
Renault API

HACF (created and maintained by @epenet)

Jun 09, 2026

CONTENTS

1	Features	3
2	Requirements	5
3	API Usage	7
4	Avoiding password storage	9
5	CLI Usage	11
6	Contributing	13
7	License	15
8	Disclaimer	17
9	Issues	19
10	Credits	21
	Python Module Index	83
	Index	85

FEATURES

This Python package manages the communication with the private Renault API used by the official MyRenault application.

The client is able to read various vehicle attributes, such as:

- mileage
- GPS location
- fuel autonomy (for fuel vehicles)
- battery autonomy (for electric vehicles)
- contracts associated to the vehicle (warranty and connected services)

For some vehicles, it is also possible to manage:

- hvac/pre-conditioning of the vehicle
- charge schedule

This package has been developed to be used with Home-Assistant, but it can be used in other contexts

REQUIREMENTS

- Python (≥ 3.10)

API USAGE

You can install *Renault API* via pip from PyPI:

```
$ pip install renauld-api
```

```
import aiohttp
import asyncio

from renauld_api.renauld_client import RenaultClient

async def main():
    async with aiohttp.ClientSession() as websession:
        client = RenaultClient(websession=websession, locale="fr_FR")
        await client.session.login('email', 'password')
        print(f"Accounts: {await client.get_person()}") # List available accounts, make a
↳note of kamereon account id

        account_id = "Your Kamereon account id"
        account = await client.get_api_account(account_id)
        print(f"Vehicles: {await account.get_vehicles()}") # List available vehicles, make
↳a note of vehicle VIN

        vin = "Your vehicle VIN"
        vehicle = await account.get_api_vehicle(vin)
        print(f"Cockpit information: {await vehicle.get_cockpit()}")
        print(f"Battery status information: {await vehicle.get_battery_status()}")

loop = asyncio.get_event_loop()
loop.run_until_complete(main())
```


AVOIDING PASSWORD STORAGE

The password is only needed once: `session.login` exchanges it for a Gigya login token, which acts as a refresh token used to mint access tokens (JWTs). Read it from `session.login_token` after login, store it securely *instead of the password*, and restore it on the next session with `session.set_login_token`:

```
await client.session.login('email', 'password')
login_token = client.session.login_token # persist this, not the password

# later, in a new session:
client.session.set_login_token(login_token) # no password required
```


CLI USAGE

The `renault-api` is also available through a CLI, which requires additional dependencies. For the added dependencies, you can install *Renault API* via `pip` from `PyPI`:

```
$ pip install renauld-api[cli]
```

Once installed, the following command prompts for credentials and settings, displays basic vehicle status information, and generates traces:

```
$ renauld-api --log status
```

- Credentials will automatically be stored in the user home directory (`~/.credentials/renault-api.json`)
- Logs will automatically be generated in `logs` subfolder

It is also possible to use raw http GET/POST commands, for example to get odometer and others data on a Windows machine:

```
C:> renauld-api http get /commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/  
↪cars/{vin}/cockpit
```

Or to request a battery status refresh on a Windows Machine (internal quotes escaped, whole payload enclosed in un-escaped quotes):

```
C:> renauld-api http post /commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/  
↪cars/{vin}/actions/refresh-battery-status "{\\"data\\": {\\"type\\": \\  
↪"RefreshBatteryStatus\\"}}"
```

Please see the [Command-line Reference](#) for full details.

CONTRIBUTING

Contributions are very welcome. To learn more, see the [Contributor Guide](#).

LICENSE

Distributed under the terms of the [MIT](#) license, *Renault API* is free and open source software.

DISCLAIMER

This project is not affiliated with, endorsed by, or connected to Renault. I accept no responsibility for any consequences, intended or accidental, as a result of interacting with Renault's API using this project.

CHAPTER
NINE

ISSUES

If you encounter any problems, please [file an issue](#) along with a detailed description.

CREDITS

This project was generated from [@cjolowicz's Hypermodern Python Cookiecutter](#) template. This project was heavily based on [@jamesremuscat's PyZE](#) python client for the Renault ZE API.

10.1 CLI Usage

10.1.1 renault-api

Main entry point for the Renault CLI.

Usage

```
renault-api [OPTIONS] COMMAND [ARGS]...
```

Options

--version

Show the version and exit.

--debug

Display debug traces.

--log

Log debug traces to file.

--json

Return data as JSON

--locale <locale>

API locale (eg. fr_FR)

--account <account>

Kamereon account ID to use

--vin <vin>

Vehicle VIN to use

accounts

Display list of accounts.

Usage

```
renault-api accounts [OPTIONS]
```

charge

Charge functionality.

Usage

```
renault-api charge [OPTIONS] COMMAND [ARGS]...
```

history

Display charge history.

Usage

```
renault-api charge history [OPTIONS]
```

Options

--period <period>

Period over which to aggregate.

Options

day | month

--to <end>

Required Date to finish showing history at (cannot be in the future)

--from <start>

Required Date to start showing history from

mode

Display or set charge mode.

Usage

```
renault-api charge mode [OPTIONS]
```

Options

--set <set>

Target charge mode (schedule_mode/always/always_schedule)

schedule

Display or update charge schedules.

Usage

```
renault-api charge schedule [OPTIONS] COMMAND [ARGS]...
```

activate

Activate charging schedule {ID}.

Usage

```
renault-api charge schedule activate [OPTIONS] ID
```

Arguments

ID

Required argument

deactivate

Deactivate charging schedule {ID}.

Usage

```
renault-api charge schedule deactivate [OPTIONS] ID
```

Arguments

ID

Required argument

set

Update charging schedule {ID}.

Usage

```
renault-api charge schedule set [OPTIONS] ID
```

Options

--monday <monday>

Monday schedule in format *hh:mm,duration* (for local timezone) or *Thh:mmZ,duration* (for utc) or *clear* to unset.

--tuesday <tuesday>

Tuesday schedule in format *hh:mm,duration* (for local timezone) or *Thh:mmZ,duration* (for utc) or *clear* to unset.

--wednesday <wednesday>

Wednesday schedule in format *hh:mm,duration* (for local timezone) or *Thh:mmZ,duration* (for utc) or *clear* to unset.

--thursday <thursday>

Thursday schedule in format *hh:mm,duration* (for local timezone) or *Thh:mmZ,duration* (for utc) or *clear* to unset.

Renault API

--friday <friday>

Friday schedule in format *hh:mm,duration* (for local timezone) or *Thh:mmZ,duration* (for utc) or *clear* to unset.

--saturday <saturday>

Saturday schedule in format *hh:mm,duration* (for local timezone) or *Thh:mmZ,duration* (for utc) or *clear* to unset.

--sunday <sunday>

Sunday schedule in format *hh:mm,duration* (for local timezone) or *Thh:mmZ,duration* (for utc) or *clear* to unset.

Arguments

ID

Required argument

show

Display charge schedules.

Usage

```
renault-api charge schedule show [OPTIONS]
```

sessions

Display charge sessions.

Usage

```
renault-api charge sessions [OPTIONS]
```

Options

--to <end>

Required Date to finish showing history at (cannot be in the future)

--from <start>

Required Date to start showing history from

soclevels

Display or update charge SoC levels.

Usage

```
renault-api charge soclevels [OPTIONS] COMMAND [ARGS]...
```

set

Set battery soc levels.

Usage

```
renault-api charge soclevels set [OPTIONS]
```

Options

--min <min>

Required Minimum state of charge level to set

--target <target>

Required Target state of charge level to set

show

Display battery soc levels.

