Integration Guide

Feature Exchange Service

June 1, 2022

AUTODATA SOLUTIONS

TABLE OF CONTENTS

DOCUMENT OVERVIEW

Recognizing that you may be new to the Developer Portal, this guide provides step-by-step instructions on how to make requests using the Test Client within the Portal.

The Request Object section provides examples that you can use to make requests in the Developer Portal so that you can jump start your understanding of the Feature Exchange product.

The Response Object section provides a brief description of the Feature Exchange response object and informs you about where you can go to get more information about the response.

The final section directs you to where you can get more integration information once you are ready to start developing your own application.

Related Documentation

Document	Description
API Reference	The Feature Exchange API reference is available in a Swagger UI format within the Portal. It describes the service, each endpoint, each input attribute and each output attribute.
Batch Guide	The Feature Exchange Batch Integration guide describes how to submit vehicles in input files to be processed. As well, it describes the output files that are generated by the service.
Feature List	The Feature List document provides a comprehensive list of the features returned by the Feature Exchange service. Included in this list, along with other data, are the Feature Id and Feature Key values for each feature.
Simple Model Walk	Two types of documentation are available for the Simple Model Walk service. API Reference – A Simple Model Walk API reference is available in a Swagger UI format within the Portal. It describes the service, each endpoint, each input attribute and each output attribute. Tutorial – This guide provides step-by-step instructions on how to use the Simple Model Walk service to identify a style id for a vehicle. Each step provides an example request and response. You will require the Simple Model Walk service and its documentation when you do not have a VIN and you need a different way in which to identify vehicles in requests within the Feature Exchange service.
Portal Guide	The Portal Orientation guide provides step-by-step instructions on how to navigate and use the Portal. For example, it explains how you can make requests using the Portal's Test Client. Once you understand how to find and use the Test Client, you can use the example requests in this guide within the Test Client to send requests.
Security Guide	The Shared Secret Security Protocol document describes how to build and integrate a security token protocol into the Authorization header of the request to the service. You would need this information after you have finished testing a service in the Portal and are ready to begin your development work.

CONTACTING CLIENT SUPPORT

Client Support is available by phone toll-free at (800) 937-3661, Monday through Friday, from 6:00 a.m. to 5:00 p.m. Pacific Time, or you can reach Client Support by email at css@autodata.net. This team can help you with product support, billing questions, and other inquiries.

PRODUCT OVERVIEW

The Feature Exchange product is built as a RESTful service that takes a VIN or Style ID as a request input and returns vehicle information beneficial to the vehicle quoting, rating and underwriting processes.

Today, the insurance industry primarily uses risk factors associated with the driver (i.e. age, marital status, credit history) when providing an insurance quote. With new driver-assist features that can prevent accidents, insurance carriers can start including vehicle factors in their premium pricing algorithms. This service allows insurance providers to quickly evaluate the vehicle being quoted, without interviewing their customer, to determine if there are any cost reductions or increases that are required based on the features of the vehicle.

The data returned focuses on features that prevent the frequency or severity of accidents / insurance losses. The data includes passive safety features, active safety features, features that prevent theft as well as other features that help describe the vehicle.

This API has 2 endpoints:

PUT /study - Returns feature information for the requested vehicle.

PUT /multistudy – Returns feature information for up to five requested vehicles.

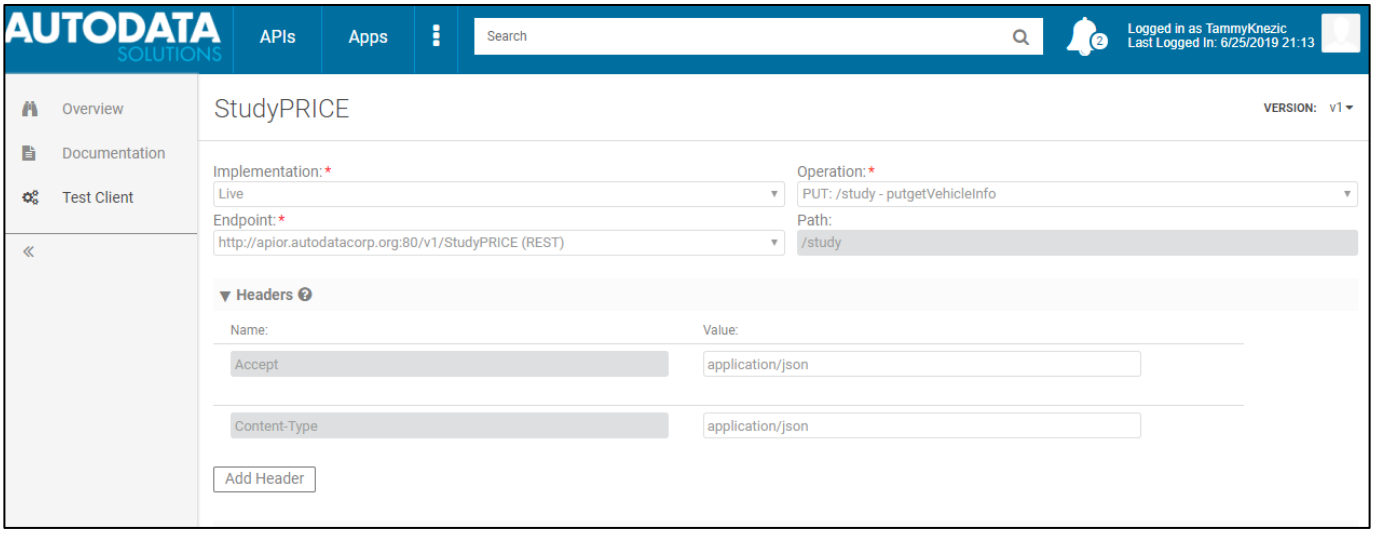
Note: Weight and measurement related data is returned as imperial for the US and metric for Canada. Also, weight data is returned as rounded figures. In addition, prices returned are in United State (US) dollars for US styles and Canadian (CA) Dollars for CA styles.

