JDPOWER

StudyPrice

Integration Guide

February 2026

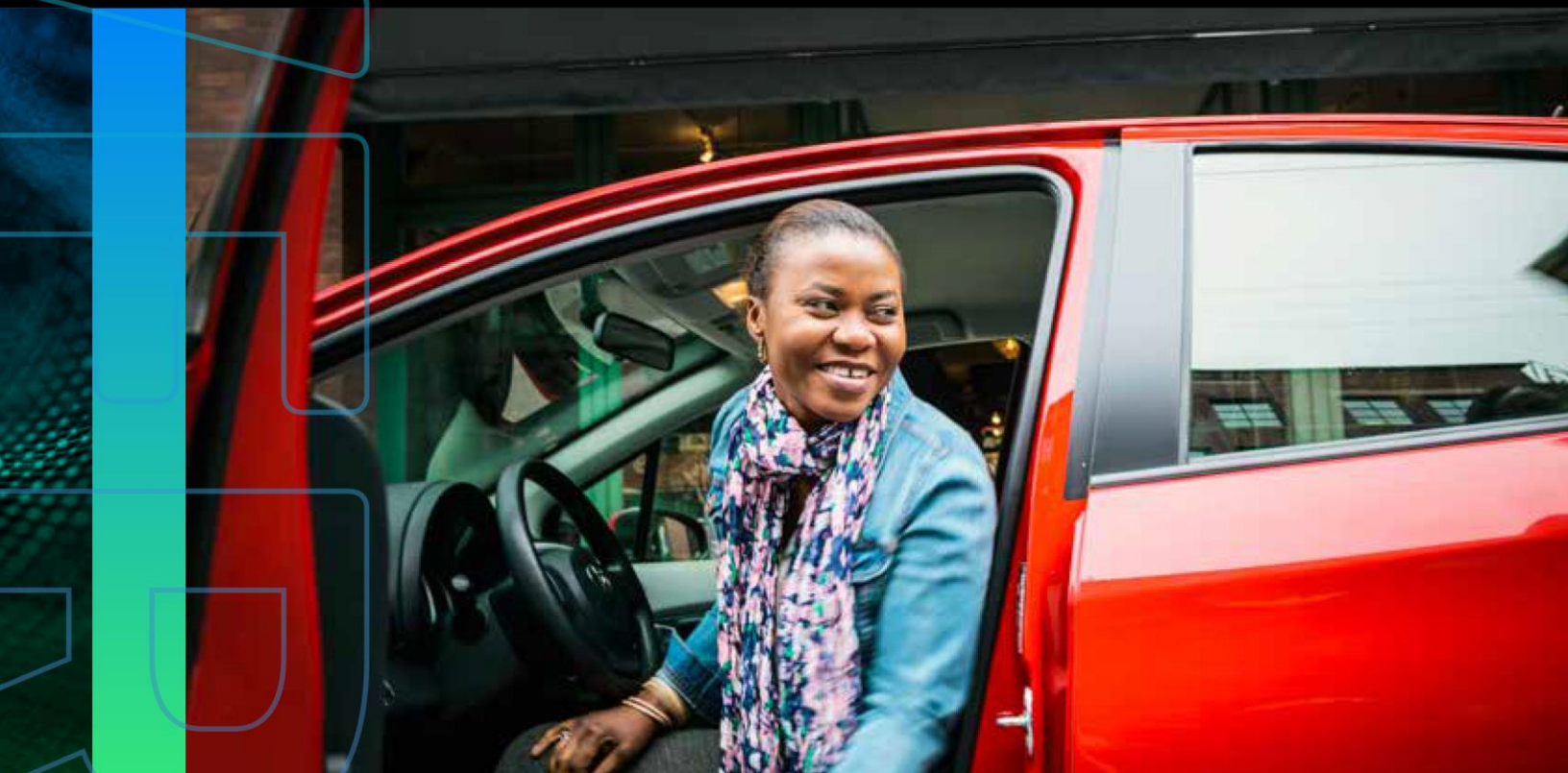


TABLE OF CONTENTS

DOCUMENT OVERVIEW

This guide provides step-by-step instructions on how to make requests using the Test Client within the Portal.

It also describes how to:

- Get a list of feature key answers
- Get feature benefit statements for feature key answers
- Get a list of features for a vehicle or vehicles

The Integrating with the Service section directs you to where you can get additional information for integrating StudyPRICE with your application.

Related Documentation

Document	Description
API Reference	The StudyPRICE API reference is available in a Swagger UI format within the Portal. It describes the service, each operation, each input attribute, and each output attribute.
Batch Guide	The StudyPRICE Batch Integration guide describes how to submit VINs in input files to be processed. It also describes the output files generated by the service.
Feature List	The Feature List document provides a comprehensive list of the features returned by the StudyPRICE service. This list includes feature ID's and feature keys for each feature in addition to other important feature information such as feature section and sub-section.
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 operation, 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 but require a way to identify vehicles in requests within the StudyPRICE 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 the StudyPRICE Integration 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 will 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 StudyPRICE 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 preventing theft as well as other features that help describe the vehicle.

This API has the following operations:

PUT /featuredictionary OR **GET /featuredictionary** – Returns definitions for the feature key answers used in generating the feature descriptions in the service.

GET /featuredictionary/csv – Returns all the feature key answers in csv format.

PUT /featuredefinition - Returns feature benefit statements for feature key answers.

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

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

Note: For weight and measurement related data returned, standardWeightUnits attribute defines the unit of measure for the vehicle's standard weight attribute and numericUnits defines the unit of measure for the numeric feature, if applicable. In addition, weight data is returned as rounded figures and prices returned are in United States (US) dollars for US styles and Canadian (CA) dollars for CA styles.

