

# Package ‘vindecodr’

October 12, 2022

**Title** Provides an Interface to the Department of Transportation VIN Decoder

**Version** 0.1.1

**Description** Provides a programmatic interface in R for the US Department of Transportation (DOT) National Highway Transportation Safety Administration (NHTSA) vehicle identification number (VIN) API, located at [<https://vpic.nhtsa.dot.gov/api/>](https://vpic.nhtsa.dot.gov/api/).

The API can decode up to 50 vehicle identification numbers in one call, and provides manufacturer information about the vehicles, including make, model, model year, and gross vehicle weight rating (GVWR).

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 4.0.0)

**Imports** httr

**Suggests** purrr, stringr

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Christopher Burch [aut, cre] ([<https://orcid.org/0000-0001-6934-3325>](https://orcid.org/0000-0001-6934-3325))

**Maintainer** Christopher Burch <christopher.m.burch@gmail.com>

**Repository** CRAN

**Date/Publication** 2020-11-25 14:00:03 UTC

## R topics documented:

build_nhtsa_url . . . . .	2
check_vin . . . . .	3
check_vin_no_purrr . . . . .	3
check_vin_purrr . . . . .	4
decode_vin . . . . .	4
swap_letter . . . . .	5

swap_map . . . . .	5
valid_check_digit . . . . .	6
valid_vin_format . . . . .	6

<b>Index</b>	<b>8</b>
--------------	----------

---

build_nhtsa_url	<i>Build a NHTSA URL</i>
-----------------	--------------------------

---

## Description

A family of functions to build URLs for the National Highway Transportation Safety Administration (NHTSA) vehicle identification number (VIN) decoder API.

The `build_nhtsa_url()` function returns a closure containing the appropriate endpoint and file format request to pass to the NHTSA VIN API.

- `build_vin_url()` takes a single VIN in a character string and returns an appropriately-formatted url for a NHTSA API request via the `/vehicles/DecodeVINValues/` endpoint.
- `build_vin_batch_url()` takes up to 50 VINs in a character vector and returns appropriately-formatted url for a NHTSA API request via the `/vehicles/DecodeVINBatchValues/` endpoint.

## Usage

```
build_nhtsa_url(endpoint, format = "json", ...)
```

```
build_vin_url(vin, ...)
```

```
build_vin_batch_url(vin, ...)
```

## Arguments

endpoint	a string containing the appropriate endpoint. Candidate endpoints can be found at <a href="https://vpic.nhtsa.dot.gov/api/">https://vpic.nhtsa.dot.gov/api/</a>
format	the file format to return from the API, one of 'json', 'xml', or 'csv'. Defaults to 'json'.
...	additional arguments to passed on to derived builder functions
vin	a string containing the VIN to query.

## Value

- `build_nhtsa_url()` returns a function which will in turn build a url which points to the specified endpoint on the NHTSA API
- `build_vin_url()` returns a url as a string, formatted to query the NHTSA DecodeVinValues endpoint and decode a single VIN.
- `build_vin_batch_url()` returns a url as a string, formatted to query the NHTSA DecodeVinBatch Values endpoint and decode multiple VINs in one call.

**Examples**

```
vin_url_xml <- build_nhtsa_url("/vehicles/DecodeVINValues/", format = "xml")
build_vin_url("3VWLL7AJ9BM053541")
build_vin_batch_url(c("3VWLL7AJ9BM053541", "JH4KA3140KC015221"))
```

---

check_vin	<i>Verify VIN Validity</i>
-----------	----------------------------

---

**Description**

Examines provided VINs for valid length, characters, and check digit.

**Usage**

```
check_vin(vin, guess = FALSE)
```

**Arguments**

vin	A character vector of VINs to check. Wildcards (e.g. *) are NOT allowed.
guess	Logical. Should values for illegal characters be guessed?

**Value**

A logical vector of same length as the input vector.

**Examples**

```
vins <- c("WDBEA30D3HA391172", "3VWLL7AJ9BM053541")
check_vin(vins)
```

---

check_vin_no_purrr	<i>Verify VIN Validity Without Purrr</i>
--------------------	--

---

**Description**

Verify VIN Validity Without Purrr

**Usage**

```
check_vin_no_purrr(vin, guess = FALSE)
```

**Arguments**

vin	A character vector of VINs to check. Wildcards (e.g. *) are NOT allowed.
guess	Logical. Should values for illegal characters be guessed?