Usage

```
renault-api charge soclevels show [OPTIONS]
```

start

Start charging with optional delay.

Usage

```
renault-api charge start [OPTIONS]
```

Options

--at <at>

Date/time at which to start charging. (defaults to immediate if not given). You can use times like 'in 5 minutes' or 'tomorrow at 9am'.

stop

Stop charge.

Usage

```
renault-api charge stop [OPTIONS]
```

contracts

Display vehicle contracts.

Usage

```
renault-api contracts [OPTIONS]
```

horn

Start horn.

Usage

```
renault-api horn [OPTIONS]
```

http

Raw HTTP.

Usage

```
renault-api http [OPTIONS] COMMAND [ARGS]...
```

get

Process HTTP GET request on endpoint.

Usage

```
renault-api http get [OPTIONS] ENDPOINT
```

Arguments

ENDPOINT

Required argument

post

Process HTTP POST request on endpoint.

Usage

```
renault-api http post [OPTIONS] ENDPOINT JSON_BODY
```

Arguments

ENDPOINT

Required argument

JSON_BODY

Required argument

post-file

Process HTTP POST request on endpoint.

Usage

```
renault-api http post-file [OPTIONS] ENDPOINT JSON_BODY
```

Arguments

ENDPOINT

Required argument

JSON_BODY

Required argument

hvac

HVAC functionality.

Usage

```
renault-api hvac [OPTIONS] COMMAND [ARGS]...
```

cancel

Cancel air conditioning.

Usage

```
renault-api hvac cancel [OPTIONS]
```

history

Display air conditioning history.

Usage

```
renault-api hvac history [OPTIONS]
```

Options

--period <period>

Period over which to aggregate.

Options

day | month

--to <end>

Required Date to finish showing history at (cannot be in the future)

--from <start>

Required Date to start showing history from

sessions

Display air conditioning sessions.

Usage

```
renault-api hvac sessions [OPTIONS]
```

Options

--to <end>

Required Date to finish showing history at (cannot be in the future)

--from <start>

Required Date to start showing history from

start

Start air conditioning.

Usage

```
renault-api hvac start [OPTIONS]
```

Options

--temperature <temperature>

Required Target temperature (in Celsius)

--at <at>

Date/time at which to complete preconditioning (defaults to immediate if not given). You can use times like 'in 5 minutes' or 'tomorrow at 9am'.

lights

Start lights.

Usage

```
renault-api lights [OPTIONS]
```

login

Login to Renault.

Usage

```
renault-api login [OPTIONS]
```

Options

--user <user>

--password <password>

reset

Clear all credentials/settings from the credential store.

Usage

```
renault-api reset [OPTIONS]
```

set

Store specified settings into credential store.

Usage

```
renault-api set [OPTIONS]
```

Options

--locale <locale>

API locale (eg. fr_FR)

--account <account>

Kamereon account ID to use for future calls

--vin <vin>

Vehicle VIN to use for future calls

settings

Display the current configuration keys.

Usage

```
renault-api settings [OPTIONS]
```

status

Display vehicle status.

Usage

```
renault-api status [OPTIONS]
```

vehicle

Display vehicle details.

Usage

```
renault-api vehicle [OPTIONS]
```

vehicles

Display list of vehicles.

Usage

```
renault-api vehicles [OPTIONS]
```

10.2 Renault high-level API Reference

10.2.1 Session

Session provider for interaction with Renault servers.

```
class renault_api.renault_session.RenaultSession(websession, locale=None, country=None,  
                                                locale_details=None, credential_store=None)
```

Renault session for interaction with Renault servers.

Parameters

- **websession** (*ClientSession*)
- **locale** (*str* | *None*)
- **country** (*str* | *None*)
- **locale_details** (*dict*[*str*, *str*] | *None*)
- **credential_store** (*CredentialStore* | *None*)

```
async get_account_vehicles(account_id)
```

GET to /accounts/{account_id}/vehicles.

Parameters

account_id (*str*)

Return type

[KamereonVehiclesResponse](#)

```
async get_person()
```

GET to /persons/{person_id}.

Return type

[KamereonPersonResponse](#)

```
async get_vehicle_contracts(account_id, vin)
```

GET to /v{endpoint_version}/cars/{vin}/contracts.

Parameters

- **account_id** (*str*)
- **vin** (*str*)

Return type

[KamereonVehicleContractsResponse](#)

```
async get_vehicle_data(account_id, vin, endpoint, params=None, * (Keyword-only parameters  
separator (PEP 3102)), adapter_type='kca')
```

GET to /v{endpoint_version}/cars/{vin}/{endpoint}.

Parameters

- **account_id** (*str*)
- **vin** (*str*)
- **endpoint** (*str*)
- **params** (*dict*[*str*, *str*] | *None*)
- **adapter_type** (*str*)

Return type[KamereonVehicleDataResponse](#)**async get_vehicle_details**(*account_id*, *vin*)

GET to /accounts/{account_id}/vehicles/{vin}/details.

Parameters

- **account_id** (*str*)
- **vin** (*str*)

Return type[KamereonVehicleDetailsResponse](#)**async http_request**(*method*, *endpoint*, *json=None*, *, *schema=None*)

GET to specified endpoint.

Parameters

- **method** (*str*)
- **endpoint** (*str*)
- **json** (*dict*[*str*, *Any*] | *None*)
- **schema** (*Schema* | *None*)

Return type[KamereonResponse](#)**async login**(*login_id*, *password*)

Attempt login on Gigya.

Parameters

- **login_id** (*str*)
- **password** (*str*)

Return type

None

property login_token: *str* | *None*

Return the current Gigya login token.

This token is obtained from the password on *login*, and is used to mint JWTs (access tokens) without re-supplying the password. Store it securely instead of the password, and restore it with *set_login_token* (or by pre-populating the credential store) on the next session.

set_login_token(*login_token*)

Restore a Gigya login token obtained from a previous session.

This avoids storing and reusing the user password: a session restored this way can mint JWTs without ever calling *login*.

Parameters**login_token** (*str*)**Return type**

None

async set_vehicle_action(*account_id, vin, endpoint, attributes, *, adapter_type='kca'*)

POST to /v{endpoint_version}/cars/{vin}/{endpoint}.

Parameters

- **account_id** (*str*)
- **vin** (*str*)
- **endpoint** (*str*)
- **attributes** (*dict[str, Any]*)
- **adapter_type** (*str*)

Return type[KamereonVehicleDataResponse](#)

10.2.2 Client

Client for Renault API.

class `renault_api.renault_client.RenaultClient`(*session=None, websession=None, locale=None, country=None, locale_details=None, credential_store=None*)

Proxy to a Renault profile.

Parameters

- **session** ([RenaultSession](#) | *None*)
- **websession** ([ClientSession](#) | *None*)
- **locale** (*str* | *None*)
- **country** (*str* | *None*)
- **locale_details** (*dict[str, str]* | *None*)
- **credential_store** ([CredentialStore](#) | *None*)

async get_api_account(*account_id*)

Get account proxy for specified account id.

Parameters**account_id** (*str*)**Return type**[RenaultAccount](#)**async get_api_accounts**()

Get account proxies.

Return typelist[[RenaultAccount](#)]

async get_person()

GET to /persons/{person_id}.

Return type

`KamereonPersonResponse`

property session: `RenaultSession`

Get session provider.

10.2.3 Account

Client for Renault API.

class `renault_api.renault_account.RenaultAccount`(*account_id, session=None, websession=None, locale=None, country=None, locale_details=None, credential_store=None*)

Proxy to a Renault account.

