

# Package: altfuelr (via r-universe)

September 18, 2024

**Title** Provides an Interface to the NREL Alternate Fuels Locator

**Version** 0.1.0

**Description** Provides a number of functions to access the National Energy Research Laboratory Alternate Fuel Locator API [<https://developer.nrel.gov/docs/transportation/alt-fuel-stations-v1/>](https://developer.nrel.gov/docs/transportation/alt-fuel-stations-v1/). The Alternate Fuel Locator shows the location of alternate fuel stations in the United States and Canada. This package also includes the data from the US Department of Energy Alternate Fuel database as a data set.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**Depends** R (>= 3.1.0)

**Imports** httr, jsonlite, purrr, lubridate, dplyr, magrittr, stringr

**Repository** <https://burch-cm.r-universe.dev>

**RemoteUrl** <https://github.com/burch-cm/altfuelr>

**RemoteRef** HEAD

**RemoteSha** a767c53c4d23c95b85f19617413898e7d86a36b4

## Contents

all_stations . . . . .	2
altfuel . . . . .	2
altfuel_api . . . . .	5
alt_fuel_near . . . . .	6
count_results . . . . .	7
last_updated . . . . .	7
nearest_stations . . . . .	8
nrel_params . . . . .	9

stations	12
station_by_id	12
update_params	13

## Index 14

---

all_stations	<i>Find All Alternate Fuel Stations by Criteria</i>
--------------	---

---

### Description

Find All Alternate Fuel Stations by Criteria

### Usage

```
all_stations(api_key, params = nrel_params(limit = "all"))
```

### Arguments

api_key	Character. An authorized API key for the NREL API service. API keys can be requested at <a href="https://developer.nrel.gov/signup/">https://developer.nrel.gov/signup/</a>
params	A list of parameters for the API call. See ?nrel_params for options.

### Value

An object of class nrel\_api, containing content, parameters, and response.

### Examples

```
## Not run:
my_params <- nrel_params(fuel_type = "ELEC", ev_connector_type = "TESLA")
my_stations <- get_all_stations(MY_API_KEY, my_params)
stations(my_stations)

## End(Not run)
```

---

altfuel	<i>Alternate Fuel Stations in the US.</i>
---------	---

---

### Description

A dataset containing the alternate fuel stations in the United States and Canada tracked by the US DOE Alternate Fuels Data Center

### Usage

```
altfuel
```

**Format**

A data frame with 40279 observations of 64 variables:

**Fuel\_Type\_Code** The coded fuel type available: BD = BioDiesel, CNG = Compressed Natural Gas, ELEC = electric, E88 = Ethanol (E85), HY = Hydrogen, LNG = Liquefied Natural Gas, LPG = Propane.

**Station\_Name** The name of the fuel station.

**Street\_Address** The physical street address of the station.

**Intersection\_Directions** Brief additional information about how to locate the station.

**City** The name of the city in which the station is located.

**State** The two character code for the U.S. state or Canadian province/territory of the station's location.

**ZIP** The postal code for the station.

**Plus4** The plus-4 portion of the station's ZIP code (U.S. ZIP codes only).

**Station\_Phone** The telephone number of the fuel station

**Status\_Code** The current status of the station, given as code values: E = Available, P = Planned, T = Temporarily Unavailable

**Expected\_Date** For planned stations, the date the station is expected to open or start carrying alternative fuel. For temporarily unavailable stations, the expected date is the date the station is expected to reopen. This date is estimated but not guaranteed.

**Access\_Code** A description of who is allowed to access the station, given as 'Public' or 'Private'.

**Access\_Detail\_Code** A description of other station access information, given as code values: CALL = call ahead, KEY\_AFTER\_HOURS = card key after hours, KEY\_ALWAYS = card key at all times, CREDIT\_CARD\_AFTER\_HOURS = credit card after hours, CREDIT\_CARD\_ALWAYS = credit card at all times

**Access\_Days\_Time** Hours of operation for the station.

**Cards\_Accepted** A space-separated list of payment methods accepted. Payment methods are given as code values: A = American Express, Debit = Debit, D = Discover, M = Mastercard, V = Visa, Cash = Cash, Checks = Check, ALLIANCE = Alliance AutoGas, ARI = ARI, CleanEnergy = Clean Energy, Comdata = Commercial Fueling Network, EFS = EFS, FleetOne = FleetOne, FuelMan = Fuelman, GasCard = GASCARD, PacificPride = Pacific Pride, PHH = PHH, Proprietor = Proprietor Fleet Card, Speedway = Speedway, SuperPass = SuperPass, TCH = TCH, Tchek = T-Chek T-Card, Trillium = Trillium, Voyager = Voyager, Wright\_Exp = WEX