---

check_vin_purrr	<i>Verify VIN Validity Using Purrr</i>
-----------------	--

---

**Description**

Verify VIN Validity Using Purrr

**Usage**

```
check_vin_purrr(vin, guess = FALSE)
```

**Arguments**

vin	A character vector of VINs to check. Wildcards (e.g. *) are NOT allowed.
guess	Logical. Should values for illegal characters be guessed?

---

decode_vin	<i>Use the NHTSA API to Decode VINs</i>
------------	---

---

**Description**

Use the NHTSA API to Decode VINs

**Usage**

```
decode_vin(vin, ...)
```

**Arguments**

vin	either a single vehicle identification number in a character string, or multiple vehicle identification numbers in a character vector.
...	additional arguments passed to the url builder functions.

**Value**

a data frame with the VIN, Make, Model, Model Year, Fuel Type, and Gross Vehicle Weight Rating (GVWR) for the specified VINs.

**Examples**

```
## Not run:
# Decode a single VIN:
decode_vin("JHLRD68404C018253")

# Decode multiple VINs:
decode_vin(c("JHLRD68404C018253", "JH4DA9450MS001229"))

## End(Not run)
```

---

`swap_letter`*Replace a Letter in a Character Vector*

---

**Description**

Replace a Letter in a Character Vector

**Usage**

```
swap_letter(.string, .target, .replacement)
```

**Arguments**

`.string`        character vector  
`.target`        character to replace  
`.replacement`   character to substitute

**Value**

the modified string

---

`swap_map`*Replace Multiple Letters in a Character Vector*

---

**Description**

Replace Multiple Letters in a Character Vector

**Usage**

```
swap_map(.string, .targets, .replacements)
```

**Arguments**

`.string`        character vector  
`.targets`       characters to replace  
`.replacements`   characters to substitute

**Value**

the modified string

---

valid\_check\_digit      *Check for Valid VIN Check Digit*

---

### Description

Calculates the VIN check digit and compares it to VIN position 9. For US-based VINs, this determines if the VIN is valid. This may not apply to VINs from outside of the United States.

### Usage

```
valid_check_digit(vin, value = FALSE, guess = FALSE)
```

### Arguments

vin	character. The VIN to check. VINs must be complete, i.e. 17 digits with no wildcards.
value	logical. Should the calculated check digit be returned instead?
guess	logical. Should incorrect characters be replaced by the best guess at corrected characters? O -> 0 I -> 1 Q -> 0

### Value

If value is FALSE, a logical value is returned. If value is TRUE, a character is returned.

### Examples

```
valid_check_digit("WDBEA30D3HA391172") # True
valid_check_digit("WDBEA30D3HA391172", value = TRUE)
valid_check_digit("WDBEA3QD3HA39I172", guess = TRUE)
```

---

valid\_vin\_format      *Check VIN Length and Characters*

---

### Description

Checks that VINs are 17 characters long and will optionally check that disallowed characters (I, O, Q) are not present.

### Usage

```
valid_vin_format(vin, check_chars = FALSE)
```

### Arguments

vin	A character. Should be a properly formatted Vehicle Identification Number. Wildcards (e.g., '*') are acceptable.
check_chars	Logical. Should an error be thrown if the VIN contains illegal characters?

*valid\_vin\_format*

7

**Value**

Logical.

**Examples**

```
# Random VIN
valid_vin_format("3VWLL7AJ9BM053541")
# With wild card
valid_vin_format("3VWLL7AJ9BM*53541")
```

# Index

`build_nhtsa_url`, 2  
`build_vin_batch_url (build_nhtsa_url)`, 2  
`build_vin_url (build_nhtsa_url)`, 2

`check_vin`, 3  
`check_vin_no_purrr`, 3  
`check_vin_purrr`, 4

`decode_vin`, 4

`swap_letter`, 5  
`swap_map`, 5

`valid_check_digit`, 6  
`valid_vin_format`, 6