 'hit' against a user's license, decrementing the remaining lookups available. Unless otherwise noted, valuation calls that are flagged as license decrementing will not decrement the license if the vehicle's base values are all $0.

Some endpoints may decrement the license by more than 1 on successful lookups (see buildVehicleAndValuesByVin for example.)

5.5. Common parameters for all methods and responses

5.5.1.period

Virtually all methods include a required parameter named “period” representing the effective date for the request. In other words, to get a value for a vehicle as of May 5, 2018, you would set the period = 2018-05-01 (or any other date inside May 2018.)

Legacy conversion: The use of a specifically formatted string in the REST API replaces the integer formatted year/month of the SOAP API, i.e., 2019-02-25 replaces 201902.

Legacy conversion: The use of special literal period value 0 to indicate the current period is extended in the REST API to indicate “use the current date” vs. the SOAP API’s “use the current year/month”. Any endpoint that does not require a Period input parameter does not support this special Period parameter value but will also simply ignore any period parameter.

Developer note: ***period=0 uses the API application server’s clock to determine the effective date. The servers are Amazon Web Services-based and use UTC (Universal Time), not Eastern time as previously noted.***

Legacy conversion: Based on the underlying data platform, there is no support for the SOAP API’s special period value of 1, which was used to indicate ‘use the newest available period.’ A new REST API endpoint (newestAvailablePeriod) has been added and can be called to identify whether the upcoming month’s data has become available yet.

5.5.2.vehicletype

This parameter is optional and applies a filter to the data set being searched. The options are:

- **`UsedCar`** for cars and light-duty trucks
- **`CommercialTruck`** for medium- and heavy-duty trucks
- **`Motorcycle`** (added Q1/2020) which will filter to only Motorcycles
- **`NoMotorcycle`** (added Q1/2020) which will filter to cars and trucks only (including Commercial Truck vehicles) (optional alternative: NoMotorcycles)
- or do not include the parameter, which will not apply a filter (i.e., all vehicle types.)

Legacy conversion: Most of the SOAP API methods had a similar vehicle type field in the input, but some of them worked by defaulting to UsedCar if the type wasn’t selected. The REST API allows you to continue to select the vehicle type as before, or to remove the vehicle type filter completely.

5.5.3.requestId

This is part of every response and is critical in debugging. It is not an input parameter.

Developer note: It is very important to keep track of the requestId for debugging purposes, especially during development of your application. With it, our staff can track the progress of your entire request from when it first is submitted to the API layer until a response is returned.

5.5.4.authId

This is part of every response and displays the “user name” associated with the input api-key. It is not an input parameter.

5.5.5.userinfo

This is an optional* string input parameter for all endpoints except those noted in the individual endpoint description. It allows an API user to assign and control unique identifiers to individual users, locations, or in the case of our Value-Added Resellers, individual customers. The **`userinfo`** values will be shown on the usage reports.

If no input is passed, the default value of **`Default`** is used/injected.

The **`userinfo`** value is returned in each response outside the query result array, just before the **`authId`** value.

> ***** - This parameter is optional at the API level. However, Value-Added Resellers (VARs) are required to make use of this parameter to identify each of their end-users for billing and tracking purposes. If you are a VAR and have any questions on the use of this field and how it impacts your billing, please reach out to your sales representative to discuss.

5.5.6.Notes on parameters

Input parameters may be passed in several ways, and both the parameter names and values are case-sensitive:

1. Passing inputs via querystring (parameters)

   This is the most common approach for calling RESTful APIs and is used in all the examples below.

   Because the parameters are passed as part of the complete URL, if any of the parameters include characters not valid for use in web addresses, you must perform URL-encoding on those special characters. For instance, replace any spaces inside the parameter with %20 or replace an ampersand with &

   Ex: use ?model=Town%20&%20Country

   instead of ?model=Town & Country

2. Passing inputs as form data inside the body

   Less common when calling RESTful APIs, you can use the HTML form data format as below; this will only work on methods which support POST.

```
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="vin"
1FMCU9J9JUB98016
------WebKitFormBoundary7MA4YWxkTrZu0gW--,
Content-Disposition: form-data; name="vin"
1FMCU9J9JUB98016
------WebKitFormBoundary7MA4YWxkTrZu0gW--
Content-Disposition: form-data; name="period"
2019-01-01
------WebKitFormBoundary7MA4YWxkTrZu0gW—
```

3. Hidden API parameters

   Starting with the late February 2024 update allowing reverting **Early Release Values** and continuing with the late March 2024 Trim Reduction update, some API endpoints now allow for “un-documented” parameters to enable alternate processing without impacting existing applications.

   These parameters will only be defined in the documentation itself (see updated endpoint definitions for vehiclesByVin for instance) and omitting them or providing anything but the defined “special” parameter value(s), will cause the endpoint to work exactly as before.

4. Additional undefined parameters

   The API will accept additional parameters beyond those defined in endpoint. In general, those parameters will be ignored, unless they are classified as the new hidden/optional parameters (above). These additional parameters will be logged in the API logs.

5.6. Design Notes

5.6.1. Query Per Second / Throttling

In order to provide consistent response times to all users, our API implements a Query Per Second (API call per second, abbreviated QPS) throttle to prevent sudden spikes in demand by one or two users from negatively impacting all other platform users. If a single API Key submits more API requests in any second than its defined limit, the call will fail with an HTTP 429 (Rate Limit Exceeded) error message. Please note that in practice, HTTP 429 is not a server error message; it is a request to slow down your requests to reduce impact on the server.

Each user's API Key has a defined QPS setting; our system default is 30 QPS, allowing over 100,000 calls per hour. If your application will need to exceed this rate, please reach out to JD Power's support for assistance.

Each response now includes several custom HTTP headers which can be used to monitor your key activity to allow self-throttling. See the notes in the QPS error message section for details.

5.6.2.Premium endpoint groups

Premium endpoints are enhanced features available in the REST API. These enhancements offer more detailed data or other enhanced functionality compared to the standard endpoints.

Access to the Premium endpoints is restricted and are subject to additional licensing and fees. Please contact your sales representative for access and additional information. You can use UCG_Sales@jdpa.com if you don't have your specific contact's information.

Premium endpoints will have a different path such as `/valuationservices/build/` in place of the standard endpoint paths of `/valuationservices/valuation/` and will be noted as ***Premium*** in the endpoint definitions in the documentation.

5.6.3.Service Response Times

Our development and platform teams are constantly working to improve our API endpoint response times. Response times for the following scenarios will typically be higher than otherwise:

5. Historical data periods
   Over 98% of all API calls to our service reference the current or immediately future month when it is available. API calls to historical data periods will typically be accessing data that has fallen out of the database cache and so may be anywhere from slightly to significantly slower than calls to the current data.

6. Batch endpoints (including Vehicle List)
   Any batch endpoints, plus the vehicleList endpoint, can incur both additional database processing time as well as potentially significantly longer data transmission time due to the size of the response payload.

5.6.4. VIN Decoding approaches

Our API supports multiple approaches to decoding VINs for the purposes of valuing vehicles. The typical approaches are covered below.

Please see the definitions section for details on our VIN decoding processes.

7. Weekly Used Values variations

   Some of the listed endpoints in this section have alternates to be used when your application consumes our Weekly Used Values. These will be listed separately below (as WUV variant).

   If no direct Weekly Used Values endpoint exists, you may consider using the corresponding "monthly" endpoint, then using the weeklyUsedValuesByVehicleId endpoint to retrieve the Weekly Used Values by vehicle ID instead of by VIN.

8. **Decoding/valuing VINs in the background / non-interactive**

   This approach is typically used in applications which do not interact with a user, or with a user who has access to details of the vehicle being valued.

   In this scenario, the application typically will utilize one of our single-call VIN Decode + Valuation endpoints, which use different business rules to select and value a single vehicle when a VIN is shared/ambiguous.

   The application will select from the following VIN decode/valuation endpoints:

   - defaultVehicleAndValuesByVin
     - Values the vehicle selected by Valuation Services' analysts as the most representative trim/model for the VIN in question
     - WUV variant: weeklyDefaultVehicleAndUsedValuesByVin
   - highVehicleAndValuesByVin or lowVehicleAndValuesByVin
     - These value the vehicle that is the highest or lowest Clean Trade value for the VIN. The Low endpoint ignores any unvalued vehicles in this logic
   - msrpVehicleAndValuesByVin
     - Values the vehicle with the Base MSRP in our data which is closest to, without going over, the input MSRP.

3. Decoding/valuing VINs in an interactive application

This approach is typically used when there is a user interacting with the application who can be asked for additional details of the VIN being valued.

In this scenario, the application calls our basic VIN decoding endpoint and values the vehicle using the selected vehicles ID.

Developers Note: Because the final valuation is performed by vehicle ID and not VIN, this approach will not provide the VIN in the detailed usage report unless you pass the VIN in the API call. Passing VIN in an "byId" call will not impact the API response but will be logged with the call in our systems.

1. vehiclesByVin or vehiclesWithTrimByVin
    a. If this endpoint returns only one vehicle, skip directly to step 3
2. Present the list of vehicles to the user for selection
    a. Please note that while in most cases, the vehicles returned will only differ at the final bodystyle (trim, etc.) level, in some cases they may differ at the model level as well.
3. valueByVehicleId or vehicleAndValueByVehicleId
    a. Use the vehicleAndValueByVehicleId if your application needs the Base MSRP or any of the other vehicle attributes in addition to the values.
    b. WUV variants: weeklyUsedValuesByVehicleId or weeklyVehicleAndValueByVehicleId

4. VIN Precision+ decoding/valuation process

The VIN Precision+ system (VP+) provides additional levels of detail and accuracy, and can be used in either the interactive or non-interactive approaches

VIN Precision+ data is not available for all VINs. As of the time this section of the API documentation was created, we were seeing an approximate coverage of 80-85% of VINs passed through our systems by users, for model year 2015 and newer.

As a result, using the VIN Precision+ will require application code branches to handle “no results available” in either interactive or non-interactive systems.

Also please note that the VIN Precision+ endpoints are classified as premium endpoints, requiring additional licensing and potentially additional costs.

Please see the main documentation section for VIN Precision+ for more details.

1. Check for VP+ availability for the VIN using the buildDataAvailableByVin endpoint
    a. If this returns false, return to the non-VP+ code path (above)
2. If this returns true, value the VIN using the buildVehicleAndValuesByVin endpoint
    a. WUV variant: weeklyBuildVehicleAndValuesByVin
    b. Note that in uncommon cases, even with a true response from the buildDataAvailableByVin call, you will get an empty response from the Build Value call. Your application should handle this possibility and branch back to the non-VP+ code path as if the original Build Available call had been false.
3. If you are retrieving the accessory data in your application and received a successful response from the Build Value call (step 2 above), please remember to use the VIN Precision+ Accessory endpoint - buildAccessoriesByVin

5.6.5.Per Valuation Licensing and Avoiding Duplicate Lookup Charges

Many of our users are licensed on a per-valuation (or per-lookup) basis. Each endpoint below includes a callout ("License decrement:") that details whether the use of the endpoint potentially causes a license/lookup to be charged. In general – if the endpoint returns a non-$0 vehicle value, it gets charged.

Our API usage reporting process will perform a limited degree of de-duplication checking – when an API Key calls the same endpoint in consecutive calls with identical inputs, within two minutes, the 2nd call is considered a duplicate and will not be charged. (See the Weekly Used Values endpoint section for a note on how these are sometimes considered the same as the 'standard' or monthly endpoints for the purposes of this license lookup de-duplication process.)

The safest way to avoid multiple charges for the same vehicle is to store a 'last valued' date value for each vehicle in your system, and only perform a valuation API call if the new valuation date is in a different month (for monthly endpoints) or Monday – Sunday week (for weekly endpoints.)

You can also reference the FAQ section for additional details.

6. Service Methods

6.1. Vehicle Selection Methods

Vehicles may be selected for valuation using either UCG's VIN decode methodologies, or via selection by description (Year, Make, Model, etc.) In general, VIN decoding offers several advantages over description selection:

- Better verification/validation that the correct vehicle is being selected/valued
- More straightforward use in 'black box' systems

VIN selection methods are covered in the following section.

6.1.1.years

Request a list of the available Model Years in the data period and (optionally) vehicle type input.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/years

Parameters:

• period	string	required	Valuation date as YYYY-MM-DD
• vehicletype	string	optional	Filter to specific vehicle types Default: empty string, treated as "no filter"
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces the getYears() method in the legacy/SOAP API.

Developer note: Be aware that if you do not specify the vehicletype, you will see combined results, i.e., the total list of available model years for both UsedCar and CommercialTruck vehicles, and the UsedCar and CommercialTruck data sets do not always fully align on the available model years.

```
curl -X GET \
'https://cloud.jdpower.ai/data-api/valuationservices/valuation/years?
period=2018-01-01' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
{
  "requestId": "abc123defg56789",
  "result":
  [
      {
        "modelyear": "2019"
      },
      {
        "modelyear": "2018"
      },
      {
        "modelyear": "2017"
```

```
      }
  ]
  , "userinfo": "Default"
  , "authId": "UCG"
}
```

6.1.2.makes

Request a list of the applicable Makes in the data set for the period, (optional) vehicle type, and selected Model Year.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/makes

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• modelyear	int	required	Model year selected
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces the getMakes() method in the legacy/SOAP API.

Developer note: Be aware that if you do not specify the vehicletype, you will see combined results, i.e., the total list of available makes for both UsedCar and CommercialTruck vehicles for the input modelyear value. As a result, if you, for instance, specify a modelyear where there are no Commercial Truck vehicles available (ex: CommercialTruck covers model years 2000 - 2018 as of February 2019 data, while UsedCar covers 2000 – 2019.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/makes?
period=2018-01-01&modelyear=2017' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'

{
  "requestId": "abc123defg56789",
  "result": [
    {
      "make": "Acura"
    },
    {
      "make": "Alfa Romeo"
    },
    {
      "make": "Acura"
    }
  ]
  , "userinfo": "Default"
  , "authId": "UCG"
}
```

6.1.3.models

Request a list of the applicable models for the selected inputs.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/models

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• modelyear	int	required	Model year selected
• make	string	required	Make selected
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces the getSeries() method in the legacy/SOAP API. There is a slight difference between the Model (new) and Series (legacy) data.

Developer note: The vehicletype field is very optional here, as there are currently no instances where UCG uses the same Make description for both UsedCar and CommercialTruck types (cf. Ford vs. Ford Commercial, etc.)

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/models?
period=2018-01-01&modelyear=2017&make=Acura' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
    {
      "model": "ILX"
    },
    {
      "model": "MDX"
    },
    {
      "model": "NSX"
    }
   ]
   , "userinfo": "Default"
   , "authId": "UCG"
 }
```

6.1.4.bodies

Request a list of the applicable Bodystyles (vehicles) for the inputs. The return will include both the vehicle description and the UCG Vehicle ID, which can be used for valuation calls.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/bodies

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• modelyear	int	required	Model year selected
• make	string	required	Make selected
• model	string	required	Model selected
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces the getBodyUids() method in the legacy/SOAP API.

Developer note: The vehicletype field is very optional here, as there are currently no instances where UCG uses the same Make description for both UsedCar and CommercialTruck types (cf. Ford vs. Ford Commercial, etc.)

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
bodies?period=2018-01-01&modelyear=2017&make=Acura&model=ILX' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
    {
      "body": "Sedan 4D",
      "ucgvehicleid": "201712345",
      "vid": "88283"
    },
    {
      "body": "Sedan 4D Premium",
      "ucgvehicleid": "201712357"",
      "vid": "88284"
    }
     ]
       , "userinfo": "Default"
       , "authId": "UCG"
}
```

6.2. VIN Decode Methods

6.2.1.vehiclesByVin

Request the available Vehicles for an input VIN. If more than one vehicle is returned, the first is the UCG Default vehicle and the following are Alternate vehicles (see Definitions section.) If empty, the VIN doesn't decode in the data period or for the VehicleType filter selected. Not all VINs will decode in our systems or in all time periods.

License decrement: This method does not decrement the user's license under typical usage cases. If we see a user using this endpoint at significant volume without accompanying valuation calls, we may at our discretion require a separate license specifically for the non-valuation-related VIN decode API calls.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/vehiclesByVin

Parameters: Note the additional hidden parameters added 3/28/24 (UAT).

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be decoded
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• vpp	int	hidden	0 = Do not apply Trim Reduction. default behavior is Use Trim Reduction
• vinsource	int	hidden	1 = Add vinsource field to response outside the result array. default behavior is "do not include"
• oem	int	hidden	1 = Do not use Verified Engineered data. default behavior is Use Engineered

Legacy conversion: This replaces the getVehicles() method in the legacy/SOAP API.

Developer note: Utilizing the optional aspect of the vehicletype input field can be particularly useful here for developers who need to limit access to either Used Car and Commercial Truck datasets, or do not want to use Motorcycle data (NoMotorcycles vehicletype).

Developer note: As noted in the parameter notes for Hidden parameters, hidden parameters do not appear in the API Swagger definitions.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
vehiclesByVin?period=2018-01-01&vin= 1FMCU9J93JUB98016' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
    {
      "modelyear": "2018",
      "make": "Ford",
      "model": "Escape",
      "body": "Utility 4D Titanium EcoBoost 4WD 2.0L I4 Turbo",
      "ucgvehicleid": "201834687"",
      "vid": "93603"
    }
   ]
   , "userinfo": "Default"
   , "authId": "UCG"
}
```


Examples with Hidden Parameter usage:

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
vehiclesByVin?period=2024-03-01&vin= 1FMCU9J93JUB98016&vinsource=1' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
    {
      "modelyear": "2018",
      "make": "Ford",
      "model": "Escape",
      "body": "Utility 4D Titanium EcoBoost 4WD 2.0L I4 Turbo",
      "ucgvehicleid": "201834687"",
      "vid": "93603"
    }
   ]
   , "userinfo": "Default"
   , "vinsource": "Standard"
   , "authId": "UCG"
}
```


Note: the `vinsource` field will not appear unless you provide the hidden and optional parameter `vinsource` with value `1`.

Shared-VIN example:

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
vehiclesByVin?period=2018-01-01&vin=1N4AA51E09C800436' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
    {
      "modelyear": "2009",
      "make": "Nissan",
      "model": "Maxima",
      "body": "Sedan 4D S 3.5L V6",
      "ucgvehicleid": "200930033""",
      "vid": "53659"
    },
    {
      "modelyear": "2009",
      "make": "Nissan",
      "model": "Maxima",
      "body": "Sedan 4D SV 3.5L V6",
      "ucgvehicleid": "200930034""",
      "vid": "53660"
    },
    {
      "modelyear": "2009",
      "make": "Nissan",
      "model": "Maxima",
      "body": "Sedan 4D SV Premium 3.5L V6",
      "ucgvehicleid": "200931927""",
      "vid": "76253"
    }
   ]
   , "userinfo": "Default"
   , "authId": "UCG"
}
```

6.2.2.defaultVehicleAndValuesByVin

Request the vehicle description and values for the input VIN, based on the UCG Default decoding definition for VINs that can return more than one vehicle (see Definitions)

License decrement: This method **does** decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/defaultVehicleAndValuesByVin

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be decoded
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• vpp	int	hidden	1 = Apply VP+ Content Adjustment. 0 or omit (default) = legacy behavior
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

Legacy conversion: This replaces the getDefaultVehicleAndValuesByVin() method in the legacy/SOAP API.

Developer note: Utilizing the optional aspect of the vehicletype input field can be particularly useful here for developers who need to support access to both Used Car and Commercial Truck datasets.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
defaultVehicleAndValuesByVin?period=2018-01-01&vin=
1FMCU9J93JUB98016&region=1&mileage=7139' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
    {
      "modelyear": "2018",
      "make": "Ford",
      "model": "Escape",
      "body": "Utility 4D Titanium EcoBoost 4WD 2.0L I4 Turbo",
      "ucgvehicleid": "201834687",
      "vehicletype": "Light Truck",
      "mileageclass": "2",
      "basemsrp": "33395",
      "bodytype": "Utility",
      "doors": "4",
      "trim": "Titanium",
      "trim2": "EcoBoost",
```

```
      "drivetype": "4WD",
      "liters": "2.0",
      "engineconfiguration": "I",
      "cylinders": "4",
      "inductiontype": "Turbo",
      "transmission": "",
      "fueltype": "Gas",
      "wheels": "4",
      "curbweight": "3645",
      "gvw": null,
      "gcw": null,
      "ucgsubsegment": "Compact Utility",
      "modelnumber": "U9J",
      "rollupvehicleid": "201834687",
      "basecleantrade": "23525",
      "baseaveragetrade": "22375",
      "baseroughtrade": "20950",
      "basecleanretail": "26225",
      "baseloan": "21175",
      "averagemileage": "17500",
      "mileageadjustment": "725",
      "vinoptionstrade": "0",
      "vinoptionsretail": "0",
      "vinoptionsloan": "0",
      "adjustedcleantrade": "24250",
      "adjustedaveragetrade": "22300",
      "adjustedroughtrade": "21675",
      "adjustedcleanretail": "26950",
      "adjustedloan": "21175",
      "maxmileageadj": "11763",
      "minmileageadj": "-9410",
      "minadjretail": "420",
      "minadjcleantrade": "120",
      "minadjaveragetrade": "120",
      "minadjroughtrade": "120",
      "minadjloan": "0",
      "minadjretailforloan": "900",
      "vid": "93603"
    }
   ]
   , "userinfo": "Default"
   , "authId": "UCG"
}
```


Business rules and other notes:

- ***maxmileageadj:*** largest positive value that can be applied for mileage adjustments, if you call the MileageAdjust method. This cap is already factored into the mileageAdjustment field in this return data.
- ***minmileageadj:*** largest negative value that can be applied for mileage adjustments, if you call the MileageAdjust method. This cap is already factored into the mileageAdjustment field in this return data.
- ***minadjretail:*** minimum that mileage and option adjustments can reduce the final Adjusted Clean Retail value to, if you perform client-side accessory and/or mileage adjustment processing.

- ***minadjcleantrade:*** minimum that mileage and option adjustments can reduce the final Adjusted Clean Trade-In value to, if you perform client-side accessory and/or mileage adjustment processing.
- ***minadjavgtrade:*** minimum that mileage and option adjustments can reduce the final Adjusted Average Trade-In value to, if you perform client-side accessory and/or mileage adjustment processing.
- ***minadjroughtrade:*** minimum that mileage and option adjustments can reduce the final Adjusted Rough Trade-In value to, if you perform client-side accessory and/or mileage adjustment processing.
- ***minadjloan:*** minimum that mileage and option adjustments can reduce the final Adjusted Loan value to if you perform client-side accessory and/or mileage adjustment processing.
- ***minadjretailforloan:*** if the Adjusted Retail is reduced below this value, the Adjusted Loan should be set to $0 regardless of any other factors.

6.2.3.lowVehicleAndValuesByVin

Request the vehicle description and values for the input VIN, based on the lowest (non-$0) Clean Trade-In valued vehicle, for VINs that can return more than one vehicle (see Definitions.) If all vehicles for a given VIN are unvalued ($0), this will return the Default vehicle. The Low vehicle can be the same vehicle as the Default in some cases, or it may not. It uses the same input parameters and returns the same data structures as defaultVehicleAndValuesByVin (above.)

License decrement: This method **does** decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/lowVehicleAndValuesByVin

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be decoded
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• vpp	int	hidden	1 = Apply VP+ Content Adjustment. 0 or omit (default) = legacy behavior
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

Legacy conversion: This replaces the getLowVehicleAndValuesByVin() method in the legacy/SOAP API.

Developer note: Utilizing the optional aspect of the vehicletype input field can be particularly useful here for developers who need to support access to both Used Car and Commercial Truck datasets.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
lowVehicleAndValuesByVin?period=2018-01-01&vin=
1FMCU9J93JUB98016&region=1&mileage=7139' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

(see defaultVehicleAndValuesByVin method above for data return structure)

6.2.4.highVehicleAndValuesByVin

Request the vehicle description and values for the input VIN, based on the highest (non-$0) Clean Trade-In valued vehicle, for VINs that can return more than one vehicle (see Definitions.) If all vehicles for a given VIN are unvalued ($0), this will return the Default vehicle. The High vehicle can be the same vehicle as the Default in some cases, or it may not. It uses the same input parameters and returns the same data structures as defaultVehicleAndValuesByVin (above.)

License decrement: This method **does** decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/highVehicleAndValuesByVin

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be decoded
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• vpp	int	hidden	1 = Apply VP+ Content Adjustment. 0 or omit (default) = legacy behavior
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

Legacy conversion: This replaces the getHighVehicleAndValuesByVin() method in the legacy/SOAP API.

Developer note: Utilizing the optional aspect of the vehicletype input field can be particularly useful here for developers who need to support access to both Used Car and Commercial Truck datasets.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
highVehicleAndValuesByVin?period=2018-01-01&vin=
1FMCU9J93JUB98016&region=1&mileage=7139' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

(see defaultVehicleAndValuesByVin method above for data return structure)

6.2.5.msrpVehicleAndValuesByVin

Request the vehicle description and values for the input VIN, based on the vehicle whose base MSRP is closest to the input MSRP value, without going above it. If all vehicles for a given VIN have a base MSRP higher than the input MSRP value, this will return the Default vehicle. It returns the same data structures as defaultVehicleAndValuesByVin (above.)

License decrement: This method **does** decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/msrpVehicleAndValuesByVin

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be decoded
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• msrp	int	required	MSRP to be matched in decode logic
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• vpp	int	hidden	1 = Apply VP+ Content Adjustment. 0 or omit (default) = legacy behavior
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

Legacy conversion: This replaces the getMsrpVehicleAndValuesByVin() method in the legacy/ SOAP API.

Developer note: Utilizing the optional aspect of the vehicletype input field can be particularly useful here for developers who need to support access to both Used Car and Commercial Truck datasets.

Developer note: Note that in some cases one or more vehicles for a given VIN may not be accessible to this endpoint in two scenarios:

1. If a vehicle has no baseMsrp value in our data (not available from the manufacturer or other sources), that vehicle cannot be returned from this endpoint regardless of the input MSRP parameter, unless the (no baseMsrp) vehicle in question is the default vehicle and the input MSRP is below the lowest (non-null) baseMsrp for the vehicles available for the VIN in question.
2. In rarer cases multiple vehicles available for a given VIN may have a common baseMsrp value. In this scenario, the vehicle with the lower UcgVehicleId will be returned when either of these vehicles would be selected based on the input MSRP parameter.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
msrpVehicleAndValuesByVin?period=2018-01-01&vin=
1FMCU9J93JUB98016&region=1&msrp=26701&mileage=7139' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

(see defaultVehicleAndValuesByVin method above for data return structure)

6.2.6.batchDefaultVehicleAndValuesByVin

This endpoint accepts an array of up to 700 input rows and performs a single batch valuation call against all the inputs at once, using the current data only. This endpoint is significantly slower than the other endpoints (response times between 30 and 50 seconds) but is more timeefficient than serializing several hundred single-VIN calls when sending more than 250 VINs at a time (roughly 3x faster when sending a full 700 VIN payload.)

Developer note: As of August 2021, this endpoint will return an error if the input parameter is empty.

Developer note: This endpoint uses the POST method, not GET. Calling this endpoint via GET will return a warning message reminding you to use POST instead.

Developer note: There is a single (required) parameter – input – which contains a JSON-formatted array of Vin/state/mileage values. In theory you can include this parameter in the querystring for the endpoint as normal. However, very large querystring values can be truncated at multiple points along the internet connection, from the user's system, proxy or firewall all the way through multiple routers or the destination hosts. As a result, we recommend including the input parameter in the request Body using x-www-form-urlencoded formatting.

Developer note: The various batch endpoints are provided to streamline the use of our REST API for larger volumes – sending 1 call with 500-1,000 inputs rather than having to serialize 500 to 1,000 individual calls. However, as these endpoints require longer query processing time than the single calls, we request that users taking advantage of our batch endpoints do not submit more than 1 – 2 batch requests in parallel.

Developer note: In addition to limiting the number of simultaneous batch requests, ***please do not use the batch endpoint with a single input row, or even fewer than 10-25 rows.*** For the user, these endpoints are optimized for 100+ rows and will take nearly as long for 1 or 10 or 25 rows as it will for 100 rows, and significantly longer time than calling the single-input endpoint sequentially (not even considering parallel calls to the single-input endpoint.) From our side, even running just a single VIN through the batch endpoint uses more than twice the CPU cost compared to the single-input endpoint.

License decrement: This method **does** decrement the user's license **for every valued input row**. For instance, if you send 700 VINs, 685 decode, but only 637 have basecleantrade > $0, this will decrement your remaining lookups by 637.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/batchDefaultVehicleAndValuesByVin

```
curl -X POST
"https://cloud.jdpower.ai/data-api/UAT/valuationservices/valuation/
batchDefaultVehicleAndValuesByVin"
-H "accept: application/json"
-H "api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63"
-H "Content-Type: application/x-www-form-urlencoded"
--data-urlencode 'input=[{"vin": "5XYPG4A33GG179357","state": "TX","mileage":
37828},{"vin": "1C4SDJCT3JC474730","state": "CA","mileage": 12473}]' \
--data-urlencode 'userinfo=BatchTest'
```

This represents the URL-encoded version of the below 2-element input array:

```
[
{"vin": "5XYPG4A33GG179357","state": "TX","mileage": 37828}
,{"vin": "1C4SDJCT3JC474730","state": "CA","mileage": 12473}
]
```


```
{
    "requestId": "yqdzva6mnbplhxqkckhll",
    "result": {
        "5XYPG4A33GG179357": {
            "vin": "5XYPG4A33GG179357",
            "state": "TX",
            "mileage": "37828",
            "ucgvehicleid": "201632257",
            "modelyear": "2016",
            "make": "Kia",
            "model": "Sorento",
            "body": "Utility 4D LX 2.4L I4",
            "basemsrp": "26200",
            "basecleantrade": "13850",
            "baseaveragetrade": "12825",
            "baseroughtrade": "11575",
            "basecleanretail": "15700",
            "baseloan": "12475",
            "averagemileage": "52500",
            "mileageadjustment": "1025",
            "vinoptionstrade": "0",
            "vinoptionsretail": "0",
            "vinoptionsloan": "0",
            "adjustedcleantrade": "14875",
            "adjustedaveragetrade": "13850",
            "adjustedroughtrade": "12600",
            "adjustedcleanretail": "16725",
            "adjustedloan": "13500",
            "maxmileageadj": "6925",
            "minmileageadj": "-5540",
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
            "minadjroughtrade": "120",
            "minadjloan": "0",
            "minadjretailforloan": "900",
            "multibody": " "
```

```
        },
        "1C4SDJCT3JC474730": {
            "vin": "1C4SDJCT3JC474730",
            "state": "CA",
            "mileage": "12473",
            "ucgvehicleid": "201832984",
            "modelyear": "2018",
            "make": "Dodge",
            "model": "Durango",
            "body": "Utility 4D R/T AWD 5.7L V8",
            "basemsrp": "46295",
            "basecleantrade": "34650",
            "baseaveragetrade": "33375",
            "baseroughtrade": "31800",
            "basecleanretail": "36925",
            "baseloan": "31200",
            "averagemileage": "27500",
            "mileageadjustment": "1275",
            "vinoptionstrade": "0",
            "vinoptionsretail": "0",
            "vinoptionsloan": "0",
            "adjustedcleantrade": "35925",
            "adjustedaveragetrade": "34650",
            "adjustedroughtrade": "33075",
            "adjustedcleanretail": "38200",
            "adjustedloan": "32475",
            "maxmileageadj": "17325",
            "minmileageadj": "-13860",
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
            "minadjroughtrade": "120",
            "minadjloan": "0",
            "minadjretailforloan": "900",
            "multibody": " "
        }
    },
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


Note that the results fields are a subset of the fields returned by the individual defaultVehicleAndValuesByVin endpoint; see that section above for field descriptions as needed.