USING THE DEVELOPER PORTAL TO MAKE REQUESTS

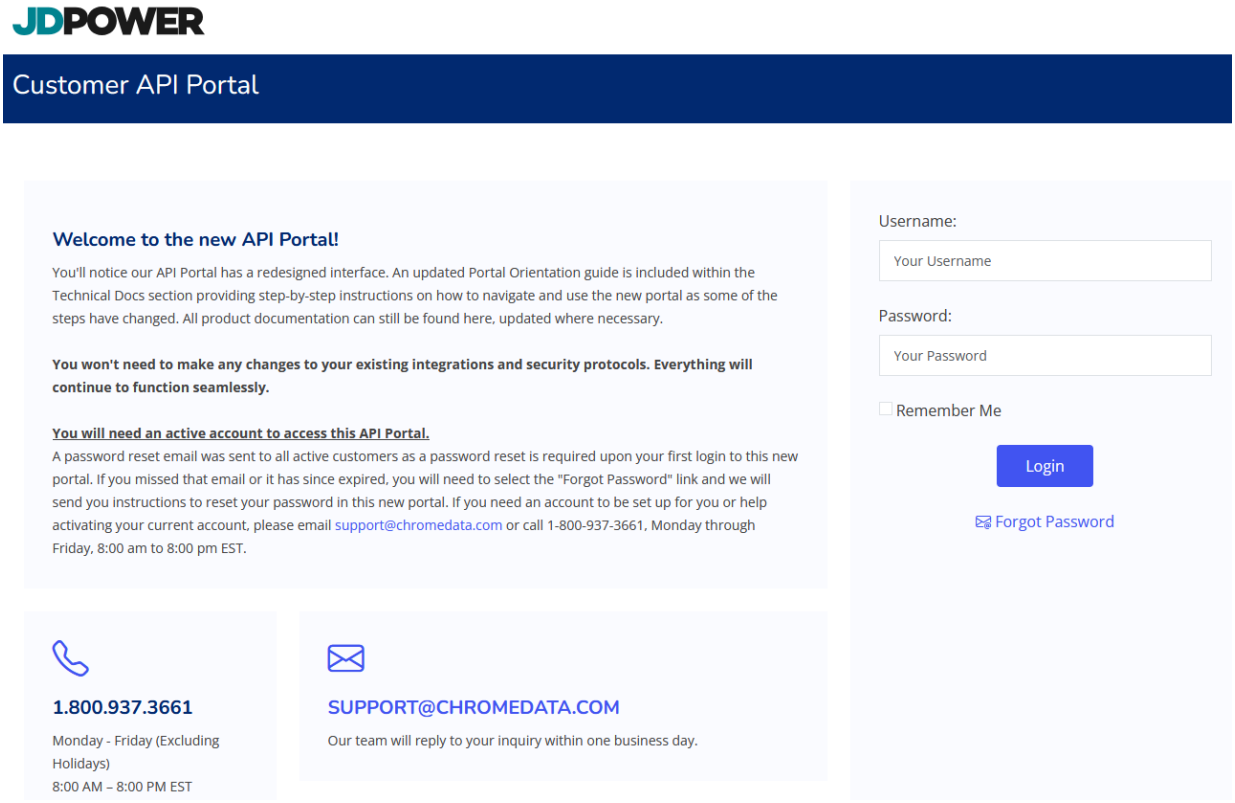
There are two ways to make requests using the JDP Developer Portal. Requests to the service can be made via the swagger documentation provided or by using the Test Client in our API Portal.

The following steps describe how to make requests using the Test Client.

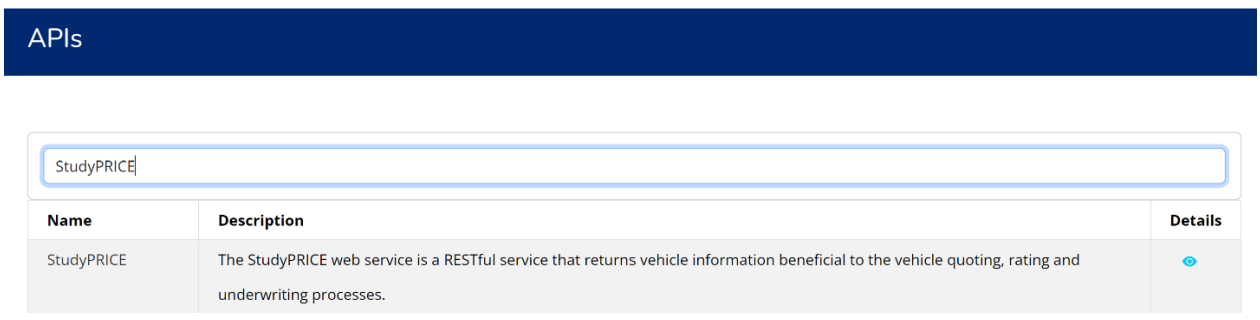
To execute a PUT request:

The following provides step-by-step instructions on how to make a PUT request.

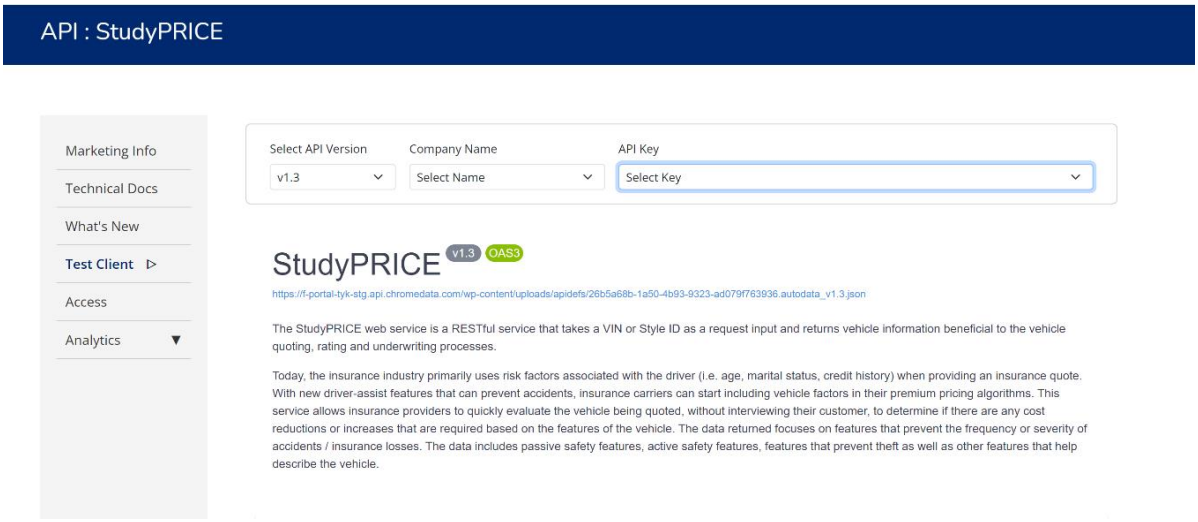
1. Log in to the JD Power API Portal: https://portal.jdpower.com/