Parameters

- **account_id** (*str*)
- **session** (`RenaultSession` | `None`)
- **websession** (`ClientSession` | `None`)
- **locale** (*str* | `None`)
- **country** (*str* | `None`)
- **locale_details** (*dict[str, str]* | `None`)
- **credential_store** (`CredentialStore` | `None`)

property account_id: `str`

Get account id.

async get_api_vehicle(*vin*)

Get vehicle proxy for specified vin.

Parameters

vin (*str*)

Return type

`RenaultVehicle`

async get_api_vehicles()

Get vehicle proxies.

Return type

`list[RenaultVehicle]`

async get_vehicles()

GET to /accounts/{account_id}/vehicles.

Return type

`KamereonVehiclesResponse`

property session: `RenaultSession`

Get session provider.

10.2.4 Vehicle

Client for Renault API.

```
class renauld_api.renauld_vehicle.RenaultVehicle(account_id, vin, *, session=None,  
                                                websession=None, locale=None, country=None,  
                                                locale_details=None, credential_store=None,  
                                                vehicle_details=None, car_adapter=None)
```

Proxy to a Renault vehicle.

Parameters

- **account_id** (*str*)
- **vin** (*str*)
- **session** (*RenaultSession* | *None*)
- **websession** (*ClientSession* | *None*)
- **locale** (*str* | *None*)
- **country** (*str* | *None*)
- **locale_details** (*dict*[*str*, *str*] | *None*)
- **credential_store** (*CredentialStore* | *None*)
- **vehicle_details** (*KamereonVehicleDetails* | *None*)
- **car_adapter** (*KamereonVehicleCarAdapterData* | *None*)

property account_id: *str*

Get account id.

async get_battery_soc()

Get vehicle battery state of charge limits

Return type

KamereonVehicleBatterySocData

async get_battery_status()

Get vehicle battery status.

Return type

KamereonVehicleBatteryStatusData

async get_car_adapter()

Get vehicle car adapter details.

Return type

KamereonVehicleCarAdapterData

async get_charge_history(*start*, *end*, *period*)

Get vehicle charge history.

Parameters

- **start** (*datetime*)
- **end** (*datetime*)
- **period** (*str*)

Return type[KamereonVehicleChargeHistoryData](#)**async get_charge_mode()**

Get vehicle charge mode.

Return type[KamereonVehicleChargeModeData](#)**async get_charge_schedule()**

Get vehicle charge schedule.

Return type

dict[str, Any]

async get_charges(start, end)

Get vehicle charges.

Parameters

- **start** (datetime)
- **end** (datetime)

Return type[KamereonVehicleChargesData](#)**async get_charging_settings()**

Get vehicle charging settings.

Return type[KamereonVehicleChargingSettingsData](#)**async get_cockpit()**

Get vehicle cockpit.

Return type[KamereonVehicleCockpitData](#)**async get_contracts()**

Get vehicle contracts.

Return type

list[KamereonVehicleContract]

async get_details()

Get vehicle details.

Return type[KamereonVehicleDetails](#)**async get_endpoint_definition(endpoint)**

From VEHICLE_ENDPOINTS / DEFAULT_ENDPOINT.

Parameters**endpoint** (str)**Return type**[EndpointDefinition](#)

async get_full_endpoint(*endpoint*)

From VEHICLE_ENDPOINTS / DEFAULT_ENDPOINT.

Parameters

endpoint (*str*)

Return type

str

async get_hvac_history(*start, end, period*)

Get vehicle hvac history.

Parameters

- **start** (*datetime*)
- **end** (*datetime*)
- **period** (*str*)

Return type

[KamereonVehicleHvacHistoryData](#)

async get_hvac_sessions(*start, end*)

Get vehicle hvac sessions.

Parameters

- **start** (*datetime*)
- **end** (*datetime*)

Return type

[KamereonVehicleHvacSessionsData](#)

async get_hvac_settings()

Get vehicle hvac settings (schedule+mode).

Return type

[KamereonVehicleHvacSettingsData](#)

async get_hvac_status()

Get vehicle hvac status.

Return type

[KamereonVehicleHvacStatusData](#)

async get_location()

Get vehicle location.

Return type

[KamereonVehicleLocationData](#)

async get_lock_status()

Get vehicle lock status.

Return type

[KamereonVehicleLockStatusData](#)

async get_notification_settings()

Get vehicle notification settings.

Return type

[KamereonVehicleNotificationSettingsData](#)

async get_res_state()

Get vehicle res state.

Return type

[KamereonVehicleResStateData](#)

async get_tyre_pressure()

Get vehicle tyre pressure.

Return type

[KamereonVehicleTyrePressureData](#)

async http_get(endpoint)

Run HTTP GET to endpoint.

Parameters

endpoint (*str*)

Return type

[KamereonResponse](#)

async http_post(endpoint, json=None)

Run HTTP POST to endpoint.

Parameters

- **endpoint** (*str*)
- **json** (*dict[str, Any] | None*)

Return type

[KamereonResponse](#)

property session: [RenaultSession](#)

Get session.

async set_ac_start(temperature, when=None)

Start vehicle ac.

Parameters

- **temperature** (*float*)
- **when** (*datetime | None*)

Return type

[KamereonVehicleHvacStartActionData](#)

async set_ac_stop()

Stop vehicle ac.

Return type

[KamereonVehicleHvacStartActionData](#)

async set_battery_soc(*, min, target)

Sets Battery Soc levels.

Parameters

- **min** (*int*)
- **target** (*int*)

Return type

[KamereonVehicleBatterySocActionData](#)

async set_charge_mode(*charge_mode*)

Set vehicle charge mode.

Parameters

charge_mode (*str*)

Return type

[KamereonVehicleChargeModeActionData](#)

async set_charge_schedules(*schedules*)

Set vehicle charge schedules.

Parameters

schedules (*list*[[ChargeSchedule](#)])

Return type

[KamereonVehicleChargeScheduleActionData](#)

async set_charge_start(*when=None*)

Start vehicle charge with optional delay.

Parameters

when (*datetime* | *None*)

Return type

[KamereonVehicleChargingStartActionData](#)

async set_charge_stop()

Start vehicle charge.

Return type

[KamereonVehicleChargingStartActionData](#)

async set_hvac_schedules(*schedules*)

Set vehicle charge schedules.

Parameters

schedules (*list*[[HvacSchedule](#)])

Return type

[KamereonVehicleHvacScheduleActionData](#)

async supports_endpoint(*endpoint*)

Check if vehicle supports endpoint.

Parameters

endpoint (*str*)

Return type

bool

property vin: **str**

Get vin.

10.3 Kamereon low-level API Reference

10.3.1 Core

Kamereon API.

`renault_api.kamereon.get_account_url(root_url, account_id)`

Get the url to the account.

Parameters

- `root_url` (*str*)
- `account_id` (*str*)

Return type

str

`async renault_api.kamereon.get_account_vehicles(websession, root_url, api_key, gigya_jwt, country, account_id)`

GET to `/accounts/{account_id}/vehicles`.

Parameters

- `websession` (*ClientSession*)
- `root_url` (*str*)
- `api_key` (*str*)
- `gigya_jwt` (*str*)
- `country` (*str*)
- `account_id` (*str*)

Return type

[KamereonVehiclesResponse](#)

`renault_api.kamereon.get_car_adapter_url(root_url, account_id, version, vin, *, adapter_type='kca')`

Get the url to the car adapter.