6.2.7.vehiclesWithTrimByVin

New endpoint added in March 2022, this duplicates the existing vehiclesByVin endpoint functionality as a means of identifying the possible VIN decode results without valuation data but adds the vehicle(s)' Trim descriptions as unique fields (Trim and Trim2). As with the existing endpoint, the first vehicle returned is our Default decode if more than one row is returned.

License decrement: This method does not decrement the user's license under typical usage cases. If we see a user using this endpoint at significant volume without accompanying valuation calls, we may at our discretion require a separate license specifically for the non-valuation-related VIN decode API calls.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/vehiclesWithTrimByVin

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be decoded
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• vpp	int	hidden	0 = Do not apply Trim Reduction. default behavior is Use Trim Reduction
• vinsource	int	hidden	1 = Add vinsource field to response outside the result array. default behavior is "do not include"
• oem	int	hidden	1 = Do not use Verified Engineered data. default behavior is Use Engineered

The new hidden parameters as well as Trim Reduction functionality, are available initially on the UAT API environment (3/28/2024.) They should be available on the Production API in mid-April.

Legacy conversion: This replaces and enhances the getVehicles() method in the legacy/SOAP API.

Developer note: Utilizing the optional aspect of the vehicletype input field can be particularly useful here for developers who need to limit access to either Used Car and Commercial Truck datasets, or do not want to use Motorcycle data (NoMotorcycles vehicletype).
Developer note: Only Cars and Light Trucks will return usable Trim/Trim2 fields. Others will return null or N/A values for these fields.

Example:

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
vehiclesWithTrimByVin?period=0&vin= 19XFG3B1XEE012345' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
        {
```

```
                "modelyear": " 2014",
                "make": "Honda",
                "model": "Civic",
                "body": "Coupe 2D EX-L Navigation 1.8L I4 Auto",
                "trim": "EX-L",
                "trim2": "Navigation",
                "ucgvehicleid": "201427990",
                "vid": "79542"
            },
            {
                "modelyear": "2014",
                "make": "Honda",
                "model": "Civic",
                "body": "Coupe 2D EX-L 1.8L I4 Auto",
                "trim": "EX-L",
                "trim2": "N/A",
                "ucgvehicleid": "201427988",
                "vid": "79541"
            }
        ]
        , "userinfo": "Default"
        , "authId": "UCG"
    }
```


Please see vehiclesByVin definition for example with hidden parameter usage.

6.3. Vehicle Valuation Methods

These endpoints return editorialized Used valuation (and sometimes vehicle) data; these values are typically updated on a monthly basis. You can also retrieve our Weekly Used Values, which are similar to the "regular" or monthly Used values only updated more frequently, and/or Weekly Auction Values, which are more representative of wholesale/auction activity, from endpoints covered in those sections.

6.3.1.valueByVehicleId

Request the vehicle values for the input UcgVehicleId. It would be used either after the vehiclesByVin decode method above, after selecting the vehicle by description (years/makes/models/bodies calls above), or when the vehicle has already been identified in a prior period or other application.

License decrement: This method **does** decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/valueByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• ucgvehicleid	int	required	Vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This roughly corresponds to the getBaseVehicleValueByUid() and getVehicleValueByUid methods in the legacy/SOAP API.

Developer note: The vehicletype field is provided here for legacy support purposes only, as the vehicle ID will not exist in multiple vehicle type data sets.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
valueByVehicleId?period=2018-01-01&ucgvehicleid=
201834687&region=1&mileage=7139' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

```
{
  "requestId": "abc123defg56789",
  "result": [
    {
      "basecleantrade": "23525",
      "baseaveragetrade": "22375",
      "baseroughtrade": "20950",
      "basecleanretail": "26225",
      "baseloan": "21175",
     "averagemileage": "17500",
      "mileageadjustment": "725",
      "adjustedcleantrade": "24250",
      "adjustedaveragetrade": "22300",
      "adjustedroughtrade": "21675",
      "adjustedcleanretail": "26950",
      "adjustedloan": "21175",
      "maxmileageadj": "21175",
      "minmileageadj": "21175",
      "minadjretail": "21175",
      "minadjcleantrade": "21175",
      "minadjaveragetrade": "21175",
      "minadjroughtrade": "21175",
      "minadjloan": "0",
      "vid": "93603"
    }
   ]
   , "userinfo": "Default"
   , "authId": "UCG"
 }
```


(see defaultVehicleAndValuesByVin method above for business rules and notes on the final 7 fields in the return structure)

6.3.2.vehicleAndValueByVehicleId

Request the vehicle description and values for the input UcgVehicleId. It would be used either after the vehiclesByVin decode method above, or when the vehicle has already been identified in a prior period or other application. It is functionally identical to the valueByVehicleId method above with the addition of vehicle description and other vehicle attributes in the return data.

License decrement: This method **does** decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/vehicleAndValueByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• ucgvehicleid	int	required	Vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This roughly corresponds to the getVehicleAndValueByUid method in the legacy/SOAP API.

Developer note: The vehicletype field is provided here for legacy support purposes only, as the vehicle ID will not exist in multiple vehicle type data sets.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
vehicleAndValueByVehicleId?period=2018-01-01&ucgvehicleid=
201834687&region=1&mileage=7139' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
 {
     "requestId": "8a14jg1f4vr094eh9stz0rr",
     "result": [
         {
             "modelyear": "2018",
             "make": "Ford",
             "model": "Escape",
             "body": "Utility 4D Titanium EcoBoost 4WD 2.0L I4 Turbo",
             “ucgvehicleid”: "201834687",
             "vehicletype": "Light Truck",
             "mileageclass": "2",
             "basemsrp": "33395",
             "bodytype": "Utility",
             "doors": "4",
             "trim": "Titanium",
             "trim2": "EcoBoost",
             "drivetype": "4WD",
             "liters": "2.0",
             "engineconfiguration": "I",
             "cylinders": "4",
             "inductiontype": "Turbo",
             "transmission": "N/A",
             "fueltype": "Gas",
             "wheels": "4",
             "curbweight": "3645",
             "gvw": null,
             "gcw": null,
             "ucgsubsegment": "Compact Utility",
             "modelnumber": "U9J",
             "rollupvehicleid": "201834687",
             "basecleantrade": null,
             "baseaveragetrade": null,
             "baseroughtrade": null,
             "basecleanretail": null,
             "baseloan": null,
             "averagemileage": "12500",
             "mileageadjustment": null,
             "adjustedcleantrade": null,
             "adjustedaveragetrade": null,
             "adjustedroughtrade": null,
             "adjustedcleanretail": null,
             "adjustedloan": null,
             "maxmileageadj": null,
             "minmileageadj": null,
```

```
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
            "minadjroughtrade": "120",
            "minadjloan": "0",
            "minadjretailforloan": "900"",
            "vid": "93603"
        }
    ]
  , "userinfo": "Default"
  , "authId": "UCG"
}
```


(see defaultVehicleAndValuesByVin method above for business rules and notes on the final 7 fields in the return structure)

6.3.3.historicalValuesByVehicleId

This endpoint allows you to retrieve historical values for a single vehicle across multiple consecutive valuation periods. It would be used either after the vehiclesByVin decode method above, or when the vehicle has already been identified in a prior period or other application. It is like the valueByVehicleId method above, but does not include a mileage input parameter or adjusted values in the output, and returns a single row of values per month to a maximum number of months as defined by the input maxage parameter.

Only months with non-$0 Used values will be returned, and no more than 84 months' data will be returned regardless of the input maxage parameter value.

License decrement: This method decrements the user's license **by one per each month's values returned**.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/historicalValuesByVehicleId

Parameters:

• period	string	required	Starting valuation/definition date
• ucgvehicleid	int	required	vehicle ID to be valued
• region	int	required	Region ID for valuation (1-10)
• maxage	int	required	Maximum age in months to be returned
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

```
curl -L -X GET 'https://cloud.jdpower.ai/data-api/UAT/valuationservices/
valuation/historicalValuesByVehicleId?
period=0&ucgvehicleid=201834687&region=1&maxage=13&userinfo=UCG testing' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

This API call is requesting 13 months' base monthly Used values starting with the current date (`period=0`) and including no more than 13 months' data (`maxage=13`.)

```
{
    "requestId": "sca88cxdxbe2oah62pslh7",
    "result": [
        {
            "beginperiod": "2020-10-01",
            "endperiod": "2020-10-31",
            "basecleanretail": "22200",
            "basecleantrade": "20225",
            "baseaveragetrade": "19100",
            "baseroughtrade": "17725",
            "baseloan": "18225"
        },
        {
            "beginperiod": "2020-11-01",
            "endperiod": "2020-11-30",
            "basecleanretail": "21325",
            "basecleantrade": "19375",
            "baseaveragetrade": "18275",
            "baseroughtrade": "16900",
            "baseloan": "17450"
        },
         …etc
        }
    ],
    "userinfo": "UCG testing",
    "authId": "UCG UAT"
}
```

6.4. Weekly Used Values Methods

J.D. Power Valuation Services has added weekly updates. These weekly updates are based on analysis of wholesale and retail market price performance during the prior week, expected price movements, and economic conditions. Incorporating the most recent available data, J.D. Power's goal is to provide clients with up-to-date valuations that will help them make informed business decisions.

These endpoints and values are available starting mid-April 2020 and correspond to existing endpoints for the monthly Used values.

** Please note: any endpoint that doesn't include "weekly" in the endpoint name will return the "regular" monthly data.

At this time, vehicles without updated Weekly Values (specifically vehicles in the Older Used Car Guide and Commercial Truck Guide) will return empty/null Weekly Values fields; use the existing (monthly) endpoints for these vehicles.

6.4.1. License Decrement/Charge notes:

At the time of release (April 23, 2020), these endpoints do not impact the license usage tracking. **However, these endpoints will become charged (license-decrementing) endpoints starting in September 2020.**

For the purposes of determining a license charge, these endpoints will be tracked as noncharged duplicates if they are called immediately prior to or immediately after the corresponding Monthly Used endpoint with identical input parameters.

In other words, if you look up the Monthly Used values and the Weekly Used Values for the same inputs within 2 minutes of each other, only the Monthly endpoint will be charged against your license.

This only impacts users on a per-valuation or valuation block pricing model.

6.4.2.weeklyDefaultVehicleAndUsedValuesByVin

This is the “Weekly Used Values” endpoint corresponding to the “monthly” defaultVehicleAndValuesByVin endpoint and functions identically other than using the Weekly Used Values.

License decrement: This method **does** decrement the user’s license unless immediately before or after an otherwise-identical call to the monthly endpoint.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/weeklyDefaultVehicleAndUsedValuesByVin

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be decoded
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• vpp	int	hidden	1 = Apply VP+ Content Adjustment. 0 or omit (default) = legacy behavior
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

```
curl -L -X GET 'https://cloud.jdpower.ai/data-api/UAT/valuationservices/
weeklyDefaultVehicleAndUsedValuesByVin?
period=2020-04-13&vehicletype=UsedCar&vin=1FMCU9J93JUB98016&region=1&mileage
=12345' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

The fields in this response correspond to those in the defaultVehicleAndValuesByVin (monthly) endpoint, with the exception that the base and adjusted values (highlighted below) include “weekly” in the field names and represent the weekly updates to the Used values. The ‘business value’ fields (maxmileageadj through minadjretailforloan) apply equally to these Weekly Used Values.

```
{
    "requestId": "go4vyyw4pnn3ubb2o1szih",
    "result": [
        {
            "modelyear": "2018",
            "make": "Ford",
            "model": "Escape",
            "body": "Utility 4D Titanium EcoBoost 4WD 2.0L I4 Turbo",
            "ucgvehicleid": "201834687",
            "vehicletype": "Light Truck",
            "mileageclass": "2",
            "basemsrp": "33395",
            "bodytype": "Utility",
            "doors": "4",
            "trim": "Titanium",
            "trim2": "EcoBoost",
            "drivetype": "4WD",
            "liters": "2.0",
            "engineconfiguration": "I",
            "cylinders": "4",
            "inductiontype": "Turbo",
            "transmission": "N/A",
            "fueltype": "Gas",
            "wheels": "4",
            "curbweight": "3645",
            "gvw": null,
            "gcw": null,
            "ucgsubsegment": "Compact SUV",
            "modelnumber": "U9J",
            "rollupvehicleid": "201834687",
            "baseweeklycleantrade": "19675",
            "baseweeklyaveragetrade": "18575",
            "baseweeklyroughtrade": "17200",
            "baseweeklycleanretail": "21825",
            "baseweeklyloan": "17725",
            "averagemileage": "37500",
            "mileageadjustment": "1750",
            "vinoptionstrade": "0",
            "vinoptionsretail": "0",
            "vinoptionsloan": "0",
            "adjustedweeklycleantrade": "21425",
            "adjustedweeklyaveragetrade": "20325",
            "adjustedweeklyroughtrade": "18950",
            "adjustedweeklycleanretail": "23575",
            "adjustedweeklyloan": "19475",
            "maxmileageadj": "9838",
            "minmileageadj": "-7870",
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
            "minadjroughtrade": "120",
            "minadjloan": "0",
            "minadjretailforloan": "900",
            "vid": "93603"
        }
    ],
    "userinfo": "Default",
    "authId": "TestAdmin"
```

```
}
```

6.4.3.weeklyVehicleAndValueByVehicleId

This is the “Weekly Used Values” endpoint corresponding to the “monthly” vehicleAndValueByVehicleId endpoint and functions identically other than using the Weekly Used Values.

License decrement: This method **does** decrement the user’s license unless immediately before or after an otherwise-identical call to the monthly endpoint.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/weeklyVehicleAndValueByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• ucgvehicleid	int	required	vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

```
curl -L -X GET 'https://cloud.jdpower.ai/data-api/UAT/valuationservices/valuation/weeklyVehicleAndValueByVehicleId?period=2020-04-13&vehicletype=UsedCar&ucgvehicleid=201730198&region=1&mileage=12345' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

The fields in this response correspond to those in the vehicleAndValueByVehicleId with the exception that the base and adjusted values (highlighted below) include “weekly” in the field names and represent the weekly updates to the Used values. The ‘business value’ fields (maxmileageadj through minadjretailforloan) apply equally to these Weekly Used Values.


```
{
    "requestId": "btoaguphhd0iwgsxv6o02",
    "result": [
        {
            "modelyear": "2017",
            "make": "Ford",
            "model": "Taurus",
            "body": "Sedan 4D SE 3.5L V6",
            "ucgvehicleid": "201730198",
            "vehicletype": "Car",
            "mileageclass": "2",
            "basemsrp": "27345",
            "bodytype": "Sedan",
            "doors": "4",
            "trim": "SE",
            "trim2": "N/A",
            "drivetype": "FWD",
            "liters": "3.5",
            "engineconfiguration": "V",
            "cylinders": "6",
```

```
            "inductiontype": "Standard",
            "transmission": "N/A",
            "fueltype": "Gas",
            "wheels": "4",
            "curbweight": "3917",
            "gvw": null,
            "gcw": null,
            "ucgsubsegment": "Large Car",
            "modelnumber": "2D8",
            "rollupvehicleid": "201730198",
            "baseweeklycleantrade": "11975",
            "baseweeklyaveragetrade": "11000",
            "baseweeklyroughtrade": "9800",
            "baseweeklycleanretail": "13950",
            "baseweeklyloan": "10800",
            "averagemileage": "47500",
            "mileageadjustment": "2650",
            "adjustedweeklycleantrade": "14625",
            "adjustedweeklyaveragetrade": "13650",
            "adjustedweeklyroughtrade": "12450",
            "adjustedweeklycleanretail": "16600",
            "adjustedweeklyloan": "13450",
            "maxmileageadj": "5988",
            "minmileageadj": "-4790",
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
            "minadjroughtrade": "120",
            "minadjloan": "0",
            "minadjretailforloan": "900"
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```

6.4.4.weeklyUsedValuesByVehicleId

This is the “Weekly Used Values” endpoint corresponding to the “monthly” valueByVehicleId endpoint and functions identically other than using the Weekly Used Values.

License decrement: This method **does** decrement the user’s license unless immediately before or after an otherwise-identical call to the monthly endpoint.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/weeklyVehicleAndValueByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• ucgvehicleid	int	required	vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

```
curl -L -X GET ' https://cloud.jdpower.ai/data-api/UAT/valuationservices/
valuation/weeklyUsedValuesByVehicleId?
period=0&vehicletype=UsedCar&ucgvehicleid=201730198&region=1&mileage=12345'
\
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

The fields in this response correspond to those in the valueByVehicleId with the exception that the base and adjusted values (highlighted below) include “weekly” in the field names and represent the weekly updates to the Used values. The ‘business value’ fields (maxmileageadj through minadjretailforloan) apply equally to these Weekly Used Values.


```
{
    "requestId": "xfoaj74qcevfe97oqmpk",
    "result": [
        {
            "baseweeklycleantrade": "11975",
            "baseweeklyaveragetrade": "11000",
            "baseweeklyroughtrade": "9800",
            "baseweeklycleanretail": "13950",
            "baseweeklyloan": "10800",
            "averagemileage": "47500",
            "mileageadjustment": "2650",
            "adjustedweeklycleantrade": "14625",
            "adjustedweeklyaveragetrade": "13650",
            "adjustedweeklyroughtrade": "12450",
            "adjustedweeklycleanretail": "16600",
            "adjustedweeklyloan": "13450",
            "maxmileageadj": "5988",
            "minmileageadj": "-4790",
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
```

```
            "minadjroughtrade": "120",
            "minadjloan": "0",
            "minadjretailforloan": "900"
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


6.4.5.weeklyBuildVehicleAndValuesByVin

This is the “Weekly Used Values” endpoint corresponding to the “monthly” buildVehicleAndValuesByVin endpoint and functions identically other than using the Weekly Used Values.

This is a Premium endpoint.

Please see the endpoint definition in the Build endpoint group below.

6.4.6.(deprecated) weeklyUsedValuesByLegacyId

Deprecated October 2022.

For details on this endpoint, see its entry in the Deprecated Endpoint document section.

6.5. Combined Monthly and Weekly Used Values methods

These endpoints, released for testing in January 2022, provide both the "standard" monthly and new (as of 2020) weekly Used values in a single API call. Not all vehicles will return Weekly Used values, and all historical valuation calls with period parameters before the release of the Weekly Used Values (April 13, 2020) will also return 0 for the weekly used values.

Developer note: Some adjustment field names are shared between the monthly and weekly endpoints. In these cases, the fields containing the Weekly values will include "weekly" in the field name; see usedValuesByVehicleId example return below.

These adjustments will usually be the same but may vary for dates where a given week crosses a month boundary (i.e., Monday is in January, but Friday is in February.) In those scenarios, the weekly mileage and option adjustments are based on the month that the week's Wednesday falls into.

Developer note: We have not included the vehicletype parameter filter in the two combined Used Values by ID endpoints; it is assumed the developer will have applied any needed vehicletype filter previously, when determining the vehicle's ID initially. The combined Used Values by VIN endpoint still contains the vehicletype parameter since it will often be the initial lookup made for any given VIN.

6.5.1.usedValuesByVehicleId

This endpoint acts as a combination of the valueByVehicleId and weeklyUsedValuesByVehicleId endpoints. As with those endpoints, it is assumed the user has determined the correct vehicle ID previously.

License decrement: This method does decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/usedValuesByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• ucgvehicleid	int	required	Vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This is a new method; there is no corresponding legacy/SOAP method.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
usedValuesByVehicleId?period=0&ucgvehicleid= 201834687&region=4&mileage=27435'
\
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

```
{
    "requestId": "lre0styacg9wo5gcodjg",
    "result": [
        {
            "basecleantrade": "25925",
            "baseaveragetrade": "24750",
            "baseroughtrade": "23275",
            "basecleanretail": "28425",
            "baseloan": "23350",
            "baseweeklycleantrade": "24750",
            "baseweeklyaveragetrade": "23575",
            "baseweeklyroughtrade": "22125",
            "baseweeklycleanretail": "27200",
            "baseweeklyloan": "22275",
            "averagemileage": "57500",
            "averageweeklymileage": "57500",
            "mileageadjustment": "2000",
            "weeklymileageadjustment": "2000",
            "adjustedcleantrade": "27925",
            "adjustedaveragetrade": "26750",
            "adjustedroughtrade": "25275",
            "adjustedcleanretail": "30425",
            "adjustedloan": "25350",
            "adjustedweeklycleantrade": "26750",
            "adjustedweeklyaveragetrade": "25575",
            "adjustedweeklyroughtrade": "24125",
            "adjustedweeklycleanretail": "29200",
            "adjustedweeklyloan": "24275",
            "maxmileageadj": "12963",
            "minmileageadj": "-10370",
            "maxweeklymileageadj": "12375",
            "minweeklymileageadj": "-9900",
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
            "minadjroughtrade": "120",
            "minadjloan": "0",
            "minadjretailforloan": "900",
            "vid": "93603"
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```

6.5.2.usedValuesAndVehicleByVehicleId

This endpoint acts as a combination of the vehicleAndValueByVehicleId and weeklyVehicleAndValueByVehicleId endpoints, adding the vehicle description and attribute fields along with the values. As with those endpoints, it is assumed the user has determined the correct vehicle ID previously.

License decrement: This method does decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/usedValuesAndVehicleByVehicleId

Parameters:

- period string required Valuation/definition date
- ucgvehicleid int required Vehicle ID to be valued
- region int required Region ID (see Definitions and Regions)
- mileage int optional Used to determine mileage adjustment. No mileage adjustment returned if empty
- userinfo string optional User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This is a new method; there is no corresponding legacy/SOAP method.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
usedValuesAndVehicleByVehicleId?period=0&ucgvehicleid=
201834687&region=4&mileage=27435' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
    "requestId": "lre0styacg9wo5gcodjg",
    "result": [
        {
            "modelyear": "2018",
            "make": "Ford",
            "model": "Escape",
            "body": "Utility 4D Titanium EcoBoost 4WD 2.0L I4 Turbo",
            "ucgvehicleid": "201834687",
            "vehicletype": "Light Truck",
            "mileageclass": "2",
            "basemsrp": "33395",
            "bodytype": "Utility",
            "doors": "4",
            "trim": "Titanium",
            "trim2": "EcoBoost",
            "drivetype": "4WD",
            "liters": "2.0",
            "engineconfiguration": "I",
            "cylinders": "4",
            "inductiontype": "Turbo",
            "transmission": "N/A",
            "fueltype": "Gas",
            "wheels": "4",
```

```
            "curbweight": "3645",
            "gvw": null,
            "gcw": null,
            "ucgsubsegment": "Compact SUV",
            "modelnumber": "U9J",
            "rollupvehicleid": "201834687"
            , "basecleantrade": "25925",
            "baseaveragetrade": "24750",
            "baseroughtrade": "23275",
            "basecleanretail": "28425",
            "baseloan": "23350",
            "baseweeklycleantrade": "24750",
            "baseweeklyaveragetrade": "23575",
            "baseweeklyroughtrade": "22125",
            "baseweeklycleanretail": "27200",
            "baseweeklyloan": "22275",
            "averagemileage": "57500",
            "averageweeklymileage": "57500",
            "mileageadjustment": "2000",
            "weeklymileageadjustment": "2000",
            "adjustedcleantrade": "27925",
            "adjustedaveragetrade": "26750",
            "adjustedroughtrade": "25275",
            "adjustedcleanretail": "30425",
            "adjustedloan": "25350",
            "adjustedweeklycleantrade": "26750",
            "adjustedweeklyaveragetrade": "25575",
            "adjustedweeklyroughtrade": "24125",
            "adjustedweeklycleanretail": "29200",
            "adjustedweeklyloan": "24275",
            "maxmileageadj": "12963",
            "minmileageadj": "-10370",
            "maxweeklymileageadj": "12375",
            "minweeklymileageadj": "-9900",
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
            "minadjroughtrade": "120",
            "minadjloan": "0",
            "minadjretailforloan": "900",
            "vid": "93603"
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```

6.5.3.defaultVehicleAndUsedValuesByVin

This endpoint acts as a combination of the vehicleAndValueByVehicleId and weeklyVehicleAndValueByVehicleId endpoints, adding the vehicle description and attribute fields along with the values. As with those endpoints, it is assumed the user has determined the correct vehicle ID previously.

Unlike the combined Used values by ID endpoints, this endpoint does provide the optional vehicletype filter parameter.

License decrement: This method does decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/defaultVehicleAndUsedValuesByVin

Parameters:

• period	string	required	Valuation/definition date
• vin	string	required	Vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• vehicletype	string	optional	Filter to specific vehicle types
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• vpp	int	hidden	1 = Apply VP+ Content Adjustment. 0 or omit (default) = legacy behavior
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

Legacy conversion: This is a new method; there is no corresponding legacy/SOAP method.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
defaultVehicleAndUsedValuesByVin?
period=0&vin=1GT42TCB9KZ063980&vehicletype=UsedCar&region=8&mileage=41094' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
    "requestId": "vc9a93kvbjifbufjg0zylr",
    "result": [
        {
            "modelyear": "2019",
            "make": "GMC Light Duty",
            "model": "Sierra 3500",
            "body": "Crew Cab 4WD DRW 6.0L V8",
            "ucgvehicleid": "201917849",
            "vehicletype": "Light Truck",
            "mileageclass": "3",
            "basemsrp": "43600",
            "bodytype": "Crew Cab",
            "doors": "N/A",
            "trim": "Base",
            "trim2": "N/A",
            "drivetype": "4WD",
            "liters": "6.0",
            "engineconfiguration": "V",
            "cylinders": "8",
            "inductiontype": "Standard",
            "transmission": "N/A",
            "fueltype": "Gas",
            "wheels": "6",
            "curbweight": "6717",
            "gvw": "13025",
            "gcw": null,
            "ucgsubsegment": "Large Pickup - Heavy Duty",
            "modelnumber": "(1/4)2T",
```

```
"rollupvehicleid": "201917849",
"basecleantrade": "46350",
"baseaveragetrade": "44750",
"baseroughtrade": "42800",
"basecleanretail": "50700",
"baseloan": "41725",
"baseweeklycleantrade": "41600",
"baseweeklyaveragetrade": "40025",
"baseweeklyroughtrade": "38125",
"baseweeklycleanretail": "45850",
"baseweeklyloan": "37450",
"averagemileage": "42500",
"averageweeklymileage": "42500",
"mileageadjustment": "0",
"weeklymileageadjustment": "0",
"vinoptionstrade": "0",
"vinoptionsretail": "0",
"vinoptionsloan": "0",
"weeklyvinoptionstrade": "0",
"weeklyvinoptionsretail": "0",
"weeklyvinoptionsloan": "0",
"adjustedcleantrade": "46350",
"adjustedaveragetrade": "44750",
"adjustedroughtrade": "42800",
"adjustedcleanretail": "50700",
"adjustedloan": "41725",
"adjustedweeklycleantrade": "41600",
"adjustedweeklyaveragetrade": "40025",
"adjustedweeklyroughtrade": "38125",
"adjustedweeklycleanretail": "45850",
"adjustedweeklyloan": "37450",
"maxmileageadj": "23175",
"minmileageadj": "-18540",
```

```
            "maxweeklymileageadj": "20800",
            "minweeklymileageadj": "-16640",
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
            "minadjroughtrade": "120",
            "minadjloan": "0",
            "minadjretailforloan": "900",
            "vid": "96629"
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


In addition to the monthly/weekly adjustments noted in the “by ID” endpoints, this endpoint includes both weekly and monthly VIN Option adjustment totals (highlighted above.)

6.5.4.buildVehicleAndUsedValuesByVin

This endpoint acts as a combination of the buildVehicleAndValuesByVin and weeklyBuildVehicleAndValuesByVin endpoints.

For full details, see the endpoint definition in the Build (VIN Precision+) section.

6.6. Vehicle Information methods

6.6.1.vehicleInformationByVehicleId

Request the vehicle attributes for the input UcgVehicleId. It would be used either after the vehiclesByVin decode method above, after selecting the vehicle by description (years/makes/models/bodies calls above), or when the vehicle has already been identified in a prior period or other application, when the additional data elements/attributes are needed.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/vehicleInformationByVehicleId

Parameters:

- period string required Valuation/definition date
- vehicletype string optional Filter to specific vehicle types
- ucgvehicleid int required Vehicle ID to be valued
- userinfo string optional User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This is a new method; there is no corresponding legacy/SOAP method.

Developer note: The vehicletype field is provided here for legacy support purposes only, as the vehicle ID will not exist in multiple vehicle type data sets.

Developer note: Not every vehicle will have every attribute defined in the results.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
vehicleInformationByVehicleId?period=2018-01-01&ucgvehicleid=201834687' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