USING THE DEVELOPER PORTAL TO MAKE REQUESTS

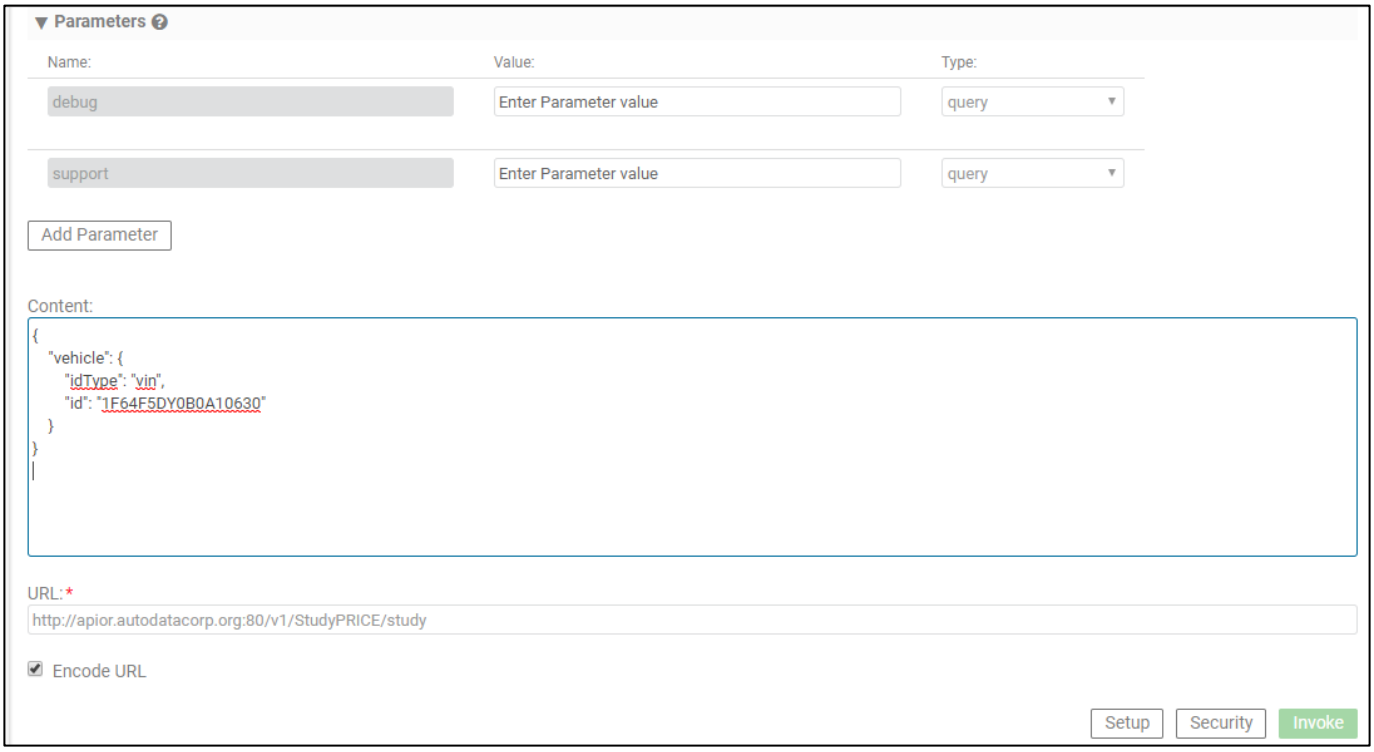
You access and use the two Feature Exchange endpoints using Autodata's Developer Portal. There are two ways to make requests using the Portal. You can access the swagger documentation and make a request from there or preferably you can use the Test Client. This section describes how to use the Test Client.

To execute a request:

1. On the Dashboard, click the **APIs** button.
2. On the APIs page, click the hyperlink name for the StudyPRICE API.
3. On the web service page, in the left navigation menu, click **Test Client**.



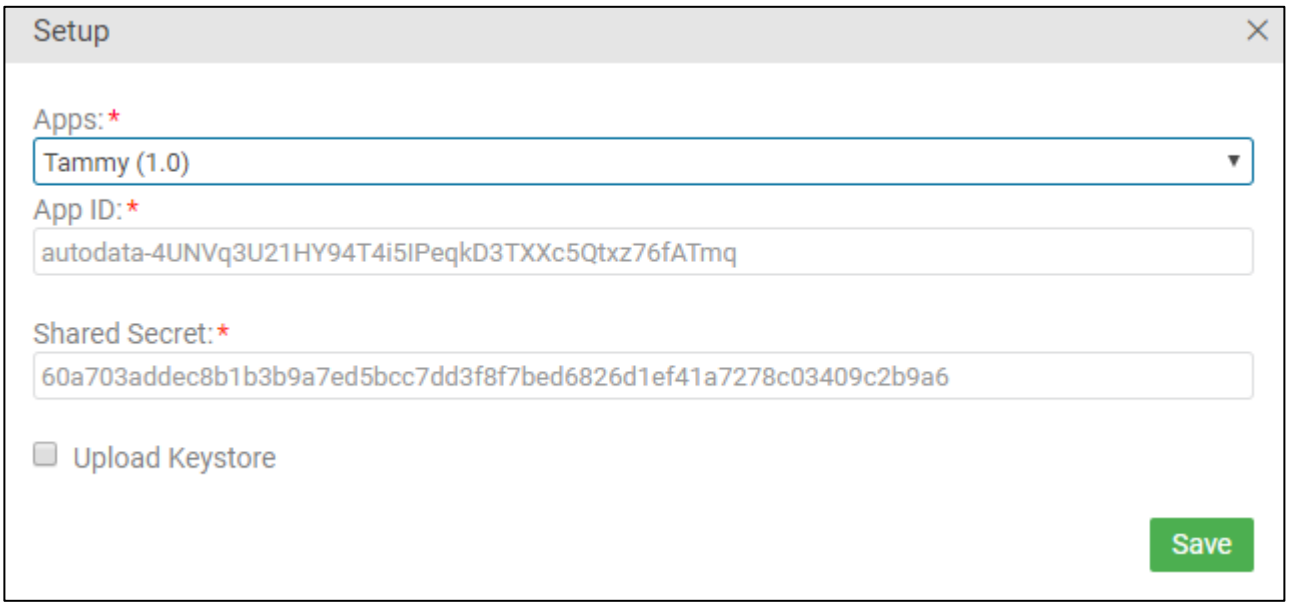
4. On the Test Client page, from the Endpoint drop-down menu, select an endpoint.
5. From the Operation drop-down list, select an operation.
6. In the Content field, enter the body of the request.