Parameters

- `root_url` (*str*)
- `account_id` (*str*)
- `version` (*int*)
- `vin` (*str*)
- `adapter_type` (*str*)

Return type

str

`renault_api.kamereon.get_commerce_url(root_url)`

Get the Kamereon base commerce url.

Parameters

`root_url` (*str*)

Return type

str

`renault_api.kamereon.get_contracts_url(root_url, account_id, vin)`

Get the url to the car contracts.

Parameters

- **root_url** (*str*)
- **account_id** (*str*)
- **vin** (*str*)

Return type

str

`async renault_api.kamereon.get_person(webssession, root_url, api_key, gigya_jwt, country, person_id)`

GET to `/persons/{person_id}`.

Parameters

- **webssession** (*ClientSession*)
- **root_url** (*str*)
- **api_key** (*str*)
- **gigya_jwt** (*str*)
- **country** (*str*)
- **person_id** (*str*)

Return type

[KamereonPersonResponse](#)

`renault_api.kamereon.get_person_url(root_url, person_id)`

Get the url to the person.

Parameters

- **root_url** (*str*)
- **person_id** (*str*)

Return type

str

`async renault_api.kamereon.get_vehicle_contracts(webssession, root_url, api_key, gigya_jwt, country, locale, account_id, vin)`

GET to `/accounts/{accountId}/vehicles/{vin}/contracts`.

Parameters

- **webssession** (*ClientSession*)
- **root_url** (*str*)
- **api_key** (*str*)
- **gigya_jwt** (*str*)
- **country** (*str*)
- **locale** (*str*)
- **account_id** (*str*)
- **vin** (*str*)

Return type[KamereonVehicleContractsResponse](#)

async `renault_api.kamereon.get_vehicle_data`(*websession, root_url, api_key, gigya_jwt, country, account_id, vin, endpoint, endpoint_version=None, params=None, *, adapter_type='kca'*)

GET to `/v{endpoint_version}/cars/{vin}/{endpoint}`.

Parameters

- **websession** (*ClientSession*)
- **root_url** (*str*)
- **api_key** (*str*)
- **gigya_jwt** (*str*)
- **country** (*str*)
- **account_id** (*str*)
- **vin** (*str*)
- **endpoint** (*str*)
- **endpoint_version** (*int | None*)
- **params** (*dict[str, str] | None*)
- **adapter_type** (*str*)

Return type[KamereonVehicleDataResponse](#)

async `renault_api.kamereon.get_vehicle_details`(*websession, root_url, api_key, gigya_jwt, country, account_id, vin*)

GET to `/accounts/{account_id}/vehicles/{vin}/details`.

Parameters

- **websession** (*ClientSession*)
- **root_url** (*str*)
- **api_key** (*str*)
- **gigya_jwt** (*str*)
- **country** (*str*)
- **account_id** (*str*)
- **vin** (*str*)

Return type[KamereonVehicleDetailsResponse](#)

async `renault_api.kamereon.request`(*websession, method, url, api_key, gigya_jwt, params, json=None, schema=None, *, wrap_array_in=None*)

Process Kamereon HTTP request.

Parameters

- **websession** (*ClientSession*)
- **method** (*str*)

- **url** (*str*)
- **api_key** (*str*)
- **gigya_jwt** (*str*)
- **params** (*dict[str, str]*)
- **json** (*dict[str, Any] | None*)
- **schema** (*Schema | None*)
- **wrap_array_in** (*str | None*)

Return type[KamereonResponse](#)

async `renault_api.kamereon.set_vehicle_action`(*websession, root_url, api_key, gigya_jwt, country, account_id, vin, endpoint, attributes, endpoint_version=None, data_type=None, *, adapter_type='kca'*)

POST to `/v{endpoint_version}/cars/{vin}/{endpoint}`.

Parameters

- **websession** (*ClientSession*)
- **root_url** (*str*)
- **api_key** (*str*)
- **gigya_jwt** (*str*)
- **country** (*str*)
- **account_id** (*str*)
- **vin** (*str*)
- **endpoint** (*str*)
- **attributes** (*dict[str, Any]*)
- **endpoint_version** (*int | None*)
- **data_type** (*dict[str, Any] | None*)
- **adapter_type** (*str*)

Return type[KamereonVehicleDataResponse](#)

10.3.2 Data models

Kamereon models.

class `renault_api.kamereon.models.ChargeDaySchedule`(*raw_data, startTime, duration*)

Kamereon vehicle charge schedule for day.

Parameters

- **raw_data** (*dict[str, Any]*)
- **startTime** (*str | None*)
- **duration** (*int | None*)

for_json()

Create dict for json.

Return type

dict[str, Any]

get_end_time()

Get end time.

Return type

str | None

```
class renault_api.kamereon.models.ChargeSchedule(raw_data, id, activated, monday, tuesday,
                                                wednesday, thursday, friday, saturday, sunday)
```

Kamereon vehicle charge schedule for week.

Parameters

- **raw_data** (dict[str, Any])
- **id** (int | None)
- **activated** (bool | None)
- **monday** (ChargeDaySchedule | None)
- **tuesday** (ChargeDaySchedule | None)
- **wednesday** (ChargeDaySchedule | None)
- **thursday** (ChargeDaySchedule | None)
- **friday** (ChargeDaySchedule | None)
- **saturday** (ChargeDaySchedule | None)
- **sunday** (ChargeDaySchedule | None)

for_json()

Create dict for json.

Return type

dict[str, Any]

```
class renault_api.kamereon.models.EndpointDefinition(endpoint: str, mode: str = 'default')
```

Parameters

- **endpoint** (str)
- **mode** (str)

```
class renault_api.kamereon.models.HvacDaySchedule(raw_data, readyAtTime)
```

Kamereon vehicle hvac schedule for day.

Parameters

- **raw_data** (dict[str, Any])
- **readyAtTime** (str | None)

for_json()

Create dict for json.

Return type

dict[str, str | None]

class `renault_api.kamereon.models.HvacSchedule`(*raw_data, id, activated, monday, tuesday, wednesday, thursday, friday, saturday, sunday*)

Kamereon vehicle hvac schedule for week.

Parameters

- **raw_data** (*dict[str, Any]*)
- **id** (*int | None*)
- **activated** (*bool | None*)
- **monday** (*HvacDaySchedule | None*)
- **tuesday** (*HvacDaySchedule | None*)
- **wednesday** (*HvacDaySchedule | None*)
- **thursday** (*HvacDaySchedule | None*)
- **friday** (*HvacDaySchedule | None*)
- **saturday** (*HvacDaySchedule | None*)
- **sunday** (*HvacDaySchedule | None*)

`for_json()`

Create dict for json.

Return type

dict[str, Any]

class `renault_api.kamereon.models.KamereonPersonAccount`(*raw_data, accountId, accountType, accountStatus*)

Kamereon person account data.

Parameters

- **raw_data** (*dict[str, Any]*)
- **accountId** (*str | None*)
- **accountType** (*str | None*)
- **accountStatus** (*str | None*)

class `renault_api.kamereon.models.KamereonPersonResponse`(*raw_data, errors, accounts*)

Kamereon response to GET on `/persons/{gigya_person_id}`.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errors** (*list[KamereonResponseError] | None*)
- **accounts** (*list[KamereonPersonAccount] | None*)

class `renault_api.kamereon.models.KamereonResponse`(*raw_data, errors*)

Kamereon response.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errors** (*list[KamereonResponseError] | None*)

raise_for_error_code()

Raise exception if errors found in the response.