```
{
  "requestId": "abc123defg56789",
  "result": [
    {
      "vehicletype": "Light Truck",
      "mileageclass": "2",
      "basemsrp": "33395",
      "bodytype": "Utility",
      "doors": "4",
      "trim": "Titanium",
      "trim2": "EcoBoost",
      "drivetype": "4WD",
      "liters": "2.0",
      "engineconfiguration": "I",
      "cylinders": "4",
      "inductiontype": "Turbo",
      "transmission": "",
      "fueltype": "Gas",
      "wheels": "4",
      "curbweight": "3645",
      "gvw": "",
      "gcw": "",
      "ucgsubsegment": "Compact Utility",
      "modelnumber": "U9J",
      "rollupvehicleid": "201834687"
    }
   ]
   , "userinfo": "Default"
   , "authId": "UCG"
 }
```

6.6.2.vehicleList

This endpoint returns a list of the vehicles for the input parameters, which apply some mandatory and some optional filter parameters. This can be used to retrieve and cache the full list of vehicles available in the API for customers who want to be able to populate vehicle selection drop-down controls but do not want to hit the API to populate the controls every time they are needed.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/vehicleList

Legacy conversion: This is a new method; there is no corresponding legacy/SOAP method. Also, see the Developer Note below regarding potential parameter filter issues.

Legacy conversion: An additional Legacy Vehicle List has been added (mid-July 2021) which includes the Legacy vehicle IDs and Descriptions. Please see its definition below.

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• modelyear	int	optional	Model year selected; leave off to receive all model years available
• make	string	optional	Make selected; leave off to receive all makes available for the selected Model Year
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

```
curl -X GET \ 'https://cloud.jdpower.ai/data-api/valuationservices/valuation/vehicleList?period=2018-01-01&modelyear= 2018&make=Acura' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
    "requestId": "3ralodu9gj195womhwcs5n",
    "result": [
        {
            "ucgvehicleid": "201838421",
            "year": "2018",
            "make": "Acura",
            "model": "TLX",
            "bodystyle": "Sedan 4D A-Spec 3.5L V6",
            "msrp": "42800",
            "vehicletype": "Car",
            "vid": "95239"
        },
        {
            "ucgvehicleid": "201836489",
            "year": "2018",
            "make": "Acura",
            "model": "TLX",
            "bodystyle": "Sedan 4D Technology AWD 3.5L V6",
```

```
            "msrp": "41900",
            "vehicletype": "Car",
            "vid": "95241"
        }, … (more rows)
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


Developer note: Since the response data for this endpoint can be very large (up to several MB for an un-filtered request of over 60k result rows) the Average Response Time for this endpoint can significantly exceed the normal endpoint response times.

Please include the HTTP Header for Accept-Encoding: gzip to enable compression of the results during network transfer.

Developer note: Like our other endpoints, this endpoint is case-sensitive on the input parameters. Additionally, the model year/make filters must align with data in our system. For instance, as of September 1, 2019, you will get an empty response if you set the modelyear parameter to 2021, or to 2000, and you will also get an empty response if you set the make to Ram (or Ford Truck, which exists as a Make in our legacy system but is replaced with simply Ford in the REST API) or to an exotic make not in our systems (i.e., Lamborghini, etc.)

6.6.3.statusByVehicleId

This endpoint can be used to help determine the status of a given vehicle ID in terms of first and last available periods, first valued, and optionally the status for a given period value.

For example, if your requirements typically retrieve the value of a given vehicle to be on January 1 of the current year but to retrieve the oldest available data if nothing is available on January 1, rather than “walking” up through each period until you find data, you can call this endpoint to pinpoint the oldest period you can use.

Alternatively, because the Used Car Guide data covers a rolling 20 model years, some vehicles are no longer available but may have been available previously. You can use this endpoint to determine the last available period for the vehicle in this case.

License decrement: This method does not decrement the user’s license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/statusByVehicleId

Parameters:

• ucgvehicleid	int	required	Vehicle ID to be queried
• targetperiod	string	optional	Specific period for status
• erv	int	hidden	1 = Treat Early Release Values as Valued
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

```
curl -X GET \ 'https://cloud.jdpower.ai/data-api/valuationservices/valuation/ statusByVehicleId?ucgvehicleid=201834687 ' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
    "requestId": "3ralodu9gj195womhwcs5n",
    "result": [
        {
            "firstperiod": "2017-08-01",
            "lastperiod": "2021-03-31",
            "firstmsrp": "2017-08-01",
            "firstvalued": "2018-07-01",
            "targetperiodstatus": "Undefined"
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```

Developer note: By default, this endpoint and the statusByVin endpoint look at the Base Clean Trade value when determining if an input (ID or VIN) is valued in any given month/period. Since February 2024, some newer vehicles fall into our Early Release Values process and have only Base Average Trade and Base Clean Retail values, but no Clean Trade. As a result, they are treated as "unvalued" while in the ERV process. The new (July 2025+) **`erv=1`** parameter changes this behavior so an ERV vehicle will be treated as Valued instead of Unvalued. The response may also have one or more notification messages in an added **`ervMessage`** field (**`response.ervMessage`**).

Response fields:

- firstperiod — First date where the input vehicle ID is available
- lastperiod — Last date which can be used to find data for the vehicle ID
    - Note: in most cases this will be the end of the currently newest data period. This does not imply that the vehicle will not be available in the next period, only that this is the latest date with available data.
- firstmsrp — First date where there is non-$0 BaseMsrp available
- firstvalued — First date where there is non-$0 Used values available
- targetperiodstatus — Status of the ID for the optional input targetperiod parameter. You will always receive Undefined if you do not provide the optional targetperiod parameter
    - Undefined — Input ID doesn't exist in target period
    - Defined — Input ID exists with $0 MSRP and Used values
    - MSRP — Input ID exists with non-$0 BaseMsrp
    - Valued — Input ID exists with non-$0 Used values
    - MSRP + Valued — Input ID exists with both BaseMsrp and Used values

6.6.4.statusByVin

This endpoint can be used to help determine the status of a given VIN in terms of first and last available periods that we decode the VIN, first valued, and optionally the status for a given period value. See the statusByUcgVehicleId endpoint above for additional usage scenarios.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/statusByVin

Parameters:

vin	string	required	VIN to be queried
targetperiod	string	optional	Specific period for status
erv	int	hidden	1 = Treat Early Release Values as Valued
userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

```
curl -X GET \ 'https://cloud.jdpower.ai/data-api/valuationservices/valuation/ statusByVin?vin=1FMCU9J93JUB98016&targetperiod=2020-01-01' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
    "requestId": "3ralodu9gj195womhwcs5n",
    "result": [
        {
            "firstperiod": "2017-08-01",
            "lastperiod": "2021-03-31",
            "firstmsrp": "2017-08-01",
            "firstvalued": "2018-07-01",
            "targetperiodstatus": "MSRP + Valued"
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


Response fields: see statusByUcgVehicleId endpoint above.

Developer note: By default, this endpoint and the statusByVehicleId endpoint look at the Base Clean Trade value when determining if an input (ID or VIN) is valued in any given month/period. Since February 2024, some newer vehicles fall into our Early Release Values process and have only Base Average Trade and Base Clean Retail values, but no Clean Trade. As a result, they are treated as "unvalued" while in the ERV process. The new (July 2025+) **`erv=1`** parameter changes this behavior so an ERV vehicle will be treated as Valued instead of Unvalued. The response may also have one or more notification messages in an added **`ervMessage`** field (**`response.ervMessage`**).

6.7. Vehicle Adjustment Methods

6.7.1.accessoryDataByVehicleId

Request the list of available accessories, including the option logic (Includes/Excludes) for the input UcgVehicleId. It would called be after selecting the vehicle by description (years/makes/models/bodies calls above), or when the vehicle has already been identified in a prior period or other application.

Developer note: This endpoint, added in early 2021, combines the data from the accessoriesByVehicleId and accessoryLogicByVehicleId endpoints (below) into a single call. Those endpoints should be considered deprecated at this time and existing application users should migrate to this new endpoint as soon as possible.

Developer note: This endpoint removes the optional vehicletype filter parameter for performance purposes. If needed, apply this filter in previous steps (when defining the vehicle being valued.)

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/accessoryDataByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• ucgvehicleid	int	required	Vehicle ID to be valued
• region	int	required	Region ID (see l or Regions method)
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces the getAccessories() method in the legacy/SOAP API, and also replaces both the accessoriesByVehicleId and accessoryLogicByVehicleId REST endpoints.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
accessoryDataByVehicleId?period=2018-01-01&ucgvehicleid= 201834687&region=6'
\
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
   {
         "accdesc": "Parking Aid",
         "acccode": "0D9",
         "tradein": null,
         "retail": null,
         "loan": null,
         "isincluded": 1,
         "isadded": 0,
```

```
            "accessorycategory": "Electronics",
            "includes": "",
            "excludes": ""
        },
… more rows
    }
```

Business Rules and notes on fields.

- ***acccode:*** Accessory ID (UCG VAC) used to identify each accessory when using the Mutual Exclude and Package Inclusive logic (see accessoryLogicByVehicleId method below.)
- ***tradein, retail, loan:*** Accessory adjustment values for each value type. TradeIn applies to all Trade value types (Clean, Average and Rough)
- ***isincluded***: If 1, this accessory's value is already included in the vehicle's Base values. This accessory should be displayed as selected to your user but can be (temporarily) deselected by Mutual Exclusive logic (see accessoryLogicByVehicleId method below) Included options always have $0 values.
- ***isadded:*** If 1, this accessory is pre-selected based on the inputs. Included in this return object for compatibility with the VIN accessory methods, below. This is primarily used in the accessoriesByVinAndVehicleId and buildAccessoriesByVin methods.
- ***includes:*** A comma-delimited array of accessories (by AccCode) which, when this accessory is selected, should be treated as Included (selected but $0 value impact.) See Package Include in the Definition section.
    - *In the example above, the Safe & Smart Pkg "Includes" AccCodes 0C7, 0C8, 0LE, & 1AP.*
- ***excludes:*** A comma-delimited array of accessories (by AccCode) which, when this accessory is selected, are Excluded. See Mutual Excludes in the Definition section.

Developer note: The Includes and Excludes fields will often be empty.

Developer note: In general, you should treat any accessories flagged as `isincluded` or `isadded` as if the user had selected them, i.e., check for any "child" accessories in the Includes and Excludes fields and process them as if the user had selected the `isincluded` accessory.

As noted above, any 'parent' `isincluded` accessories will not have any **`Excludes`** to check for.

6.7.2.accessoryDataByVinAndVehicleId

Request the list of available accessories for the input VIN and Vehicle ID. It would be used either after the one of the VIN decode methods above.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/accessoryDataByVinAndVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN for vehicle being valued
• ucgvehicleid	int	required	Vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions)
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• vpp	int	hidden	1 = Apply VP+ Content Adjustment. 0 or omit (default) = legacy behavior
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

Legacy conversion: This replaces the getAccessories() method in the legacy/SOAP API when supplying the (SOAP-system's) optional VIN input parameter and also replaces both the earlier accessoriesByVinAndVehicleId and accessoryLogicByVehicleId REST endpoints.

Developer note: In general, you should treat any accessories flagged as `isincluded` or `isadded` as if the user had selected them, i.e., check for any "child" accessories in the Includes and Excludes fields and process them as if the user had selected the `isincluded` accessory. As noted above, any 'parent' `isincluded` accessories will not have any **`Excludes`** to check for.

Developer note: Both VIN and Vehicle ID are required as the VIN does not always decode to a unique Vehicle ID, and the list of options, their values, and Included status may vary by vehicle.

Developer note: This endpoint, added in early 2021, combines the data from the accessoriesByVinAndVehicleId and accessoryLogicByVehicleId endpoints (below) into a single call. Those endpoints should be considered deprecated at this time and existing application users should migrate to this new endpoint as soon as possible.

Developer note: This endpoint removes the optional vehicletype filter parameter for performance purposes. If needed, apply this filter in previous steps (when defining the vehicle being valued.)

Developer notes continue below

Developer note – VIN Precision+ Content Adjustment: Because this endpoint includes both a VIN and Vehicle ID input parameter, there is additional logic to verify that the input ucgvehicleid parameter aligns with the Build Data result (if available.) If Build Data is available and the input ID does not align with the Build result, you will receive the Standard results for your input, not the VIN Precision+ results, and the response will include a vinerror field (outside the result[] array) with a warning message.

Always use the returned ucgvehicleid from the VIN Valuation response when using the optional vpp=1 VIN Precision+ Content Adjustment enhancement to avoid this issue.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
accessoryDataByVinAndVehicleId?period=2018-01-01&vin=
1FMCU9J93JUB98016&ucgvehicleid=201834687&region=1' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
    {
          "accdesc": "Parking Aid",
          "acccode": "0D9",
          "tradein": null,
          "retail": null,
          "loan": null,
          "isincluded": 1,
          "isadded": 0,
          "accessorycategory": "Electronics",
          "includes": "",
          "excludes": ""
     },
     …
     {
          "accdesc": "Safe & Smart Pkg.",
          "acccode": "0J0",
          "tradein": 800,
          "retail": 900,
          "loan": 800,
          "isincluded": 0,
          "isadded": 0,
          "accessorycategory": "Package",
          "includes": "0C7,0C8,0LE,1AP",
          "excludes": ""
 },
    , "userinfo": "Default"
    , "authId": "UCG"
}
```


See accessoryDataByVehicleId method above with additional notes as below.

isadded: This accessory is pre-selected based on the input VIN using UCG's standard VIN decode (VIN Pattern-based) or Build data (when calling buildAccessoriesByVehicleId)

See Developer Note above re: handing `isadded` accessories.

6.7.3.(Deprecated) accessoryLogicByVehicleId

Developer note: This endpoint should be considered deprecated for development purposes as of mid-February 2021 and may be removed from the API in Q3/Q4 of 2025; please start using one of the new consolidated accessory data endpoints above except when using the Build Accessory endpoint; a replacement consolidated Build Accessory Data endpoint is under development but is not ready for testing yet.

Request the list of available accessories for the input VIN and Vehicle ID. It would be used either after the vehiclesByVin decode method above, after selecting the vehicle by description (years/ makes/models/bodies calls above.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/accessoryLogicByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• ucgvehicleid	int	required	Vehicle ID to be valued
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces and consolidates the getExclusiveAccessories() and getInclusiveAccessories() methods in the legacy/SOAP API.

Developer note: The vehicletype field is provided here for legacy support purposes only, as the vehicle ID will not exist in multiple vehicle type data sets.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
accessoryLogicByVehicleId?period=2018-01-01& ucgvehicleid= 201834687' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

```
{
  "requestId": "abc123defg56789",
  "result": [
    {
          "acccode": "01A",
          "childacccode": "081",
          "logic": "E"
    },
    {
          "acccode": "081",
          "childacccode": "01A",
          "logic": "E"
    }
 ]
    , "userinfo": "Default"
    , "authId": "UCG"
 }
```

Business Rules and notes on fields.

This method returns an array of options in a 'parent'/'child' relationship. When an accessory that matches the 'parent' AccCode value is selected, the specified Logic (E for Exclude or I for Include) is applied to the 'child' ChildAccCode accessory.

Not all vehicles will have Accessory Logic available.

Accessories flagged as IsIncluded=True in the main accessory methods above will NOT have a corresponding E (Exclude) row for themselves, but they may appear as ChildAccCode with Exclude rows. In other words, Included options do not Exclude but may themselves be excluded.

- **Logic E – Mutual Exclude:** The ChildAccCode accessory is de-selected and disabled until the AccCode (parent) accessory becomes de-selected by the user.
- **Logic I – Package Include:** The ChildAccCode accessory is selected and not de-selectable by the user until the AccCode (parent) accessory becomes de-selected, but the child accessory value is treated as $0 (its value is already factored into the package/parent accessory record's values.)

6.7.4.(Deprecated) accessoriesByVehicleId

This endpoint has been deprecated as of March 2021. Please use the new accessoryDataByVehicleId endpoint. For endpoint definition, see the deprecated endpoint reference.

6.7.5.(Deprecated) accessoriesByVinAndVehicleId

This endpoint has been deprecated as of March 2021. Please use the new accessoryDataByVinAndVehicleId endpoint. For endpoint definition, see the deprecated endpoint reference.

6.7.6.mileageByVehicleId

Request the available mileage adjustments for the input Vehicle ID. It would be used either after the vehiclesByVin decode method above, after selecting the vehicle by description (years/ makes/models/bodies calls above, when client-side mileage adjustment needs to be performed, in lieu of the standard valuation calls (above) where a specific mileage adjustment value is returned when an input mileage value is provided.

License decrement: This method does not decrement the user's license based on the additional VIN Content data in use.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/mileageByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• ucgvehicleid	int	required	Vehicle ID to be valued
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces the corresponding legacy/SOAP getMileageAdj method for the most part. The results contain only the mileage_struc portion of the legacy API results; the remaining data elements from the getMileageAdj have been moved into the vehicle valuation result objects instead.

Developer note: The vehicletype field is provided here for legacy support purposes only, as the vehicle ID will not exist in multiple vehicle type data sets.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
mileageByVehicleId?period=2018-01-01& ucgvehicleid= 201834687' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
    {
          "lowlevel": "0",
          "highlevel": "5000",
          "adjamount": "1125",
          "ratepermile": "0"
    },
    {
          "lowlevel": "5001",
          "highlevel": "10000",
          "adjamount": "725",
          "ratepermile": "0"
    }
 ]
    , "userinfo": "Default"
 , "authId": "UCG"
 }
```


Business Rules and notes on fields.

The results array represents the table of possible mileage adjustments for the input vehicle ID. The mileage adjustment is determined by finding the appropriate row where LowLevel <= (input miles) <= HighLevel and adding the AdjAmount plus RatePerMile x (input miles). You must then apply the maxMileageAdj/minMileageAdjust caps from the vehicle valuation method results.

For cars and light-duty trucks, there will be many rows in 5,000-mile increments, and the ratepermile will be 0, so the mileage adjustment will always be the AdjAmount value.

For medium- and heavy-duty trucks, there will be three rows returned, representing low, typical, and high mileage bands respectively.

6.8. Weekly Auction Values methods

These methods will provide access to the Weekly Auction Values data. The vehicle must be defined through a previous method call.

Please note that as of late April 2019, these methods do include a license decrement (charge) for calling. Our legacy service required the vehicle be looked up in the Used data prior to looking up the Weekly values; this is no longer the case.

Developer note: Prior to April 4, 2022, the Weekly Auction Values system used a straight calculation approach to determine the impact of mileage on the Weekly Auction Values. Starting April 2022, the Weekly Auction Values will use the same Mileage approach as our Weekly Used Values, i.e., there is a published table of mileage ranges for each vehicle, in 5,000-mile blocks, each block has a 'base' mileage adjustment, and there are caps to the maximum allowable mileage adjustment. In the case of Weekly Auction Values, these mileage caps will be based on the (base) High Auction Value rather than the Base Clean Trade-In value.

The WAV endpoints have been updated with an optional "start early date" to allow a user to have access to the new mileage process either for testing, or to use in their applications; it must be noted that the J.D. Power Values Online web application will not provide this same option, so users who require their values to align with the J.D. Power website values should only use the "start early date" field during testing.

The new, optional `v2MileageStartDate` parameter is only effective on the UAT API currently. Expect it to be available on the Production API in late January 2022.

6.8.1.weeklyAuctionValuesByVehicleId

Request the vehicle's Weekly Auction Values (WAV) for the input UcgVehicleId. It would be used either after the vehiclesByVin decode method above, after selecting the vehicle by description (years/makes/models/bodies calls above), or when the vehicle has already been identified in a prior period or other application.

License decrement: This method **does** decrement the user's license. This differs from the SOAP system which did not decrement the license but required retrieving the monthly/Used values for the vehicle ID before you could return WAV data.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/weeklyAuctionValuesByVehicleId

Parameters:

- period string required Valuation/definition date
- vehicletype string optional Filter to specific vehicle types
- ucgvehicleid int required Vehicle ID to be valued
- region int required Region ID (see Definitions and Regions)
- mileage int optional Used to determine mileage adjustment. No mileage adjustment returned if empty
- accessorylist string optional Comma-delimited array of VAC codes for options to be valued with the vehicle. Not all selected options impact WAV results.
- v2MileageStartDate optional Optional ISO date to over-ride the system default (2022-04-04) for when the new WAV mileage process starts.
- userinfo string optional User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This corresponds to the getAuctionValues() method in the legacy/SOAP API.

Developer note: The vehicletype field is provided here for legacy support purposes only, as the input vehicle ID will not exist in multiple vehicle type data sets. As Commercial Truck Guide vehicles do not receive Weekly Auction Values currently, there is no point in requesting Weekly Auction Values for any vehicle in the CommercialTruck vehicletype data set.

Developer note: Only cars/light trucks 8 model years old or newer (often excluding the current calendar year) will return Weekly Auction Values.

Developer note: Only Trim, Drive, Engine and Transmission-type accessories will impact the Weekly Auction Values.

```
Curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
weeklyAuctionValuesByVehicleId?
period=0&ucgvehicleid=201423113&region=4&mileage=15924&accessorylist=0D0,1BJ
,0D3,0C8,294,1BR,1AU,137,059,1DK,1BZ,0LE,09J,346,139,081,145,0C9,025,1AT,0D9
' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'

{
  "requestId": "abc123defg56789",
  "result": [
    {
      "lowauction": "23525",
      "avgauction": "22375",
      "highauction": "20950",
      "daterange": "11/12/2018 – 11/18/2018",
      "auctionaccessories": "294,081",
      "acceptablemileage": "23025",
      "ratepermile": -0.105,
      "mileageadjust": "-995",
      "minauctionvalue": "225"

    }
   ]
   , "userinfo": "Default"
   , "authId": "UCG"
 }
```

Response fields notes:

- The returned auction values already include any accessory and/or mileage adjustments.
- The **`daterange`** field indicates the week that the values are active for
- **`acceptablemileage`** corresponds to the (monthly/weekly Used) **`averagemileage`** field.
- **`mileageadjust`** represents the calculated mileage adjustment for the input mileage parameter. Its value is already included in the Auction valus.
- **`minauctionvalue`** represents the minimum value you can adjust any of the Auction Values to when using the "after-the-fact" mileage adjustment approach.

Developer note: To recalculate the mileage adjusted Auction values after calling this endpoint; his method ***cannot be used*** if any of the Auction values is already equal to the **`minauctionvalue`**

- Subtract the provided **`mileageadjust`** value from the three Auction values
- Find a new **`mileageadjust`** value
  - Prior to April 4, 2022 using the formula below:
    - **`mileageadjust`** = **`ratepermile`** x (**`inputMileage - acceptablemileage`**)
  - On/After April 4, 2022, or before if using the v2MileageStartDate (see below):

- Call the weeklyAuctionValueMileageByVehicleId endpoint and look up the appropriate adjustment in the mileage chart/table

- Add the new **`mileageadjust`** value to each Auction value
- Set any new Auction value to the **`minauctionvalue`** if the mileage adjustment reduced the Auction value below this value

Developer notes regarding new WAV mileage and the `v2MileageStartDate` parameter:

- Effective April 4, 2022 (or any date on/after the date passed in the `v2MileageStartDate` parameter), the `ratepermile` field will be `null`, and the `acceptablemileage` field will only be useful for displaying the "average" mileage.
- Values passed in the `v2MileageStartDate` parameter later than 2022-04-04 will generate an error response; you cannot use this parameter to retrieve old-format WAV mileage values after the new approach is fully live on the production API as the data needed will not exist.

6.8.2.weeklyAuctionValuesByVinAndVehicleId

Request the vehicle's Weekly Auction Values for the input VIN and UcgVehicleId (needed to process Shared VINs so required for all requests)

License decrement: This method **does** decrement the user's license

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/weeklyAuctionValuesByVinAndVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	Vin to be decoded
• ucgvehicleid	int	required	Vehicle ID
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• accessorylist	string	optional	Comma-delimited array of VAC codes for options to be valued with the vehicle. Not all selected options impact WAV results.
• v2MileageStartDate		optional	Optional ISO date to over-ride the system default (2022-04-04) for when the new WAV mileage process starts.
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Response fields notes:

See weeklyAuctionValuesByVehicleId (above) for response information

Legacy conversion: This expands on the to the getAuctionValues() method in the legacy/SOAP API.

Developer note: This endpoint has the same overall developer notes as the weeklyAuctionValuesByVehicleID endpoint does.

Developer note: This endpoint is primarily intended for SOAP conversion compatibility, as the only difference from weeklyAuctionValuesByVehicleID is in theory the addition of some VIN-indicated accessory values. However, in the master content, these accessories will be factored into the base vehicle values in almost all cases.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
weeklyAuctionValuesByVinAndVehicleId?period=2018-01-01&vin=
1FD8X3BT0F1234567&ucgvehicleid= 201834687&region=1&mileage=7139' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

See weeklyAuctionValuesByVehicleId endpoint for response example

6.8.3.weeklyAuctionValueMileageByVehicleId

Added (to UAT) in December 2021, this endpoint provides the developer a means of re-calculating the Weekly Auction Value mileage adjustments under the new (as of April 2022) mileage adjustment approach. Values prior to April 2022 used a linear calculation approach whose values were supplied in the WAV valuation endpoint, but the new approach uses a mileage table approach, and requires access to the base High Auction Value to apply caps to the mileage adjustment.

Developer note: This endpoint is only needed if you wish to adjust the Weekly Auction Values for mileage after the initial valuation call; otherwise simply use the initial valuation call's results.

License decrement: This method **does not** decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/weeklyAuctionValuesMileageByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• ucgvehicleid	number	required	Vehicle ID to be valued
• region	number	required	The RegionID being used for valuation. Required due to regional variation in the High Auction Value (for mileage adjustment caps)
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Response fields notes:

- see `mileageByVehicleId` endpoint for field definitions
- Note: this WAV mileage endpoint does not return the `ratepermile` field
- The `adjamount` values returned in this endpoint already implement the `minmileageadj` and `maxmileageadj` mileage cap values based on the vehicles Base High Auction Value (not exposed to the user).

Developer note: This endpoint does not support Period values older than 2021-10-04. The new mileage model itself starts in April 2022, but developers may reference it earlier for testing.
Developer note: Please note the additional required parameter (region) compared to the standard/monthly Mileage endpoint.

Legacy conversion: This functionality was not needed in the legacy/SOAP API; this is a new endpoint.