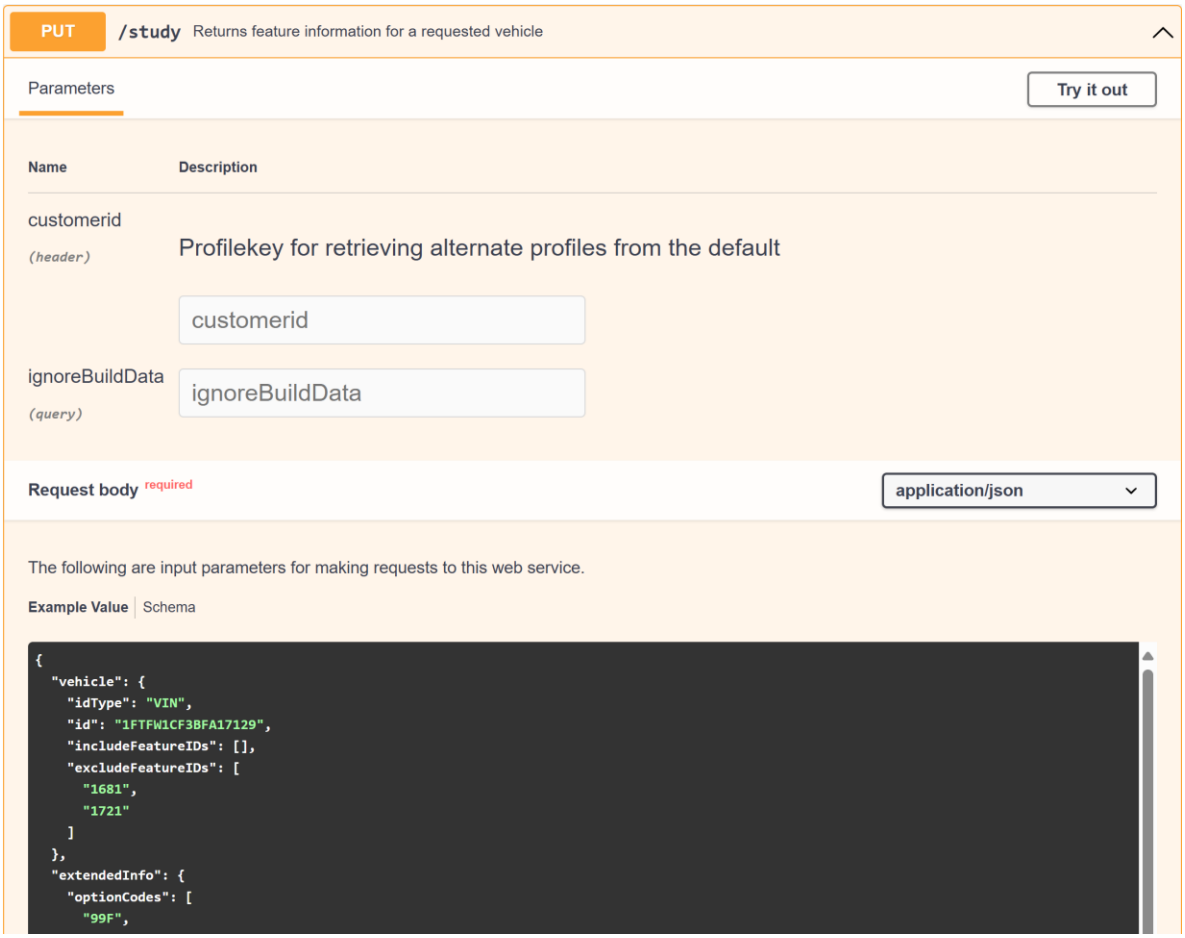
2. Find the StudyPRICE API and click the eye icon associated with it (under the Details column).



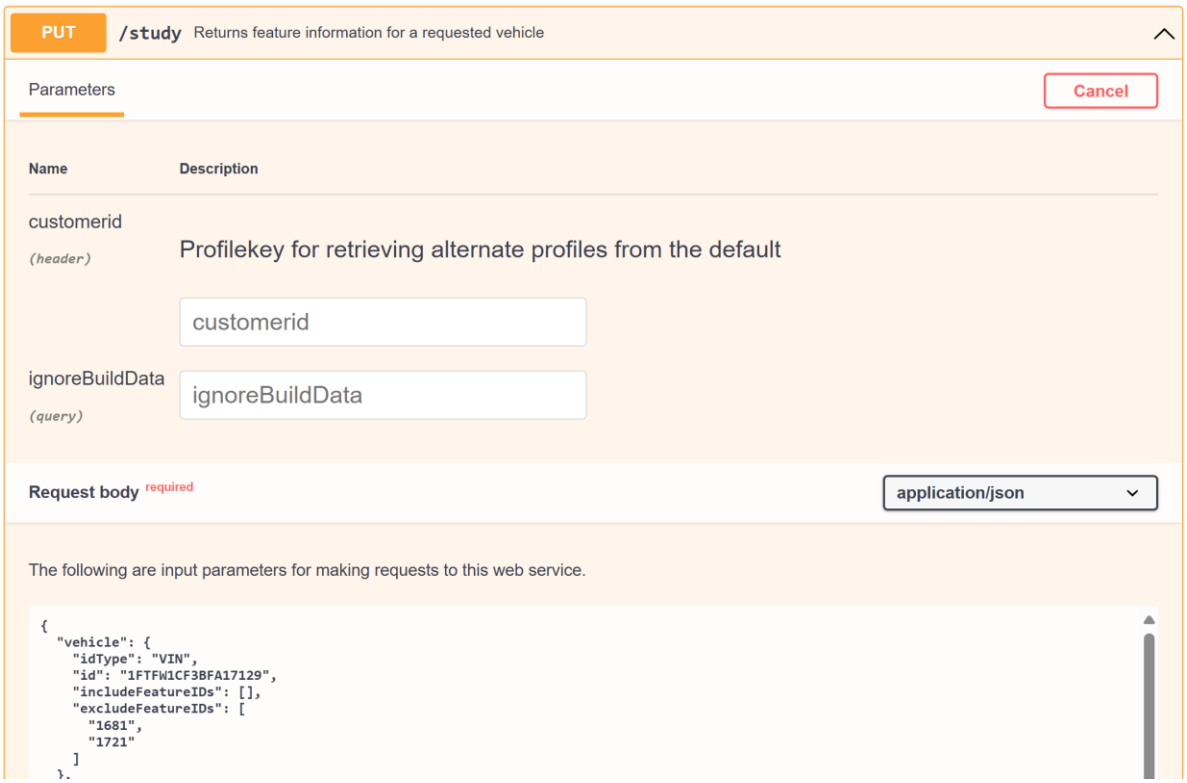
3. In the left navigation menu, click Test Client.
4. On the Test Client page, select the API Version, company name, and API Key.