Return type

None

class `renault_api.kamereon.models.KamereonResponseError`(*raw_data, errorCode, errorMessage*)

Kamereon response error.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errorCode** (*str | None*)
- **errorMessage** (*str | None*)

get_error_details()

Extract the error details sometimes hidden inside nested JSON.

Return type

str | None

raise_for_error_code()

Raise exception from response error.

Return type

None

class `renault_api.kamereon.models.KamereonVehicleBatterySocActionData`(*raw_data*)

Kamereon vehicle action data soc attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleBatterySocData`(*raw_data, lastEnergyUpdateTimestamp, socMin, socTarget*)

Kamereon vehicle battery state of charge limits data.

Parameters

- **raw_data** (*dict[str, Any]*)
- **lastEnergyUpdateTimestamp** (*str | None*)
- **socMin** (*int | None*)
- **socTarget** (*int | None*)

class `renault_api.kamereon.models.KamereonVehicleBatteryStatusData`(*raw_data, timestamp, batteryLevel, batteryTemperature, batteryAutonomy, batteryCapacity, batteryAvailableEnergy, plugStatus, chargingStatus, chargingRemainingTime, chargingInstantaneousPower, chargingRemainingTimeLastUpdateDateTime, V2L_SystemStatusDisplay*)

Kamereon vehicle battery-status data.

Parameters

- **raw_data** (*dict[str, Any]*)
- **timestamp** (*str | None*)
- **batteryLevel** (*int | None*)
- **batteryTemperature** (*int | None*)
- **batteryAutonomy** (*int | None*)
- **batteryCapacity** (*int | None*)
- **batteryAvailableEnergy** (*int | None*)
- **plugStatus** (*int | None*)
- **chargingStatus** (*float | None*)
- **chargingRemainingTime** (*int | None*)
- **chargingInstantaneousPower** (*float | None*)
- **chargingRemainingTimeLastUpdateDateTime** (*str | None*)
- **V2L_SystemStatusDisplay** (*int | None*)

get_charging_status()

Return charging status.

Return type

ChargeState | None

get_plug_status()

Return plug status.

Return type

PlugState | None

```
class renault_api.kamereon.models.KamereonVehicleCarAdapterData(raw_data, vin, vehicleId,  
batteryCode, brand,  
canGeneration, carGateway,  
deliveryCountry, deliveryDate,  
energy, engineType, familyCode,  
firstRegistrationDate, gearbox,  
modelCode, modelCodeDetail,  
modelName, radioType, region,  
registrationCountry,  
registrationNumber, tcuCode,  
versionCode, privacyMode,  
privacyModeUpdateDate,  
svtFlag, svtBlockFlag)
```

Kamereon vehicle data hvac-status attributes.

Parameters

- **raw_data** (*dict[str, Any]*)
- **vin** (*str | None*)
- **vehicleId** (*int | None*)

- **batteryCode** (*str* | *None*)
- **brand** (*str* | *None*)
- **canGeneration** (*str* | *None*)
- **carGateway** (*str* | *None*)
- **deliveryCountry** (*str* | *None*)
- **deliveryDate** (*str* | *None*)
- **energy** (*str* | *None*)
- **engineType** (*str* | *None*)
- **familyCode** (*str* | *None*)
- **firstRegistrationDate** (*str* | *None*)
- **gearbox** (*str* | *None*)
- **modelCode** (*str* | *None*)
- **modelCodeDetail** (*str* | *None*)
- **modelName** (*str* | *None*)
- **radioType** (*str* | *None*)
- **region** (*str* | *None*)
- **registrationCountry** (*str* | *None*)
- **registrationNumber** (*str* | *None*)
- **tcuCode** (*str* | *None*)
- **versionCode** (*str* | *None*)
- **privacyMode** (*str* | *None*)
- **privacyModeUpdateDate** (*str* | *None*)
- **svtFlag** (*bool* | *None*)
- **svtBlockFlag** (*bool* | *None*)

controls_action_via_kcm(*action*)

Return True if model uses endpoint via kcm.

Parameters

action (*str*)

Return type

bool

reports_charging_power_in_watts()

Return True if model reports chargingInstantaneousPower in watts.

Return type

bool

uses_electricity()

Return True if model uses electricity.

Return type

bool

uses_fuel()

Return True if model uses fuel.

Return type

bool

class `renault_api.kamereon.models.KamereonVehicleChargeHistoryData`(*raw_data*)

Kamereon vehicle data charge-history attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleChargeModeActionData`(*raw_data*)

Kamereon vehicle action data charge-mode attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleChargeModeData`(*raw_data*, *chargeMode*)

Kamereon vehicle data charge-mode attributes.

Parameters

- **raw_data** (*dict[str, Any]*)
- **chargeMode** (*str | None*)

class `renault_api.kamereon.models.KamereonVehicleChargeScheduleActionData`(*raw_data*)

Kamereon vehicle action data charge-schedule attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleChargesData`(*raw_data*)

Kamereon vehicle data charges attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleChargingSettingsData`(*raw_data*, *mode*,
schedules,
startDateTime, *dateTime*,
delay)

Kamereon vehicle data charging-settings attributes.

Parameters

- **raw_data** (*dict[str, Any]*)
- **mode** (*str | None*)
- **schedules** (*list[ChargeSchedule] | None*)
- **startDateTime** (*str | None*)
- **dateTime** (*str | None*)
- **delay** (*int | None*)

update(*args*)

Update schedule.

Parameters

args (*dict[str, Any]*)

Return type

None

class `renault_api.kamereon.models.KamereonVehicleChargingStartActionData`(*raw_data*)

Kamereon vehicle action data charging-start attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleCockpitData`(*raw_data*, *fuelAutonomy*, *fuelQuantity*, *totalMileage*)

Kamereon vehicle data cockpit attributes.

Parameters

- **raw_data** (*dict[str, Any]*)
- **fuelAutonomy** (*float | None*)
- **fuelQuantity** (*float | None*)
- **totalMileage** (*float | None*)

class `renault_api.kamereon.models.KamereonVehicleContract`(*raw_data*, *type*, *contractId*, *code*, *group*, *durationMonths*, *startDate*, *endDate*, *status*, *statusLabel*, *description*)

Kamereon vehicle contract.

Parameters

- **raw_data** (*dict[str, Any]*)
- **type** (*str | None*)
- **contractId** (*str | None*)
- **code** (*str | None*)
- **group** (*str | None*)
- **durationMonths** (*int | None*)
- **startDate** (*str | None*)
- **endDate** (*str | None*)
- **status** (*str | None*)
- **statusLabel** (*str | None*)
- **description** (*str | None*)

class `renault_api.kamereon.models.KamereonVehicleContractsResponse`(*raw_data*, *errors*, *contractList*)

Kamereon response to GET on `/accounts/{accountId}/vehicles/{vin}/contracts`.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errors** (*list[KamereonResponseError] | None*)
- **contractList** (*list[KamereonVehicleContract] | None*)

class `renault_api.kamereon.models.KamereonVehicleData`(*raw_data, type, id, attributes*)

Kamereon vehicle data.

Parameters

- **raw_data** (*dict[str, Any]*)
- **type** (*str | None*)
- **id** (*str | None*)
- **attributes** (*dict[str, Any] | None*)

class `renault_api.kamereon.models.KamereonVehicleDataAttributes`(*raw_data*)

Kamereon vehicle data attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleDataResponse`(*raw_data, errors, data*)

Kamereon response to GET/POST on `.../cars/{vin}/{type}`.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errors** (*list[KamereonResponseError] | None*)
- **data** (*KamereonVehicleData | None*)

get_attributes(*schema*)

Return jwt token.