```
curl -L -X GET
'https://cloud.jdpower.ai/data-api/valuationservices/valuation/mileageByVehicleId?
period=0&ucgvehicleid=201506496&region=3' \

-H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63
```


```
{
    "requestId": "aff8920aehl86qku230il7",
    "result": [
        {
            "lowlevel": "1",
            "highlevel": "5000",
            "adjamount": 7350,
        },
        {
            "lowlevel": "5001",
            "highlevel": "10000",
            "adjamount": 6725,
        },
        … ,
        {
            "lowlevel": "240001",
            "highlevel": "245000",
            "adjamount": -5175,
        },
        {
            "lowlevel": "245001",
            "highlevel": "99999999",
            "adjamount": -5300,
        }
    ],
    "userinfo": "Default",
    "authId": "UCG Admin"
}
```

6.9. Build methods

These methods provide data on specific VIN content, i.e., while the 'standard' methods above provide details on a typical vehicle for a given VIN (based on a VIN pattern), the Build methods provide data on unique/individual VINs.

Because of the increased size of the VIN data and the possible use of external APIs in these methods, you may see slightly increased response times for these methods.

Use of these endpoints typically entails additional license decrement (cost) per method, as noted in the individual endpoint documentation below.

Premium endpoint: As of September 1, 2020, the endpoints in this group fall under our Premium Endpoint definition, and the endpoint URLs have been updated. Customers will need to request access to these (or other Premium endpoints) from your Sales representative.

For example, the "Build VIN Precheck" endpoint (buildDataAvailableByVin) has moved from

https://cloud.jdpower.ai/data-api/valuationservices/valuation/buildDataAvailableByVin

to

https://cloud.jdpower.ai/data-api/valuationservices/build/buildDataAvailableByVin

Other functionality is unchanged.

6.9.1.VIN Precision+

VIN Precision+ from J.D. Power Valuation Services makes the blind spots associated with traditional VIN decoding a thing of the past. By combining the unbiased and accurate values from J.D. Power Valuation Services with industry-leading Chrome Data "as built" data, VIN Precision+ decodes VINs based on actual OEM build data, where available, or incorporates vehicle configuration information derived from a sophisticated algorithm developed by J.D. Power, to reveal exact trim and equipment specifications at the time the vehicle rolled off the assembly line.

1. Chrome Verified Engineered update (Q3 2024)

   This enhancement adds Chrome's Verified Engineered build data as a source for the VIN Precision+ functionality. This high-quality engineered data is based on near-OEM quality data sources and will be used only when OEM sources are not available for the VIN in question. An optional and hidden (not in the Swagger definitions) parameter allows a user to disable this enhancement on a per-call basis.

6.9.2. Technical considerations in VIN Precision+ endpoints

5. Possible false negatives in Build Available endpoint

5.1. A recent (April 2021) internal enhancement added a timeout when calling external APIs. As a result, in rare cases you may get a False response because of the timeout.

6. Possible empty responses in Build Value (and accessories)

6.1. The valuation endpoints include additional data validation steps that are not used in the Build Available endpoint. As a result, it is possible (but unlikely) to get an empty response in the valuation endpoint even after a True response from the Build Available endpoint. Scenarios where this can occur are:

- The input VIN is not available in the valuation period but is available in the Build data; the valuation data has a rolling 20 model year filter which is not applicable to the Build data.
- There was a timeout in the call to the (external) Chrome service when retrieving the Build details
- Rarely, the vehicle data returned does not align with our internal VIN decoding. This will reduce in frequently over time.

3. Higher latency in these endpoints

3.1. Due to the nature of the VIN Precision+ endpoint functionality, the response time for these endpoints may be longer than for our other endpoints.

3.2. To reduce the potential impact of intermittent upstream network latency, these endpoints include a timeout, which can result in a 3-second empty response.

6.9.3.buildDataAvailableByVin

This method provides a simple true/false response on whether we currently have Build Data available for the VIN in question and should be called before calling the other Build methods to ensure you don't waste time on empty results. If the response comes back as false, please call the corresponding 'standard' method instead of the Build method.

License decrement: This method **does not** decrement the user's license.

Premium endpoint: This endpoint falls into the Premium classification and requires explicit access rights to use.

Endpoint: https://cloud.jdpower.ai/dataapi/valuationservices/build/buildDataAvailableByVin

Parameters:

• vin	string	required	VIN to be used in future Build calls
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• oem	int	hidden	1 = Do not use Verified Engineered data. Default behavior is Use Engineered

Legacy conversion: This is an entirely new area of functionality with no corresponding SOAP call

```
curl -X GET \
  'https://cloud.jdpower.ai /data-api/UAT/valuationservices/build/
buildDataAvailableByVin?vin=1FMCU9J93JUB98016' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
    "requestId": "j6hkjcnm1biqymh9126g9",
    "result": [
        {
            "builddataavailable": "true"
        }
         ]
      , "userinfo": "Default"
      , "authId": "UCG"
    }
```

6.9.4.buildVehicleAndValuesByVin

This method corresponds to the 'standard' defaultVehicleAndValuesByVin method, but it uses the Build data to both get down to a specific trim/vehicle on shared VINs, and also includes adjustments for all options which the Build data indicates is present based on the unique VIN, beyond the 'standard' method's engine/drive options.

Developers should call the buildDataAvailableByVin method to determine if Build Data is available for the VIN in question and use one of the 'standard' VIN decode/valuation methods above if that method returns a false response.

License decrement: This method decrements the user's license **by two** on successful calls.

Premium endpoint: This endpoint falls into the Premium classification and requires explicit access rights to use.

Endpoint: https://cloud.jdpower.ai/dataapi/valuationservices/build/buildVehicleAndValuesByVin

Parameters:

• period	string	required	Valuation period as YYYY-MM-DD
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be used in future Build calls
• region	int	required	Region ID (see Definitions and Regions method)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• oem	int	hidden	1 = Do not use Verified Engineered data. Default behavior is Use Engineered
• vinsource	int	hidden	1 = Return VIN Source in response 0 or omit (default) = omit this field
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

Legacy conversion: This is an entirely new area of functionality but corresponds to the Build implementation of the getDefaultVehicleAndValueByVin

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/UAT/valuationservices/build/
buildVehicleAndValuesByVin?
period=2019-04-01&vehicletype=UsedCar&vin=1N4BZ0CP0HC300077&region=10&mileage=1
000' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

(see defaultVehicleAndValuesByVin method above for data return structure)

6.9.5.(Deprecated) buildAccessoriesByVin

Developer note: This endpoint is deprecated as of 3/1/2024. Use the buildAccessoryDataByVin endpoint instead.

This method corresponds to the 'standard' accessoriesByVinAndVehicleID method, but it uses the Build data to flag the isAdded field for all options which the Build data indicates is present based on the unique VIN, beyond the 'standard' method's engine/drive options.

Developers should call the buildDataAvailableByVin method to determine if Build Data is available for the VIN in question and use one of the 'standard' VIN decode/valuation methods above if that method returns a false response.

License decrement: This method does not decrement the user's license.

Premium endpoint: This endpoint falls into the Premium classification and requires explicit access rights to use

Endpoint: https://cloud.jdpower.ai/dataapi/valuationservices/build/buildAccessoriesByVin

Parameters:

• period	string	required	Valuation period as YYYY-MM-DD
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be valued
• region	int	required	Region ID (see Definitions and Regions method)
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• oem	int	hidden	1 = Do not use Verified Engineered data. Default behavior is Use Engineered

Legacy conversion: This is an entirely new area of functionality but corresponds to the Build implementation of the getAccessories method with (sometimes) additional accessories being flagged as isAdded=1

```
curl -X GET \
  ' https://cloud.jdpower.ai/data-api/UAT/valuationservices/build/
buildAccessoriesByVin?
period=2019-04-01&vehicletype=UsedCar&vin=1FMCU9J93JUB98016&region=1' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

(see accessoriesByVehicleId method above for data return structure)

Developer note: In general, you should treat any accessories flagged as `isincluded` or `isadded` as if the user had selected them, i.e., check for any "child" accessories in the Includes and Excludes fields and process them as if the user had selected the `isincluded` accessory. As noted above, any 'parent' `isincluded` accessories will not have any **`Excludes`** to check for.

6.9.6.weeklyBuildVehicleAndValuesByVin

This is the “Weekly Used Values” endpoint corresponding to the “monthly” buildVehicleAndValuesByVin endpoint and functions identically other than using the Weekly Used Values.

License decrement: This method decrements the user’s license **by two** on successful calls unless immediately before or after an otherwise-identical call to the monthly endpoint.

Premium endpoint: This endpoint falls into the Premium classification and requires explicit access rights to use

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/build/weeklyBuildVehicleAndValuesByVin

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be valued
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• oem	int	hidden	1 = Do not use Verified Engineered data. Default behavior is Use Engineered
• vinsource	int	hidden	1 = Return VIN Source in response 0 or omit (default) = omit this field
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

```
curl -L -X GET 'https://cloud.jdpower.ai/data-api/valuationservices/build/
weeklyBuildVehicleAndValuesByVin?
period=0&vehicletype=UsedCar&vin=1FMCU9J93JUB98016&region=1&mileage=12345' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

The fields in this response correspond to those in the buildVehicleAndValuesByVin endpoint with the exception that the base and adjusted values include “weekly” in the field names and represent the weekly updates to the Used values. The ‘business value’ fields (maxmileageadj through minadjretailforloan) apply equally to these Weekly Used Values.

1/2/2025 Note: The response for this endpoint has had the vid field in the middle of the result array element, rather than being the last field as intended. This is updated on 1/2/25 (UAT) and will be included in the production deployment later in January.

6.9.7.buildAccessoryDataByVin

This method corresponds to the 'standard' accessoryDataByVin method, but it uses Build data to flag the isAdded field for all options which the Build data indicates is present based on the unique VIN, beyond the 'standard' method's engine/drive options. As compared to the older/ deprecated buildAccessoriesByVin endpoint, this new version adds the Accessory Logic (Includes/Excludes data), removing the need to use the accessoryLogicByVehicleId endpoint.

Developers should call the buildDataAvailableByVin method to determine if Build Data is available for the VIN in question and use one of the 'standard' VIN decode/valuation methods above if that method returns a false response.

License decrement: This method does not decrement the user's license.

Premium endpoint: This endpoint falls into the Premium classification and requires explicit access rights to use

Endpoint: https://cloud.jdpower.ai/dataapi/valuationservices/build/buildAccessoryDataByVin

Parameters:

• period	string	required	Valuation period as YYYY-MM-DD
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be used in future Build calls
• region	int	required	Region ID (see Definitions and Regions method)
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• oem	int	hidden	1 = Do not use Verified Engineered data. Default behavior is Use Engineered
• vinsource	int	hidden	1 = Return VIN Source in response 0 or omit (default) = omit this field
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

Legacy conversion: This is an entirely new area of functionality but corresponds to the Build implementation of the getAccessories method with (sometimes) additional accessories being flagged as isAdded=1

```
curl -X GET \
  ' https://cloud.jdpower.ai/data-api/UAT/valuationservices/build/
buildAccessoryDataByVin?
period=2019-04-01&vehicletype=UsedCar&vin=1FMCU9J93JUB98016&region=1' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

(see accessoryDataByVehicleId method above for data return structure)

Developer note: In general, you should treat any accessories flagged as `isincluded` or `isadded` as if the user had selected them, i.e., check for any "child" accessories in the Includes and Excludes fields and process them as if the user had selected the `isincluded` accessory.

As noted above, any 'parent' `isincluded` accessories will not have any **`Excludes`** to check for.

6.9.8.buildVehicleAndUsedValuesByVin

This endpoint acts as a combination of the buildVehicleAndValuesByVin and weeklyBuildVehicleAndValuesByVin endpoints.

Unlike the combined Used values by ID endpoints, this endpoint does provide the optional vehicletype filter parameter.

License decrement: This method does decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/defaultVehicleAndUsedValuesByVin

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	Vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.
• oem	int	hidden	1 = Do not use Verified Engineered data. Default behavior is Use Engineered
• vinsource	int	hidden	1 = Return VIN Source in response 0 or omit (default) = omit this field
• vinsourcedetail	int	hidden	1 = Return VIN Source Detail in response 0 or omit (default) = omit this field

Legacy conversion: This is a new method; there is no corresponding legacy/SOAP method.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/build/
buildVehicleAndUsedValuesByVin?
period=0&vin=1C4RJEBG4JC358889&vehicletype=UsedCar&region=8&mileage=41094' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

```
{
  "requestId": "1709232881541-test-run",
  "result": [
    {
      "modelyear": "2018",
      "make": "Jeep",
      "model": "Grand Cherokee",
      "body": "Utility 4D Limited 3.6L V6",
      "ucgvehicleid": "201830712",
      "vehicletype": "Light Truck",
      "mileageclass": "2",
      "basemsrp": "38195",
      "bodytype": "Utility",
      "doors": "4",
      "trim": "Limited",
      "trim2": "N/A",
      "drivetype": "RWD",
      "liters": "3.6",
      "engineconfiguration": "V",
      "cylinders": "6",
      "inductiontype": "Standard",
      "transmission": "N/A",
      "fueltype": "Gas",
      "wheels": "4",
      "curbweight": "4576",
      "gvw": null,
      "gcw": null,
      "ucgsubsegment": "Midsize SUV",
      "modelnumber": "JEB",
      "rollupvehicleid": "201830712",
      "basecleantrade": "17425",
      "baseaveragetrade": "16350",
      "baseroughtrade": "15025",
```

```
"basecleanretail": "20325",
"baseloan": "15700",
"baseweeklycleantrade": "17050",
"baseweeklyaveragetrade": "15975",
"baseweeklyroughtrade": "14650",
"baseweeklycleanretail": "19950",
"baseweeklyloan": "15350",
"averagemileage": "82500",
"averageweeklymileage": "82500",
"mileageadjustment": "2925",
"weeklymileageadjustment": "2925",
"vinoptionstrade": "250",
"vinoptionsretail": "275",
"vinoptionsloan": "250",
"weeklyvinoptionstrade": "250",
"weeklyvinoptionsretail": "275",
"weeklyvinoptionsloan": "250",
"adjustedcleantrade": "20600",
"adjustedaveragetrade": "19525",
"adjustedroughtrade": "18200",
"adjustedcleanretail": "23525",
"adjustedloan": "18875",
"adjustedweeklycleantrade": "20225",
"adjustedweeklyaveragetrade": "19150",
"adjustedweeklyroughtrade": "17825",
"adjustedweeklycleanretail": "23150",
"adjustedweeklyloan": "18525",
"maxmileageadj": "8713",
"minmileageadj": "-6970",
"maxweeklymileageadj": "8525",
"minweeklymileageadj": "-6820",
"minadjretail": "420",
"minadjcleantrade": "120",
```

```
        "minadjaveragetrade": "120",
        "minadjroughtrade": "120",
        "minadjloan": "0",
        "minadjretailforloan": "900",
        "vid": "94737"
      }
    ],
    "userinfo": "Default"
  }
```


Note: This new endpoint adds the gvw, gcw, and vid return fields as compared to the standard/monthly endpoint's return.

6.10.Other methods

Endpoints that do not fall into any of the other categories. These are listed in the order they were added to our API.

6.10.1.regions

Request the list of Regions in use by the Used Car Guide valuation processes. All method calls that return a Used value, including the various accessory retrieval methods, require a Region ID in the input. The return array includes the list of Regions as well as the different States with their corresponding Region ID values.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/regions

Parameters:

- userinfo string optional User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This a replacement for the legacy/SOAP method getRegions() method.

Developer note: The vehicletype field is provided here for legacy support purposes only.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/regions \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'

{
  "requestId": "abc123defg56789",
  "result": [
    {
          "region": "Eastern",
          "regionId":"1"
    },
    {
          "region": "Pacific Northwest",
          "regionId":"2"
    },
    {
          "region": "Washington",
          "regionId":"2"
    }

 ]
   , "userinfo": "Default"
   , "authId": "UCG"
 }
```

6.10.2.regionIdByStateCode

Request the Region ID for the input State Code (i.e., VA for Virginia, NY for New York, etc.) All method calls that return a Used value, including the various accessory retrieval methods, require a Region ID in the input. The return array includes the list of Regions as well as the different States with their corresponding Region ID values.

License decrement: This method does not decrement the user's license based.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/regionIdByStateCode

Parameters:

• statecode	string	required	2-character US Postal Code for the state
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This the legacy/SOAP method getRegionByStateCode() method.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
regionIdByStateCode?statecode=VA' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'

{
  "requestId": "abc123defg56789",
  "result": [
    {
          "region": "Eastern",
          "regionId":"1"
    }
 ]
    , "userinfo": "Default"
    , "authId": "UCG"
 }
```

Developer note: This endpoint will now return an HTTP 400/Bad Request status and accompanying error message if an invalid StateCode is passed; scheduled for production API on 4/8/2022.

6.10.3.newestAvailablePeriod

Determine the latest data period that is available in the API. This will allow you to programmatically determine whether the next months' data is available for instance.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/newestAvailablePeriod

Parameters:

- userinfo string optional User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces the legacy functionality custom Period value of 1 by allowing the user to determine if the next month's "early values" are available yet.

Developer note: The vehicletype field is provided here for legacy support purposes only.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
newestAvailablePeriod' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'

{
  "requestId": "abc123defg56789",
  "result": [
    {
          "period": "201909",
          "beginperiod": "2019-09-01",
          "endperiod": "2019-09-30",
          "week": "201934",
          "beginweek": "2019-09-09",
          "endweek": "2019-09-15"
    }
  ]
    , "userinfo": "Default"
    , "authId": "UCG"
}
```

Response fields notes:

- period: Corresponds to the legacy SOAP period, i.e., valuation data period as YYYYMM
- beginperiod: earliest date available for the data period
- endperiod: latest date available for the data period
    - ***Developer note:*** New as of 2019-09-13: Note the following weekly fields which provide the latest available period value for Weekly Auction Values endpoints
- week: Represents the ISO Week, i.e., YYYY + (week number)
- beginweek/endweek: first and last date in the latest available weekly data

6.10.4.apiUsageReport

This method provides a user access to their own REST API license activity for the input date range, showing all license-impacting API calls for the current user. It also includes the (optional) userinfo data which will allow you to review the results received on a per-user basis; any requests made without a userinfo parameter value will return as userinfo=Default.

For security purposes, the api-key value returned in this report is partially masked.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/apiUsageReport

Parameters:

- startdate string optional Begin date in yyyy-mm-dd format
  leave blank to use the 1st day of the current month
- enddate string optional End date in yyyy-mm-dd format
  leave blank to use 31 days after the startdate

Legacy conversion: This is similar in functionality to the online License Usage Detail Report available through the nada.com Customer Maintenance web application. Please note that this information (the REST API usage data) will also be available through that application in the upcoming months.

Date range update (October 2022).
This endpoint has been updated to return data only for the past 3 calendar months (current calendar month plus the prior two calendar month). In other words, if you call this endpoint on October 20h, 2022, you can only see data for August 1st through October 19th, 2022.

If you need prior month's detailed reports, please contact your Sales representative.

To prevent breaking code for applications already coded to request older data, this endpoint does not return an error code if you request older (unavailable) data; you will simply receive an empty result[] array.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/apiUsageReport?
startdate=2019-08-01&enddate=2019-08-31' \
  -H 'api-key: 3f3090a0-0a7b-40a7-9998-39d3efe40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
{

  "Package": "UCG Build Value",
  "LookupDate": "2019-08-27 11:25:04.000 America/New_York",
  "Username": "UCG UAT",
  "ApiKey": "xxxxxxxx-xxxx-xxxx-xxxx-6d697f8c86eb",
  "UserInfo": "Default",
  "RequestId": "sdv8fdmh9zqba671y1hx",
  "ApiMethod": "defaultVehicleAndValuesByVin",
  "VehicleDescription": "2010 GMC Light Duty Canyon Crew Cab SLT 3.7L I5",
  "UcgVehicleId": "201009676",
  "Vin": "1GTJSFDE8A8123456",
  "Period": "201908",
  "Region": "Eastern",
  "LicenseCoefficient": "1"
 },
```


Developers note: Due to the amount of data behind this method, the response time can be up to several seconds and may time out due to transmission times if there are exceptionally large counts. If this happens, please reduce the date range, or contact your sales representative.

Business Rules and notes on fields.

- **Package**: This will always be "UCG Build Value" and represents how the license record will appear in the NADA Online report (when available)
- **LookupDate:** The date the lookup occurred; this system reports using US Eastern time
- **UserName:** The company name associated with the API Key (as established when the API Key was generated)
- **ApiKey:** A partially masked value of the API Key passed in this Report request
- **UserInfo:** The (optional) userinfo input parameter passed in the original request. If no userinfo was passed, this will return the literal string Default
- **RequestId:** The requestId returned for the original request
- **ApiMethod:** The endpoint called in the original request
- **VehicleDescription:** A description of the vehicle valued in the original request
- **UcgVehicleId:** The internal UCG vehicle ID returned in the original request
- **VIN:** The VIN passed in the original request (for VIN methods only)
- **Period:** The valuation period (as YYYYMM) in the original request

- **Region:** The text description of the Used Car Guide region used in the original request
- **LicenseCoefficient:** How many “lookups” were charged against the license by the original request. Values range from 0 (for unvalued vehicles) to 2 (valued vehicle + Build data)

6.10.5.vinCheckSum

This endpoint verifies the input VIN against the US NHTSA VIN Checksum calculation formula and returns a simple true/false as to whether the VIN passes checksum.

Please note that this has no bearing on the other endpoints' decoding methods; the 'standard' VIN decode endpoints for instance depend primarily on a masked VIN pattern, not the full 17-character VIN. However, if a VIN doesn't pass the checksum calculation it will not exist in our Build endpoints' data.

License decrement: This method does not decrement the user's license based on the additional VIN Content data in use.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/vinCheckSum

Parameters:

- vin string required VIN to be validated

Legacy conversion: This the replacement for the legacy SOAP Web Service's VinCheckSum() method.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/vinCheckSum?
vin=3VWVH69M23M170818' \
  -H 'api-key: 3f3090a0-0a7b-40a7-9998-39d3efe40d63'
```


```
{
    "requestId": "rcl9px90v6dvlrfsdozqsp",
    "result": [
        {
            "vin": "3VWVH69M23M170818",
            "result": "true"
        }
    ],
    "authId": "UCG Testing"
}
```


6.10.6.apiLatencyReport

This endpoint will return internal latency figures for several time ranges, both for your API Key and for all users. It reports the Average and 90th Percentile values for the internal latency tracked within our system for the following time ranges:

Latest 8 hours	Latest 24 hours
Latest 7 days	Latest 30 days
Latest 90 days	

If your API has access to, and has used, any premium folders, the values will be broken down into one row of data for the primary (/valuation/) path, one for each premium path, and one (All) for all endpoints.

The latency values are updated every two hours, at approximately ½ past the hour.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/apiLatencyReport

Parameters: (none) The endpoint references the api-key used to invoke it.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/apiLatencyReport' \
  -H 'api-key: 3f3090a0-0a7b-40a7-9998-39d3efe40d63'
```


```
{
    "requestId": "qugqkw0kholwijbbb315h",
    "result": [
        {
            "timerange": "8 hours",
            "endpointpath": "All",
            "myAvgLatency": "125",
            "my90PercentLatency": "215",
            "allAvgLatency": "115",
            "all90PercentLatency": "349"
        },
        {
            "timerange": "8 hours",
            "endpointpath": "valuation",
            "myAvgLatency": "39",
            "my90PercentLatency": "35",
            "allAvgLatency": "114",
            "all90PercentLatency": "349"
        },
        {
            "timerange": "8 hours",
            "endpointpath": "build",
            "myAvgLatency": "209",
            "my90PercentLatency": "272",
            "allAvgLatency": "216",
            "all90PercentLatency": "298"
        },
          …etc.
       }
],
    "authId": "Site24x7 Monitor"
}
```


Notes on fields:

- timerange — How far back does this data cover (from last 2-hour update)
- endpointpath — Group of endpoints covered. If you do not have access to any of premium endpoints, you will only see the `valuation` folder
- myAvgLatency — The average (internal) latency, in milliseconds, for your API calls
- my90PercentLatency — 90 percent of your API calls process in this value or less
- allAvgLatency — The average latency for all API calls to this path/time range
- all90PercentLatency — 90 percent of all calls to this path/time range process in this value or less.

6.10.7.apiUsageSummary

This endpoint provides license tracking data for the most recent 12 months' usage, like the existing apUsageReport endpoint but is summarized at monthly levels. As with the prior endpoint, this data is not updated in real time, so please do not call either endpoint more than once daily. The license activity report data is updated overnight (approx. 0600 UTC). Also, the data from this endpoint is for informational purposes only; the official billing data is maintained offline and can be requested periodically from your Sales contact.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/apiUsageSummary

Parameters:

- startdate string optional Begin date in yyyy-mm-dd format
  leave blank to use the 1st day of the current month
- enddate string optional End date in yyyy-mm-dd format
  leave blank to use 31 days after the startdate

Developer note: The data behind this endpoint only covers the most recent 12 months' activity. If you need report data for older months, please contact your sales representative. The report is also capped at no more than 100 distinct UserInfo values for reporting. If you use more than 100 distinct UserInfo values for license tracking and need this reporting data at the userinfo level, please contact your sales representative.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/apiUsageSummary?
startdate=2022-01-01&enddate=2022-02-28' \
  -H 'api-key: 3f3090a0-0a7b-40a7-9998-39d3efe40d63'
```


```
{
    "requestId": "5vo9lnrytop9dp81ehlv7g",
    "result": [
        {
            "lookupMonth": "2022-01",
            "UserInfo": "Default",
            "valuationCalls": "15954",
            "licenseUsage": "11886"
        },
        {
            "lookupMonth": "2022-01",
            "UserInfo": "UCG Postman Tests",
            "valuationCalls": "68",
            "licenseUsage": "28"
        },
        {
            "lookupMonth": "2022-01",
            "UserInfo": "Key monthly total",
            "valuationCalls": "16022",
            "licenseUsage": "11914"
        },
        …etc…
    ],
```

```
    "message": "These figures are for your informational purposes only, are
not official, and do not include recent activity. Please contact your
account manager if you need the official counts.",
    "authId": "UCG Admin"
}
```


6.10.8.distributionCount

When required by license, this endpoint allows a user to self-report activity when they are reselling/redistributing our values to multiple end-users. The input is the requestId from the original valuation API call, and the number of "re-distributions" made for those values.

If you pass the values from our API to more than a single end-user, please check with your account manager to see if this endpoint will be required.

License decrement: This method does decrement the user's license based on the number of distributions listed for the original valuation call.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/distributionCount

Parameters:

• requestId	string	required	The requestId value from the original valuation API call
• distributionCount	int	required	The number of times you redistributed the original values
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This is a new endpoint with no corresponding legacy functionality.

Developer note: If you are required to use this endpoint for reporting purposes, you must store the original valuation API call's requestId from its response to pass as one of the inputs for this endpoint.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/distributionCount?
requestId= abc123defg56789&distributionCount=8' \
  -H 'api-key: 3f3090a0-0a7b-40a7-9998-39d3efe40d63'
```


```
{
    "requestId": "qugqkw0kholwijbbb315h",
    " "result": [
    {
      "target_requestId": "abc123defg56789",
      "distributionCount": 7,
      "success": "true"
    }
  ],
  "userinfo": "Default"
  "authId": "Site24x7 Monitor"
}
```

6.11.Legacy Support Endpoints

6.11.1.General note on Legacy Support Endpoints

These endpoints are not intended to be permanent; they will be removed shortly after we have confirmed all licensed users of our legacy data and API have finished converting to our REST API.

Developer note: All endpoints in this section have been deprecated as of October 2022 unless noted below. Data updates are slated to be available through June 2023. These endpoints will be removed from the UAT API around July 2023, and from the production API shortly thereafter.

6.11.2.*(deprecated)* masterVehiclesByLegacyId

Deprecated October 2022.

For details on this endpoint, see its entry in the Deprecated Endpoint document section.

6.11.3.*(deprecated)* Legacy Build endpoint (buildLegacyVehicleAndIdsByVin)

Deprecated and removed from UAT instance as of July 26, 2020

Removal from Production instance scheduled for August 28, 2020

For details on this endpoint, see its entry in the Deprecated Endpoint document section.

6.11.4.*(deprecated)* masterToLegacyVehicleMapping

Deprecated October 2022.

For details on this endpoint, see its entry in the Deprecated Endpoint document section.

6.11.5.*(deprecated)* weeklyUsedValuesByLegacyId

Deprecated October 2022.

For details on this endpoint, see its entry in the Deprecated Endpoint document section.

6.11.6.*(deprecated)* masterVehiclesByLegacyIdAndVin (GET method)

Deprecated October 2022.

For details on this endpoint, see its entry in the Deprecated Endpoint document section.

6.11.7. *(deprecated)* masterVehiclesByLegacyIdAndVin (POST method)

Deprecated October 2022.