**Owner\_Type\_Code** The type of organization that owns the fueling infrastructure. Owner types are given as code values: FG = Federal Government Owned, J = Jointly Owned, LG = Local/Municipal Government Owned, P = Privately Owned, SG = State/Provincial Government Owned, T = Utility Owned

**BD\_Blends** For biodiesel stations, the level of biodiesel blends the station provides.

**EV\_Level1\_EVSE\_Num** For electric stations, the number of Level 1 EVSE (standard 110V outlet).

**EV\_Level2\_EVSE\_Num** For electric stations, the number of Level 2 EVSE (J1772 connector).

**EV\_DC\_Fast\_Count** For electric stations, the number of DC Fast Chargers.

- EV\_Other\_Info** For electric stations, the number and type of additional EVSE.
- EV\_Network** For electric stations, the name of the EVSE network, if applicable.
- EV\_Network\_Web** For electric stations, the EVSE network Web site, if applicable.
- Geocode\_Status** A rating indicating the approximate accuracy of the latitude and longitude for the station's address, given as code values: GPS = GPS, 200-9 = Point, 200-8 = Address, 200-7 = Intersection, 200-6 = Street, 200-5 = Neighborhood, 200-5 = Postal Code - Extended, 200-5 = Postal Code, 200-4 = City/Town, 200-3 = County, 200-2 = State/Province, 200-1 = Country, 200-0 = Unknown
- Latitude** The latitude of the station's address.
- Longitude** The longitude of the station's address.
- Date\_Last\_Confirmed** The date the station's details were last confirmed.
- ID** A unique identifier for this specific station.
- Updated\_At** The time the station's details were last updated (ISO 8601 format).
- Federal\_Agency\_ID** A unique identifier for the federal agency.
- Federal\_Agency\_Name** The name of the federal agency.
- Open\_Date** The date that the station began offering the fuel. Please note that most LPG (propane) stations do not have open dates. Some open dates are approximate. Also note that electric station records that are imported to the Station Locator on a daily basis through a network API do not have open dates.
- Hydrogen\_Status\_Link** For hydrogen stations, a link to a website that provides up-to-date information about the current status of this hydrogen station.
- LPG\_Primary** For propane stations, the type of station, given as a boolean.
- E85\_Blender\_Pump** For E85 stations, an indication of whether the station has a blender pump on site, given as a boolean.
- EV\_Connector\_Types** For electric stations, an array of strings identifying the connector types available at this station.
- Country** The two character country code of the station's location.
- Hydrogen\_Is\_Retail** For hydrogen stations, whether a station offers the retail sale of hydrogen by accepting payment at the point of sale.
- Federal\_Agency\_Code** The abbreviation for the Federal Agency, if relevant.
- Facility\_Type** The type of facility at which the station is located.
- CNG\_Dispenser\_Num** For CNG stations, the number of CNG dispensers installed.
- CNG\_On-Site\_Renewable\_Source** For CNG stations, the type of renewable energy used to generate CNG on-site.
- CNG\_Total\_Compression\_Capacity** For CNG stations, the total compressor capacity per compressor, measured in standard cubic feet per minute (scfm)
- CNG\_Storage\_Capacity** For CNG stations, the total storage capacity, measured in standard cubic feet (scf).
- LNG\_On-Site\_Renewable\_Source** For LNG stations, the type of renewable energy used to generate LNG on-site.

- E85\_Other\_Ethanol\_Blends** For E85 stations, an array of strings identifying the range(s) of blends other than E85 available at the station.
- EV\_Pricing** For EVSE stations, information about whether and how much users must pay to use the EVSE.
- LPG\_Nozzle\_Types** For LPG stations, an array of strings, identifying the type of nozzles available at the station, given as code values: ACME = ACME, QUICK\_CONNECT = Quick-connect.
- Hydrogen\_Pressures** For hydrogen stations, any array of strings identifying the pressures of the hydrogen available (in bar).
- Hydrogen\_Standards** For hydrogen stations, any array of strings identifying which SAE International fueling protocol standard(s) the infrastructure meets.
- CNG\_Fill\_Type\_Code** For CNG stations, the type of dispensing capability available, given as code values: B = Fast-fill and time-fill, Q = Fast-fill, T = Time-fill.
- CNG\_PSI** For CNG stations, a space separated list of PSI pressures available.
- CNG\_Vehicle\_Class** For CNG stations, the maximum vehicle size that can physically access the fueling infrastructure, given as code values: LD = Passenger vehicles (class 1-2), MD = Medium-duty (class 3-5), HD = Heavy-duty (class 6-8)
- LNG\_Vehicle\_Class** For LNG stations, the maximum vehicle size that can physically access the fueling infrastructure, given as code values: LD = Passenger vehicles (class 1-2), MD = Medium-duty (class 3-5), HD = Heavy-duty (class 6-8)
- EV\_On-Site\_Renewable\_Source** For EVSE stations, the type of renewable energy used to generate electricity on-site.