Parameters

schema (*Schema*)

Return type

KamereonVehicleDataAttributes

class `renault_api.kamereon.models.KamereonVehicleDetails`(*raw_data, vin, registrationNumber, radioCode, brand, model, energy, engineEnergyType, assets*)

Kamereon vehicle details.

Parameters

- **raw_data** (*dict[str, Any]*)
- **vin** (*str | None*)
- **registrationNumber** (*str | None*)
- **radioCode** (*str | None*)
- **brand** (*KamereonVehicleDetailsGroup | None*)
- **model** (*KamereonVehicleDetailsGroup | None*)
- **energy** (*KamereonVehicleDetailsGroup | None*)
- **engineEnergyType** (*str | None*)
- **assets** (*list[dict[str, Any]] | None*)

controls_action_via_kcm(*action*)

Return True if model uses endpoint via kcm.

Parameters

action (*str*)

Return type

bool

get_asset(*asset_type*)

Return asset.

Parameters

asset_type (*str*)

Return type

dict[*str*, *Any*] | None

get_brand_label()

Return vehicle model label.

Return type

str | None

get_endpoint(*endpoint*)

Return model endpoint

Parameters

endpoint (*str*)

Return type

[EndpointDefinition](#) | None

get_endpoints()

Return model endpoints.

Return type

Mapping[*str*, [EndpointDefinition](#) | None]

get_energy_code()

Return vehicle energy code.

Return type

str | None

get_model_code()

Return vehicle model code.

Return type

str | None

get_model_label()

Return vehicle model label.

Return type

str | None

get_picture(*size=AssetPictureSize.LARGE*)

Return vehicle picture.

Parameters

size (*AssetPictureSize*)

Return type
str | None

reports_charge_session_durations_in_minutes()

Return True if model reports history durations in minutes.

Return type
bool

reports_charging_power_in_watts()

Return True if model reports chargingInstantaneousPower in watts.

Return type
bool

supports_endpoint(endpoint)

Return True if model supports specified endpoint.

Parameters
endpoint (str)

Return type
bool

uses_electricity()

Return True if model uses electricity.

Return type
bool

uses_fuel()

Return True if model uses fuel.

Return type
bool

class renaul_t_api.kamereon.models.**KamereonVehicleDetailsGroup**(raw_data, code, label, group)

Kamereon vehicle details group data.

Parameters

- **raw_data** (dict[str, Any])
- **code** (str | None)
- **label** (str | None)
- **group** (str | None)

class renaul_t_api.kamereon.models.**KamereonVehicleDetailsResponse**(raw_data, vin, registrationNumber, radioCode, brand, model, energy, engineEnergyType, assets, errors)

Kamereon response to GET on /accounts/{account_id}/vehicles/{vin}/details.

Parameters

- **raw_data** (dict[str, Any])
- **vin** (str | None)
- **registrationNumber** (str | None)

- **radioCode** (*str* | *None*)
- **brand** (*KamereonVehicleDetailsGroup* | *None*)
- **model** (*KamereonVehicleDetailsGroup* | *None*)
- **energy** (*KamereonVehicleDetailsGroup* | *None*)
- **engineEnergyType** (*str* | *None*)
- **assets** (*list[dict[str, Any]]* | *None*)
- **errors** (*list[KamereonResponseError]* | *None*)

class `renault_api.kamereon.models.KamereonVehicleHvacHistoryData`(*raw_data*)

Kamereon vehicle data hvac-history attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleHvacModeActionData`(*raw_data*)

Kamereon vehicle action data hvac-mode attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleHvacScheduleActionData`(*raw_data*)

Kamereon vehicle action data hvac-schedule attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleHvacSessionsData`(*raw_data*)

Kamereon vehicle data hvac-sessions attributes.

Parameters

raw_data (*dict[str, Any]*)

class `renault_api.kamereon.models.KamereonVehicleHvacSettingsData`(*raw_data*, *mode*, *schedules*)

Kamereon vehicle data hvac-settings (mode+schedules) attributes.

Parameters

- **raw_data** (*dict[str, Any]*)
- **mode** (*str* | *None*)
- **schedules** (*list[HvacSchedule]* | *None*)

update(*args*)

Update schedule.

Parameters

args (*dict[str, Any]*)

Return type

None

class `renault_api.kamereon.models.KamereonVehicleHvacStartActionData`(*raw_data*)

Kamereon vehicle action data hvac-start attributes.

Parameters

raw_data (*dict[str, Any]*)

```
class renault_api.kamereon.models.KamereonVehicleHvacStatusData(raw_data, lastUpdateTime,
                                                                externalTemperature,
                                                                internalTemperature, hvacStatus,
                                                                nextHvacStartDate,
                                                                socThreshold)
```

Kamereon vehicle data hvac-status attributes.

Parameters

- **raw_data** (*dict[str, Any]*)
- **lastUpdateTime** (*str | None*)
- **externalTemperature** (*float | None*)
- **internalTemperature** (*float | None*)
- **hvacStatus** (*str | None*)
- **nextHvacStartDate** (*str | None*)
- **socThreshold** (*float | None*)

```
class renault_api.kamereon.models.KamereonVehicleLocationData(raw_data, lastUpdateTime,
                                                                gpsLatitude, gpsLongitude)
```

Kamereon vehicle data location attributes.

Parameters

- **raw_data** (*dict[str, Any]*)
- **lastUpdateTime** (*str | None*)
- **gpsLatitude** (*float | None*)
- **gpsLongitude** (*float | None*)

```
class renault_api.kamereon.models.KamereonVehicleLockStatusData(raw_data, lockStatus,
                                                                doorStatusRearLeft,
                                                                doorStatusRearRight,
                                                                doorStatusDriver,
                                                                doorStatusPassenger,
                                                                hatchStatus, lastUpdateTime)
```

Kamereon vehicle data lock-status attributes.

Parameters

- **raw_data** (*dict[str, Any]*)
- **lockStatus** (*str | None*)
- **doorStatusRearLeft** (*str | None*)
- **doorStatusRearRight** (*str | None*)
- **doorStatusDriver** (*str | None*)
- **doorStatusPassenger** (*str | None*)
- **hatchStatus** (*str | None*)
- **lastUpdateTime** (*str | None*)

```
class renaul_api.kamereon.models.KamereonVehicleNotificationSettingsData(raw_data)
```

Kamereon vehicle data notification-settings attributes.

Parameters

raw_data (*dict[str, Any]*)

```
class renaul_api.kamereon.models.KamereonVehicleResStateData(raw_data, details, code)
```

Kamereon vehicle data res-set attributes.

Parameters

- **raw_data** (*dict[str, Any]*)
- **details** (*str | None*)
- **code** (*str | None*)

```
class renaul_api.kamereon.models.KamereonVehicleTyrePressureData(raw_data, flPressure,  
frPressure, rlPressure,  
rrPressure, flStatus, frStatus,  
rlStatus, rrStatus)
```

Kamereon vehicle tyre-pressure data.

Parameters

- **raw_data** (*dict[str, Any]*)
- **flPressure** (*int | None*)
- **frPressure** (*int | None*)
- **rlPressure** (*int | None*)
- **rrPressure** (*int | None*)
- **flStatus** (*int | None*)
- **frStatus** (*int | None*)
- **rlStatus** (*int | None*)
- **rrStatus** (*int | None*)

```
class renaul_api.kamereon.models.KamereonVehiclesLink(raw_data, vin, vehicleDetails)
```

Kamereon vehicles link data.