For details on this endpoint, see its entry in the Deprecated Endpoint document section.

6.11.8. *(deprecated)* vehicleList_Legacy

Deprecated October 2022.

For details on this endpoint, see its entry in the Deprecated Endpoint document section.

7. Definitions

7.1. VIN Decoding definitions

7.1.1. Shared VINs

The primary VIN decode methodology used in our systems uses a VIN pattern, where certain portions of multiple 17-character VINs are masked out to allow one record to represent many thousands (or hundreds of thousands) of unique VINs. For instance, the following VIN pattern could represent many VINs:

```
Pattern      1N4AA51E _ 9C _ _ _ _ _ _

Unique VINs  1N4AA51E09C800131
             1N4AA51E09C800436
             1N4AA51E09C800467
```

Some manufacturers will share some of these VIN patterns with multiple trims or other combinations that UCG treats as separate vehicles. For instance, in some cases, you can tell from a VIN that a vehicle is a V6 or 4-Cylinder engine and Sedan or Coupe, but you cannot tell which of 3 or 4 trim levels it could be.

7.1.2. Default decode

When a VIN can decode to more than one vehicle in UCG's data, our analysts will determine which vehicle is the best representative for that VIN pattern. This vehicle is referred to as the Default.

7.1.3. Alternate decode

When a VIN can decode to more than one vehicle in UCG's data, our analysts will determine which vehicle is the best representative for that VIN pattern. This vehicle is referred to as the Default. The remaining vehicles which are available for the input VIN pattern are referred to as Alternates.

Not every VIN has alternates.

There is no defined maximum for the number of alternates for any given VIN.

7.1.4. Build decode

Available through the VIN Precision+ premium endpoints, this decoding system uses the full, exact 17-character VIN to determine the specific trim level, and when available, the installed option/accessory content.

Updated January 30, 2025 Build decode results are also available in the legacy VIN Decode endpoints via the VIN Precision+ Trim Reduction enhancement. This is an opt-out update.

Updated January 30, 2025 Build decode results are also available in the legacy VIN Valuation endpoints via (opt-in) the VIN Precision+ Content Adjustment enhancement.

7.1.5.VIN Source field

Available as an optional response field when using Trim Reduction, or a standard response field when using VIN Precision+ Content Adjustment, this field indicates whether the response is based on our standard VIN decoding approach (Standard) or Build data-driven (VIN Precision+).

For Trim Reduction, non-Shared VINs will return Standard even when Build Data is available unless documented otherwise in the specific endpoint definitions.

For VP+ Content Adjustment, all Build results will the VIN Precision+ vinsource response.

7.1.6.VIN Source Detail field

Available as an optional response field when using Trim Reduction, VIN Precision+ Content Adjustment, or most VIN Precision+ (/build/*) endpoints, this field indicates the source of the build data for the input VIN. Options are (currently): OEM, Verified Engineered, UCG Legacy, and Pattern (for non-VIN Precision+ results.)

7.2. Vehicle data definitions

7.2.1.modelyear

The Model Year of the vehicle in question

7.2.2.make

Our description for the manufacturer

7.2.3.model

Our description for the model of the vehicle.

Legacy conversion: This replaces the legacy Series description

7.2.4.body

Our description of the vehicle at the final level of body type, trim, etc.

7.2.5.ucgvehicleid

The new (master content) vehicle ID. The first four digits represent the model year, and the remaining digits identify a unique vehicle within that model year. If a given vehicle (at the level of trim, doors, body type, engine and transmission etc.) exists across multiple model years, the right-most five digits represent that same vehicle across those model years. I.e., (artificial example) UcgVehicleID 200012345 is the same 'vehicle' but a different model year compared to UcgVehicleID 200112345

Legacy conversion: This is roughly analogous to the SOAP API's UID field, meaning it is a persistent unique vehicle record identifier. As the REST API's content is more granular, there are instances where there is a one-to-many relationship between SOAP vehicles and their UIDs and REST vehicles and their UcgVehicleIDs.

7.2.6.vehicletype

Options are Car, Light Truck, Medium Truck and Heavy Truck. Medium and Heavy Trucks are returned for the input parameter CommercialTruck, while Car and Light Truck are returned for the input parameter UsedCar

7.2.7.mileageclass

A single character representing our internal Mileage Class, representing (currently) values of 1-5 for UsedCar and A-E for CommercialTruck. Some vehicles do not have a mileageclass value.

7.2.8.basemsrp

An integer representing the MSRP as published in our print Guides. This represents the vehicle's Base MSRP, at the time of introduction, and does not factor in typical equipment.

7.2.9.bodytype

E.g., Sedan, Coupe, Convertible, etc.

7.2.10.doors

Number of passenger (not cargo) doors

7.2.11.trim

Description of the vehicle's Trim level, i.e., LX, LE, Premium, etc.

7.2.12.trim2

In uncommon cases we will provide a second trim indicator to acheive a unique name/ description where the common trim is shared with slightly different vehicles.

7.2.13.drivetype

E.g., Front wheel (FWD), rear wheel (RWD), Four Wheel Drive (4WD) etc.

7.2.14.liters

Displacement of the engine in liters

7.2.15.engineconfiguration

Type of engine/cylinder configuration; for example, a V6 would have “V” here, while an I4 would have “I”.

7.2.16.cylinders

Number of cylinders in the engine

7.2.17.inductiontype

E.g., Turbo, Supercharged, Standard, etc.

7.2.18.transmission

E.g., Manual, Automatic, etc.

7.2.19.fueltype

E.g., Gas, Diesel, Electric, Hybrid, etc.

7.2.20.wheels

7.2.21.curbweight

The Curb Weight listed for the vehicle in our Guide

7.2.22.gvw

The Gross Vehicle Weight for vehicles in our Commercial Truck Guide only. GVW represents the total weight of a single vehicle plus its load.

7.2.23.gcw

The Gross Combined Weight for vehicles in our Commercial Truck Guide only. GVW represents the total weight of a single vehicle plus its load plus any trailer and cargo.

7.2.24.ucgsubsegment

Represents our Market Segment description, i.e., Mid-Size Pickup, Intermediate Subcompact, etc.

7.2.25.model number

A representative portion of the VIN, as published in our print Guides

7.2.26.rollupvehicleid

A pointer to the vehicle that corresponds to the legacy/SOAP/print vehicle for the vehicle in question. The correspondence between the legacy and master vehicles is

Master Vehicle (REST) == Legacy Vehicle (SOAP) + 0-n Accessories

The Rollup Vehicle ID represents the UcgVehicleId for the corresponding Legacy Vehicle (i.e., where # of accessories for the mapping is 0 and the master vehicle is the directly corresponding vehicle.)

7.2.27.accessorycategory

A general description of the type of accessory, such as Engine, Drive, Trim, Electronics, etc.

7.2.28.vid

A legacy vehicle ID, analogous to the new system's UcgVehicleId; each unique numeric ID represents a specific vehicle definition in our system. Unlike the UcgVehicleId, the VID is simply a sequential number assigned as each vehicle is created in our content system, so there is no relationship between the VIDs of similar vehicles. The VID may be represented as an 11-character, zero-padded string, i.e., `0000012345`, or as a simple integer, i.e., `12345`

November 2019: Added to the output of several endpoints to support linking the REST API with our AuctionNet data files product, which uses the same base vehicle content as the REST API but still uses the legacy VID identifier.

7.2.29.uid

A legacy vehicle ID, analogous to the new system's UcgVehicleId; each unique numeric ID represents a specific vehicle definition in our system. Unlike the UcgVehicleId, the UID is simply a sequential number assigned as each vehicle is created in our content system, so there is no relationship between the VIDs of similar vehicles.

The UID is the vehicle identifier used in our legacy/SOAP Web Services platform and .NET API and is directly linked to the VID; the UID is simply the VID (integer) plus 1,100,000, i.e., VID `0000012345` or `12345` == UID `1112345`.

November 2019: Added to the output of several endpoints to support linking the REST API with our AuctionNet data files product, which uses the same base vehicle content as the REST API but still uses the legacy VID identifier.

7.2.30.**Multibody**

Currently returned only by the batchDefaultVehicleAndValuesByVin endpoint, this field is used to indicate whether a decoded VIN is ambiguous/shared or not. Ambiguous VINs can decode in theory to more than one vehicle in our data sets. An ***** in this field indicates this scenario, and the Default vehicle was returned/valued.

7.3. Value data definitions

7.3.1. Used Value Types:

4. Retail

The Retail value is our projected retail value (cost to purchase) for a vehicle in clean condition. See the Trade Values section below for the definition of Clean condition.

4.1. *Note:* This field represents the **Average Retail** value for Commercial Truck vehicles and Motorcycles.

5. Loan

The suggested amount of credit that may be obtained on a vehicle based on the Clean Trade-In value. Providers of vehicle financing determine the amount of credit they are willing to extend on a vehicle.

5.1. Note: This field is presented as **Clean Loan** in the print Used Car Guide and our Values Online b2b web application (GUI and reports)

6. Trade-In

These values represent our projected wholesale value for this vehicle (see vehicle condition definitions below) with mileage falling within our assumed mileage range. They are provided for the following conditions:

7. Condition definitions

7.1. **Rough**

Significant mechanical defects requiring repairs in order to restore reasonable running condition; paint, body and wheel surfaces have considerable damage to their finish, which may include dull, faded or oxidized paint, small to medium size dents, frame damage, rust, or obvious signs of previous repairs; interior reflects above average wear, with inoperable equipment, damaged or missing trim, and heavily soiled /permanent imperfections on the headliner, carpet, and upholstery; may have a branded title; vehicle will need substantial reconditioning and repair to be made ready for resale; some existing damage may be difficult to restore.

7.2. Average

Mechanically sound but may require some repairs/servicing to pass all necessary inspections; paint, body and wheel surfaces have moderate imperfections and an average finish which can be improved with restorative repair; interior reflects some soiling and wear in relation to vehicle age, with all equipment operable or requiring minimal effort to make operable; clean title history; vehicle will need a fair degree of reconditioning to be made ready for resale.

7.3. Clean

No mechanical defects and passes all necessary inspections with ease; paint, body and wheels may have minor surface scratching with a high gloss finish; interior reflects minimal soiling and wear, with all equipment in complete working order; vehicle has a clean title history; vehicle will need minimal reconditioning to be made ready for resale.

7.3.1. *Note:* This field represents **Average Wholesale** for Commercial Truck vehicles.

7.3.2.Weekly Auction Values

8. **Low Auction Value**

This represents values for vehicles with below average auction prices. Vehicles in this category may be negatively impacted by short-term oversupply and/or require substantial reconditioning.

9. Average Auction Value

This represents values for with average auction price.

10. **High Auction Value**

This represents values for vehicles with higher-than-average auction market prices. Vehicles in this category may be positively impacted by a short-term shortage in supply and/or in excellent condition with attributes or option content that would improve demand.

11. daterange

This represents the week that the response's values are valid for. The Weekly Auction Values are presented for an ISO Week (Monday through Sunday)

12. auctionaccessories

If you pass an array of installed/present options, this field will return an array of the accessories (by acccode) that impact the Auction values. Currently only engine, drive, and trim accessories will impact Auction values.

7.3.3.Base values

Represent the value of the vehicle by itself, prior to mileage or option/accessory adjustments.

7.3.4.averagemileage

The middle of the 'acceptable mileage' range, i.e., where the mileage adjustment is $0. For Weekly Auction Values, see acceptablemileage

7.3.5.vinoptionsorade/retail/loan

The total adjustment values for any accessories flagged as present based on the input VIN. Primarily valued in Build endpoints, and sometimes in older data periods.

7.3.6.maxmileageadj/minmileageadj

The maximum adjustment range for mileage for this particular vehicle. Regardless of the value printed in the Guide's mileage charts, there is a capped adjustment value based on the vehicle's Base Clean Trade value. The maxMileageAdj represents the maximum Add for low mileage (i.e., positive adjustment value) while minMileageAdj represents the maximum Deduct for high mileage (i.e., negative adjustment value.)

Legacy conversion: These values appear in the vehicle valuation results now instead of the getMileageAdj method results.

7.3.7. minadjretail / minadjcleantrade / minadjaveragetrade / minadjroughtrade / minadjloan / minadjretailforloan / minauctionvalue

These represent the Minimum Adjusted Values Rules figures for the vehicle in question to allow more accurate values when client-side option handling is used.

- minadjretail / minadjcleantrade / minadjaveragetrade / minadjroughtrade / minadjloan, minauctionvalue fields: Your application should not reduce the adjusted value below the listed value(s).
- minadjretailforloan field: If your application reduces the final Adjusted Retail value below this figure, set the Adjusted Loan value to $0 regardless of other factors.

Legacy conversion: These values appear in the vehicle valuation results now instead of the getTotalAdjFloorValues method results.

7.3.8. Adjusted values

These figures represent the vehicle's adjusted values, factoring in mileage and VIN option adjustments, if any, and after applying the Adjusted Values business rules.

7.3.9. acceptablemileage

Added to the Weekly Auction Values endpoints in November 2019; this figure represents the point where there is $0 mileage adjustment applied in the weekly auction values. Analogous to the **`averagemileage`** field for the monthly/Used values.

Legacy conversion: This value corresponds to the **`intercept`** field in the Weekly Auction Values data files products.

7.3.10.ratepermile

This figure represents the mileage adjustment value applied in the Weekly Auction Values endpoints based on the difference between the input **`mileage`** parameter (if any) and the **`acceptablemileage`** value.

Legacy conversion: This value corresponds to the **`slope`** field in the Weekly Auction Values data files products.

Developer note: `ratepermile` will be NULL for valuation periods on/after April 4, 2022, when the Weekly Auction Values mileage update is applied.

7.3.11.Weekly Used Values

The (new as of April 2020) Weekly Values returned represent the most current values available and are updated on the Monday of each week. They otherwise correspond to updated versions of the corresponding “monthly” Used values of the same value types.

7.3.12.Early Release Values

Added in February 2024; Average Trade and Clean Retail values for newer vehicles made available earlier in the valuation process than typical. No option or mileage adjustments. See FAQ for details.

7.4. Other definitions

7.4.1.Region

The Used Car Guide data is broken into ten regions for valuation purposes, so any method that returns a value requires a Region ID in the input parameters. If your business is limited to a particular state/region, you can look up the Region ID once and hard-code it into your application, or you can find the appropriate Region ID using the region (get) or regionIdForStateCode methods.

7.4.2.Package Inclusive

An Accessory Logic condition. Some of the options listed in our data “include” the values of others which can themselves be selected individually without the “package” or “parent” option. For instance, a vehicle may list Aluminum Wheels as a $100 add individually, but then also list a Sport Package for $500; if the Sport Package “includes” the Wheels, the Wheels should be displayed as selected whenever the user selects the Sport Package accessory, but the Wheels accessory’s effective adjustment values become $0.

7.4.3.Mutually Exclusive

An Accessory Logic condition. Some options in our data may “exclude” others; only one of the two (or more) exclusive accessories may be selected and valued by the end user. When the first exclusive accessory is “selected” by the user, any accessory flagged as Excluded by the first accessory should be de-selected and disabled/un-selectable.

Note that it is possible to have 1-way exclusions – for instance, any accessory flagged as Included in the data (representing a “standard” accessory) does not Exclude other accessories, but may itself be excluded by another accessory selection.

Example: a vehicle may list a Power Sunroof accessory as standard (Included = 1) but may also show a w/o Power Sunroof (-$500) accessory. The w/o accessory will de-select the Included option in this case.

7.4.4.Accessory Codes (acccode)

This three-character code represents our internal identifier for an accessory. Not all accessories available for a given vehicle will be listed in our data; only those which our valuation team determines have meaningful impact on the used values.

7.4.5.Body Inclusive accessories (isincluded:1)

Accessories which are flagged as isincluded:1 are factored into the base values of the selected vehicle, i.e., our analysts consider them to be “typically equipped” on the vehicle in question. In some cases, there will be a corresponding “W/out …” accessory available for the less common “not actually on this specific vehicle” scenario; selecting the “W/out” option will typically Exclude the originally included accessory. This is expected behavior.

In general, you should treat any accessories flagged as `isincluded` or `isadded` as if the user had selected them, i.e., check for any “child” accessories in the Includes and Excludes fields and process them as if the user had selected the `isincluded` accessory. As noted above, any ‘parent’ `isincluded` accessories will not have any **`Excludes`** to check for.

8. Exceptions and Error Codes

At this time, the only exceptions or error messages returned are provided in the results as below. More detailed error messages may be added in future enhancements.

Please note that some endpoints will return a successful but empty response which may include an error message (**`response.error`**); this is a warning message rather than an actual error indicator but the "error" name has been in use too long to change.

8.1.1. Invalid API Key:

```
{
    "requestId": "41yd2zayrywt48a3lp0bob",
    "success": false,
    "error": "Invalid api-key."
}
```

See section 4.1 above, or the FAQ section below.

8.1.2.No results for input:

```
{
    "requestId": "41yd2zayrywt48a3lp0bob",
    "result": []
}
```

Developer note: This can be an expected response for many requests, as for example if you attempt to decode a VIN for a model year too old to be in the requested period's data, or you pass a Commercial Truck Guide VIN but set the request's vehicletype parameter to UsedCar.

Please note that for many endpoints, you will also receive an "error" element with a warning message outside the `result[]` array. See 8.1.7 below.

8.1.3.Malformed input or missing (required) parameter:

```
{
    "success": false,
    "requestId": "41yd2zayrywt48a3lp0bob",
    "error": "Invalid input received.  Please check your input and
formatting."
}
```

Developer note: Please note that the API system is typically case-sensitive, so be sure to check case for parameters and input values i.e., "VIN" is not the same to the API as "vin" or sending "usedcar" vs. "UsedCar" in the vehicletype parameter will result in empty results.

This error can also be returned if an error occurs in the database layer of the API.

8.1.4.api-key is temporarily throttled due to exceeding account queries per second setting.

As of early April 2020, we have implemented a Query Per Second (QPS) throttle to prevent high bursts of calls from individual users from overloading the system. The default setting for the QPS throttle is 30, corresponding to over 100,000 calls in an hour. If you encounter this error frequently or anticipate needing a higher QPS limit, please reach out to UcgServiceDeskIT@jdpa.com.

Additionally, each API response contains 3 custom HTTP headers:

- `x-ratelimit-limit` `30` `Your API Key's QPS setting`
- `x-ratelimit-remaining` `99` `Calls left in current polling cycle before limit is hit`
- `x-ratelimit-reset` `1` `Seconds remaining till quota limit resets`

8.1.5.Input validation errors

Platform validation error messages

The following error messages are defined in the API platform itself and are sent when required parameters are not provided, or when parameters defined as numeric input are not able to be parsed as numbers. They also provide an `errorType` field in addition to the `error` field.

13. Required parameter not supplied
    - Applies to: All endpoints with required input parameters
    - errorType: REQUIRED_FIELD_MISSING
    - Error text: *`parameter_name`* `is a required field. Value passed was: Undefined`

14. Parameter type mismatch
    - Applies to: All endpoints with numeric input parameters
    - errorType: NOT_A_NUMBER
    - Error text: *`parameter_name`* `is a number field but the value \"five\" was passed and is not a number.`

8.1.6. Endpoint-specific error messages

Additional error checking and error responses have been added to most API endpoints as below. These messages will be returned in the HTTP response as below along with an HTTP 400/Bad Request status:

```
{
  "success": false,
  "requestId": "some13random94guid",
  "error": "Invalid Period input, period must be in the format of
  YYYY-MM-DD or literal value of 0 for current date."
}
```

If multiple errors occur, only one error message per request will be returned, with the Platform (missing or mismatch parameter) errors taking precedence.

Not every endpoint applies these input validation steps.

15. Confirm valid period

    Validates **`period`** input formatting

    Applies to: All endpoints which require a **`period`** input parameter

    Error text: `Invalid Period input, period must be in the format of YYYY-MM-DD or literal value of 0 for current date.`

16. **Confirm valid region**

    Validates **`region`** input values (1-10)

    Applies to: All endpoints which require a **`region`** input parameter

    Error text: `The region parameter must be an integer between 1 and 10; please see the regions or regionIdByStateCode endpoint for valid region values.`

17. **Confirm valid date range (standard endpoints)**

    Validates **`period`** input values for oldest available data in the API.

    Applies to: All endpoints which require a **`period`** input parameter

    Error text: `The oldest available Period is 2000-01-01 for Used Car Guide or 2003-03-01 for Commercial Truck Guide vehicles.`

18. **Confirm valid date range (legacy support endpoints)**

    Validates **`period`** input values for oldest available data in the API.

    Applies to: All endpoints which require a `period` input parameter

    Error text: `Error: No historical data for this endpoint prior to` *`2019-06-01`*

    `** date may vary**`

19. **Confirm valid vehicle type**

    Validates **`vehicletype`** input values for valid selections

    Applies to: All endpoints which require a **`vehicletype`** input parameter

    Error text: `The optional vehicletype input parameter must be UsedCar or CommercialTruck.`

    `** valid vehicletype values may vary by endpoint **`

    Depending on endpoint, this may also list `Motorcycle` or `NoMotorcycles` as valid input options.

20. Confirm valid mileage

Validates **`mileage`** input formatting

Applies to: All endpoints which require a **`mileage`** input parameter

Error text: `The optional mileage parameter must be a non-negative integer value.`

** Also see Input mileage rounded to integer Warning message below

21. Confirm valid UcgVehicleId

Validates **`ucgvehicleid`** input formatting

Applies to: All endpoints which require a **`ucgvehicleid`** input parameter

Error text: `The input UcgVehicleId must be 9 digits long (10 for motorcycles) and start with the Model Year of the selected vehicle.`

`** Endpoints which do not apply to Motorcycles will not include the (10 for motorcycles) text.`

22. Confirm valid VIN

Validates **`vin`** input size

Applies to: All endpoints which require a **`vin`** input parameter

Error text: `The input VIN must be 17 characters long.`

23. Confirm valid State Code

Validates **`statecode`** input size and contents

Applies to: regionIdByStateCode

Error text: `The input statecode must be two characters representing a US State or District of Columbia; no APO, Zip Codes, or other inputs at this time`

8.1.7. Warning messages

Warning messages have been added **outside** the results[] array for the following conditions. These will appear as below and ***may use `error` as the warning field name***:

```
{
    "requestId": "some13random94guid",
    "result": [],
    "userinfo": "Default",
    "warning_field": "warning_text",
    "authId": "your_user"
}
```

24. No results available

Warning provided when no data is found for the input parameters.

Please note: due to the design of the API platform, it isn't possible to determine what specific parameter(s) caused the empty results.

Warning field name: **`error`**
Warning text: ***`varies`*; examples below**

No data available for the input *parameter(s)* in the input period/date

The input VIN was not able to be decoded for the input period/date

The input VIN was not able to be decoded for the input period/date or no VIN Configuration/Build data was available

25. Input mileage rounded to integer

Non-critical warning provided when a non-integer numeric **`mileage`** value was passed to an endpoint with a **`mileage`** input parameter. Mileage adjustment values returned were based on truncating the input mileage parameter to an integer.

Warning field name: **`AdjustedMileage`**
Warning text: ***The optional mileage parameter was converted to an integer value of NNN***

26. **Result did not match vehicletype filter**

Non-critical warning provided when data was available for all input parameters except the (optional) vehicletype parameter. Will accompany an empty result.

Warning field name: ***error***
Warning text: ***Result did not match vehicletype filter***

As of the 2022-04-08 release (available earlier on UAT platform) this message applies to the following endpoints, but developers should plan on it being a potential warning message for any endpoint with a vehicletype parameter available over the remainder of Q2 2022.

- defaultVehicleAndValuesByVin
- highVehicleAndValuesByVin
- lowVehicleAndValuesByVin
- msrpVehicleAndValuesByVin
- valueByVehicleId
- vehicleAndValueByVehicleId
- vehicleInformationByVehicleId
- weeklyAuctionValuesByVehicleId
- weeklyAuctionValuesByVinAndVehicleId
- vehiclesWithTrimByVin

9. Examples

See cUrl code samples in each endpoint documentation.

10.Frequently Asked Questions

10.1.I receive an error response saying "error": "Invalid api-key."

10.1.1.Make sure you pass the api-key as an HTTP Header, not as a parameter

See section 4.1.1 above.

cUrl Example: `-H "api-key: your_api_key_here"`

C# example:

```
request.Headers.Add("api-key", "your_api_key_here");
```

10.1.2.Make sure you are using the (test vs. production) api-key and URL combination.

See section 5.2 above.

Each api-key is assigned access to either the test or production API instance and will not work against endpoints in the other instance, so if you have a test key, be sure your endpoints have /UAT before the API root space, i.e.

Test endpoint: https://cloud.jdpower.ai/data-api/UAT/valuationservices/valuation/years

If you are using a production key, you would instead use

Prod endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/years

10.2. I'm converting from the legacy/SOAP Web Service product and the value field names have changed. Is there a value field mapping available?

		REST field	SOAP Field		
Value Type		(common naming)	VehicleValue_Struc	Value_Struc	StandardValue_Struc
Base	Retail	basecleanretail		Retail	Retail
	Loan	baseloan		Loan	Loan
	Clean Trade-In	basecleantrade		TradeIn	TradeIn
	Average Trade-In	baseaveragetrade		AvgTradeIn	AvgTradeIn
	Rough Trade-In	baseroughtrade		RoughTradeIn	RoughTradeIn
Adjusted	Retail	adjustedcleanretail	RetailPlusVinAccMileage	MileAdjRetail	
	Loan	adjustedloan	LoanPlusVinAccMileage	MIleAdjLoan	
	Clean Trade-In	adjustedcleantrade	TradeInPlusVinAccMileage	MileAdjTradeIn	
	Average Trade-In	adjustedaveragetrade	AvgTradeInPlusVinAccMileage	MileAdjAvgTradeIn	
	Rough Trade-In	adjustedroughtrade	RoughTradeInPlusVinAccMileage	MileAdjRoughTradeIn	
Other	MSRP	basemsrp	Msrp		
	Mileage Adjustment	mileageadjustment	MileageAdj	MileageAdj	
	Average Mileage	averagemileage	AveMileage	AveMileage	
	Unmapped*				
	Retail + Vin options, w/o mileage	n/a	RetailPlusVinAcc		
	Loan + Vin options, w/o mileage	n/a	LoanPlusVinAcc		
	Clean Trade-In + Vin options, w/o mileage	n/a	TradeInPlusVinAcc		
	Average Trade-In + Vin options, w/o mileage	n/a	AvgTradeInPlusVinAcc		
	Rough Trade-In + Vin options, w/o mileage	n/a	RoughTradeInPlusVinAcc		

* for these, pass 0 for mileage parameter or leave mileage parameter off.

10.3. I'm receiving the error message "Invalid input received. Please check your input and formatting."

- Ensure you're sending all **required** parameters for the endpoint, that they're the expected data type (integer or character), and if you're setting the request URL programmatically, that there are no trailing spaces in any of the parameters.
- You may also receive this error during the monthly database software patch restart which takes about 2-5 minutes and typically occurs on the first (and sometimes third) Monday of each month between 0400 and 0500 UTC. Each DB server uses a different 15-minute time slot and the individual server restarts take about 2 minutes each.
    - J.D. Power maintains an email distribution list to send maintenance notifications as well as updated copies of this API documentation. If you would like to be added to this email list, please contact your Sales representative.

- You can forward the response's `requestId` field to Tim.Draper@jdpa.com for additional debugging/support.

10.4.My network needs your API's IP address(es) for whitelisting.

Since the UCG REST API resides in the cloud, the below values may change over time. The IP addresses below are current as of November 8, 2019.

cloud.jdpower.ai (the primary host) has a static IP of 99.86.66.92

Each API request may be routed through load balancers at 3.217.156.101 and 52.44.64.155

Since we use Cloudfront as Content Delivery Network (CDN), the IP address of cloud.jdpower.ai will have different values based on which region user makes request from. Because of that, some clients may need to whitelist an IP range instead of just a static IP address.

Amazon publishes an updated IP range list for their AWS services at https://ip-ranges.amazonaws.com/ip-ranges.json - if you download this file and filter for "service": "CLOUDFRONT" you can find the current IP ranges.

Once you have the latest CloudFront IP subnet range list, you can find the network(s) you connect through by running the `nslookup` command from a command shell, then finding the matching network ranges from the ip-ranges.json file.

Ex:

```
C:\> nslookup cloud.jdpower.ai

Server:  --your local domain DNS host--
Address:  --xx.xx.xx.xx--

Non-authoritative answer:
Name:    d1ihzp1zrdhr75.cloudfront.net
Addresses:99.84.191.80
          99.84.191.87
          99.84.191.111
          99.84.191.65
Aliases:  cloud.jdpower.ai
```

The 99.84.x.x IP addresses align with the below IP range from the AWS CloudFront list:

ip_prefix	region	service
99.84.0.0/16	GLOBAL	CLOUDFRONT

Your network team would need to whitelist the 99.84.0./16 subnet.

You may need to run the nslookup command several times to get all the CloudFront networks you connect to.