5. Scroll down, select a PUT operation from the list of available operations.



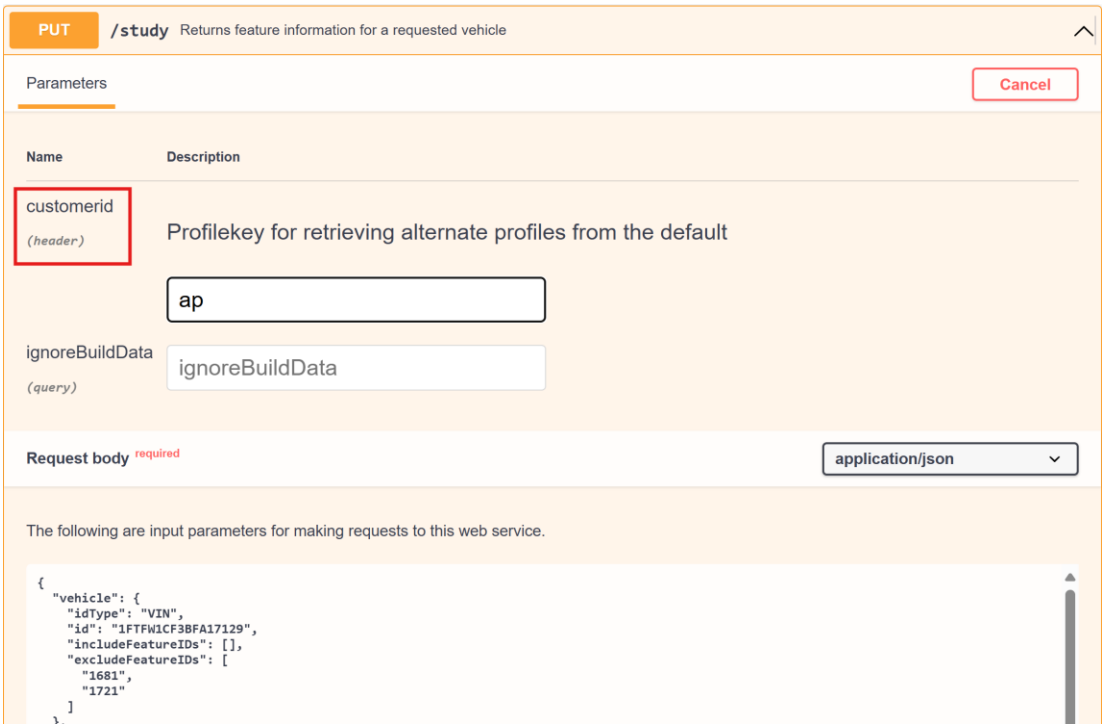
6. Click Try it out. The text fields are now editable.



7. Fill in the relevant information and request parameters.

   Note: Refer to the various Operation sections later in this guide for details on the parameters for each operation.

8. Optionally for clients who have more than one profile, to filter the feature set returned in the response by a specific profile, specify the customer profile identifier in the “customerid” field as part of the header parameters being supplied in the service request.



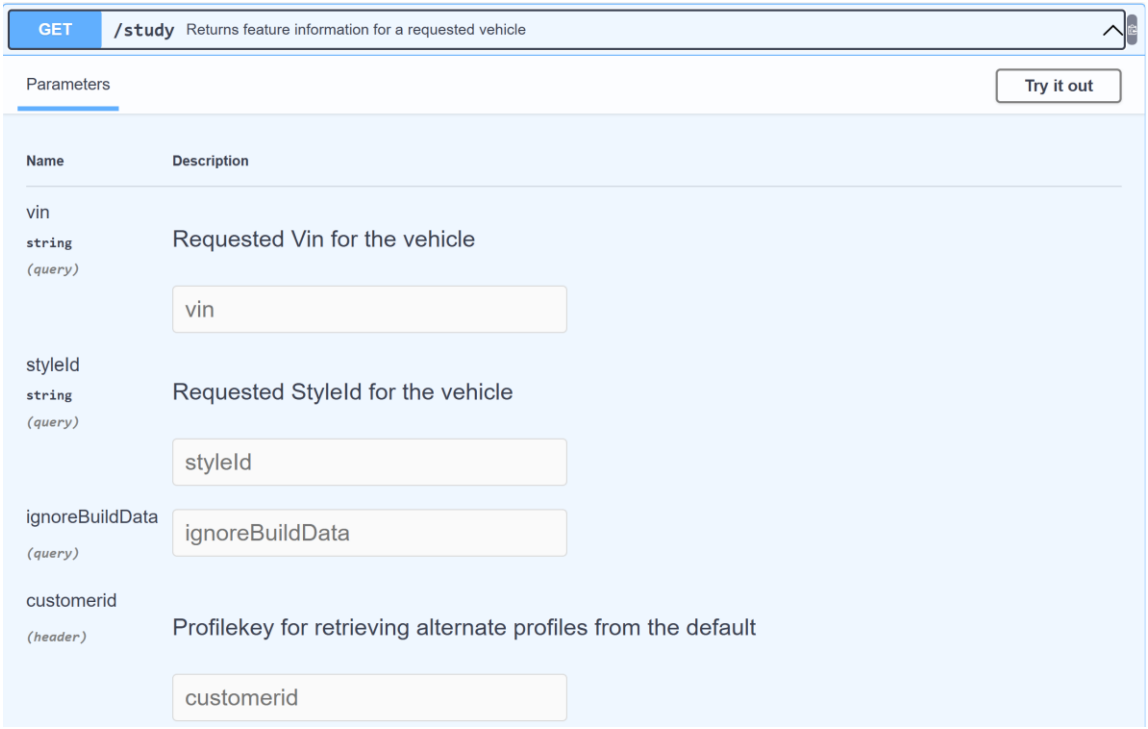
Note: If the customer profile identifier does not exist, the default profile is used.

9. Click Execute.
10. The request is sent, and response details are returned below on the page.

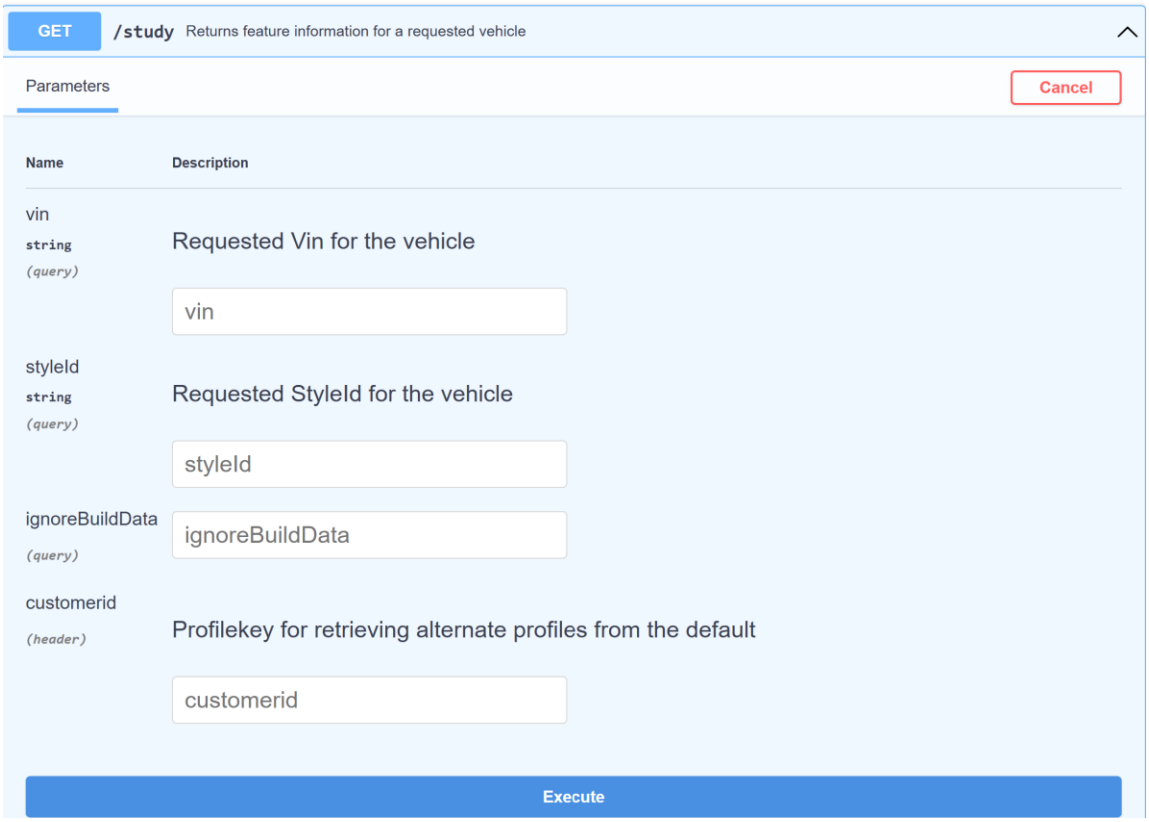
To execute a GET request:

The 1-4 steps for executing a GET request are the same as executing a PUT request.