Parameters

- **raw_data** (*dict[str, Any]*)
- **vin** (*str | None*)
- **vehicleDetails** ([KamereonVehicleDetails](#) | *None*)

```
class renaul_api.kamereon.models.KamereonVehiclesResponse(raw_data, errors, accountId, country,  
vehicleLinks)
```

Kamereon response to GET on /accounts/{account_id}/vehicles.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errors** (*list[KamereonResponseError] | None*)
- **accountId** (*str | None*)
- **country** (*str | None*)

- **vehicleLinks** (*list[KamereonVehiclesLink] | None*)

`renault_api.kamereon.models.get_model_endpoint(model_code, endpoint)`

Return model endpoint

Parameters

- **model_code** (*str | None*)
- **endpoint** (*str*)

Return type

[EndpointDefinition](#) | None

`renault_api.kamereon.models.get_model_endpoints(model_code)`

Return model endpoints.

Parameters

model_code (*str | None*)

Return type

[Mapping](#)[*str*, [EndpointDefinition](#) | None]

10.4 Gigya low-level authentication API Reference

10.4.1 Core

Gigya API.

async `renault_api.gigya.get_account_info(websession, root_url, api_key, login_token)`

Send POST to `/accounts.getAccountInfo`.

Parameters

- **websession** (*ClientSession*)
- **root_url** (*str*)
- **api_key** (*str*)
- **login_token** (*str*)

Return type

[GigyaGetAccountInfoResponse](#)

async `renault_api.gigya.get_jwt(websession, root_url, api_key, login_token)`

Send POST to `/accounts.getJWT`.

Parameters

- **websession** (*ClientSession*)
- **root_url** (*str*)
- **api_key** (*str*)
- **login_token** (*str*)

Return type

[GigyaGetJWTResponse](#)

async `renault_api.gigya.login(websession, root_url, api_key, login_id, password)`

Send POST to /accounts.login.

Parameters

- **websession** (*ClientSession*)
- **root_url** (*str*)
- **api_key** (*str*)
- **login_id** (*str*)
- **password** (*str*)

Return type

[GigyaLoginResponse](#)

async `renault_api.gigya.request(websession, method, url, data, schema)`

Send request to Gigya.

Parameters

- **websession** (*ClientSession*)
- **method** (*str*)
- **url** (*str*)
- **data** (*dict[str, Any]*)
- **schema** (*Schema*)

Return type

[GigyaResponse](#)

10.4.2 Data models

Gigya models.

class `renault_api.gigya.models.GigyaGetAccountInfoData(raw_data, personId)`

Gigya GetAccountInfo data details.

Parameters

- **raw_data** (*dict[str, Any]*)
- **personId** (*str | None*)

class `renault_api.gigya.models.GigyaGetAccountInfoResponse(raw_data, errorCode, errorDetails, data)`

Gigya response to POST on /accounts.getAccountInfo.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errorCode** (*int*)
- **errorDetails** (*str | None*)
- **data** ([GigyaGetAccountInfoData](#) | *None*)

get_person_id()

Return person id.

Return type

str

class `renault_api.gigya.models.GigyaGetJWTResponse`(*raw_data, errorCode, errorDetails, id_token*)

Gigya response to POST on /accounts.getJWT.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errorCode** (*int*)
- **errorDetails** (*str | None*)
- **id_token** (*str | None*)

get_jwt()

Return jwt token.

Return type

str

class `renault_api.gigya.models.GigyaLoginResponse`(*raw_data, errorCode, errorDetails, sessionInfo*)

Gigya response to POST on /accounts.login.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errorCode** (*int*)
- **errorDetails** (*str | None*)
- **sessionInfo** (*GigyaLoginSessionInfo | None*)

get_session_cookie()

Return cookie value from session information.

Return type

str

class `renault_api.gigya.models.GigyaLoginSessionInfo`(*raw_data, cookieValue*)

Gigya Login sessionInfo details.

Parameters

- **raw_data** (*dict[str, Any]*)
- **cookieValue** (*str | None*)

class `renault_api.gigya.models.GigyaResponse`(*raw_data, errorCode, errorDetails*)

Gigya response.

Parameters

- **raw_data** (*dict[str, Any]*)
- **errorCode** (*int*)
- **errorDetails** (*str | None*)

raise_for_error_code()

Checks the response information.

Return type

None

10.5 Renault endpoints

This is a list of the endpoints available, and their characteristics.

10.5.1 Vehicle information endpoints

alerts

Base url:

/commerce/v1/accounts/{account_id}/kamereon/vehicles/{vin}/alerts

Sample return:

```
[
  {
    "id": "xxxxxxxxxx",
    "alertType": "proactive",
    "canArchitecture": "204",
    "alertCode": "crashairbagmalfunction",
    "trueValues": "1",
    "priority": 0,
    "minAge": 1,
    "minMileage": 100,
    "generateTime": 1440,
    "generateMileage": 15,
    "closeTime": 1440,
    "closeMileage": 36,
    "criticity": 3,
    "generateOccurrence": 3,
    "closeOccurrence": 10,
    "generateLead": 1,
    "notifSMS": 0,
    "notifMail": 1,
    "notifMybrand": 1,
    "notifSourceEvent": "CRIT3.RDVDEALER",
    "leadContactDays": 90,
    "notifContactDays": 5,
    "notifReminderDay": 15,
    "notifReminderCount": 0,
    "notifReminderCounter": 0,
    "notifReminderDate": "2024-03-15T00:00:00.000Z",
    "vin": "xxxxxxxxxx",
    "registrationNumber": "xxxxxxxxxx",
    "countryCode": "IT",
    "deliveryDate": "2020-10-30T00:00:00.000Z",
    "brandComCode": "REN",
    "modelCode": "HJB",
    "partyId": "xxxxxxxxxxxxxxxx",
  }
]
```

(continues on next page)

```
"partyCountry": "IT",
"partyOrg": 0,
"partyStaff": 0,
"partyPhone": 1,
"partySms": null,
"partyMail": null,
"partyPostal": null,
"localeDefault": "it-IT",
"dealerNumber": "xxxxxxxxxxx",
"dealerNumberOrigin": "dealer_myr",
"isConnected": 1,
"mileageDate": "2024-02-29T00:00:00.000Z",
"mileage": 63593,
"yearMileage": 16064,
"operationCode": null,
"operationName": null,
"operationType": null,
"nextDate": null,
"nextMileage": null,
"reconductCount": 0,
"isConnectedOperation": 0,
"genReason": null,
"endMileage": 70068,
"status": 0,
"creationDateTime": "2024-02-29T14:21:02.382Z",
"lastUpdateDateTime": "2024-10-29T02:08:45.671Z",
"inVehicleDateTime": "2024-02-29T14:21:15.000Z",
"closureReason": "archived",
"function": "the airbag",
"shortDescription": "WARNING: Check Airbag",
"longDescription": "WARNING: Check Airbag",
"criticityName": "Major alert",
>alertCodeTitle": "Defective airbag",
>alertCodeSubTitle": "Details of the event",
>alertCodeDescription": "The airbag must be inspected",
>alertCodeDetailDescription": "We have detected a malfunction in your airbag.
↪system.\nYou must inspect it to ensure that the front passengers are protected if
↪there is an impact. For further information, refer to the user manual of your
↪vehicle below.",
"iconUrl": "https://cap-static-medias.ope.apps.renault.com/images/crashAirbag.
↪png",
"detailImageUrl": "https://cap-static-medias.ope.apps.renault.com/images/
↪crashAirbag.png",
"userGuideUrl": "https://it.e-guide.renault.com/ita/",
"mileageUnit": "km"
}
]
```