10.5. I'm receiving the error "One of the preprocessor(s) failed" on my requests.

- Ensure that you are sending all **required** parameters for the endpoint, that they're the expected data type (integer or character), and if you're setting the request URL programmatically, that there are no trailing spaces in any of the parameters.
- The API platform is case sensitive, so make sure your request parameters are provided as shown in the documentation – **`Vin`** or **`VIN`** when **`vin`** is expected will generate this error on every VIN endpoint because they are all set to convert the input vin parameter value to uppercase, but the pre-processor will fail if the input is passed as **`Vin`** or **`VIN`** because JavaScript sees those as different values.
- Ensure that you have passed the period parameter as either the literal value **`0`** or as a date formatted as **`YYYY-MM-DD`**, i.e., **`period`**: **`2019-12-1`** will fail because the system expects **`2019-12-01`** for instance.
- Be sure you are sending the request parameters via the URI querystring for GET endpoints (i.e., except for the batch endpoints where the input should be via the request Body)

10.6. Which HTTPS protocols are supported in your API?

Our API platform supports TLS 1.1 and TLS 1.2 only currently. An HTTP 301 permanent redirect is used if an HTTP (not HTTPS) request is sent; be sure your specific application "follows" the 301 Redirect vs. displaying the 301 message text, or simply use HTTPS instead.

TLS 1.0, SSLv3 and older HTTPS security protocols are not supported.

10.7. What is the difference between the masterVehiclesByLegacyId and masterToLegacyMapping endpoints?

- The masterVehiclesByLegacyId endpoint returns the list of Master vehicles based on a Legacy vehicle ID.
    - Use this endpoint if you have an existing Legacy data source (SOAP web service, .NET API, Data Files, etc.) and want to use the REST API for valuation or to take advantage of the more granular vehicle definitions, so you need to convert the Legacy vehicle ID (source) to the Master vehicle ID (destination.)
- The masterToLegacyMapping endpoint returns the single Legacy vehicle which corresponds to the input Master vehicle ID.
    - Use this endpoint if you are already using the REST API (with Master content) but need to feed data back to a legacy system, so you need to take the REST API's Master vehicle ID (source) and find the corresponding Legacy vehicle information (destination).

10.8.Which API endpoints impact my license usage tracking?

While many of our API users license this service on a flat fee or per-rooftop basis, many others pay based on an annual block of valuations, or may pay a flat cost per valuation. This block is consumed as each license performs valuation calls and expires at the end of the subscription time or when the block of valuations is 'exhausted'. In other words, if your license is paid for 10,0000 lookups through 12/31/2020, your license will be exhausted on 1/1/2021 or when you have used 10,000 lookups, whichever occurs first.

A license decrement occurs when a successful valuation call returns a non-$0 vehicle value, which currently can occur for the following endpoints. Premium endpoints are **highlighted**.

- defaultVehicleAndValuesByVin
- lowVehicleAndValuesByVin
- highVehicleAndValuesByVin
- msrpVehicleAndValuesByVin
- valueByVehicleId
- vehicleAndValueByVehicleId
- These endpoints return both monthly + weekly Used values in a single call:
    - usedValuesAndVehicleByVehicleId
    - usedValuesByVehicleId
    - defaultVehicleAndUsedValuesByVin
- weeklyAuctionValuesByVehicleId
- weeklyAuctionValuesByVinAndVehicleId
- The below endpoints will incur one license charge per returned vehicle valued (batch endpoint) or per month returned with non-$0 values (historical endpoint)
    - batchDefaultVehicleAndValuesByVin
    - historicalValuesByVehicleId
- These endpoints require additional license and may incur an additional lookup charge for Build data:
    - **buildVehicleAndValuesByVin**
    - **weeklyBuildVehicleAndValuesByVin**
- The below Weekly Used Values endpoints will be charged if they are not called immediately after the corresponding 'monthly Used value endpoint above, with the same parameters. To avoid issues with 'close but not duplicate' calls not being treated as sequential, see the "monthly + weekly used" endpoints listed above.
    - weeklyUsedValuesByVehicleId
    - weeklyDefaultVehicleAndUsedValuesByVin
    - weeklyVehicleAndValueByVehicleId

The remaining endpoints do not impact the license decrement process **unless otherwise noted in the endpoint documentation.**

10.9. The masterVehiclesByLegacyId endpoint sometimes returns more than one vehicle. How can I determine which to select?

One of the return fields from this endpoint is named “islegacyvehicle”; this is the vehicle in the new system that exactly corresponds to the legacy vehicle **without applying any options**. The other vehicle(s), if any, correspond to the legacy vehicle **with** typically one or more options applied; these options are supplied via the rollupaccessories field. In some cases (typically much older vehicles) the options have “aged out” of the system and no longer have a valuation impact, but the vehicle was defined in a time period when they did exist.

An example for a 2018 Nissan Titan is shown below.

https://cloud.jdpower.ai/data-api/UAT/valuationservices/valuation/masterVehiclesByLegacyId?period=0&uid=1196089


```
{
    "requestId": "a0e8r6932j7spw9tfgbip",
    "result": [
        {
            "ucgvehicleid": "201837641",
            "modelyear": "2018",
            "make": "Nissan",
            "model": "Titan XD",
            "body": "Crew Cab Platinum Reserve 4WD 5.6L V8",
            "islegacyvehicle": "true",
            "rollupaccessories": ""
        },
        {
            "ucgvehicleid": "201837378",
            "modelyear": "2018",
            "make": "Nissan",
            "model": "Titan XD",
            "body": "Crew Cab Platinum Reserve 4WD 5.0L V8 T-Diesel",
            "islegacyvehicle": "false",
            "rollupaccessories": "0GD"
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


For this vehicle (UID 1196089 - 2018 NISSAN Titan XD Crew Cab Platinum Reserve 4WD):

- UcgVehicleId 201837641 (Crew Cab Platinum Reserve 4WD 5.6L V8), with legacyvehicle=true, is the direct corresponding vehicle in the new system
- UcgVehicleId 201837378 (Crew Cab Platinum Reserve 4WD 5.0L V8 T-Diesel) corresponds to UID 1196089 **plus** VAC 0GD (5.0L V8 Turbo Diesel Engine)

If you have the list of selected options for any vehicle record saved with the legacy IDs, you can compare them to the rollupaccessories data returned.

Alternatively, if you have the VIN for the vehicle in question, you can call the vehiclesByVin endpoint and typically only one of the vehicles returned by the masterVehiclesByLegacyId endpoint will appear in both data sets; this is the best fit to use.

2020-06-27: A new endpoint was added on the UAT platform that implements this UID + VIN approach natively inside the API. See matching examples below.

1. 2018 Nissan Titan XD Platinum Reserve (UID 1196089)
    a. masterVehicle results:
    …/valuation/masterVehiclesByLegacyId?period=0&uid=1196089
        i. 201837641 (5.6L V8); legacy=true
        ii. 201837378 (5.0L Turbo Diesel)
    b. Ex: VIN points to legacy vehicle:
        i. Example VIN: 1N6AA1F46JN511649
        …/valuation/vehiclesByVin?period=0&vehicletype=UsedCar&vin=5N1AT2MV5GC829888
        ii. vehiclesByVin results
            1. 201837648 (SV 5.6L V8)
            2. 201837641 (Platinum Reserve 5.6L V8)
            3. 201837643 (Pro-4X 5.6L V8)
            4. 201837644 (S 5.6L V8)
            5. 201837646 (SL 5.6L V8)
        iii. Only common UcgVehicleID is 201837641 (Platinum Reserve 5.6L V8)
    c. Ex: VIN points to non-legacy vehicle:
        i. Example VIN: 1N6BA1F46JN526219
        …/valuation/vehiclesByVin?period=0&vehicletype=UsedCar&vin=1N6BA1F46JN526219
        ii. vehiclesByVin results
            1. 201837385 (SV Turbo Diesel)
            2. 201837378 (Platinum Reserve Turbo Diesel)
            3. 201837379 (Pro-4X Turbo Diesel)
            4. 201837381 (S Turbo Diesel)
            5. 201837383 (SL Turbo Diesel)
        iii. Only common UcgVehicleID is 201837378 (Platinum Reserve Turbo Diesel)

New endpoint examples:

1.b) VIN 1N6AA1F46JN511649

```
https://cloud.jdpower.ai/data-api/UAT/valuationservices/valuation/
masterVehiclesByLegacyIdAndVin?period=0&uid=1196089&vin=1N6AA1F46JN511649
```


```
{
    "requestId": "sn73t9adllua3wtfaznw",
    "result": [
        {
            "selectpriority": "1",
            "ucgvehicleid": "201837641",
            "modelyear": "2018",
            "make": "Nissan",
            "model": "Titan XD",
            "bodystyle": "Crew Cab Platinum Reserve 4WD 5.6L V8",
            "rollupvacs": "",
            "decodepriority": "0",
            "legacypriority": 1
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


1.c) VIN 1N6BA1F46JN526219

```
https://cloud.jdpower.ai/data-api/UAT/valuationservices/valuation/
masterVehiclesByLegacyIdAndVin?period=0&uid=1196089&vin=1N6BA1F46JN526219
```


```
{
    "requestId": "fz2mh64h7kei9y237noti",
    "result": [
        {
            "selectpriority": "0",
            "ucgvehicleid": "201837378",
            "modelyear": "2018",
            "make": "Nissan",
            "model": "Titan XD",
            "bodystyle": "Crew Cab Platinum Reserve 4WD 5.0L V8 T-Diesel",
            "rollupvacs": "0GD",
            "decodepriority": "0",
            "legacypriority": 0
        }
    ],
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


10.10. I've valued a vehicle successfully, but I get an empty response on one or more of the accessory endpoints.

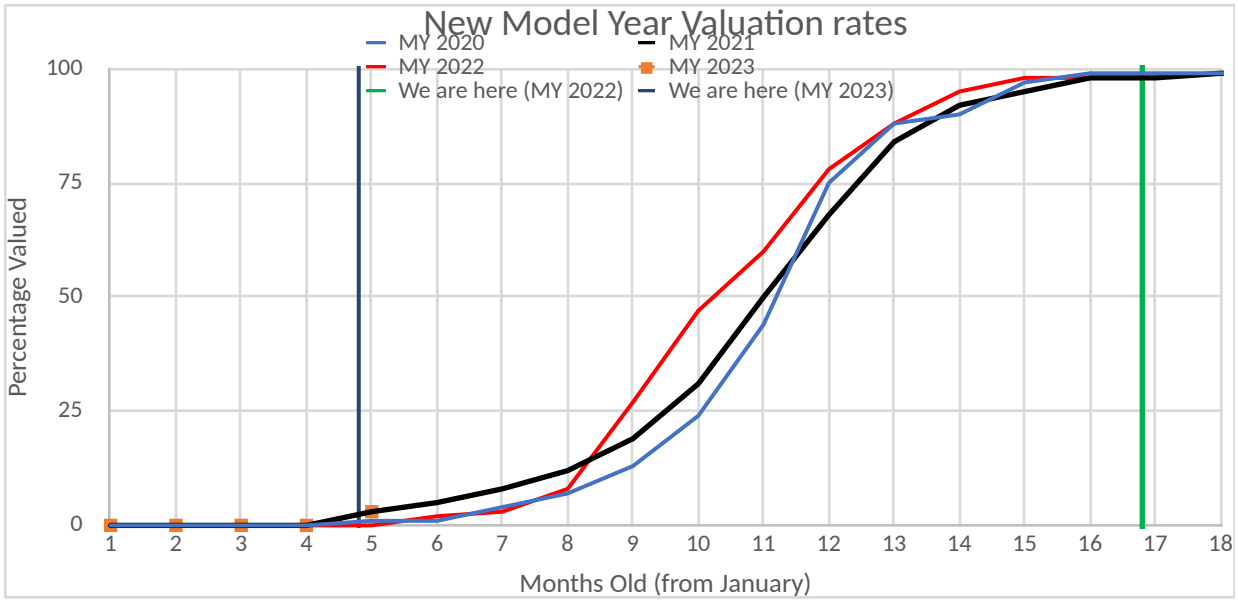
This is normal behavior, particularly for older vehicles. Our data does not list every accessory potentially available for a vehicle, only those which our analysts feel have a measurable impact on the used value of a vehicle. Additionally, accessory values depreciate more quickly over time than vehicle values do, and once an accessory is no longer value-impacting, it is removed from the data for the vehicle in question.

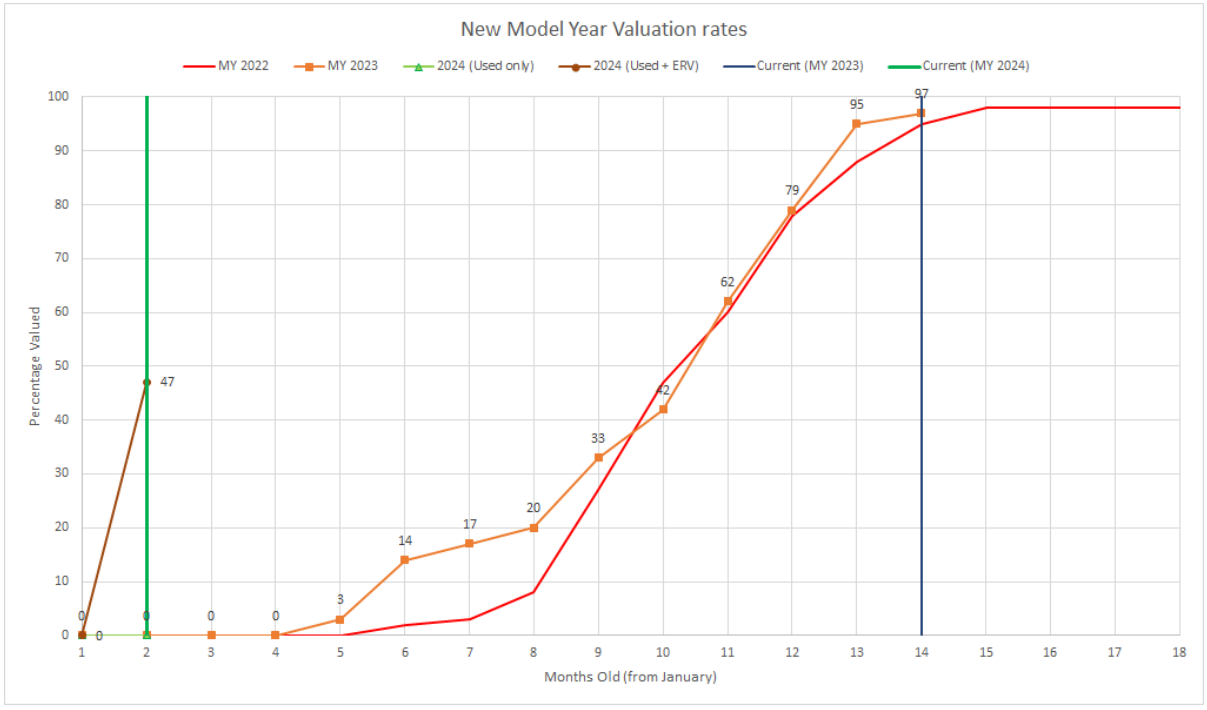
10.11.Why do I get a null or $0 value for newer vehicles?

Updated 2021-09-09: *Currently you will receive a NULL for base values for most endpoints other than vehicleAndValueByVehicleId, which returns $0 base values. Over the next several months, all vehicle value endpoints will be updated to return 0 (zero) rather than NULL base values.*

Updated 2024-02-01: Chart below now includes separate “Used” vs. “Used + Early Release Values” to include coverage from this new set of values.

Our valuation models depend in large part on resale transaction data rather than a simple depreciation formula. As such, vehicles will be listed in our data for some time before they have Used values available for them. The chart below shows recent (MY 2022-2024) “valued over time” data through February 2024:


New Model Year Valuation rates
MY 2020
MY 2021
MY 2022
MY 2023
We are here (MY 2022)
We are here (MY 2023)
Percentage Valued
Months Old (from January)


New Model Year Valuation rates
MY 2022
MY 2023
2024 (Used only)
2024 (Used + ERV)
Current (MY 2023)
Current (MY 2024)
Percentage Valued
Months Old (from January)

10.12. I'm on a per-valuation charge basis in my license. How can I tell which API calls will trigger a license charge?

Each endpoint definition above includes a line starting with "**License Decrement:**"; this identifies which endpoints can potentially impact a per-lookup license cost. Unless otherwise described on a specific endpoint, to trigger the license decrement, a non-$0 vehicle value (Used retail, trade, loan, or Weekly Auction Value) must be returned in the response.

As described briefly in the Design Notes above, there is a limited amount of de-duplication logic performed in our API usage reporting system, but you should provide for deduplication in your own application to be safe since the internal de-dupe process considers only (a) consecutive (per API Key) calls, (b) to the same endpoint, (c) with the same inputs, (d) within two minutes of the first call. So, if your application has multiple users, and user A values vehicle 1, then a second later user B values vehicle 2, if user A re-values vehicle 1 even within two minutes, it is not considered a duplicate because the valuation calls were not sequential.

You can store a 'last valued' date for each vehicle being valued in your system, and only actually call the valuation API if the new valuation date falls into a different valuation period (year/month for most endpoints, ISO week (Monday – Sunday) for Weekly Auction and Weekly Used Values endpoints; check the newestAvailablePeriod endpoint periodically to check for newer valuation data.

10.13. My application/business model involves redistributing the results of your API to a variable number of users. Is there an easier way to accurately report my usage other than making the same valuation call multiple times?

Yes, we've just (7/15/22) added a new endpoint to allow you to report your redistribution volume. The new endpoint, distributionCount, requires you to pass in the requestId from the API response of the initial valuation call, plus the number of times you redistributed the values from this initial valuation call.

In some cases, the cost for this type of usage is already factored into your licensing, but please check with your account manager to verify.

10.14. Early Release Values – what are they and how to identify them in a response (released for February 2024)

We're excited to announce an upcoming enhancement to our Used Car valuation products, slated for release with our February 2024 data publications: **Early Release Values**.

- **What are they?**
  - It has historically taken 12-18 months (or more) for newer model year vehicles to have sufficient resale transaction data available for our valuation experts to be able to generate reliable and consistent Used values.
  - Our new Early Values leverage our existing valuation expertise enhanced by our PIN new & used transactional data to get you used values you can rely on, only faster.
- **When do they start?**
  - Our Early Release Values are available in our REST API and Values Online b2b web application starting with February 2024's monthly updates.
- **What do I have to do to access them?**
  - **Nothing: you will automatically receive our Early Release Values when you request the "regular" monthly Used values from our API for a vehicle which has Early Release Values available in the selected valuation period.**
  - Cars and Light Trucks with Early Release Values will return Clean Retail and Average Trade values only.
    - Clean Trade, Rough Trade and Loan values will be returned as $0 (meaning "N/A" not a literal zero-dollar value.) This will be how you can identify an Early Release Values result vs. a 'regular' used result.
    - Mileage and Accessory adjustments will not be available/ applicable for these vehicles, so the Base and Adjusted values will be the same.

Value Types available by Vehicle Type (Feb 2024):

Vehicle Type	Clean Retail	Clean Trade	Average Trade	Rough Trade	Loan
Car/Light Truck (Early Release Values)	**X** *(ERV)*	n/a	**X** *(ERV)*	n/a	n/a
Car/Light Truck	**X**	**X**	**X**	**X**	**X**
Medium/Heavy Truck	**X** *(Avg Retail)*	**X** *(Avg Wholesale)*	n/a	n/a	**X**
Motorcycle	**X**	**X**	n/a	**X**	n/a

10.14.1.Is there a way to avoid getting the Early Release Values (2/9/24)

Available for testing (UAT API) on/about February 12, 2024, if you wish to receive $0 instead of the Early Release Values, add a new parameter **`&erv=0`** to the valuation calls. You will receive an additional warning message after the result[] array as below when this parameter overrode the Early Release Values.

https://cloud.jdpower.ai/data-api/UAT/valuationservices/valuation/defaultVehicleAndValuesByVin?period=0&vin=19UDE4G7*RA******&region=1&erv=0