### Source

[https://afdc.energy.gov/data\\_download/alt\\_fuel\\_stations\\_format](https://afdc.energy.gov/data_download/alt_fuel_stations_format)

---

altfuel\_api

*Query the NREL Alternative Fuel API*

---

### Description

Query the NREL Alternative Fuel API

### Usage

```
altfuel_api(api_key, endpoint, params = list(NULL))
```

### Arguments

- |          |  |
|----------|--|
| api_key  | Character. An authorized API key for the NREL API service. API keys can be requested at <a href="https://developer.nrel.gov/signup/">https://developer.nrel.gov/signup/</a>  |
| endpoint | Character. Path to the specific API endpoint. Options available at <a href="https://developer.nrel.gov/docs/transportation/alt-fuel-stations-v1/">https://developer.nrel.gov/docs/transportation/alt-fuel-stations-v1/</a> |
| params   | A list of parameters for the API call. See ?nrel_params for options.   |

**Value**

An object of class `nrel_api`, containing content, parameters, and response.

**Examples**

```
## Not run:
altfuel_api(MY_API_KEY, endpoint = "/api/alt-fuel-stations/v1")

## End(Not run)
```

---

alt_fuel_near	<i>Check For Alternate Fuel Nearby</i>
---------------	--

---

**Description**

Check For Alternate Fuel Nearby

**Usage**

```
alt_fuel_near(api_key, location = NULL, miles = 5, fuel_type = NULL)
```

**Arguments**

api_key	Character. An authorized API key for the NREL API service. API keys can be requested at <a href="https://developer.nrel.gov/signup/">https://developer.nrel.gov/signup/</a>
location	A free-form input describing the address of the location. This may include the address given in a variety of formats, such as: street, city, state, postal code, etc.
miles	Numeric. The radius (in miles) around the search location to search for stations within. An explicit radius of up to 500.0 miles may be passed in, or the special infinite string may be passed in to find the nearest stations regardless of distance. Defaults to 5.
fuel_type	Optional. Filter results by a specific fuel type. Options: BD (biodiesel), CNG (compressed natural gas), ELEC (electric), E85 (ethanol/E85), HY (hydrogen), LNG (liquified natural gas), LPG, (propane).

**Value**

Logical value indicating whether at least one alternative fuel station exists for the given fuel in the given radius around the specified location.

**Examples**

```
## Not run:
alt_fuel_near(MY_API_KEY, "Nome, AK")
alt_fuel_near(MY_API_KEY, "Raleigh, NC", radius = 10, fuel_type = "LNG")

## End(Not run)
```

---

count_results	<i>Count Station Results</i>
---------------	------------------------------

---

**Description**

Count Station Results

**Usage**

```
count_results(x)
```

**Arguments**

x                    A nrel\_api object

**Value**

A data frame of station count by fuel station type

**Examples**

```
## Not run:  
x <- nearest_stations(MY_API_KEY, location = "Fort Worth, TX", radius = 5)  
count_results(x)  
  
## End(Not run)
```

---

last_updated	<i>Retrieve the Date of Last API Data Update.</i>
--------------	---

---

**Description**

Retrieve the Date of Last API Data Update.

**Usage**

```
last_updated(api_key)
```

**Arguments**

api\_key              Character. An authorized API key for the NREL API service. API keys can be requested at <https://developer.nrel.gov/signup/>

**Value**

An object of class POSIXct containing the date and time of the most recent data update.

**Examples**

```
## Not run:
last_updated(MY_API_KEY)

## End(Not run)
```

---

nearest\_stations      *Retrieve Nearest Alternate Fuel Stations by Criteria*

---

**Description**

Location parameters can either be passed directly as arguments, or imported using the `nrel_params()` function to generate a list. Location passed as arguments takes precedence.