5. Scroll down, select a GET Operation from the list of available operations.



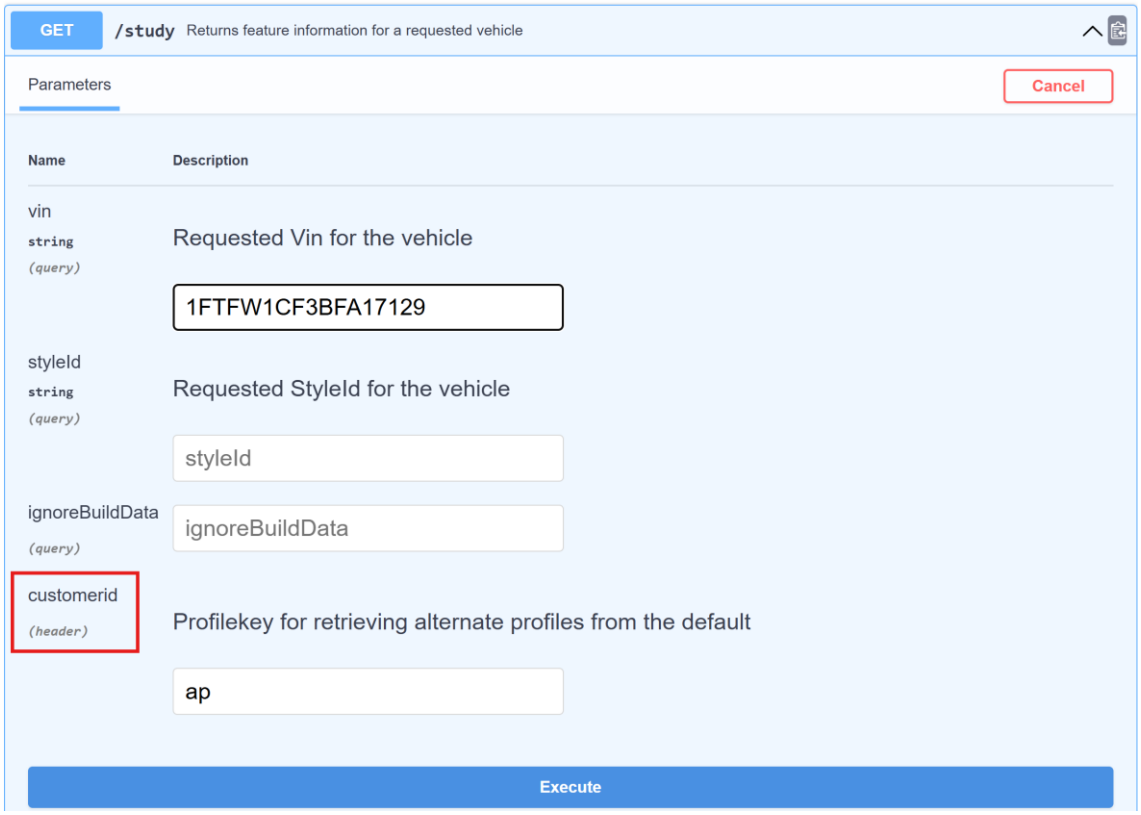
6. Click Try it out. The text fields are now editable.



7. Fill in the relevant information into the text fields.

   Note: Refer to the various Operation sections later in this guide for details on the parameters for each operation.

8. Optionally for clients who have more than one profile, to filter the feature set returned in the response by a specific profile, specify the customer profile identifier in the “customerid” field as part of the header parameters being supplied in the service request.



   Note: If the customer profile identifier does not exist, the default profile is used.

9. Click Execute.

9. The request is sent, and response details are returned below on the page.

GET ALL THE FEATURE KEY ANSWERS

Use the/featuredictionary operation to get definitions for the feature key answers used in generating the feature descriptions in the service.

Request Parameters

This operation supports both GET and PUT methods.

GET

This operation does not take any parameters in the request object;

PUT

In the request body, provide a JSON object with the following parameter.

Attribute	Description
featureKeyAnswers	Identifies the feature key answers for which to return feature benefits information. Use the /featuredictionary operation to retrieve a list of featureKeyAnswers that can be listed here.
adasOnly	Returns all the ADAS features.

Request Example

```
{
    "featureKeyAnswers": ["FDA0203","OIC1502"]
}
```

However, you can use the operation **GET featuredictionary/csv** to get all the feature key answers return in csv format. **Note:** While making the request, set headers to Accept to "text/csv" and Content-Type to “text/csv”.

Response

If successful, a 200 response code is returned and the master list of feature key answers is in the response body.

The response object for the /featuredictionary operation includes the following attributes.

Attribute	Description
featureKeyAnswer	Contains a feature key answer.
definition	Contains the feature definition.

Attribute	Description
status	Contains the status.
isNumeric	Contains a flag that when set to T indicates that the feature returns a numeric value. Features that return numeric values do not currently return information in the /featuredefinition operation.
sectionName groupName	Contains a user-friendly name for the main feature category. Contains a user-friendly name for a feature sub-category.
familyName	Contains a user-friendly name describing the feature key answer.
featureKeyName	Contains a user-friendly name describing the feature key.
featureKeyValue	Contains the value returned for the feature key.
isAdas genericLocation	Contains a flag that when set to True indicates that feature is an ADAS feature. Contains the generic repairing location of the vehicle.