```
        …
          }
    ],
    "userinfo": "Default",
    "earlyReleaseValueSuppressed": "Early Release Values overridden by user request.",
    "authId": "Your Company Here"
}
```

This parameter does not appear in the Swagger API definitions, and does not impact any of the non-valuation endpoints (i.e., mileage, accessories, etc.)

10.15.What value type names should I use in my application and/or reports to align with the Values Online b2b web tool?

Vehicle Type:	Car, Light Truck	Medium Truck, Heavy Truck	Motorcycle
API Value Type	**Used Car**	**Commercial Truck**	**Motorcycles**
Clean Retail	Clean Retail	Average Retail	Average Retail
Clean Trade	Clean Trade-In	Average Wholesale	Clean Trade-In/Wholesale
Average Trade	Average Trade-In	*n/a*	*n/a*
Rough Trade	Rough Trade-In	*n/a*	Rough Trade-In/Wholesale
Loan	Clean Loan	Loan	*n/a*
Base MSRP	Base MSRP	Base MSRP	Suggested List

10.16.Trim Reduction in the VIN Decode endpoints (3/28/24)

Our standard VIN Decoding endpoints – vehiclesByVin and vehiclesWithTrimByVin – are being updated to take advantage of our available Build data sources to greatly improve the percentage of VIN decode results that return a single vehicle ("Trim Reduction".) Once the update is completed, these endpoints will return a single vehicle about 9 times out of 10, compared to the current 2 out of 3. Results will vary by make and model year. When build data is available, a VIN which previously would have returned two, three or more vehicles can now return a single vehicle/trim, reducing vehicle selection errors and improving the overall valuation experience.

- VINs using any sort of wildcard/placeholder or any other character not valid in a US VIN will not be able to take advantage of the Trim Reduction since this process is based on VIN-specific Build data.
- VINs which would previously return multiple trim levels will see their response times increase by around 150 - 200ms due to additional upstream processing.
- The existing VIN valuation endpoints (xxxVehicleAndValuesByVin etc.) will continue to function as before, using their own specific internal business rules to return single vehicle selection/ valuations.
- VINs which decode to a single vehicle in our "standard" or pattern-based process will skip the extra "Build Data" check for performance purposes, so this option should not be used as a shortcut to the buildDataAvailableByVin endpoint.
    - Correspondingly, these VINs will always return "Standard" as the VIN Decode Source when using the optional/hidden VIN Source parameter.
- Users looking to get installed options from the build data, including installed option total adjustments, will still need to license and use the VIN Precision+ endpoints.
- Developers may, at their discretion, provide optional flags to these endpoints to force the previous behavior, or to add a VIN Source field to the response.
    - Default behavior will be to take advantage of Build data but not add the VIN Source field to the response, to minimize impact on the users' applications.
    - The new optional/hidden parameters are not listed in the API Swagger definitions to remove impact on existing applications.

10.17.VIN Precision+ Content Adjustment Update (Q4 2024 – Q1 2025)

When this update goes live in late 2024, we will automatically return the corresponding (Monthly, Weekly, or Combined) VIN Precision+ (/build/*) result for calls made to the legacy VIN Valuation endpoints when we have data for the specific input VIN. This will impact the returned vehicle and base values as well as "as-built" content adjustments for identified factory-installed optional equipment.

To assist with users retrieving the as-built Accessory information after VIN valuation, we have included the accessoryDataByVinAndVehicleId endpoint in the 2nd round of this update.

Legacy Endpoint	Build Response when available
valuation/defaultVehicleAndValuesByVin	build/buildVehicleAndValuesByVin
valuation/highVehicleAndValuesByVin	build/buildVehicleAndValuesByVin
valuation/lowVehicleAndValuesByVin	build/buildVehicleAndValuesByVin
valuation/msrpVehicleAndValuesByVin	build/buildVehicleAndValuesByVin
valuation/ weeklyDefaultVehicleAndUsedValuesByVin	build/ weeklyBuildVehicleAndValuesByVin
valuation/defaultVehicleAndUsedValuesByVin	build/ buildVehicleAndUsedValuesByVin
valuation/accessoryDataByVinAndVehicleId	build/buildAccessoryDataByVin

Note: The Batch Default VIN Valuation endpoint will not receive this update.

11/21/24 update - Based on user feedback, these endpoints require the use of a hidden/optional parameter (**`&vpp=1`**) to enable the update on a per-call basis. This parameter will also add the additional `vinsource` field to the response.

1/3/25 update – We've deployed the remaining enhanced endpoints to the UAT API today, including the Accessory Data endpoint. User testing will be available through 1/10/25 and production deployment is anticipated on 1/14/25.

3/7/25 update (UAT) – Added a new optional/hidden parameter (**`&vinsourcedetail=1`**) to return an indicator of the Build Data source when a VIN Precision+ response is returned.

The return values will be:

- OEM
- Verified Engineered
- UCG Legacy
- Pattern
    - Pattern indicates a non-build or "standard" result.

11.SOAP API method to REST Endpoint conversions

This section will provide additional details on converting from our legacy SOAP API (webservice.jdpowervalues.com) to the new REST API at a SOAP/legacy method level. If you are in a hurry and want to just 'lift and shift' without necessarily taking advantage of some of the new functionality, you can look up the SOAP methods you use below and link to the REST endpoint documentation.

11.1.SecureLogin service

11.1.1.getToken

There is no REST equivalent; the username/password/token approach is replaced with a static API Key

11.2.Vehicle service

11.2.1. ping

No direct equivalent; use the regions endpoint as a replacement "health check"

11.2.2. getYears

11.2.3. getMakes

11.2.4. getSeries

11.2.5. getBodyUids

Use years, makes, models, and bodies. Optionally, call vehicleList monthly and cache.

11.2.6. getVehicles

Use vehiclesByVin or vehiclesWithTrimByVin

11.2.7. getVehicle

No direct replacement. Consider vehicleAndValuesByVehicleId or vehicleInformationByVehicleID endpoints.

11.2.8. getVehicleByVic

No replacement; the VIC is a deprecated vehicle ID.

11.2.9. getAccessories

11.2.10. getInclusiveAccessories

11.2.11. getExclusiveAccessories

Replace all 3 with accessoryDataByVehicleId or accessoryDataByVinAndVehicleId

11.2.12. getBaseVehicleValueByUid

11.2.13. getVehicleValueByUid

Replace either with valueByVehicleId and use the base value fields

11.2.14. getVehicleAndValueByUid

Use vehicleAndValueByVehicleID

11.2.15. getDefaultVehicleAndValueByVin

11.2.16. getMsrpVehicleAndValueByVin

11.2.17. getHighVehicleAndValueByVin

11.2.18. getLowVehicleAndValueByVin

Use defaultVehicleAndValuesByVin, msrpVehicleAndValuesByVin, highVehicleAndValuesByVin or lowVehicleAndValuesByVin

11.2.19. getAuctionValues

Use weeklyAuctionValuesByVehicleId or weeklyAuctionValuesByVinAndVehicleId

11.2.20. getMileageAdj

Use mileageByVehicleId

11.2.21. getRegions

Use regions and/or regionIdByStateCode

11.2.22. getTotalAdjFloorValues

No direct replacement. The business rules values are supplied in the results from the various valuation endpoints.

12.Change control history

For more recent updates, please see Change Control section near top of this document.

2024-04-08 **API updates:** Deploying Trim Reduction to the Production API April 9, 2024.

2024-03-28 **API updates:**

Updated the existing vehiclesByVin and vehiclesWithTrimByVin endpoints to reduce the number of times you receive multiple vehicles back for VINs, when we have available "As Built" data for the specific VIN (Trim Reduction). The updated endpoints also allow you to (optionally) request a vinsource value in the response.

Documentation updates:

Both updated endpoints above offer optional/hidden parameters not exposed through the Swagger interface to provide additional functionality for edge case users without impacting users happy with the primary functionality. We've added a section on these hidden parameters to the Parameters documentation section. We've also added an FAQ on the new Trim Reduction and added references to this FAQ in the VIN Decode documentation sections.

2024-02-29 **API updates:**

Added two new VIN Precision+ endpoints: **buildAccessoryDataByVin**, replacing the current buildAccessoriesByVin and adding accessory logic data, and **buildVehicleAndUsedValuesByVin**, which combines the existing buildVehicleAndValuesByVin (monthly used values) and weeklyBuildVehicleAndValuesByVin (weekly used values.)

CRITICAL NOTE: with the release of the buildAccessoryDataByVin endpoint, both the previous buildAccessoriesByVin and accessoryDataByVehicleId endpoint are now considered deprecated, and no new development should reference them. All accessory calls should be migrated to accessoryDataByVehicleId and/or accessoryDataByVinAndVehicleId (non-build), or buildAccessoryDataByVin (build calls.)

Documentation updates:

Updated License Decrement notes from the build Accessory endpoints; the buildAccessory endpoints do NOT impact your license usage tracking.

2024-02-09 **API updates:**

Added a method for reverting **Early Release Values** in the valuation endpoints.
Updated the **vehicletype** parameter description in the Swagger file to match the valid input values.
Both updates available for testing on UAT February 12, 2024.

2024-02-01 **Documentation updates:**

Added note for display/reports that want to align with the value types displayed in our Values Online web application, including updates in the Values Definitions.
Updated 'values over time' chart for newer vehicle valuation rates.

2024-01-22 **Documentation updates:**
Early Release Values information added to FAQ and Definitions

2023-11-29 **Documentation updates:**
Updated the FAQ on 'which endpoint(s) impact our license usage charges?'

2023-07-31 **Documentation updates:**
Corrected the documentation bullet/numbering for error messages in sec. 8.1.5

2023-04-24 **Documentation updates:**
Updated the "new model year valuation rate" chart

2023-02-22 **API updates:**
Corrected or updated input validation error messages as below:
- defaultVehicleAndValuesByVin and defaultVehicleAndUsedValuesByVin: VIN not 17 chars: reverts to "The input VIN must be 17 characters long."
- years: Invalid period: Corrected to standard error message rather than text from Invalid Date Range.
- Weekly auction values endpoints, defaultVehicleAndUsedValuesByVin, valueByVehicleId, vehicleAndValueByVehicleId, weeklyUsedValuesByLegacyId: Corrected mileage input validation error message to new standard
- Most endpoints: re-enabled Date Range validation

Documentation updates:
Updated the mileage input validation error message text.
Added SOAP Method to REST Endpoint list for "lift and shift" moves

2023-01-01 **Documentation updates:**
Corrected the title/tag for the apiUsageSummary endpoint in the documentation (was "apiSummaryReport".)

2022-11-30

API updates:

Fixed a sorting issue with the vehiclesWithTrimByVin endpoint on UAT; it will return the default decode first as expected.

We will be deploying phase 3 of our data layer upgrade to the UAT API next Tuesday (12/6/2022). This will improve the performance and stability of the following endpoints:

- **vehiclesByVin**
- **vehiclesWithTrimByVin**
- **regions**
- **regionIdByStateCode**
    - *Please note that the last two endpoints' results are cached so the overall performance will not improve significantly.*

Expect both updates to be deployed to production on December 13th, 2022.

2022-11-09

API updates:

The legacy endpoints flagged as deprecated in the 2022-10-18 update, along with the accessoriesByVehicleId and accessoriesByVinAndVehicleId, will now return a warning message (via response.error field, outside the result[] array) indicating their deprecated status, but only on the UAT instance at this time, in order to nudge developers (using the UAT instance) to update to the newer endpoint(s).
The next set of data migration endpoints are also live on the UAT instance (Weekly Used Values and combined Used Values by Vehicle ID endpoints) and should be promoted to the production API in the next week or two.

2022-11-03

API updates:

A bug fix has been applied to the UAT instance for the Weekly Auction Values endpoints. This bug would cause false "no data available"/empty responses for Weekly Auction Values calls made with Period parameters of 2022-05-29 or 2022-10-30. This fix is planned to be deployed to the Production instance on Tuesday November 8th.

Documentation updates:

Corrected the note on Period=0, which was previously documented as using Eastern time. The correct time zone is UTC not Eastern.

2022-10-18 **API updates:**

The following legacy support endpoints are officially deprecated, may not return usable data after June 2023, and will be removed from the API in early Q3 2023 (UAT) and Q4 2023 (PROD).

- **masterVehiclesByLegacyId**
- **masterToLegacyVehicleMapping**
- **weeklyUsedValuesByLegacyId**
- **masterVehiclesByLegacyIdAndVin**
- **vehicleList_Legacy**

Documentation updates:

Moved the endpoints noted above into the Deprecated Endpoint section of the product documentation

2022-10-18 **API updates:**

The updates noted for 10/7/22 below were delayed until today, but the mileage bug noted in the 10-07 update below has been addressed for all impacted endpoints other than the build endpoint on the UAT API today rather than some now and the3 rest in a future update.

Planned deployment to Production API is next Tuesday, 10/25/2022.

2022-10-07 **API updates:**

A bug in the mileage adjustment calculation for very old data has been identified and a fix is rolling out in stages. The mileage adjustment is being incorrectly returned as $0 for 8 through 20 model year old cars and light trucks, with low mileage, for valuation dates in calendar years 2000 through 2004 only. A bug fix will be in place on the UAT API for the valueByVehicleId and vehicleAndValueByVehicleId endpoints by 10/11 and should be promoted to the Production API on 10/18/2022. We will then apply this fix for the remaining endpoints on UAT for 1-2 weeks' testing before applying the fix to the remaining endpoints on Production.

This bug affects the following endpoints:

- defaultVehicleAndValuesByVin
- highVehicleAndValuesByVin
- lowVehicleAndValuesByVin
- msrpVehicleAndValuesByVin
- valueByVehicleId
- vehicleAndValueByVehicleId
- /build/buildVehicleAndValuesByVin

The Customer-facing API Usage Summary endpoint is returning data on the UAT API again. This should be functional on the Production API on 10/18/2022.

The Customer-facing API Usage (detailed) Report endpoint is being limited to the most recent 3 months' activity. This limitation is live on the UAT instance starting 10/7/2022 and should be live on the production instance on 10/18/2022. See endpoint documentation for additional information.

2022-09-30

API updates:

Due to the change in the data layer for some endpoints (see 9/22 notes), we are treating any input mileage parameter value which exceeds 9,999,999 as 9,999,999. Previously most endpoints would return empty results if provided mileage of 10 million or higher. You will now receive the normal results with our maximum mileage adjustment range. If this occurs, you will also receive the "mileage adjustment warning" message. ***This update is live on both test/UAT (September 26) and PROD instance (September 27.)***

2022-09-22

API updates:

We are adding some additional input validation code to some endpoints and migrating three endpoints to a new data platform.
Validation updates will be applied to the **UAT API** on **9/27**, while the data platform updates have been on the UAT API since 9/20 and will be applied to the **production API** on **9/27**.
The endpoints affected are:

- Production API updates: migrate following endpoints to new data platform
    - valueByVehicleId
    - vehicleAndValueByVehicleId
    - vehicleInformationByVehicleId
- UAT API updates: additional input validation; please note that this will update the SWAGGER response schema for some endpoints by adding additional (optional) response fields between the result array and the userinfo field.
    - batchDefaultVehicleAndValuesByVin
        - Fail if input contains single-quote characters
    - weeklyDefaultVehicleAndUsedValuesByVin
        - Confirm valid region
        - Confirm valid period
        - Confirm valid date range
        - Confirm valid vehicle type
        - Confirm valid mileage
        - Empty results

- weeklyVehicleAndValueByVehicleId and weeklyUsedValuesByVehicleIdweeklyUsedValuesByLegacyId
  - Confirm valid region
  - Confirm valid period
  - Confirm valid date range
  - Confirm valid vehicle type
  - Confirm valid mileage
  - Confirm valid ucgvehicleid
  - Empty results
- weeklyUsedValuesByLegacyId
  - Confirm valid region
  - Confirm valid period
  - Confirm valid date range
  - Confirm valid vehicle type
  - Confirm valid mileage
  - Empty results
- usedValuesByVehicleId, usedValuesAndVehicleByVehicleId, and defaultVehicleAndUsedValuesByVin
  - Confirm valid date range

Documentation updates:
Updated the New Model Year Valuation graph in FAQ

2022-07-14

API updates:
Added endpoint to allow for self-reporting valuation redistribution volume

Documentation updates:
Added FAQ section discussing valuation redistribution

2022-06-23

API updates:
Updated functionality on apiUsageSummary endpoint, limited to most recent 12 months' report data.

Documentation updates:
Added endpoint documentation for apiUsageSummary endpoint.

2022-06-22

API updates:
Hotfix to both UAT and Prod APIs to temporarily disable functionality of the apiUsageSummary endpoint.

2022-04-13

API updates:
Hotfix to both UAT and Prod APIs to remove inadvertent case-sensitivity in recent regionIdByStateCode update

2022-04-12 **API updates: Live on Production April 13, 2022:**

vehiclesWithTrimByVin release to Production

regionIdByStateCode invalid input warning release to Production

Performance optimizations to following endpoints:

- valueByVehicleId
- vehicleAndValueByVehicleId
- defaultVehicleAndValuesByVin
- highVehicleAndValuesByVin
- lowVehicleAndValuesByVin
- msrpVehicleAndValuesByVin
- accessoriesByVinAndVehicleId
- vehicleInformationByVehicleId
- regions

2022-04-01 **Documentation updates:**

Fixed typo in Rate Limit header section of document.

2022-03-14 **API updates:**

New endpoint: vehiclesWithTrimByVin; adds Trim and Trim2 fields to the output of the existing vehiclesByVin endpoint. Available immediate on UAT, planned to be available on Prod with April 8, 2022 update.

Documentation updates:

Updated the empty due to vehicletype warning section to include the response field name and text.

Added VID response field to the vehiclesByVin Shared VIN example results.

2022-03-10 **API updates:**

The following API updates are available immediately or by 3/14/22 on the UAT API, and are planned for Production release on 4/8/22:

New input validation for regionIdByStateCode endpoint including **error** response.

Added warning message to several endpoints when vehicletype parameter filtered out potential response data. See the list of endpoints impacted in the new warning message section below.

Performance optimizations on multiple endpoints; no input/output changes.

Documentation updates:

Removed invalid Developer note (VehicleType filter) and added new failure/error response for **regionIdByStateCode endpoint**

Split "empty due to vehicletype" warning message" to its own section and provided list of endpoints this currently applies to.

Updated the per-endpoint input validation errors to include HTTP 400 status and new State Code validation message.

2022-01-21

API updates:
Added new Combined Monthly + Weekly Used endpoints (UAT API)
Moved the Historical Values endpoint out of the premium path; this endpoint is now a "generally available" endpoint.

Documentation updates:
Added section for new Combined Monthly + Weekly Used endpoints
Added new "empty results" warning message examples.

2022-01-19

Documentation updates:
Added additional notes in the Weekly Auction Values section as well as WAV definitions.

2022-01-13

Documentation updates:
Added Used and Weekly Auction Values Value Types and Conditions to the Definitions section
Made several minor updates to the Exceptions and Error Codes section
Fixed several typos throughout.

2021-12-16

API updates:
Added new Weekly Auction Values Mileage endpoint on the UAT* API instance to support developers who modify the input mileage value after the initial WAV valuation call has been completed. ***Promoted to Prod API 12/30/2021**

2021-12-10

API updates:
Updated the Weekly Auction Values endpoints on the UAT* API instance to make the upcoming (April 2022) change in WAV mileage calculations available, including an optional parameter to use the new mileage approach early.
***Promoted to Prod API 12/30/2021**

Documentation updates:
Updated the Weekly Auction Values endpoints documentation including notes on the optional "start early" parameter added (on the UAT instance only at this time).
Fixed the section numbering of the Design Note regarding avoiding charges for duplicate lookups and added a link from the Design Note to the FAQ on the same topic.
Removed references to former /valuationservices/build* endpoint paths

2021-10-11

API updates:

New Historical Used Values Trend endpoint has been added to the UAT API. This endpoint is a Premium endpoint which may require additional licensing terms and/or increased cost to use.

New endpoint for checking API performance metrics (latency) is available on the UAT API.

Documentation updates:

Updated the list of license-impacting endpoints in FAQ

Corrected the endpoint URL text for the weeklyDefaultVehicleAndUsedValuesByVin endpoint (replaced BuildApiTest with correct API root of valuationservices)

2021-09-09

API updates:

Performance optimizations will be applied to the vehicleAndValueByVehicleId and vinCheckSum endpoints. The changes will be transparent to users.

The updates noted in the 8/18 update below for the batchDefaultVehicleAndValuesByVin endpoint will be promoted to the Production API.

Documentation updates:

Updated the FAQ on unvalued vehicles to reflect the change noted in the Developer Notes below.

Developer Notes on upcoming changes:

Currently several endpoints return NULL for base values when a vehicle is unvalued or is part of a product which doesn't contain all value types (Commercial Trucks, Motorcycles). Over the upcoming months, these endpoints will be updated to return 0 in place of these NULLs.

We will communicate the specific endpoint updates as the code is updated, but please take this time to start verifying that your code will work identically for a NULL or 0 (zero) dollar base value response.

2021-09-03

Documentation updates:

Updated the listed DB maintenance window time in an FAQ

2021-08-18

API updates:

Added Swagger Response Schema to additional endpoints (phase 4) on UAT

Documentation updates:

Added FAQs on empty accessory results and unvalued vehicles.

Added a note in the Design Notes covering License Lookups and DeDuplication.

Added an FAQ discussing License Lookups/charges

Added note to vehiclesByVin endpoint clarifying its License Decrement status

Added additional Developer Notes to the batch VIN value endpoint: (a) will now return an error for empty input, (b) please don't use for less than 50-100 VINs at a time. This second may be enforced at the API level in the future if needed.

Replaced [base_url] and [url_base] in endpoint definitions with the production URL path.

2021-08-06 **Updated Batch VIN Value endpoint** to return an error if an empty input is provided.
Added the optional Error field to most endpoints' Swagger Response Schema.

2021-07-15 **Added new Legacy Vehicle List endpoint**
Added note to the Legacy endpoint section – the Legacy Support endpoints will be removed from the API after all current legacy customers have converted away from our legacy data/API offerings.

2021-06-24 **Added VIN Decoding usage and program flow** to the Design Notes section
Cleaned up some subsection misnumbering
Updated TLS levels supported – now TLS 1.0 through TLS 1.3

2021-06-15 **Added Response Schema in swagger.json for additional set of endpoints**

2021-05-28 **Promoting UAT version to Production to deploy the updated swagger.json info (response schema for some endpoints).**

2021-05-11 **Corrections made to the JSON Response Schemas recently added to the UAT API's swagger.json file for the Weekly Used values endpoints; the original schemas used the monthly value field names rather than the weekly value fields.** Also added additional endpoints to list.

2021-05-10 **Additional JSON Response Schemas added to swagger.json file on UAT; updated list of endpoints with defined response schema is here.**
This version is planned to be promoted to Production on/about May 15th, 2021. Be sure to contact us immediately if modifying the swagger.json will disrupt your production application.

2021-04-25 **Added JSON Response Schema to swagger.json file on (UAT deployment) for endpoints noted here.**

2021-04-22 **Added some technical considerations when accessing VIN Precision+ endpoints**

2021-03-18 **Clarified "required for VAR users" for userinfo endpoint parameter.**

2021-03-09 **Added new status endpoints (by ID and by VIN) to UAT instance to check first/ last period available, first valued, etc.**
Performance optimizations on the Weekly Auction Values endpoints.

These updates are planned were promoted to the Production API instance on Monday, 3/15/2021.

2021-03-07 **The non-premium Build endpoints (/valuation/build*) were removed from the production API today.**
The Consolidated Accessory endpoints added to the UAT API in February were promoted to the production API today.

2021-02-24 **Non-Premium Build endpoints (/valuation/build*) will be removed from the Production API instance the weekend of March 5th – 7th, 2021.** These endpoints were removed from the UAT instance in January.

The new consolidated Accessory Data by ID and by VIN endpoints released to UAT on 2/12 and 2/17 will be promoted to the production API instance at the same time.

2021-02-17

Added new consolidated Accessory Data by VIN endpoint to the UAT instance. This is like the "by Vehicle ID" endpoint added on 2/12.
Corrected the example URLs in the new consolidated data endpoint documentation and added note re: deprecation of current endpoints.
Moved the accessoriesByVehicleId and accessoriesByVinAndVehicleId documentation to the Deprecated Endpoints section of this documentation.
Plan on these endpoints being removed from the UAT deployment in March and from the Production deployment in the 2nd half of 2021.

2021-02-12

Added new consolidated Accessory Data endpoint to the UAT instance which combines the functionality of the accessoriesByVehicleId and accessoryLogicByVehicleId endpoints.
Expect this to be promoted to production in the end of February 2021.
Expect the old endpoints to be retired later in 2021 (Q2 on UAT and 2nd half of the year for the Production API instance).
Added definitions for Package Inclusive and Mutual Exclusive logic
Updated the notes in the existing accessoryLogicByVehicleId endpoint

2021-02-05

Added notes in the Weekly Used and "regular" Used valuation endpoint documentation to help differentiate monthly vs. weekly endpoint usage.
Added note re: Period=0; this uses the server's clock to determine the date. Assume US Eastern time zone.

2021-01-03

Removed /valuation/build* endpoints from UAT deployment. No other changes to this documentation.
The build endpoints in the "main" /valuation/ path will be removed from the Production API over the weekend of January 15-17, 2021.

2020-12-23

Split section 5, formerly Service Methods and Design Notes into two sections, moving the Service Methods into section 6.

The build endpoints in the "main" /valuation/ path will be removed from the Production API over the weekend of January 15-17, 2021.
They will be removed from the UAT API instance over the weekend of January 1-3.

2020-11-19

The URL changes for premium Build endpoints noted in the 2020-10-28 update are being reverted. The Premium Build endpoints will instead continue to reside under the `/build/` path instead of the `/vpplus/` path.
As a reminder, the build endpoints in the "main" /valuation/ path will be removed on or shortly after January 1, 2021.
"General" access to the premium `/build/` endpoints has been removed; if you need access to the `/build/` endpoints moving forward, please reach out to your Sales representative or UCG_Sales@jdpa.com email distribution list.

Updated the example cUrl code for the batchDefaultVehicleAndValuesByVin endpoint to use the `--data-urlencode` switch for clarity.

2020-10-28

The URLs for the Build endpoints are being updated. The Premium path noted for the Build endpoints in last month's release notes has been changed from `/valuationservices/build/` to `/valuationservices/vpplus/` to align with the new system's marketing designation of VIN Precision+

The Weekly Build endpoint (weeklyBuildVehicleAndValuesByVin) will be updated to VIN Precision+ technology and moved from the `/build/` to the new `/vpplus/` path in the next week or two.

The previous Build endpoints (`/valuationservices/valuation/build*` and `/valuationservices/build/build*`) will be removed by December 31st, 2020, and only the (premium/restricted) endpoints will be available at that time.

General availability access to the existing /build/* endpoints will be removed by November 2, 2020. Contact your account representative for access

2020-10-06

Corrected documentation error in bodies endpoint cUrl example

2020-09-04

The URLs for the Build endpoints are being updated. The Build endpoints are being moved from the main `/valuation/` path to a new `/build/` path. Initially these endpoints will be available through both paths, but they will be removed from the `/valuation/` path.

The UAT instance will have the `/valuation/build*` endpoints removed in midSeptember, and the Production instance will have them removed shortly thereafter, so all calls to the build endpoints will need to use the new `/build/` path URLs at that point.

The build endpoints fall into the first group of Premium endpoints whose functionality falls outside the primary set of endpoints, and access to these endpoints may require additional license terms and/or costs.

Moved the Weekly Used Values Build endpoint documentation from the Weekly Used Values group to the Build endpoint group in the documentation.

Updated the License Decrement notes for the Weekly Used Values endpoints. These endpoints are now classified as Charged endpoints with the exception that they count as a "duplicate" for charging purposes when called immediately before or after the corresponding Monthly endpoints.

Correct the numbering of several of the Error/Warning Message details in the documentation.

Corrected the response example in the Weekly Auction Values endpoints documentation. Replaced (incorrect) JSON array `":[{"acccode": "2A6"}, {"acccode": "081"}]` with (correct) comma-delimited text array `"294,081"`

Corrected the vehicleAndValueByVehicleId endpoint definition documentation originally copy/pasted from the valueByVehicleId endpoint (URL naming update).

2020-08-24

Removed unused Legacy Build endpoint from Production API. See Deprecated Endpoint section at end of documentation for details

2020-08-12

Updated Definitions section for VehicleType parameter – the definition section incorrectly listed the `Motorcycle` option as `Motorcycles`

2020-08-04

Motorcycle values will be added to the various valuation endpoints starting with the UAT instance this weekend; work will start this Friday 8/7 at 7 PM EDT. Barring any issues after this update, the motorcycle values data will be pushed to the Production API the following weekend (8/14 – 8/16).

Added Design Notes section to documentation to cover general design considerations such as the API Query Per Second throttle added in April 2020.

A repeat warning on the pending upcoming change to the Charged or noncharged (license cost) status of the new Weekly Used Values endpoints. See note in that section for details.

2020-07-22

Promoted last round of UAT changes to PROD instance; see notes below for 20200626 and 2020-06-01 for details. Highlights include Motorcycle and NoMotorcycle vehicletype filters and enhanced error checking and warnings.

Removed Legacy Build Endpoint from UAT instance; will be removed from the Production instance at the end of August 2020. This endpoint has seen virtually no usage.

Added Deprecated Endpoint section at end of documentation to provide visibility into deprecated functions.

2020-06-26

Added additional legacy support endpoint to the UAT instance based on the recent legacy vehicle endpoints' FAQ, which has also been updated**.** This new endpoint combines the legacy vehicle ID and the VIN to get to a single master vehicle in about 97% of the cases tested to date, still provides the additional possible matches for review, and includes batch support for up to 1,000 records per call.

Added developer notes on the two batch endpoints regarding batch use and requesting keeping 1 – 2 parallel/simultaneous batch calls.

2020-06-17

Applied patch to production and UAT instances to resolve uncommon duplicate data in results for accessoriesByVinAndVehicleId endpoint.

2020-06-01

Upcoming change to the Charged or non-charged (license cost) status of the new Weekly Used Values endpoints. See note in that section for details.

The following changes are live on the test/UAT instance as of 6/1/2020. They are planned to be promoted to the production API instance the weekend of June 13-15, pending user feedback.

Additional Motorcycle enhancements:

- Added vehicletype parameter values of Motorcycle, NoMotorcycle (or NoMotorcycles) to remaining applicable endpoints (all but Mileage, Build, Weekly Auction and Weekly Used endpoints.)
- Vehicle Lookup and Description endpoints (except Vehicle List and VehicleInformation endpoints) already include Motorcycle data
- Valuation endpoints (except Build and Weekly as noted above) will begin to return Motorcycle data in the next update around mid- or late June.
    - ***As this will be a data-only update, not an API/implementation update, there will not be the usual 1- or 2-week delay between UAT and Production rollout of this enhancement.***
- **To not receive motorcycle data**, replace the empty (optional) `vehicletype` input parameter with a value of `NoMotorcycles` to receive both Used Car and Commercial Truck data, or use UsedCar if you only want cars/light trucks.

Added VID field to results of four additional endpoints: vehicleList, vehicleInformationByVehicleId, valueByVehicleId and vehicleAndValueByVehicleId.

Added additional business rule field minAuctionValues to both Weekly Auction Values endpoints.

Added additional error checking and warnings; these can potentially add a new field to the response outside the `results[]` array.
Please note: The missing parameter and parameter type mismatch errors are platform-defined and are already live in the Production API at this time.

2020-05-26 **Updated Developers Notes for weeklyAuctionValuesByVinAndVehicleId endpoint**

2020-04-23 **Added additional Weekly Used Values endpoints:**

Weekly Used Build endpoint that provides the Weekly Used Values corresponding to the buildVehicleAndValuesByVin endpoint

Weekly Used Values by Legacy ID endpoint that provides the Weekly Used Values based on the legacy UID or VID vehicle keys; this endpoint is intended for legacy/transition support only.

2020-04-15 **Updated FAQ on IP addresses**

2020-04-13 **New endpoints to support the new Weekly Used Values are available on both the Production and UAT API instances**

Updated Value Definitions section to include new Weekly values

Removed the trailing slash in the endpoints' URL documentation throughout

2020-04-09 **New endpoints for our new Weekly Used Values will be coming next week; plan is for them to be available for testing early in the week of April 13th.**

API Query Per Second (QPS) throttling has been enabled; please see the Exceptions and Error Codes section for details.

Two additional stability enhancements to the API platform this past weekend impacted non-standard URL usage.

1. The API endpoint URL handler is now fully case-sensitive, i.e., a call using vehiclesByVIN will now fail where it used to work previously; please use vehiclesByVin, etc. as documented via the Online Documentation Page
2. The API handler allowed a trailing slash between the endpoint URL and the parameter querystring, i.e., …/years/?period=0 would work previously where the expected URL is …/years?period=0.
    a. This second change was reverted at this time as it has smaller performance/stability impact on the API handler but may be re-enabled if needed in the future.

Added an additional FAQ regarding the legacy support endpoint masterVehiclesByLegacyId covering options on how to select when it returns more than one vehicle.

Online Documentation URL has been updated; the original URLs will redirect to the new ones. The documentation page UI will be updated back to the old look and feel shortly after the 2020-04-05 update changed it.

2020-04-03 **Corrected paragraph numbering for Notes on Parameters from 5.4.6.1 to 5.5.6.1**

2020-03-26 **Updated wording on VehicleType optional input parameter from "both data sets" to "multiple vehicle type data sets" to address addition of Motorcycle data**

2020-03-13 **Added FAQ on License Decrement functionality**

2019-12-18 Corrected another error in the minAdjRetailForLoan field description (in the defaultVehicleAndValuesByVin endpoint; other endpoints reference this description). It now reads that "if the Adjusted Retail is reduced below this value, the Adjusted Loan should be set to $0 regardless of any other factors." instead of that "if the Adjusted Loan is reduced below this value, the Adjusted Retail should be set to $0 regardless of any other factors."

Also added the field usage/business logic notes to the field definition section.

2020-03-07 **Promoted following endpoints with motorcycle enhancements from UAT to production:**

- vehiclesByVin
- years
- makes
- models
- bodies

The valuation endpoints (below) will be updated for the motorcycle data in a near-term future update

- valuebyVehicleId
- vehicleAndValueByVehicleId
- defaultVehicleAndValuesByVin
- lowVehicleAndValuesByVin
- highVehicleAndValuesByVin

2019-12-10	Updated all cUrl sample code from HTTP to HTTPS (see 2019-12-05 note below)
	Added warning message if you call the legacyBuildVehicleAndIdsByVin endpoint with a period parameter before the oldest available data (2019-06-01)
	Resolved "empty result" issue in lowVehicleAndValuesByVin and highVehicleAndValuesByVin when all trims for the VIN are unvalued.
	Replace single space (" ") with empty string ("") in masterVehiclesByLegacyId rollupaccessories field when no data is in this field. Also updated the query to return the "islegacyvehicle" first then sort the remaining vehicles by UcgVehicleId.
	Additional legacy support endpoint added: masterToLegacyVehicleMapping. This new endpoint is the logical opposite of the existing legacy endpoint masterVehiclesByLegacyId.
	Added FAQ discussing the two main legacy support endpoints and when to use either one.
2019-12-05	Updated FAQ for "One of the preprocessor(s) failed" error message to include scenario where test users try to send request parameters in request Body rather than URI querystring.
	Updated to remove HTTP (vs. HTTPS) references as API platform now requires HTTPS via TLS 1.1 and TLS 1.2 (no SSL-SSLv3 support) and added to FAQs
2019-12-02	Added FAQ for "One of the preprocessor(s) failed" error message.
	Added VID, UID, acceptablemileage and ratepermile to the field definitions.
2019-11-14	Corrected typo in field definitions and defaultVehicleAndValuesByVin – replaced "minadjloanforretail" (incorrect) with "minadjretailforloan" (correct).
2019-11-08	Added the current IP addresses in use in this API to the FAQ section.

2019-11-05 The following enhancements planned for the weekend of 11/08-11/10 (UAT); these are expected to be promoted to the production instance by 11/17. These enhancements are adding data field(s) to the replies of several endpoints.

- Adding mileage calculation fields to Weekly Auction Values endpoints to assist with Data Files user conversion
    - acceptablemileage (intercept from Data Files product)
    - ratepermile (slope from Data Files product)
    - mileageadjust (already factored into the reported values)
- Adding legacy vehicle ID VID to following six endpoints to better support integration of the REST API with our AuctionNet product, which also uses the same Master Vehicle Content and uses the VID vehicle ID.
    - bodies
    - vehiclesByVin
    - defaultVehicleAndValuesByVin
    - lowVehicleAndValuesByVin
    - highVehicleAndValuesByVin
    - msrpVehicleAndValuesByVin

  Added documentation notes about special scenarios which can be encountered in the msrpVehicleAndValuesByVin endpoint which can prevent certain vehicles from being returned by that endpoint.

2019-11-01 Enhancements listed below for 2019-10-16 are live in the Production instance as of 11/1.
In addition, case sensitivity for the following input parameters has been removed (UAT instance on 2019-10-18, PROD instance on 11/1.)

- vehicletype (most endpoints)
- make (models, bodies endpoints)
- model (bodies endpoint)
- statecode (regionIdByStateCode endpoint only)

2019-10-16 API enhancements being deployed to UAT instance on 10/18; these should be live in the Production instance by 10/25

- accessoriesByVehicleId and accessoriesByVinAndVehicleId are getting an additional field added per accessory row. This new field – accessorycategory – represents a general grouping of the accessory in question.
  The same field will be added to the buildAccessoriesByVin endpoint in a future update.

- A warning message will be added to the masterVehiclesByLegacyId and legacyBuildVehicleAndIdsByVin endpoints if the input period is older than 20150801
- The results in the vehiclesByVin are returned in a more consistent sort order (default vehicle first, then alternates in ascending UcgVehicleId order)
- Fields for gvw and gcw (Gross Vehicle Weight and Gross Combined Weight) to the below endpoints, immediately after the curbweight field. In general, these fields will be empty for non-commercial vehicles, and either or both will often be empty even for commercial trucks.
  Impacted endpoints:
  - vehicleAndValueByVehicleId
  - defaultVehicleAndValuesByVehicleId
  - lowVehicleAndValuesByVehicleId
  - highVehicleAndValuesByVehicleId
  - msrpVehicleAndValuesByVehicleId

  These fields were also previously available in the vehicleInformationByVehicleId endpoint.

2019-10-10 Added new batch valuation endpoint to UAT instance

Added minor clarification to masterVehiclesByLegacyId to cover scenarios where there is a vehicle with no "rollupaccessories" because the corresponding options are no longer valued for the year/vehicle in question but the master vehicle still exists., and added legacy vehicle description to better show the legacy vs. master vehicle definitions.

2019-10-02 Added note about new XML parent tag added to XML responses
Moved XML vs JSON into separate Table of Contents section

2019-09-30 swagger.json link now available via online documentation page

2019-09-10 Next round of enhancements (releasing to UAT by 2019-09-13)
+ added Weekly Auction Values date range to newestAvailablePeriod endpoint
+ new endpoint: vinCheckSum
+ new endpoint: buildLegacyVehicleAndIdsByVin

Moved the Legacy Support endpoints into a separate Legacy Support Endpoint section in the documentation.

2019-09-09 Corrected period parameter definition in most endpoints from int to string

2019-09-04 Started FAQ section at end; invalid api-key and value field mapping added to FAQ.

2019-09-03 Added period parameter to **`masterVehiclesByLegacyId`** endpoint documentation, updated sample code.**2019-08-30**

	Added **`apiUsageReport`** and **`vehicleList`** endpoints. Minor edits, including API-Key header examples for cUrl and C#. Added note in parameters section about case sensitivity of parameter names and values. Added note that an **upcoming** platform enhancement will provide an OpenAPIcompatible swagger.json definition file for the API to the Docs page section. *(now available as of 2019-09-30)*
2019-08-20	Added note on HTTPS recommendation
2019-08-19	Corrected endpoint URL in **`vehicleValueByVehicleId`** example cURL text
2019-08-16	Adding optional **userinfo** input parameter to standard field definitions Added reference to HTTP header used to return XML instead of JSON Added reference to URL-encoding parameters containing special characters Removed **period**, **vehicletype** from **`region`** and **`regionIdByStateCode`** endpoint input parameter lists in documentation. Minor reformatting throughout
2019-08-15	Updated response object samples to align with actual API results 'case' of the field names (i.e., **regionid** vs. **regionId** or **regionID**). Corrected "`results`" to "`result`" in the return example data to match actual response from the API.
2019-08-09	Added Current Period **`(period=0)`** support to most endpoints (all but /build endpoints at this time) Added **authId** response field definition to standard field definitions
2019-07-26	Added **`masterVehiclesByLegacyId`** endpoint documentation for legacy vehicle mapping
2019-04-26	Updated Weekly Auction Values endpoints to indicate that they will decrement the user's license. Added online documentation URLs.
2019-04-11	Added Build endpoints; corrected field list/order for two methods, fixed most lowercase field names, minor formatting corrections
2019-02-25	Initial external-use draft

13.Deprecated endpoints

13.1.1.Legacy Build endpoint (buildLegacyVehicleAndIdsByVin)

- **Deprecated as of 2020-07-26**
- Removal from Production API occurred with the 2020-08-24 update

This endpoint provides access to UCG's new VIN Configuration/Build data for users who have not completed migrating off UCG's legacy data or API platforms. Unlike the primary build endpoint above (buildVehicleAndValuesByVin) this endpoint does not provide valuation data; instead it returns the corresponding legacy Vehicle description, legacy vehicle IDs (VIC, VID, UID), a small subset of the vehicle attributes, and the list of Build-indicated accessories.

For performance purposes, this endpoint returns IDs and descriptions only. You will need to tie this data to your local copy of the legacy data/API for valuation purposes.

License decrement: As with the other Build endpoints, this method **does** decrement the user's license even though no values are returned.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/legacyBuildVehicleAndIdsByVin/

Parameters:

• period	string	required	Valuation period as YYYY-MM-DD
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN to be looked up
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/buildLegacyVehicleAndIdsByVin?period=2019-09-01&vin=1FMCU9J93JUB98016' \
  -H 'api-key: 3f3090a0-0a7b-40a7-9998-39d3efe40d63'
```


```
{
 "requestId": "nsb3i14avjj2olwojw0zc",
 "result": [
     {
         "modelyear": "2018",
         "make": "Ford Truck",
         "series": "Escape",
         "body": "Utility 4D Titanium EcoBoost 4WD",
         "vid": "93603",
         "vic": "34201862YQ",
         "uid": "1193603",
         "vehicletype": "Light Truck",
         "mileageclass": "2",
         "basemsrp": "33395",
         "curbweight": "3645",
         "ucgsubsegment": "Compact Utility",
         "modelnumber": "U9J",
         "rollupvehicleid": "201834687",
```

```
            "vac": "0C7,081,1AP,076,0LE,0D0,0D9,346",
            "description": "Blind Spot Monitor,Aluminum/Alloy Wheels,Adaptive Cruise
             Control,Leather Seats,Collision Avoidance System,Remote Engine
      Starter,Parking
             Aid,Navigation System"
        }
    ],
    "userinfo": "UCG UAT",
    "authId": "UCG Testing"
}
```


