Decode This API Documentation

Version: 1.1, Date: 3/13/2016

Decode This is exposing the decoder engine that powers the web site. This decoder can provide basic information on many vehicles from the 1950s through the current model year. By accessing the service you will be able to pass a VIN and in return get an XML document of decoded information related to that VIN.

Acceptable Use

The API is provided as a means of allowing the subscriber to use the decoded information in their websites and products. It is not to be resold as a standalone VIN decoder.

Getting Access to the API

In order to access the API you must subscribe to the site at the API Access level. This subscription will give you monthly access to the API for use in your web site. You can subscribe to API Access by visiting our Subscription page. Once your subscription is complete you will receive an API Key and will have immediate access to the API. You can view your API key on your User Info page.

Usage Cap

Note while there is no hard cap to monthly usage there is an implicit soft cap of 30,000 decoded VINs per month for the API subscription and 15,000 for the Basic API subscription.

The soft cap means that you will be alerted to the overage but the service will not immediately cease upon reaching that figure. If you require more than that number of decodes please contact us for additional support.

How to Access the API

The first thing you will need in order to use the API is your API Key. You can find you key by logging into Decode This with your username and password and then click the Account link. In the Settings section you will see your API Key. Please protect your key as if it was a password. You are responsible for all usage by your key.

REST Service URL:

```
https://www.decodethis.com/webservices/decodes/[vin]/[api_key]/[details].[format]
```

The `REST` service returns XML, JSON as well as JSONP.

The parameters are:

- **[vin]** - The VIN you wish to decode
- **[api_key]** - The API key for your account
- **[details]** - 0 = No equipment information, 1 = full equipment data
- **[format]** - xml = XML, json = JSON, jsonp = JSONP, defaults to XML

Note that JSONP requests can have an additional parameter: `callback`, which specifies the name of the callback method in the response. Without this parameter the method will be “callback"

REST Service

The REST service is quite easy, simply call the API URL and pass it your VIN and API Key. As an example:

https://www.decodethis.com/webservices/decodes/1P3EW65F4VV300946/Up4YFTid9CcPP3j3FfrQ/1.jsonp?callback=ajax_func

Parameters:

- **vin** - 1P3EW65F4VV300946
- **api_key** - Up4YFTid9CcPP3j3FfrQ
- **details** - 1 = Detailed information
- **format** - jsonp = JSONP
- **?callback** - name of callback function = ajax_func

Note that JSONP requests can have an additional parameter: callback, which specifies the name of the callback method in the response. Without this parameter the method will be “callback"

Responses

The XML Response

Using the XML format will result in a document that has the structure of a root node with zero or more child nodes.

Example:

```
<decode status='SUCCESS' version="DECODE" Valid="True" VIN="1P3EW65F4VV300946">
  <vehicle id="110153" year="1997" make="Plymouth" model="Prowler" trim="Base"
body="CONVERTIBLE 2-DR" engine="Gas V6" driveline="RWD"></vehicle>
</decode>
```

The root node has several key attributes:

- **VIN** - The VIN decoded
- **status** - The status code of the decode
- **version** - What type of decode this is
- **valid** - Whether the VIN passed is valid

Possible `status` codes are:

- **SUCCESS** - The VIN was successfully decoded
- **PARAMERR** - The decode request has invalid parameters
- **VINERR** - Invalid VIN (bad format)
- **SECERR** - Invalid API Key
- **CHECKERR** - Checksum digit is invalid
- **NOTFOUND** - VIN was well-formed but not found in our database

Any code other than “SUCCESS” does not count as a decode.

Possible `version` codes are:

- **CLASSIC** - A Classic VIN (A vehicle manufactured prior to 1981, less than 17 characters)
- **DECODE** - A standard 17-character VIN

THE JSON Response

Using a `.json` or `.jsonp` format type will return a valid JSON document. For example, the request:

```
https://www.decodethis.com/webservices/decodes/1P3EW65F4VV300946/Up4YFTid9CcPP3j3FfrQ/1.json
```

Will result the JSON document:

```
{
    "decode" : {
        "status" : "SUCCESS",
        "Valid" : "True",
        "vehicle" : [
            {
                "body" : "CONVERTIBLE 2-DR",
                "driveline" : "RWD",
                "engine" : "Gas V6",
                "Equip" : [
                    {
                        "name" : "Engine Type",
                        "unit" : "",
                        "value" : "Gas V6"
                    },
                    {
                        "name" : "Model Year",
                        "unit" : "",
                        "value" : "1997"
                    },

                    ...

                ],
                "id": "110153",
                "make": "Plymouth",
                "model": "Prowler",
                "trim": "Base",
                "year": "1997"

            }
        ],
        "version" : "DECODE",
        "VIN" : "1P3EW65F4VV300946"
    }
}
```

Possible `status` codes are:

- **SUCCESS** - The VIN was successfully decoded

- **PARAMERR** - The decode request has invalid parameters
- **VINERR** - Invalid VIN (bad format)
- **SECERR** - Invalid API Key
- **CHECKERR** - Checksum digit is invalid
- **NOTFOUND** - VIN was well-formed but not found in our database

Any code other than “SUCCESS” does not count as a decode.

Possible `version` codes are:

- **CLASSIC** - A Classic VIN (A vehicle manufactured prior to 1981, less than 17 characters)
- **DECODE** - A standard 17-character VIN

JSONP Response

JSONP output is the same as JSON output, but wrapped in a function call and returns the `Content-Type` of `application/javascript` instead of `application/json` as the `.json` format returns.

```
https://www.decodethis.com/webservices/decodes/1P3EW65F4VV300946/Up4YFTid9CcPP3j3FfrQ/1.json?callback=ajax_func
```

Results in:

```
ajax_func(
{
    "decode" : {
        "status" : "SUCCESS",
        "Valid" : "True",
        "vehicle" : [
            {
                "body" : "CONVERTIBLE 2-DR",
                "driveline" : "RWD",
                "engine" : "Gas V6",
                "Equip" : [

                ...

                "id": "110153",
                "make": "Plymouth",
                "model": "Prowler",
                "trim": "Base",
                "year": "1997"

            }
        ],
        "version" : "DECODE",
        "VIN" : "1P3EW65F4VV300946"
    }
}
);
```

Support

If you have any issues implementing the API please contact us. We’ll respond quickly to assist you in your implementation.