Note: You can copy the examples above into the Content field. Select the /multistudy endpoint if copying from the Multistudy Request Examples section on page 14.

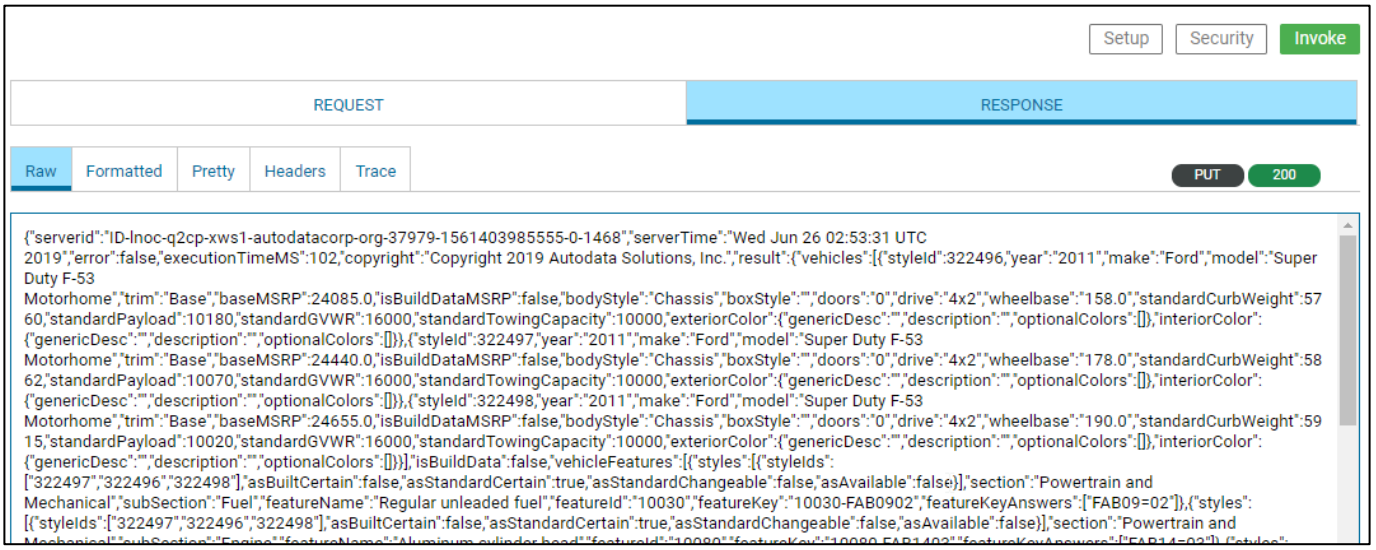
7. Click the **Setup** button.

The Setup window displays. It contains the Appid and secret key that was set up for you app. If these fields are blank, refer to section "**Error! Reference source not found.**" on page **Error! Bookmark not defined.** to learn where to locate this information within the Developer Portal.