Response Examples

Response in .json Format Example 1


```
"featureKeyList": [
            {
                "featureKeyAnswer": "FDA0203",
                "definition": "This transmission feature allows the driver to shift gears manually without depressing the clutch pedal.",
                "status": 200,
                "isNumeric": "F",
                "sectionName": "Mechanical",
                "groupName": "Transmission extras",
                "familyName": "Feature",
                "featureKeyName": "sequential shift controls",
                "featureKeyValues": "with steering wheel controls",
                "isAdas": "FALSE",
                "genericLocation": "Exterior front including engine"

            },

            {
                "featureKeyAnswer": "OIC1502",
                "featureKeyAnswer": "The vehicle can record front-facing video.",
                "status": 200,
```

```
                "isNumeric": "F",
                "sectionName": "Functional",
                "groupName": "Instrumentation",
                "familyName": "Display",
                "featureKeyName": "traffic video recording",
                "featureKeyValues": "front",
                "isAdas": "TRUE",
                "genericLocation": ""

            }
            …
        ]
```


Response in .json Format Example 2


```
"country": "US",
"language": "en",
"features": [
             {
                "featureKeyAnswer": "FDA0203",
                "definition": "This transmission feature allows the driver to shift gears manually without depressing the clutch pedal.",
                "status": 200,
                "isNumeric": "F",
                "sectionName": "Mechanical",
                "groupName": "Transmission extras",
                "familyName": "Feature",
                "featureKeyName": "sequential shift controls",
                "featureKeyValues": "with steering wheel controls",
                "isAdas": "FALSE",
                "genericLocation": "Exterior front including engine"

            },
            {
                "featureKeyAnswer": "OIC1502",
                "featureKeyAnswer": "The vehicle can record front-facing video.",
                "status": 200,
                "isNumeric": "F",
                "sectionName": "Functional",
                "groupName": "Instrumentation",
                "familyName": "Display",
                "featureKeyName": "traffic video recording",
                "featureKeyValues": "front",
```

```
            "isAdas": "TRUE",
            "genericLocation": ""

        }
    ]
```


Response in .csv Format Example

Request URL

```
https://api.jdpower.com/StudyPRICE/v1.3/featuredictionary/csv?adasOnly=false
```

Server response

Code	Details
200	Response body


```
featureKeyAnswer,definition,status,isNumeric,sectionName,groupName,familyName,featureKeyName,featureKeyValue,isAdas,genericLocation
"FDA0203","This transmission feature allows the driver to shift gears manually without depressing the clutch pedal.","200","F","Mechanical","Transmission extras","Feature","sequential shift control","with steering wheel controls","FALSE","Exterior front including engine",
"FDA0202","This automatic transmission feature gives shift control to the driver. It operates like a conventional automatic, but includes a program and a mechanism that lets the driver shift gears manually without depressing the clutch pedal.","200","F","Mechanical","Transmission extras","Feature","sequential shift control","yes","FALSE","Exterior front including engine",
"FMA1002","The driver can select the aggressiveness of the regenerative brakes to control the deceleration of the vehicle and the level of energy that is sent to the battery.","200","F","Mechanical","Brakes","Type","driver selectable regen levels","yes","FALSE","Exterior driver side",
"OIC1504","The vehicle can record front, side, and rear-facing video.","200","F","Functional","Instrumentation","Display","traffic video recording","surround","TRUE","",
"OIC1503","The vehicle can record front and rear-facing video.","200","F","Functional","Instrumentation","Display","traffic video recording","front and rear","TRUE","",
"OIC1502","The vehicle can record front-facing video.","200","F","Functional","Instrumentation","Display","traffic video recording","front","TRUE","",
"OQA0902","A dedicated light that illuminates the trailer hitch.","200","F","Functional","Trailering","Type","trailer hitch light","yes","FALSE","",
"LEH0402","An anti-whiplash head restraint actively cushions the head and neck in the event of a collision.","200","F","Functional","Exterior","Seats","Rear head restraint","anti-whiplash","yes","FALSE","",
```


Download

GET FEATURE BENEFIT STATEMENTS FOR FEATURE KEY ANSWERS

The /featuredefinition operation returns feature benefit statements for feature key answers.
Note: This endpoint will be sunset soon.

Request Body

In the request body, provide a JSON object with the following parameter.

Attribute	Description
featureKeyAnswers	Identifies the feature key answers for which to return feature benefits information. Use the /featuredictionary operation to retrieve a list of featureKeyAnswers that can be listed here.

Request Example


```
{
    "featureKeyAnswers":
["HQB0411","HQB0412","HQB0413","HQB0414","HQB0415","HQB0416","HQB0417","HQB0418","HQB0419","HQB0420","HQB0
421","HQB0422","HQB0423","HQB0424","HQB0425","HQB0501","HQB0502","HQB0503","HQB0504","HQB0505","HQB0506","
HQB0507","HQB0508","HQB0509","HQB0510","HQB0511","HQB0512","HQB0513","HQB0514","HQB0515","HQB0516","HQB051
7","HQB0518","HQB0519","HQB0520","HQB0521","HQB0522","HQB0523","HQB0524","HQB0525","HQB0526","HQB0527"]
}
```


Response

If successful, a 200-response code is returned and a list of feature definitions is in the response body.

The response object for the /featuredefinition operation includes the following attributes.

Attribute	Description
country	Contains the country for the feature benefit description.
language	Contains the language for the feature benefit description.
features	Contains an array of feature definitions for the requested feature key answers.
featureKeyAnswer	Contains the requested feature key answer.
definition	Contains the feature benefit statement for the requested feature key answer.
status	Contains the status for the feature benefit statement. If a feature benefit statement exists in the catalog, a status of 200 is returned. If it does not exist in the catalog, a 404 (Not found) is returned.
message	Contains a system message indicating the reason for a status, other than 200.

Response Examples


```
"country": "US",
"language": "en",
"features": [
        {
            "featureKeyAnswer": "HQB0411",
            "definition": "The wheel size is the diameter of the rim.",
            "status": 200
        },
        {
            "featureKeyAnswer": "HQB0412",
            "definition": "The wheel size is the diameter of the rim.",
            "status": 200
        },
        {
            "featureKeyAnswer": "HQB0413",
            "definition": "The wheel size is the diameter of the rim.",
            "status": 200
        }
    ]
```

GET A LIST OF FEATURES FOR A VEHICLE

Use the GET /study or PUT /study operation to get a list of features for a vehicle.

GET /study

This method supports an inline VIN or STYLEID parameter.

Example:

- /study?vin=1G1PC5SB1E7428104
- /study?styleId=401158

- This method only support VIN or STYLEID in a single request.
- This method does not support other input parameters as there is no request body.

Request Body

PUT /study

In the request body, provide a JSON object with the following parameters.

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

Attribute	Description
vehicle	Identifies the vehicle(s) for which features information is being requested. Also, identifies the features to include or exclude in the results.
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. If the idType is styleId, the id could be 330161. **Note:** Data is available for older vehicles, but for VINs for which catalog data is not available, or where the model year is older than 1992, partial responses may return.
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 StudyPRICE 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 StudyPRICE Feature List document, which you can access through the Portal.
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.

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.
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.
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 trimId	Identifies a third-party id type for the trim. Currently, the supported value includes STYLE. 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.

Request Examples

Return All Features

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

Filter the List of Features Returned

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

Reduce the Number of Styles Returned for A VIN

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

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 need 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.

Response

If successful, a 200 response code is returned for a full response that includes vehicle details and features. Or a 206 response code is returned for a partial response that includes vehicle details.