**Usage**

```
nearest_stations(
  api_key,
  location = NULL,
  latitude = NULL,
  longitude = NULL,
  radius = NULL,
  params = nrel_params(radius = "infinite")
)
```

**Arguments**

<code>api_key</code>	Character. An authorized API key for the NREL API service. API keys can be requested at <a href="https://developer.nrel.gov/signup/">https://developer.nrel.gov/signup/</a>
<code>location</code>	A free-form input describing the address of the location. This may include the address given in a variety of formats, such as: street, city, state, postal code, etc.
<code>latitude</code>	Numeric. The latitude of the desired location.
<code>longitude</code>	Numeric. The longitude of the desired location.
<code>radius</code>	Numeric. The radius (in miles) around the search location to search for stations within. An explicit radius of up to 500.0 miles may be passed in, or the special infinite string may be passed in to find the nearest stations regardless of distance. Defaults to 'infinite'.
<code>params</code>	A list of parameters for the API call. See <code>?nrel_params</code> for options.

**Value**

An object of class `nrel_api`, containing content, parameters, and response.

## Examples

```
## Not run:
# find the 5 nearest Tesla stations to FAA HQ in Washington DC
my_params <- nrel_params(fuel_type = "ELEC", ev_connector_type = "TESLA", limit = 5)
my_loc <- c("FAA Headquarters, Washington, DC")
tesla_stns <- nearest_stations(MY_API_KEY, location = my_loc, params = my_params)
stations(tesla_stns)

## End(Not run)
```

---

nrel\_params

*Configure parameters for NREL API call*

---

## Description

Configure parameters for NREL API call

## Usage

```
nrel_params(
  status = NULL,
  access = NULL,
  fuel_type = NULL,
  cards_accepted = NULL,
  owner_type = NULL,
  federal_agency = NULL,
  cng_fill_type = NULL,
  cng_psi = NULL,
  cng_vehicle_class = NULL,
  e85_has_blender_pump = NULL,
  ev_network = NULL,
  ev_charging_level = NULL,
  ev_connector_type = NULL,
  ev_connector_type_operator = NULL,
  lng_vehicle_class = NULL,
  state = NULL,
  zip = NULL,
  country = NULL,
  limit = NULL,
  location = NULL,
  latitude = NULL,
  longitude = NULL,
  radius = NULL
)
```

**Arguments**

status	Return stations that match the given status. A single status, or a comma-separated list of multiple statuses, may be given. Options: all, E (Available), P (Planned), T (Temporarily unavailable).
access	Return stations with the given access type. Options: all, public, private.
fuel_type	Return stations that supply any of the given fuel types. A single fuel type, or a comma-separated list of multiple fuel types, may be given. Options: all, BD (biodiesel), CNG (compressed natural gas), ELEC (electric), E85 (ethanol/E85), HY (hydrogen), LNG (liquified natural gas), LPG, (propane).
cards_accepted	Return stations that accept any of the given payment methods. A single payment method, or a comma-separated list of multiple payment methods, may be given. Options: all, A, Debit, D, M, V, Cash, Checks, ALLIANCE, ARI, CleanEnergy, Comdata, CFN, EFS, FleetOne, FuelMan, GasCard, PacificPride, PHH, Proprietor, Speedway, SuperPass, TCH, Tchek, Trillium, Voyager, Wright_Exp.
owner_type	Return stations owned by the given types of owners. A single owner type, or a comma-separated list of multiple owner types, may be given. Options: all, FG (Federal Govt Owned), J (Joint Owned), LG (Local/Municipal Govt Owned), P (Privately Owned), SG (State/Provincial Govt Owned), T (Utility Owned).
federal_agency	Return stations owned by the given federal agency. A federal agency code, or a comma-separated list of multiple federal agency codes, may be given. Options: all, AAF, CH, CHSW, USACE_CW, CSOSA, DEF, DAF, DA, DOC, HHS, DHS, HUD, DOJ, DOL, DON, DOS, DOI, DOT, DTS, VA, EWDL, ECC, FCSD, FTC, FIN, FOCCG, GSA, GAFA, GAID, GAIT, HLTH, IRC, INA, IC, ISED, JUS, LGHC, NASA, NDEF, NREV, NRES, PARL, PQPCC, PM, PSEP, PSP, SCI, SBT, SI, SSA, TRANS, TREAS, USMC, CBP, USDA, DOD, DOE, EPA, FAA, IRS, USPS, VAFF.
cng_fill_type	Return only CNG stations that provide the specified type of dispensing capability. A single type, or a comma separated list of multiple types, may be given. Options: all, B (Fast-fill and time-fill), Q (Fast-fill), T (Time-fill).
cng_psi	Return only CNG stations that provide the specified PSI pressure. A single pressure level, or a comma separated list of multiple pressure levels, may be given. Options: all, 2400, 3000, 3600.
cng_vehicle_class	Return only CNG stations that can accommodate the specified vehicle class size or greater. Options: all, LD, MD, HD.
e85_has_blender_pump	Logical. If TRUE, then return only E85 stations that also have a blender pump capable of providing mid-level ethanol blends.
ev_network	Return only electric charging stations that belong to the given network. A single network, or a comma separated list of multiple networks, may be given. Options: all, BCHYDRO, Blink Network, ChargePoint Network, Circuit électrique, eCharge Network, Electrify America, Electrify Canada, EV Connect, EVGATEWAY, eVgo Network, FLO, FCN, GE WattStation, Greenlots, Non-Networked, OpConnect, PETROCAN, POWERFLEX, SemaCharge Network, Sun Country Highway, SWTCH, Tesla Destination, Tesla, Volta, Webasto.