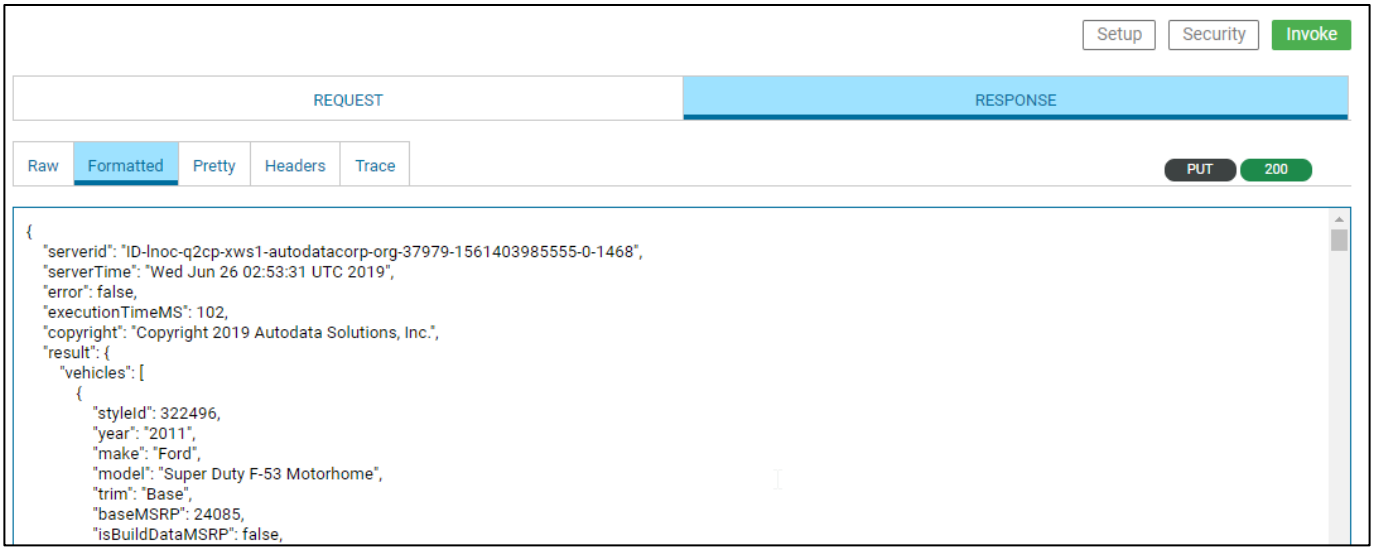


8. Click **Save**.
9. Click the **Invoke** button.

   The request is sent, and a response is returned in the Response tab.



10. To view the response in a format that is easier to read, click the **Formatted** tab.



REQUEST OBJECT

The Request object for the /study endpoint has three sections: vehicle, extendedInfo and vehicleInfo, as described in the following section.

The Request object for the /multistudy endpoint is different. For about the /multistudy request object, refer to “Multistudy Request Examples” on page 14.

Section	Description
vehicle	Identifies the vehicle(s) for which features information is being requested. Also, identifies the features to include or exclude in the results.
extendedInfo	When identifying a vehicle by its VIN, the response could contain multiple styles within the vehicles object. The extendedInfo section of the request provides a way to reduce the number of styles returned by passing options and equipment that is on the vehicle style of interest.
vehicleInfo	When identifying a vehicle by its VIN, the response could contain multiple styles within the vehicles object. The extendedInfo section of the request provides a way to reduce the number of styles returned by passing a mfgID, trim or style along with the VIN.

vehicle

Identifies the vehicle(s) for which features information is being requested. Also, identifies the features to include or exclude in the results.

Note: IdType and Id are required in each request. All other request attributes are optional.

Attribute	Description
idType	Identifies the type of vehicle identifier to be used in the request. The available idTypes are VIN and STYLEID.
Id	Identifies the vehicle according to the idType. For example, if the idType is VIN, the id could be 1FTFW1CF3BFA17129.
includeFeatureIDs	Identifies features to be included in the response. Only the features related to, or associated with, the requested featureids are returned. If no featuresids are listed, filtering does not occur. If neither the includeFeatureIDs or excludeFeatureIDs attributes have values, all features for the requested vehicle are returned. For a list of feature IDs, refer to the Feature Exchange Feature List document, which you can access through the Portal.
excludeFeatureIDs	Identifies features to be excluded in the response. Only the features related to, or associated with, the requested featureids are excluded. If no featureids are listed, filtering does not occur. If neither the includeFeatureIDs or excludeFeatureIDs attributes have values, all features for the requested vehicle are returned. For a list of feature IDs, refer to the Feature Exchange Feature List document, which you can access through the Portal.

Example 1: Returns All Features for the Requested VIN

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1F64F5DY0B0A10630"
    }
}
```

Example 2: Returns Only Features 12750 and 11540 for the Requested VIN

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1F64F5DY0B0A10630",
        "includeFeatureIDs": [
            "12750","11540"
        ]
    }
}
```

Example 3: Returns All Features Except 12750 and 11540 for the Requested VIN

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1F64F5DY0B0A10630",
        "excludeFeatureIDs": [
```

```
            "12750","11540"
        ]
    }
}
```

Note: The vehicles object in the response for all three of the above examples contain three vehicle styles 322497, 322496, 322498. The examples in the following sections explain how to reduce the number of styles returned to narrow the data in the response.

Example 4: Return All Features for the requested Partial VIN

```
{"vehicle":
{"idType": "vin","id": "2G4GS5GX4G9189000"}
}
```

OR

```
{"vehicle":
{"idType": "vin","id": "2G4GS5GX4G9189"}
}
```

Note:

Partial VINS can be used in the request body. A partial VIN at a minimum needs to consist of the first 8 characters of the VIN plus the 10th character (i.e., 1FDRF3HTK). However, for Subaru VINS, its VIN pattern requires that we pass the 11th character because this identifies the transmission, which allows for easy decode of the VIN (i.e., 4S3BNAB60J3).

Alternatively, you can pass zeros in for the missing characters (i.e., 1FDRF3HT0K0000000 or for Subaru 4S3BNAB60J3000000).The StudyPRICE service attempts to match to an existing VIN pattern and if a match is found the resulting StyleID(s) and their corresponding features are returned. Also, the vinDecoded attribute in the service response contains the new VIN value used to generate the response.

extendedInfo

When identifying a vehicle by its VIN, the response could contain multiple styles within the vehicles object.

When the service returns more than one style within the response vehicles object, you can narrow the information returned by choosing the style of interest. To achieve this, a request can specify option codes and/or equipment descriptions that may be on a specific vehicle style or a fewer number of vehicle styles.

Note: Whether you pass option codes (i.e., 1SA) or equipment descriptions (i.e., navigation) to narrow the information returned depends on the extra information you have about the vehicle. For example, you may not have option codes, but you may know the equipment descriptions and vice versa.

Attribute	Description
optionCodes	Identifies manufacturer option codes that are associated with the requested vehicle. When a response returns one-to-many styles, the option codes (i.e., 101A) can be used in conjunction with the VIN to try to reduce the styles returned in the response.
equipment	Identifies features that are on the requested vehicle. For example, "airbags", "leather seats" and so on. This can be any free form text. When a response returns one-to-many styles, the equipment descriptions can be used in conjunction with the VIN to try and reduce the styles returned in the response.

Example1: Returns All Styles

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1FTEW1EG8FFB76560"
    }
}
```

Note: The above request returns five styleIds: 372630, 372641, 372610, 372631, 372612. The following two request examples return a response that includes only styleId: 372612.