The response object for the /study operation includes the following 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.

Attribute	Description
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	Flag indicating whether build data was available at time of decode. When set to TRUE, build data is returned, if FALSE, catalog data is returned. For example, TRUE.
isEngineeredData	Flag indicating whether engineered data was available at time of decode. When set to TRUE, engineered data is returned, if FALSE, and isBuildData is FALSE, catalog data is returned. For example, TRUE.
manufacturer	Contains the vehicle manufacturer.
vehicles	Contains the vehicle description. For an example of the vehicle description, refer to Vehicles Section – Example 1: on page 26.
styleId	Contains the styleId of a vehicle.
country	Contains the country related to the styleId(s) returned for the requested vehicleId. Values can be US or CA.
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	Flag indicating whether the MSRP value came from a build data record or from catalog data. If set to true, the MSRP value came from a build data record. If set to false, the MSRP value came 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.

Attribute	Description
wheelbase	Contains the wheelbase of a truck in inches.
standardCurbWeight	Contains the standard curb weight of a vehicle.
standardPayload	Contains the standard payload of a vehicle.
standardGVWR	Contains the standard gross vehicle weight rating of a vehicle.
standardTowingCapacity	Contains the standard towing capacity of a vehicle.
standardFrontGAWR	Contains the standard front Gross Axle Weight Rating.
standardRearGAWR	Contains the standard rear Gross Axle Weight Rating.
standardFrontCurbWeight	Contains the standard front curb weight.
standardRearCurbWeight	Contains the standard rear curb weight.
standardMaxGVWR	Contains the standard maximum gross vehicle weight of the vehicle.
standardDryWeight	Contains the standard dry weight of the vehicle.
standardWeightUnits	Contains the unit of measure for the vehicle standard weight attributes. For example, 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.

Attribute	Description
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.
segments	Contains the segments returned for the vehicle.
code	Contains the classification group.
description	Contains the generic classification of a vehicle.
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.
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.
isAdasFeature	Contains a flag that when set to True indicates that feature is an ADAS feature. This data is available with additional licensing.
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.
locationId	Contains the location id of the feature. This data is available with additional licensing.
location	Contains the collision location of the vehicle. This data is available with additional licensing.

Attribute	Description
genericLocation	Contains the generic repairing location of the vehicle. This data is available with additional licensing.
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 26.
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.
isNotInstalled	Contains a flag that when set to True indicates that the feature is not installed or is optionally available on the vehicle. If multiple styleIds are returned for a vehicle, this flag will return as True only if the feature is not installed or optionally installed on any of the styles. This data is available with additional licensing.
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 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.
vehicleOptions	Contains a list of option codes returned from the build data record when isBuildData is set to true. This is no longer available and replaced with vehicleOptionsWithDescriptions.
vehicleOptionsWithDescriptions	Contains a list of option codes returned from the build data record along with their corresponding descriptions where available. This is disabled by default and available upon request.

Attribute	Description
model_secondary	When a manufacturer groups a collection of StyleIDs by Series (i.e., Mercedes S-Class) in a model name, "model_secondary" can be used to identify the model within the series (i.e., S 500). This data is disabled by default and available upon request.
cleanRetailValue	Contains national retail valuation for a given VIN with clean condition. This data requires additional licensing.
status	Contains the status of the response for each VIN sent in the request. For example, 200 or 404. 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. For all the possible status codes the service can return, refer to the Response Status Codes section on page 35.

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.

1. For features that have asStandardCertain set to True, asBuiltCertain is set to True.
2. 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.
3. Only the features where asBuiltCertain is set to True are returned.

Response Examples

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.

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

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,
    "manufacturer": General Motors,
    "vehicles": [],
    "vehicleFeatures": [],
    "status": 200     }
```

Note: In the above example, vehicles and vehicleFeatures are empty. The following examples detail what they contain when they are 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": [
```

```
{
    "buildDataMSRP": false,
    "styleId": 372610,
    "country": "US",
    "year": "2015",
    "make": "Ford",
    "model": "F-150",
    "trim": "XLT",
    "baseMSRP": 40050,
    "asBuiltEstimatedMSRP": 40845,
    "isBuildDataMSRP": false,
    "bodyStyle": "SuperCrew Cab Styleside",
    "boxStyle": "Styleside",
    "doors": "4",
    "drive": "4x4",
    "wheelbase": "145.0",
    "standardCurbWeight": 4696,
    "standardPayload": 1600,
    "standardGVWR": 6350,
    "standardTowingCapacity": 7100,
    "standardFrontGAWR": 3450,
    "standardRearGAWR": 3300,
    "standardFrontCurbWeight": 2781,
    "standardRearCurbWeight": 1915,
    "standardWeightUnits": "lbs.",
    "exteriorColor": {
        "optionalColors": [
            {
                "genericDescription": "Green",
                "description": "Green Gem Metallic"
            }
        ]
    },
    "interiorColor": {
        "optionalColors": [
            {
                "genericDescription": "Gray",
                "description": "Medium Earth Gray"
            }
        ]
    }
}
                                                                    ],
```

Note: This information is repeated for each style returned.

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,
              "isAdasFeature": false,
              "locationId": 25,
              "location": "Exterior front including engine",
              "genericLocation": "Engine Shot",
              "featureKeyAnswers": [
                  "FAB09=02"
              ],
              "styles": [
                  {
                      "styleIds": [
                          "372641",
                          "372630",
                          "372610",
                          "372631",
                          "372612"
                      ],
                      "asIccStandardCertain": true,
                      "asBuiltCertain": false,
                      "asStandardCertain": true,
                      "asStandardChangeable": false,
                      "asAvailable": false,
                      "isNotInstalled": 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, the feature Wheel hub covers is linked to styleId 372612.


```
{
                "section": "Exterior and Appearance",
                "subSection": "Wheels",
                "featureName": "Wheel hub covers",
                "featureNameNoBrand": "Wheel hub covers",
                "featureId": "12720",
                "featureKey": "12720-HQD0302",
                "isNumeric": false,
                "isAdasFeature": false,
                "locationId": 3,
                "location": "Exterior driver side",
                "genericLocation": "Profile - facing left",
                "featureKeyAnswers": [
                    "HQD03=02"
                ],
                "styles": [
                    {
                        "styleIds": [
                            "372612"
                        ],
                        "asIccStandardCertain": false,
                        "asBuiltCertain": false,
                        "asStandardCertain": false,
                        "asStandardChangeable": true,
                        "asAvailable": false,
                        "isNotInstalled": 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": "15090",
                "styleIds": [
                    "372630",
```

```
            "372641",
            "372631"
        ]
    },
    {
        "featureId": "19660",
        "styleIds": [
            "372630",
            "372610",
            "372612"
        ]
    },
    {
        "featureId": "12510",
        "styleIds": [
            "372641",
            "372630",
            "372610"
        ]
    },
    {
        "featureId": "15250",
        "styleIds": [
            "372630",
            "372641",
            "372631"
        ]
    },
    {
        "featureId": "19520",
        "styleIds": [
            "372641",
            "372630",
            "372610",
            "372631"
        ]
    },
```


vehicleFeatures Section - Example 4:

Some features return numeric values. In the following example the feature is Engine displacement: 213 cu.in. and it has three additional attributes: numericFeatureName, numericValue and numericUnits.


```
{
```

```
        "section": "Powertrain and Mechanical",
        "subSection": "Engine specs",
        "featureName": "Engine displacement: 213 cu.in.",
        "featureNameNoBrand": "Engine displacement: 213 cu.in.",
        "featureId": "10120",
        "featureKey": "10120-FAC0121340",
        "isNumeric": true,
        "isAdasFeature": false,
        "numericFeatureName": "Engine displacement",
        "numericValue": 213,
        "numericUnits": "cubic inches",
        "locationId": 25,
        "location": "Exterior front including engine",
        "genericLocation": "Engine Shot",
        "featureKeyAnswers": [
            "FAC01=21340"
        ],
        "styles": [
            {
                "styleIds": [
                    "372610",
                    "372612"
                ],
                "asIccStandardCertain": false,
                "asBuiltCertain": false,
                "asStandardCertain": false,
                "asStandardChangeable": true,
                "asAvailable": false,
                "isNotInstalled": false
            }
        ]
    },
```

Partial Response - Example 1:

If older VINs for which there is no feature data are sent in the request they may result in partial responses, as shown below.


```
{
        "id": "1P3BM18D9JY119159",
        "idType": "VIN",
        "vinDecoded": "1P3BM18D9JY119159",
        "isBuildData": false,
        "vehicles": [
            {
                "styleId": 261449,
                "year": "1988",
                "make": "Plymouth",
                "model": "Horizon",
                "trim": "America",
                "baseMSRP": -1.0,
                "asBuiltEstimatedMSRP": -1.0,
                "isBuildDataMSRP": false,
                "bodyStyle": "Hatchback 4 Dr.",
                "boxStyle": "5dr Hatchback America",
                "exteriorColor": {
                    "optionalColors": []
                },
                "interiorColor": {
                    "optionalColors": []
                }
            }
        ],
        "vehicleFeatures": [],
        "status": 206
    }
```


Note: Additional information may be returned in the service response based on optional features enabled in your client's profile

GET A LIST OF FEATURES FOR MULTIPLE VEHICLES

Use the GET /multistudy or PUT /multistudy operation to return features for up to five vehicles.

GET /mltistudy

This method supports up to five VINs or STYLEIDs delimited by comma.

Example:

- /multistudy?vin=1G1PC5SB1E7428104,1G1YY26W795110751
- /multistudy?styleId=45125,47896

- This method only support VIN or STYLEID in a single request.
- This method does not support other input parameters as there is no request body.

Request Body

PUT /mutlistudy

This operation accepts the same body request as the PUT /study operation. For more information, refer to the "Request Body" section on page 19.

The difference is that the /multistudy operation has an extra vehicles container and you can repeat the vehicle object up to five times within a single request.

Request Example

```
{
 "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

If successful, a 200 response code is returned for a full response that includes vehicle details and features for up to five vehicles. Or a 206 response code is returned for a partial response that includes vehicle details for up to five vehicles.

Response Examples

The response object for /multistudy requests provides 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.

For more information about the response, refer to the "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 need 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.

Response" section on page 20.

RESPONSE STATUS CODES

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

Code	Description
200	OK [Success]
206	The request was successfully processed, and a partial response is returned.
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 such as $, !, 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
    }
}
```

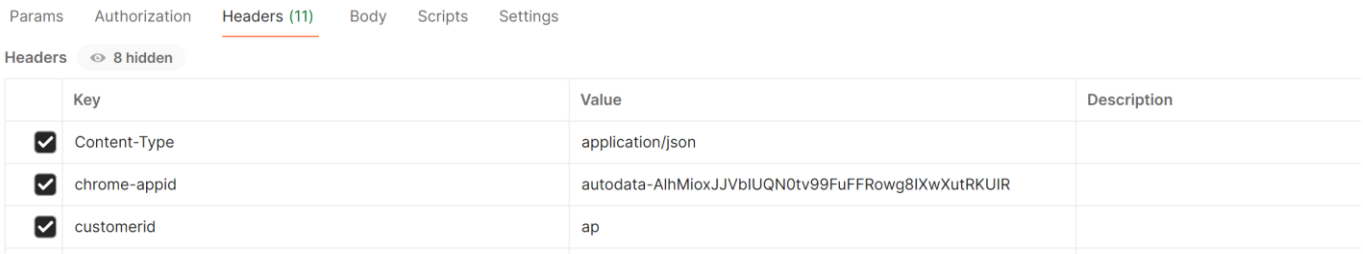

Specifying Customer Profile

To retrieve responses for a specific customer profile, the profile's unique identifier must be specified in the request. The "customerid" is the unique identifier for a customer profile. To ensure data for a specific profile is returned, there are two methods to provide the customerid: as an HTTP header **or** a URL query parameter.

If the "customerid" is omitted from a request, results associated with the account's default profile will be returned.

Using an HTTP header (Recommended Method):

In the screenshot below, the "customerid" is included directly in the header of the HTTP request, responses specifically associated with the profile "ap" will be returned.



Using a URL Query Parameter (Alternative Method): It's also possible to pass the "customerid" as a query parameter directly in the request URL.

Example: https://studyprice-api.jdpower.com/StudyPRICE/v1.3/study?customerId=ap

Note: The service response will be unique to the features enabled in your client's profile.

STYLE FLAG COMBINATION AND INTERPRETATION CHART

S/N	IsBuild Data	asIcc Standard Certain	asBuilt Certain	asStandard Certain	asStandard Changeable	asAvailable	isNot Installed	Interpretation	Additional Comment
1	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	Maybe Installed	Optional feature but there is no build data to confirm that the feature is installed
2	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	Maybe Not Installed	Optional removeable feature but there is no build data to confirm that the feature is not installed
3	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	Maybe Installed	Standard feature that can be optionally upgraded.
4	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	Maybe Not Installed	Standard not installed feature that can be optionally upgraded.
5	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	Installed	
6	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	Not Installed	
7	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	Maybe Installed	
8	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	Maybe Installed	
9	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	Installed	

10	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	Not Installed	
11	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	Not Installed	
12	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	Installed	
13	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	Installed	
14	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	Not Installed	
15	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	Installed	
16	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	Not Installed	
17	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	Installed	
18	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	Installed	
19	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	Installed	
20	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	Not Installed	

Notes: In cases where build data is false, the values of the flags need to match across all the styles returned for the Vin for the interpretation to match the chart above.

The chart above is a guide and does not represent all possible flag combinations in the service.

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, find the StudyPRICE API and click the eye icon associated with it.
3. On the web service page, in the left navigation menu, click **Technical Docs**.
4. On the Documentation page, click the **Security Guide.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>
```