ev_charging_level	Return only electric charging stations that provide the given level of electric vehicle supply equipment (EVSE). Options: all, 1, 2, dc_fast, legacy.
ev_connector_type	Return only electric charging stations that provide the given connector types. Options: all, NEMA1450, NEMA515, NEMA520, J1772, J1772COMBO, CHADEMO, TESLA.
ev_connector_type_operator	Control how multiple connector type options passed to the ev_connector_type parameter behave. The default of OR will return stations that have any of the connectors present. Specifying AND will only return stations that have all of the connectors present.
lng_vehicle_class	Return only LNG stations that can accommodate the specified vehicle class size or greater. Options: all, LD, MD, HD.
state	Return only stations within the given state. State must be given as a two character state code (eg, "CO" for Colorado). A single state, or a comma-separate list of multiple states, may be given.
zip	Return only stations within the given ZIP code. ZIP codes must be exactly 5 digits long. A single ZIP code, or a comma-separate list of multiple ZIP codes, may be given.
country	Return only stations within the given country code. Options: all, US (United States), CA (Canada).
limit	The maximum number of results to return. An explicit limit of up to 200 may be passed in, or the special all string may be passed in to return all results.
location	A free-form input describing the address of the location. This may include the address given in a variety of formats, such as: street, city, state, postal code, etc.
latitude	Numeric. The latitude of the desired location.
longitude	Numeric. The longitude of the desired location.
radius	Numeric. The radius (in miles) around the search location to search for stations within. An explicit radius of up to 500.0 miles may be passed in, or the special infinite string may be passed in to find the nearest stations regardless of distance.

### Value

A list of selected options for passing to an api call.

### Examples

```
# Which electric charging stations have Tesla connectors?
my_params <- nrel_params(fuel_type = "ELEC", ev_connector_type = "TESLA")
```

---

stations	<i>Extract station data from API object</i>
----------	---

---

**Description**

Extract station data from API object

**Usage**

```
stations(x)
```

**Arguments**

x                    An nrel\_api object.

**Value**

A data frame with the station listing from the nrel\_api object.

**Examples**

```
## Not run:
my_params <- nrel_params(fuel_type = "ELEC", ev_connector_type = "TESLA")
my_stations <- get_all_stations(MY_API_KEY, my_params)
stations(my_stations)

## End(Not run)
```

---

station_by_id	<i>Find a Specific Alternative Fuel Station by Unique ID</i>
---------------	--

---

**Description**

Find a Specific Alternative Fuel Station by Unique ID

**Usage**

```
station_by_id(api_key, station_id)
```

**Arguments**

api\_key            Character. An authorized API key for the NREL API service. API keys can be requested at <https://developer.nrel.gov/signup/>

station\_id        The unique ID of the station to find.

**Value**

An object of class `nrel_api`, containing content, parameters, and response.

**Examples**

```
## Not run:  
my_stn <- station_by_id(MY_API_KEY, station_id = 150544)  
stations(my_stn)  
  
## End(Not run)
```

---

update_params	<i>Update Non-Null Parameters</i>
---------------	-----------------------------------

---

**Description**

Update Non-Null Parameters

**Usage**

```
update_params(params, updates)
```

**Arguments**

params	list of parameters
updates	list of new parameters

**Value**

an updated list of parameters

**Examples**

```
my_params <- list(name = "Matt Foley")  
my_updates <- list(name = NULL, location = "a VAN down by the RIVER")  
new_params <- update_params(params = my_params, updates = my_updates)
```

# Index

## \* datasets

altfuel, [2](#)

all\_stations, [2](#)

alt\_fuel\_near, [6](#)

altfuel, [2](#)

altfuel\_api, [5](#)

count\_results, [7](#)

last\_updated, [7](#)

nearest\_stations, [8](#)

nrel\_params, [9](#)

station\_by\_id, [12](#)

stations, [12](#)

update\_params, [13](#)