Example 2: Returns Only the Style that Includes Option 101A

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1FTEW1EG8FFB76560"
      },
    "extendedInfo": {
        "optionCodes": [
            "101A"
        ]
    }
}
```

Note: The above request returns only one styleId: 372612 because that is the only styleId that contains an available option with an option code of 101A.

Example 3: Returns Only the Style that Includes Equipment MyKey

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1FTEW1EG8FFB76560"
      },
    "extendedInfo": {
        "equipment": [
            "MyKey"
```

```
        ]
    }
}
```

Note: The above request returns only styleId: 372612 because that is the only styleId that includes the MyKey equipment.

vehicleInfo

When identifying a vehicle by its VIN, the response could contain multiple styles within the one vehicle object.

When the service returns more than one style within the response vehicle object, you can narrow the information returned by choosing the style of interest. To achieve this, a request can specify the mfgID, trim or style along with the VIN.

Note: Whether you pass mfgId, trim or style to narrow the information returned depends on the extra information have about the vehicle. For example, you may not have mfgId or style information, but you may know the trim.

Attribute	Description
mfgId	Identifies a manufacturer style code. For example, F1C. When a response returns one-to-many styles, the mfgId parameter can be used in conjunction with the VIN to try to reduce the styles returned in the response.
trim	Identifies the vehicle trim description. When a response returns one-to-many styles, the trim parameter can be used in conjunction with the VIN to reduce the styles returned in the response.
trimIdType	Identifies a third-party id type for the trim. Currently, the supported value includes STYLE.
trimId	Identifies the vehicle trimId. When a response returns one-to-many styles, the trimId parameter can be used in conjunction with the VIN to reduce the styles returned in the response.

Example 1: Returns All Styles

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1HGCM56716A034214"
    }
}
```

Note: The above request returns four styles: 280169, 280168, 280163, 280162.

Example 2: Returns Styles for mfgId CM5676JNW

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1HGCM56716A034214"
      },
    "vehicleInfo": {
        "mfgId": "CM5676JNW"

    }
}
```

Note: The above request returns two styles 280169, 280168 because they both have the same mfgid.

Example 3: Returns Styles for Trim EX

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1HGCM56716A034214"
      },
    "vehicleInfo": {
        "trim": "EX"

    }
}
```

Note: The above request returns two styles 280163, 280162 because they both have the same trim.

Example 4: Returns Style 280168

```
{
    "vehicle": {
        "idType": "vin",
        "id": "1HGCM56716A034214"
    },
   "vehicleInfo": {
       "trimIdType": "STYLE",
```

```
        "trimId": "280168"
    }
}
```

Note: The above request returns style 280168. The above request results in the same response as the following request.

```
{
    "vehicle": {
        "idType": "styleid",
        "id": "280168"
    }
}
```

Note: You can only make the above request if you know the styleId. If you only know the VIN, you make a request using the VIN (i.e., example 1), which returns all the styles. Once you know the styleId, you can make either of the Example #4 requests to return information only for that style. This also means that if you know the styleId, you can use it to identify the vehicle instead of the VIN in any of the request examples throughout this guide.

Alternatively, if you do not have a VIN, you can use the Simple Model Walk service to retrieve a styleId and use it in a call like the one above.

Multistudy Request Examples

The Request object for the /multistudy endpoint has the same three sections: vehicle, extendedInfo and vehicleInfo as the /study endpoint. The difference is that the /multistudy endpoint has an extra vehicles container and you can repeat the vehicle object up to five times within a single request. This allows you to request information for up to five vehicles in the same call.

Example 1: Returns All Features for 3 VINs

```
{
 "vehicles":[
 {
  "vehicle":{
    "idType":"VIN",
    "id":"1F64F5DY0B0A10630"
  }
 },
  {
   "vehicle":{
    "idType":"VIN",
```

```
    "id":"1FTEW1EG8FFB76560"
  }
  },
  {
   "vehicle":{
    "idType":"VIN",
    "id":"1HGCM56716A034214"
  }
  }
 ]
}
```

Example 2: Combination of Single VIN Requests from the Request Object Sections

```
{
 "vehicles":[
 {
    "vehicle": {
        "idType": "vin",
        "id": "1F64F5DY0B0A10630",
        "includeFeatureIDs": [
            "12750","11540"
        ]
    }
  },
  {
   "vehicle": {
        "idType": "vin",
        "id": "1FTEW1EG8FFB76560"
      },
    "extendedInfo": {
        "equipment": [
            "MyKey"
        ]
    }
  },
  {
   "vehicle": {
        "idType": "vin",
        "id": "1HGCM56716A034214"
      },
```

```
    "vehicleInfo": {
        "trim": "EX"

    }
  }
 ]
}
```


RESPONSE OBJECT

The response object has one main result section. Within the result section are two main sections: vehicles and vehicleFeatures.

result section – Example 1 – Partial VIN:

In the example below a VIN less than 17 character is decoded. The Studyprice service will attempt to match to an existing vin pattern and if a match is found the resulting StyleID(s) and their corresponding features will be returned along with the new vin value used to generate the response as part of the vinDecoded attribute.