Developers note: Unlike the standard Build endpoints, this endpoint does not utilize a 3rd party API to provide data when UCG's internal VIN Configuration data does not exist, so you can call the legacy endpoint directly without calling buildDataAvailableByVin to avoid "false positives"; simply have your application ready for potential empty results for the ~35% (as of Sept 2019) of VINs which do not have data in our system at this time.

Additionally, this data is only available for period values back to June 1, 2019 (period=2010-06-01.)

13.1.2.accessoriesByVehicleId

- **Deprecated as of 2021-02-17**; replaced by accessoryDataByVehicleId endpoint
- Scheduled to be removed from UAT API in March 2021
- *Estimated* removal from Production API in 2nd half of 2021

Developer note: This endpoint should be considered deprecated for development purposes as of mid-February 2021; please start using the new consolidated accessory data endpoint above.

Request the list of available accessories for the input UcgVehicleId. It would called be after selecting the vehicle by description (years/makes/models/bodies calls above), or when the vehicle has already been identified in a prior period or other application.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/accessoriesByVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• ucgvehicleid	int	required	Vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions method)
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces the getAccessories() method in the legacy/SOAP API.

Developer note: The vehicletype field is provided here for legacy support purposes only, as the vehicle ID will not exist in multiple vehicle type data sets.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
accessoriesByVehicleId?period=2018-01-01&ucgvehicleid= 201834687&region=6' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
   {
        "accdesc": "Navigation System",
        "acccode": "346",
        "tradein": null,
        "retail": null,
        "loan": null,
        "isincluded": 1,
        "isadded": 0,
        "accessorycategory": "Electronics"
    },
    {
        "accdesc": "Leather Seats",
        "acccode": "076",
        "tradein": null,
        "retail": null,
        "loan": null,
        "isincluded": 1,
        "isadded": 0,
        "accessorycategory": "General"
    }
],
   , "userinfo": "Default"
   , "authId": "UCG"
}
```


Business Rules and notes on fields.

- ***acccode:*** Accessory ID (UCG VAC) used to identify each accessory when using the Mutual Exclude and Package Inclusive logic (see accessoryLogicByVehicleId method below.)
- ***tradein, retail, loan:*** Accessory adjustment values for each value type. TradeIn applies to all Trade value types (Clean, Average and Rough)
- ***isincluded***: If 1, this accessory's value is already included in the vehicle's Base values. This accessory should be displayed as selected to your user but can be (temporarily) deselected by Mutual Exclusive logic (see accessoryLogicByVehicleId method below) Included options always have $0 values.
- ***isadded:*** If 1, this accessory is pre-selected based on the inputs. Included in this return object for compatibility with the VIN accessory methods, below. This is primarily used in the accessoriesByVinAndVehicleId and buildAccessoriesByVin methods.

Developer note: In general, you should treat any accessories flagged as `isincluded` or `isadded` as if the user had selected them, i.e., check for any "child" accessories in the Includes and Excludes fields and process them as if the user had selected the `isincluded` accessory.

As noted above, any 'parent' `isincluded` accessories will not have any **`Excludes`** to check for.

13.1.3.accessoriesByVinAndVehicleId

- **Deprecated as of 2021-02-17**; replaced by accessoryDataByVinAndVehicleId endpoint
- Scheduled to be removed from UAT API in March 2021
- *Estimated* removal from Production API in 2nd half of 2021

Developer note: This endpoint should be considered deprecated for development purposes as of mid-February 2021; please start using the new consolidated accessory data endpoint above.

Request the list of available accessories for the input VIN and Vehicle ID. It would be used either after the one of the VIN decode methods above.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/accessoriesByVinAndVehicleId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• vin	string	required	VIN for vehicle being valued
• ucgvehicleid	int	required	Vehicle ID to be valued
• region	int	required	Region ID (see Definitions and Regions)
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This replaces the getAccessories() method in the legacy/SOAP API when supplying the (SOAP-system's) optional VIN input parameter.

Developer note: In general, you should treat any accessories flagged as `isincluded` or `isadded` as if the user had selected them, i.e., check for any "child" accessories in the Includes and Excludes fields and process them as if the user had selected the `isincluded` accessory.
As noted above, any 'parent' `isincluded` accessories will not have any **`Excludes`** to check for.

Developer note: The vehicletype field is provided here for legacy support purposes only, as the vehicle ID will not exist in multiple vehicle type data sets.

Developer note: Both VIN and Vehicle ID are required as the VIN does not always decode to a unique Vehicle ID, and the list of options, their values, and Included status may vary as well.

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
accessoryByVinAndVehicleId?period=2018-01-01&vin= 1FMCU9J93JUB98016&ucgvehicleid=
201834687&region=1' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
  "requestId": "abc123defg56789",
  "result": [
    {
      "accdesc": "Certified Pre-Owned",
      "acccode": "00A",
      "tradein": 0,
      "retail": 1325,
      "loan": 0,
      "isincluded": 0,
      "isadded": 0,
      "accessorycategory": "General"
    },
    {
      "accdesc": "Luggage Rack",
      "acccode": "043",
      "tradein": 50,
      "retail": 75,
      "loan": 50,
      "isincluded": 0,
      "isadded": 0
    },
    {
      "accdesc": "Leather Seats",
      "acccode": "076",
      "tradein": 0,
      "retail": 0,
      "loan": 0,
      "isincluded": 1,
      "isadded": 0,
      "accessorycategory": "General"
     ]
   , "userinfo": "Default"
   , "authId": "UCG"
}
```


See accessoriesByVehicleId method above with additional notes as below.

- ***isadded:*** This accessory is pre-selected based on the input VIN using UCG's standard VIN decode (VIN Pattern-based) or Build data (when calling buildAccessoriesByVehicleId)

13.1.4.masterVehiclesByLegacyId

This method provides a mapping between the legacy/SOAP vehicle content and the new master/REST vehicle content. The vehicle ID not only changed from the UID to UcgVehicleId, but also in some cases there is a 1-to-many UID to UcgVehicleId relationship because the master/REST vehicle content is more granular.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/masterVehiclesByLegacyId/

Parameters:

- period string required Valuation period as YYYY-MM-DD
- uid int required The legacy/SOAP vehicle ID in question
- userinfo string optional User-defined string used to identify a specific user of a shared API-Key.

Legacy conversion: This is similar in functionality to the legacy getVehicleByVic method, which was used to map the older legacy VIC vehicle key to the legacy UID vehicle key.

Notes for legacy Data Files users: The UID is an ID specifically used by our SOAP Web Services product, but it can be calculated from the Data Files/VIN Prefix Solution data products as `uid = (int)VID + 1100000` Legacy vehicle in example:

UID	Vic	VID	Make	Model	Series	Bodystyle
1168923	9720110T4B	68923	VOLKSWAGEN	Eos	EOS-4 Cyl. Turbo	Convertible 2D Komfort

```
curl -X GET \
  'https://cloud.jdpower.ai/data-api/valuationservices/
masterVehiclesByLegacyId?period=0&uid=1168923' \
  -H 'api-key: 3f3090a0-0a7b-40a7-9998-39d3efe40d63'

{
  "requestId": "abc123defg56789",
  "result": [
       {
            "ucgvehicleid": "201127796",
            "modelyear": "2011",
            "make": "Volkswagen",
            "model": "Eos",
            "body": "Convertible 2D 2.0T Komfort 2.0L I4 Turbo Auto",
            "islegacyvehicle": "true",
            "rollupaccessories": null
       },
       {
            "ucgvehicleid": "201128150",
            "modelyear": "2011",
            "make": "Volkswagen",
```

```
            "model": "Eos",
            "body": "Convertible 2D 2.0T Komfort 2.0L I4 Turbo Manual",
            "islegacyvehicle": "false",
            "rollupaccessories": "006"
        }
]
  , "userinfo": "Default"
  , "authId": "UCG"
}
```


Legacy Conversion notes: This method will return the corresponding vehicle(s) and, in some cases, accessory codes to map between the SOAP and REST API vehicle content. The record returned where isLegacyVehicle = true is the direct corresponding vehicle.

In many cases, additional records will be returned where isLegacyVehicle = false. In most cases, there will be data in the rollupAccessories field. This indicates that the non-legacy vehicle corresponds to the legacy vehicle (input UID) plus one or more accessories. In cases where the rollupAccessories field is empty for a non-legacy vehicle, the value difference for the non-legacy vehicle is based on adjustments not visible in the accessory list , or potentially the corresponding option has become unvalued since the vehicle was defined originally but we don't drop the granular vehicle definition even though the option is no longer a value add.

Please see the FAQ section for when to use this endpoint vs. when to use the "reverse" endpoint below (masterToLegacyVehicleMapping.)

Also added an FAQ on possible approaches to selecting one vehicle when this endpoint returns more than one.

13.1.5.masterToLegacyVehicleMapping

This method provides a mapping between the legacy/SOAP vehicle content and the new master/ REST vehicle content. It is the logical opposite of the earlier masterVehiclesByLegacyId (see 5.14.1 above) in that it takes a new master vehicle ID (UcgVehicleId) and returns the corresponding legacy vehicle information.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/masterToLegacyVehicleMapping/

Parameters:

• period	string	required	Valuation period as YYYY-MM-DD
• ucgvehicleid	int	required	The master vehicle ID in question
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

```
curl -X GET \
'https://cloud.jdpower.ai/data-api/valuationservices/
masterToLegacyVehicleMapping?period=0&ucgvehicleid= 201032477' \
  -H 'api-key: 3f3090a0-0a7b-40a7-9998-39d3efe40d63'
```


```
{
    "requestId": "gtldcth1rbwl4n22jlq8dp",
    "result": [
        {
            "vid": "63395",
            "uid": "1163395",
            "vic": "9420100C9F",
            "rollupvehicleid": "201031258",
            "vac": "092,068"
        }
    ],
    "authId": "UCG Testing"
}
```


Legacy Conversion notes: This method will return the corresponding vehicle and, in some cases, accessory codes to map between the SOAP and REST API vehicle content.

Please see the FAQ section for when to use this endpoint vs. when to use the "reverse" endpoint above (masterVehiclesByLegacyId.)

13.1.6.weeklyUsedValuesByLegacyId

This endpoint corresponds to the weeklyUsedValuesByVehicleId endpoint above, with the exception that it requires the legacy vehicle ID (UID or VID). It is intended to provide Weekly Used Values support for the deprecated Used Car Guide SOAP Web Service.

License decrement: This method **does** decrement the user's license unless immediately before or after an otherwise-identical call to the monthly endpoint.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/weeklyUsedValuesByLegacyId

Parameters:

• period	string	required	Valuation/definition date
• vehicletype	string	optional	Filter to specific vehicle types
• uid	int	required	vehicle ID (UID or VID) to be valued
• region	int	required	Region ID (see Definitions and Regions)
• mileage	int	optional	Used to determine mileage adjustment. No mileage adjustment returned if empty
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

```
curl -L -X GET 'https://cloud.jdpower.ai/data-api/valuationservices/valuation/
weeklyUsedValuesByLegacyId?
period=0&vehicletype=UsedCar&vin=1FMCU9J93JUB98016&region=1&mileage=12345' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```

The fields in this response correspond to those in the weeklyUsedValuesByVehicleId endpoint. The 'business value' fields (maxmileageadj through minadjretailforloan) apply equally to these Weekly Used Values.


```
{
    "requestId": "1rmpndy4fcyxs63w6n28ja",
    "result": [
        {
            "baseweeklycleantrade": "11950",
            "baseweeklyaveragetrade": "10975",
            "baseweeklyroughtrade": "9775",
            "baseweeklycleanretail": "13925",
            "baseweeklyloan": "10775",
            "averagemileage": "47500",
            "mileageadjustment": "2650",
            "adjustedweeklycleantrade": "14600",
            "adjustedweeklyaveragetrade": "13625",
            "adjustedweeklyroughtrade": "12425",
            "adjustedweeklycleanretail": "16575",
            "adjustedweeklyloan": "13425",
            "maxmileageadj": "5975",
            "minmileageadj": "-4780",
            "minadjretail": "420",
            "minadjcleantrade": "120",
            "minadjaveragetrade": "120",
            "minadjroughtrade": "120",
```

```
                "minadjloan": "0",
                "minadjretailforloan": "900",
                "vid": "95241"
            }
        ],
        "userinfo": "Default",
        "authId": "UCG UAT"
}
```


13.1.7.masterVehiclesByLegacyIdAndVin (GET method)

This method provides an additional mapping between the legacy/SOAP vehicle content and the new master/REST vehicle content, using the combination of the input legacy ID and an input VIN to try to reduce the number of multiple responses you can get from just the legacy ID conversion endpoint. See below for POST method for batch support.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/masterVehiclesByLegacyIdAndVin/

Parameters:

• period	string	required	Valuation period as YYYY-MM-DD
• uid	int	required	The legacy/SOAP vehicle ID in question
• vin	string	required	The full 17-character VIN of the vehicle
• userinfo	string	optional	User-defined string used to identify a specific user of a shared API-Key.

Notes for legacy Data Files users: The UID is an ID specifically used by our SOAP Web Services product, but it can be calculated from the Data Files/VIN Prefix Solution data products as

uid = (int)VID + 1100000

This endpoint accepts either the VID or UID as an input, and only returns data for input periods 2020-04-01 and newer.


```
curl -L -X GET 'https://cloud.jdpower.ai/data-api/UAT/valuationservices/valuation/masterVehiclesByLegacyIdAndVin?period=0&uid=1170439&vin=3FAHP0JG9CR222637' \
  -H 'api-key: 3f3090a0-0a7b-40a7-9998-39d3efe40d63'
{
    "requestId": "b6o3tn4kw8fsacg4h5e8ao",
    "result": [
        {
            "selectpriority": "2",
            "ucgvehicleid": "201209422",
            "modelyear": "2012",
            "make": "Ford",
            "model": "Fusion",
            "bodystyle": "Sedan 4D SEL 3.0L V6",
            "rollupvacs": "",
            "decodepriority": "1",
            "legacypriority": 1
        }
    ],
```

```
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


Result field notes:

- selectpriority Recommended match priority in descending order
- rollupvacs Accessory codes added to the 'base' legacy vehicle to achieve this particular master vehicle content
- decodepriority 1 = Default VIN decode, 0 = Alternate VIN decode
- legacypriority 1 = exact match for legacy vehicle , 0 = legacy vehicle + option content

13.1.8.masterVehiclesByLegacyIdAndVin (POST method)

This method provides batch support to the masterVehiclesByLegacyIdAndVin endpoint. Instead of the (single call) GET method, you use POST and submit a JSON array of up to 1,000 records (UID + VIN). For performance purposes, the batch endpoint does not accept a period parameter and will only return the most recent mapping results.

Developer note: This endpoint uses the POST method, not GET. Calling this endpoint via GET will return a warning message reminding you to use POST instead.

Developer note: There is a single (required) parameter – input – which contains a JSON-formatted array of Vin/state/mileage values. In theory you can include this parameter in the querystring for the endpoint as normal. However, very large querystring values can be truncated at multiple points along the internet connection, from the user's system, proxy or firewall all the way through multiple routers or the destination hosts. As a result, we recommend including the input parameter in the request Body using x-www-form-urlencoded formatting.

Developer note: The various batch endpoints are provided to streamline the use of our REST API for larger volumes – sending 1 call with 500-1,000 inputs rather than having to serialize 500 to 1,000 individual calls. However, as these endpoints require longer query processing time than the single calls we request that users taking advantage of our batch endpoints do not submit more than 1 – 2 batch requests in parallel.

Batch performance: Since this endpoint does not do any secondary lookups or match (unlike the batch VIN decode + value endpoint), its response time is often less than 2,000ms even for a large input.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/masterVehiclesByLegacyIdAndVin/ {POST not GET}

Parameters:

- input string required JSON array of input rows

```
curl -L -X POST 'https://cloud.jdpower.ai/data-api/UAT/valuationservices/
valuation/masterVehiclesByLegacyIdAndVin' \
-H 'Content-Type: application/x-www-form-urlencoded' \
-H 'api-key: aad0bbbd-35c4-44b8-96d4-6d697f8c86eb' \
--data-urlencode 'input=[{"vin":"3VWJP7AT1CM637785","uid":1171661},
{"vin":"3N1BC1CP5CK805474","uid":1171162},
{"vin":"YV1RS592592724622","uid":1153753},
{"vin":"1HGCR2F81DA190565","uid":1175654}]'
```


```
{
    "requestId": "zeckyb92d3o63d6bkr32rq",
    "result": [
        {
            "rowid": "0",
            "uid": "1171661",
            "vin": "3VWJP7AT1CM637785",
            "selectpriority": "2",
            "ucgvehicleid": "201234384",
            "modelyear": "2012",
            "make": "Volkswagen",
            "model": "Beetle",
            "bodystyle": "Coupe 2D 2.5 2.5L I5 Auto",
            "rollupvacs": "",
            "decodepriority": "1",
            "legacypriority": 1
        },
        {
            "rowid": "1",
            "uid": "1171162",
            "vin": "3N1BC1CP5CK805474",
            "selectpriority": "2",
            "ucgvehicleid": "201222707",
            "modelyear": "2012",
            "make": "Nissan",
            "model": "Versa",
            "bodystyle": "Hatchback 5D S 1.8L I4",
            "rollupvacs": "",
            "decodepriority": "1",
            "legacypriority": 1
        },
        {
            "rowid": "2",
            "uid": "1153753",
            "vin": "YV1RS592592724622",
            "selectpriority": "2",
            "ucgvehicleid": "200905773",
            "modelyear": "2009",
            "make": "Volvo",
            "model": "S60",
            "bodystyle": "Sedan 4D 2.5L I5 Turbo",
            "rollupvacs": "",
            "decodepriority": "1",
            "legacypriority": 1
        },
        {
            "rowid": "3",
            "uid": "1175654",
            "vin": "1HGCR2F81DA190565",
```

```
            "selectpriority": "2",
            "ucgvehicleid": "201318705",
            "modelyear": "2013",
            "make": "Honda",
            "model": "Accord",
            "bodystyle": "Sedan 4D EX-L 2.4L I4 Auto",
            "rollupvacs": "",
            "decodepriority": "1",
            "legacypriority": 1
        },
        {
            "rowid": "3",
            "uid": "1175654",
            "vin": "1HGCR2F81DA190565",
            "selectpriority": "0",
            "ucgvehicleid": "201308791",
            "modelyear": "2013",
            "make": "Honda",
            "model": "Accord",
            "bodystyle": "Sedan 4D EX-L Navigation 2.4L I4 Auto",
            "rollupvacs": "346",
            "decodepriority": "0",
            "legacypriority": 0
        }
    ],
    "authId": "UCG UAT"
}
```


Results field notes:

- **priority, rollupvacs***See GET method description above*
- **uid, vin** Input field values for reference
- **rowid** Cardinal row number of input array. Used to allow you to link multi-row results (see rowid = 3 in example return)

Empty/NULL results

All input rows will receive at least one result row. However, you may receive a results row with NULL results (other than the input fields and rowid).

This can happen when the VIN either doesn't decode in our system in the current data, or does not decode to the input legacy ID. This will occur primarily for VINs that were decoded in prior years for older vehicles no longer in our dataset (rolling 20 model year range), or less frequently when the VIN to vehicle ID assignment changed and the input UID (or VID) is no longer assigned as either the Default or as an Alternate vehicle decode in our legacy data.

13.1.9.vehicleList_Legacy

Similar to the 'standard' Vehicle List endpoint, this is intended to provide the corresponding legacy vehicle keys (VIC, VID, UID) and legacy descriptions. It is intended for users who must interface with both the new Master content (this REST API) and legacy systems (SOAP API, legacy data files, etc.)

This endpoint returns a list of the vehicles for the input parameters, which apply some mandatory and some optional filter parameters.

License decrement: This method does not decrement the user's license.

Endpoint: https://cloud.jdpower.ai/data-api/valuationservices/valuation/vehicleList_Legacy

Legacy conversion: This is a new method; there is no corresponding legacy/SOAP method. Also, see the Developer Note below regarding potential parameter filter issues.

Parameters:

- period string required Valuation/definition date
  CRITICAL NOTE: Only the most recent 2 months are available in this endpoint.
- vehicletype string optional Filter to specific vehicle types
- modelyear int optional Model year selected; leave off to receive all model years available
- make string optional Make selected; leave off to receive all makes available for the selected Model Year
- userinfo string optional User-defined string used to identify a specific user of a shared API-Key.

```
curl -X GET \ https://cloud.jdpower.ai/data-api/valuationservices/valuation/vehicleList_Legacy?period=0' \
  -H 'api-key: 3f309xa0-0a7b-4ga7-9998-39d9e3e40d63'
```


```
{
    "requestId": "w1p09xj811snv1v8ujd5w",
    "result": [
        {
            "ucgvehicleid": "202235683",
            "year": "2022",
            "make": "Acura",
            "model": "MDX",
            "masterbody": "Utility 4D 3.5L V6",
            "mastervid": "00000107657",
            "legacyvic": "662022020H",
            "legacyvid": "00000107657",
            "uid": "1207657",
            "legacyseries": "MDX",
            "legacybody": "Utility 4D 2WD",
            "islegacyvehicle": "true",
            "legacyaccessories": "",
            "vehicletype": "Light Truck"
        },
        {
```

```
            "ucgvehicleid": "201334148",
            "year": "Audi",
            "make": "Acura",
            "model": "A7",
            "masterbody": "Sedan 4D Prestige w/Innovation Pkg. AWD 3.0L V6
Supercharged",
            "mastervid": "00000075923",
            "legacyvic": "6520130K4C",
            "legacyvid": "00000075621",
            "uid": "1175621",
            "legacyseries": "A7",
            "legacybody": "Sedan 4D 3.0T Prestige AWD",
            "islegacyvehicle": "false",
            "legacyaccessories": "0C7,0C8,0D5",
            "vehicletype": "Car"
        },… (more rows)
    ],
    "effectivedate": "2021-07-01 - 2021-07-31",
    "userinfo": "Default",
    "authId": "UCG UAT"
}
```


Developer note: Since the response data for this endpoint can be very large (up to several MB for an un-filtered request of over 60k result rows) the Average Response Time for this endpoint can significantly exceed the normal endpoint response times. Please include the HTTP Header for Accept-Encoding: gzip to enable compression of the results during network transfer.

Developer note: Like our other endpoints, this endpoint is case-sensitive on the input parameters. Additionally, the model year/make filters must align with data in our system. For instance, as of July 1, 2021, you will get an empty response if you set the modelyear parameter to 2022, or to 2000, and you will also get an empty response if you set the make to Ram (or Ford Truck, which exists as a Make in our legacy system but is replaced with simply Ford in the REST API) or to an exotic make not in our systems (i.e., Lamborghini, etc.)

Unique response fields:

- **masterbody** The vehicle description in the Master content
- **mastervid** The legacy vehicle ID (VID) for this vehicle at the Master content level
- **legacyvic** The legacy vehicle ID (VIC) for the corresponding legacy vehicle
- **legacyvid** The legacy vehicle ID (VID) for the corresponding legacy vehicle
- **uid** The legacy SOAP API vehicle ID for the corresponding legacy vehicle
- **legacyseries** The Series description for the corresponding legacy vehicle
- **legacybody** The Bodystyle description for the corresponding legacy vehicle
- **islegacyvehicle** {true|false} Is this the identical vehicle as the corresponding legacy one
- **legacyaccessories** A comma-delimited array of accessory codes (VACs) that need to be

added to the legacy vehicle to achieve the same content as the Master vehicle. Will be empty when islegacyvehicle=true, MAY be empty for others.