```
"result": {
         "id": "2G4GS5GX4G9189",
         "idType": "VIN",
         "vinDecoded": "2G4GS5GX4G9189000",
         "isBuildData": false,
         "manufacturer": "General Motors",
         "vehicles": [...]
```


NOTE:
When there is no build data the asBuiltEstimatedMSRP can still have a different value than the baseMSRP and the reason for this has to do with the upgraded engine. In the case of a successful decode, if there is no build data and the VIN corresponds to an upgraded engine, and that engine has a price in the catalog data for the styleid, the engines MSRP value is applied to the asBuiltEstimatedMSRP value.

result section – Example 2 – Style ID:

The following example has been truncated and modified to show all the attributes of the result section.


```
    "result": {
        "id": "372610",
        "idType": "STYLEID",
        "vinDecoded": "",
        "isBuildData": false,
        "vehicles": [],
        "vehicleFeatures": [],
        "status": 200
```

```
    }
```

Note: In the above example, vehicles and vehicleFeatures are empty. The following examples detail what they contain when they're populated.

The vehicles section details each style returned in the response by providing information like, year, make, model, trim, etc.

vehicles section – Example 1:


```
"vehicles": [
            {
                "styleId": 372610,
                "year": "2015",
                "make": "Ford",
                "model": "F-150",
                "trim": "XLT",
                "baseMSRP": 40050,
                "asBuiltEstimatedMSRP": 40450,
                "isBuildDataMSRP": false,
                "bodyStyle": "SuperCrew Cab Styleside",
                "boxStyle": "Styleside",
                "doors": "4",
                "drive": "4x4",
                "wheelbase": "145.0",
                "standardCurbWeight": 4696,
                "standardPayload": 1600,
                "standardGVWR": 6350,
                "standardTowingCapacity": 7100,
                "exteriorColor": {
                    "optionalColors": [
                        {
                            "genericDescription": "Green",
                            "description": "Green Gem Metallic"
                        },
                        {
                            "genericDescription": "Silver",
                            "description": "Ingot Silver Metallic"
                        }
                    ]
                },
```


Note: This information is repeated for each style returned.

The vehicleFeatures section returns information related to each feature including which styles have the listed feature.

vehicleFeatures Section – Example 1: In the example below the feature is Regular unleaded fuel and five styles have this feature 372630, 372641, 372610, 372631 and 372612.

This information is repeated for each feature returned.


```
"vehicleFeatures": [
        {
            "section": "Powertrain and Mechanical",
            "subSection": "Fuel",
            "featureName": "Regular unleaded fuel",
            "featureNameNoBrand": "Regular unleaded fuel",
            "featureId": "10030",
            "featureKey": "10030-FAB0902",
            "isNumeric": false,
            "featureKeyAnswers": [
                "FAB09=02"
            ],
            "styles": [
                {
                    "styleIds": [
                        "372630",
                        "372641",
                        "372610",
                        "372631",
                        "372612"
                    ],
                    "asIccStandardCertain": true,
                    "asBuiltCertain": false,
                    "asStandardCertain": true,
                    "asStandardChangeable": false,
                    "asAvailable": false
                }
            ]
        },
```


vehicleFeatures Section - Example 2: Some features are linked to a specific styleId. Therefore, even if more than one styleId is returned in the response, only one styleId is listed with this feature. For example,


```
{
            "section": "Exterior and Appearance",
            "subSection": "Wheels",
            "featureName": "Wheel hub covers",
            "featureNameNoBrand": "Wheel hub covers",
            "featureId": "12720",
            "featureKey": "12720-HQD0302",
            "isNumeric": false,
```

```
            "featureKeyAnswers": [
                "HQD03=02"
            ],
            "styles": [
                {
                    "styleIds": [
                        "372612"
                    ],
                    "asIccStandardCertain": false,
                    "asBuiltCertain": false,
                    "asStandardCertain": false,
                    "asStandardChangeable": true,
                    "asAvailable": false
                }
            ]
        },
```


vehicleFeatures Section - Example 3: To support the above example, a uniqueFeatures response object may be returned at the end of a response. The uniqueFeatures response object is returned if it is enabled in your account profile settings and if the vehicle identified in the request is a VIN that returns multiple styleIds. If a feature is not available across all the styleIds for a VIN, the uniqueFeatures response object contains a list of featureids and styleids associated with that feature. The featureIds are sorted in order from the most unique (i.e., those features that return for the least number of styleIds) to the least unique (i.e., those that return for the greatest number of styleIds but not all styleIds).


```
    "uniqueFeatures": [
        {
            "featureId": "12720",
            "styleIds": [
                "372612"
            ]
        },
        {
            "featureId": "13350",
            "styleIds": [
                "372641"
            ]
        },
        {
            "featureId": "10120",
            "styleIds": [
                "372610",
                "372612"
            ]
        },
```

```
{
    "featureId": "19140",
    "styleIds": [
        "372630",
        "372641",
        "372610"
    ]
},
```


vehicleFeatures Section - Example 4: Some features return numeric values. In the following example featureId: 10120 has three additional attributes: numericFeatureName, numericValue and numericUnits. These are described in Response Attributes section.


```
{
    "section": "Powertrain and Mechanical",
    "subSection": "Engine specs",
    "featureName": "Engine displacement: 213 cu.in.",
    "featureId": "10120",
    "featureKey": "10120-FAC0121334",
    "isNumeric": true,
    "numericFeatureName": "Engine displacement",
    "numericValue": 213,
    "numericUnits": "cubic inches",
    "featureKeyAnswers": [
        "FAC01=21334"
    ],
    "styles": [
        {
            "styleIds": [
                "372630",
                "372641",
                "372631"
            ],
            "asIccStandardCertain": false,
            "asBuiltCertain": false,
            "asStandardCertain": true,
            "asStandardChangeable": false,
            "asAvailable": false
        }
    ]
},
```

Note: The response object for /multistudy requests provide the entire response for the first VIN including vehicle description and feature descriptions as described above, then provides the entire response for the second VIN and so on.

Response Attributes

Attribute	Description
id	Contains the vehicle identifier that corresponds to the submitted idType. For example, if the idType is VIN, the id could be 1FTFW1CF3BFA17129.
idType	Contains the type of vehicle identifier used in the request. The available IdTypes are VIN and STYLEID.
vinDecoded	Contains the VIN that the system decoded and used to provide the response. If you submit a full VIN, the full VIN is returned for this attribute. If you submit a valid partial VIN, it is decoded to determine a full VIN and that full VIN is used to determine the response and is returned for this attribute. If the vehicle identifier you submit is a STYLEID, this attribute returns as blank.
isBuildData	Contains a flag that when set to true, indicates that build data is available for the decoded VIN.
vehicles	Contains the vehicle description.
styleId	Contains the styleId of a vehicle.
year	Contains the vehicle model year.
make	Contains the vehicle make.
model	Contains the name of a vehicle model.
trim	Contains the vehicle trim.
baseMSRP	Contains the base price of the vehicle before adding options.
asBuiltEstimatedMSRP	When the isBuildDataMSRP flag is set to true, it indicates that the asBuiltEstimatedMSRP comes from the build record. When isBuildDataMSRP is set to false, and the isBuildData value is true, asBuiltEstimatedMSRP contains the baseMSRP plus the MSRP of all options from the build record that are contained within our Catalog data. When the isBuildDataMSRP is set to false and isBuildData is false, the asBuiltEstimatedMSRP attribute contains the baseMSRP from Catalog data.
isBuildDataMSRP	Contains a flag that when set to true, indicates the MSRP value comes from build data. If it is set to false, the MSRP value comes from Catalog data.
bodyStyle	Contains the generic classification of a vehicle.
boxStyle	Contains the box style of a truck.
doors	Contains the number of doors on a vehicle.
drive	Contains the drivetrain description.
wheelbase	Contains the wheelbase of a truck in inches.

Attribute	Description
standardCurbWeight	Contains the standard curb weight of a vehicle in lbs.
standardPayload	Contains the standard payload of a vehicle in lbs.
standardGVWR	Contains the standard gross vehicle weight rating of a vehicle in lbs.
standardTowingCapacity	Contains the standard towing capacity of a vehicle in lbs.
exteriorColor	When isBuildData is true and the build record contains color information, a single generic and OEM exterior color description is returned for the VIN. When isBuildData is false, a list of all available exterior color generic and OEM descriptions are returned.
genericDesc	Contains a category of color that represents a particular hue.
description	Contains a user-friendly name for the specific paint color.
optionalColors	Contains a list of optional exterior generic and OEM color descriptions available on the requested VIN when build data is not available or the build record did not include color information.
genericDescription	Contains a category of color that represents a particular hue for a vehicle for which there is no build data.
description	Contains a user-friendly name for the specific paint color for a vehicle for which there is no build data.
interiorColor	When isBuildData is true and the build record contains color information, a single generic and OEM interior color description is returned for the VIN. When isBuildData is false, a list of all available interior color generic and OEM descriptions are returned.
genericDesc	Contains a category of color that represents a particular hue.
description	Contains a user-friendly name for the specific paint color.
optionalColor	Contains a list of optional interior generic and OEM color descriptions available on the requested VIN when build data is not available or the build record did not include color information.
genericDescription	Contains a category of color that represents a particular hue for a vehicle for which there is no build data.
description	Contains a user-friendly name for the specific paint color for a vehicle for which there is no build data.
vehicleFeatures	Contains a feature list and feature conditions for the vehicle returned.
section	Contains a user-friendly name for the main feature category.
subsection	Contains a user-friendly name for the sub-category of each feature.

Attribute	Description
featureName	Contains a user-friendly name describing the feature name with the brand.
featureNameNoBrand	Contains a user-friendly name describing the feature name with no brand.
featureId	Contains a unique identifier for the feature.
featureKey	Contains the key for each feature.
isNumeric	Contains a flag that when set to True indicates that the feature returns a numeric value. This field defaults to False.
numericFeatureName	Contains the name of the numeric feature. For example, “Vehicle body length”. Only returns in a response if isNumeric is set to True.
numericValue	Contains the numeric value. For example, “194.4”. Only returns in a response if isNumeric is set to True.
numericUnits	Contains the unit of measure for the numeric feature, if applicable. For example, “inches”. Only returns in a response if isNumeric is set to True and if the numeric value represents a measurement like vehicle body length. If the numeric value is a count, like number of doors, this value does not return in the response.
featureKeyAnswers	Contains a list of possible answers that make up part of the featureKeyId. For example, "HQA03=14","HQA04=05","HQB03=02","HQB04=14" are possible answers for featureKeyId 19140-HQA03140405HQB030204140505HQD01110201.
styles	Contains a list of styles.
styleIds	Contains the StyleIds associated with each feature category.
asICCStandardCertain	Set to True if the feature is Standard Certain and only the brand changes the feature.
asBuiltCertain	Defaults to False. Set to True if the feature is on the vehicle as per the manufacturer's build data. For information on the overrides, refer to “AsBuiltCertain Overrides” on page 25.
asStandardCertain	Set to True if the feature is standard equipment on the vehicle per the manufacturers’ catalog and no option in the catalog can change it. In this case build data is not required to know the feature is on the vehicle.
asStandardChangeable	Set to True if the feature is standard equipment on the vehicle per the manufacturers’ catalog but options exist in the catalog that could change the feature on the vehicle. In this case, catalog data indicates that the feature is standard on the vehicle but there is no build data to confirm whether the feature has been replaced with an available optional feature.
asAvailable	Set to True if the feature is available as an option per the catalog. In this case, catalog data indicates the feature is available for the vehicle but there is no build data to confirm whether the feature is installed on the vehicle.
uniqueFeatures	The uniqueFeatures response object is only returned if enabled in your profile settings and the requested VIN returns multiple styleIds. The uniqueFeatures object contains featureIds that are not

Attribute	Description
	available for every styleid that is returned for the requested VIN. The featureIds are sorted in order from the most unique (i.e., those features that return for the least number of styleIds) to the least unique (i.e., those that return for the greatest number of styleIds but not all styleIds). You can use this information to aid in style reduction based on knowing that certain featureIds apply to the VIN.
featureId	Contains a unique identifier for the feature.
styleIds	Contains a list of styleIds associated with the featureId.
status	Contains the status of the response for each VIN sent in the request. For example, 200 or 404. See the chart below for all the possible status codes the service can return. If multiple VINs are sent in a multistudy request and at least one of the VINs returns a response status code of 200, the multistudy endpoint returns a status of 200.

AsBuiltCertain Overrides

If the request is made using a styleId, a response is generated without any override to flags. If the request is made using a VIN, the VIN information is decoded for retrieving build data and style information as follows.

If the VIN decode returns build data and one style

The response returns the features where the build data option codes match the feature optcodes for the styleId and the asBuiltCertain flag changes to true. All other flags stay the same for feature.

If the VIN decode returns one style with no build data

The response returns the engine features that apply for the VIN and the asBuiltCertain changes to True. All other flags stay the same for that feature.

If the VIN decode returns build data and multiple styles

The response returns the features where build data option codes match the feature optcodes for 'ALL' styleIds and the asBuiltCertain flag changes to True. All other flags stay the same for that feature.

If the VIN decode returns multiple styles and no build data

The response is a combination of all the styleIds that return from the decode. If the featureId is unique to one styleId that feature is added to the result. If the featureIds match, the flags for each styleId, and attributes for the feature are added to the styles node.

Notes: For vehicles that return build data, the following rules apply.

For features that have asStandardCertain set to True, asBuiltCertain is set to True.

For features that have asStandardChangeable set to True, and no other feature with the same feature id has asBuiltCertain set to True, then that feature has asBuiltCertain set to True.

Only the features where asBuiltCertain is set to True are returned.

RESPONSE STATUS CODES

The following table describes response status codes returned by the Feature Exchange service.

Code	Description
200	OK [Success]
400	The request is improperly formed. The server cannot process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).
401	The request has not been applied because it lacks valid authentication credentials for the target resource.
404	The requested vehicle (VIN/STYLEID) cannot be found.
500	Internal Server Error - The server encountered an unexpected condition that prevented it from fulfilling the request.

VIN Validation Handling

If an invalid VIN is passed in a request, an attempt to reconcile the issue is made so that a valid VIN is used in the request. For example, if the VIN is passed with an O, Q, q or o those characters are replaced with a zero (0). If the VIN is passed with an I or i those characters are replaced with a one (1). Also, if the VIN is passed with a special characters like $, !, they are removed in an attempt to create a valid VIN. In this case, the id response attribute contains the invalid VIN and the vinDecoded response attribute contains the corrected VIN and the response contains a full response for the decoded VIN.

If an invalid VIN is passed in a request that cannot be corrected, the result is a 404 response. An example is shown below.


```
{
    "error": true,
    "message": "Invalid vin submitted:  1FTEABCDI cannot be decoded",
    "executionTimeMS": 0,
    "copyright": "Copyright 2019 Autodata Solutions, Inc.",
    "result": {
        "id": "1FTEABCDI",
        "idType": "VIN",
        "isBuildData": false,
        "vehicles": [],
        "vehicleFeatures": [],
        "status": 404
    }
```

```
}
```

INTEGRATING WITH THE SERVICE

Once you are ready to start developing your application, you must properly secure your usage of services by integrating a security token protocol into the Authorization header of the request to the service. The Shared Secret Security Protocol guide describes how to do this.

To access the Shared Secret Security Protocol guide:

1. Once you have registered, as described in the next section, sign into the portal.
2. On the Dashboard, click the **APIs** button.
3. On the APIs page, click the hyperlink name for an API.
4. On the web service page, in the left navigation menu, click **Documentation**.
5. On the Documentation page, click the **Shared Secret Security Protocol.pdf** link to open the guide.

PERFORMING A HEALTH CHECK

To perform a health check, you are required to get a RSS reader plugin for your Chrome internet browser. Once that is in place go to the following link:
https://status.chromedata.com/ns/studyprice

This page shows the current response time as well as the status over the last 7 days. In the top right, click the **RSS** icon. In the window that opens, click the **Subscribe** button for status updates.
If you have an RSS reader plugin installed in Chrome, you can subscribe from the main page or if you expand the item, there's a link node:

```
<item>
<title>studyPRICE (Live) up</title>
<link>
https://www.site24x7.com/sv.do?id=OFWNBpKLR9eB29pMvbNbqfdcR88IU_1xNMk2HvWNZ1B1EHweO0BWFz3nXRFCUwnwrRtHbxD2
0_X4m5YxSqPp3DNlMurk86Ak0JtRaCA-c1U%3D&urlid=H1xmHeVFt-zDuwmMWU4HrjCq36ELC5SJGmIhKwFjVxY%3D
</link>
```

AUTODATA
SOLUTIONS