10.5.2 Vehicle data endpoints

battery-status

Base url:

```
/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v2/cars/{vin}/
battery-status
```

Sample return:

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "timestamp": "2020-01-12T21:40:16Z",
      "batteryLevel": 60,
      "batteryTemperature": 20,
      "batteryAutonomy": 141,
      "batteryCapacity": 0,
      "batteryAvailableEnergy": 31,
      "plugStatus": 1,
      "chargingStatus": 1.0,
      "chargingRemainingTime": 145,
      "chargingInstantaneousPower": 27.0
    }
  }
}
```

Note

- batteryTemperature is not always present.
- batteryCapacity appears to always return 0.

On Zoe40 (model code X101VE):

- chargingInstantaneousPower gives value in watts.
- chargingStatus uses only a subset of ChargeStatus_ enum (NOT_IN_CHARGE = 0.0, CHARGE_IN_PROGRESS = 1.0, CHARGE_ERROR = -1.0)

On Zoe50 (model code X102VE):

- batteryTemperature appears completely wrong.
- chargingInstantaneousPower seems to return values in kilowatts, but the values still appear completely wrong.

charge-history

Base url:

```
/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/
charge-history
```

Sample return:

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "chargeSummaries": [
        {
          "day": "20201208",
          "totalChargesNumber": 2,
          "totalChargesDuration": 495,
          "totalChargesErrors": 0
        },
        {
          "day": "20201205",
          "totalChargesNumber": 1,
          "totalChargesDuration": 657,
          "totalChargesErrors": 0
        }
      ]
    }
  }
}
```

charge-mode

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/
charge-mode

Sample return:

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": { "chargeMode": "always" }
  }
}
```

Note**On older vehicles, such as Zoe40 (model code X101VE):**

The return values appear to be `always_charging` and `schedule_mode`. This matches the `vehicle_action.charge-mode` action attributes.

On newer vehicles, such as Zoe50 (model code X102VE):

The return values appear to be `always` and `scheduled`. This DOES NOT match the `vehicle_action.charge-mode` action attributes.

charge-schedule**Base url:**

```
/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/
charge-schedule
```

Sample return:**Single:**

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "calendar": {
        "monday": [
          { "startTime": "2300", "duration": 480, "activationState": false }
        ],
        "tuesday": [
          { "startTime": "2300", "duration": 480, "activationState": false }
        ],
        "wednesday": [
          { "startTime": "2300", "duration": 480, "activationState": false }
        ],
        "thursday": [
          { "startTime": "2300", "duration": 480, "activationState": false }
        ],
        "friday": [
          { "startTime": "2300", "duration": 480, "activationState": false }
        ],
        "saturday": [
          { "startTime": "2300", "duration": 480, "activationState": false }
        ],
        "sunday": [
          { "startTime": "2300", "duration": 480, "activationState": false }
        ]
      }
    }
  }
}
```

charges

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/charges

Sample return:

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "charges": [
        {
          "chargeStartDate": "2020-11-11T00:31:03Z",
          "chargeEndDate": "2020-11-11T08:30:17Z",
          "chargeDuration": 479,
          "chargeStartBatteryLevel": 15,
          "chargeEndBatteryLevel": 74,
          "chargeBatteryLevelRecovered": 59,
          "chargePower": "slow",
          "chargeStartInstantaneousPower": 3100,
          "chargeEndStatus": "ok"
        }
      ]
    }
  }
}
```

charging-settings

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/charging-settings

Sample return:

Single:

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "mode": "scheduled",
      "schedules": [
        {
          "id": 1,
          "activated": true,
          "monday": {
            "startTime": "T12:00Z",
            "duration": 15
          },
          "tuesday": {
            "startTime": "T04:30Z",

```

(continues on next page)

(continued from previous page)

```

        "duration": 420
      },
      "wednesday": {
        "startTime": "T22:30Z",
        "duration": 420
      },
      "thursday": {
        "startTime": "T22:00Z",
        "duration": 420
      },
      "friday": {
        "startTime": "T12:15Z",
        "duration": 15
      },
      "saturday": {
        "startTime": "T12:30Z",
        "duration": 30
      },
      "sunday": {
        "startTime": "T12:45Z",
        "duration": 45
      }
    }
  ]
}

```

Multiple:

```

{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "mode": "scheduled",
      "schedules": [
        {
          "id": 1,
          "activated": true,
          "monday": {
            "startTime": "T00:00Z",
            "duration": 450
          },
          "tuesday": {
            "startTime": "T00:00Z",
            "duration": 450
          },
          "wednesday": {
            "startTime": "T00:00Z",
            "duration": 450
          }
        }
      ]
    }
  }
}

```

(continues on next page)

(continued from previous page)

```
    "thursday": {
      "startTime": "T00:00Z",
      "duration": 450
    },
    "friday": {
      "startTime": "T00:00Z",
      "duration": 450
    },
    "saturday": {
      "startTime": "T00:00Z",
      "duration": 450
    },
    "sunday": {
      "startTime": "T00:00Z",
      "duration": 450
    }
  },
  {
    "id": 2,
    "activated": true,
    "monday": {
      "startTime": "T23:30Z",
      "duration": 15
    },
    "tuesday": {
      "startTime": "T23:30Z",
      "duration": 15
    },
    "wednesday": {
      "startTime": "T23:30Z",
      "duration": 15
    },
    "thursday": {
      "startTime": "T23:30Z",
      "duration": 15
    },
    "friday": {
      "startTime": "T23:30Z",
      "duration": 15
    },
    "saturday": {
      "startTime": "T23:30Z",
      "duration": 15
    },
    "sunday": {
      "startTime": "T23:30Z",
      "duration": 15
    }
  },
  {
    "id": 3,
    "activated": false
```

(continues on next page)

(continued from previous page)

```
    },
    {
      "id": 4,
      "activated": false
    },
    {
      "id": 5,
      "activated": false
    }
  ]
}
}
```

cockpit

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v2/cars/{vin}/cockpit

Sample return:

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777123",
    "attributes": {
      "fuelAutonomy": 35.0,
      "fuelQuantity": 3.0,
      "totalMileage": 5566.78
    }
  }
}
```

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "totalMileage": 49114.27
    }
  }
}
```

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": { "totalMileage": 49114.27 }
  }
}
```

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "fuelAutonomy": 0,
      "fuelQuantity": 0,
      "totalMileage": 5785.75
    }
  }
}
```

hvac-history

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/
hvac-history

Sample return:

Sample data is not yet available for this endpoint.

Please check the [Contributor Guide](#) to provide a sample.

hvac-sessions

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/
hvac-sessions

Sample return:

Sample data is not yet available for this endpoint.

Please check the [Contributor Guide](#) to provide a sample.

hvac-status

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/
hvac-status

Sample return:

```
{
  "data": {
    "type": "Car",
    "id": "UU1AAAAA555777123",
    "attributes": {
      "socThreshold": 30.0,
      "hvacStatus": "off",
      "lastUpdateTime": "2020-12-03T00:00:00Z"
    }
  }
}
```

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": { "externalTemperature": 8.0, "hvacStatus": "off" }
  }
}
```

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": { "socThreshold": 40, "hvacStatus": "on" }
  }
}
```

Note

On Zoe40 (model code X101VE):

hvacStatus seems to always report off, even when preconditioning is in progress.

On Zoe50 (model code X102VE):

This endpoint seem to be unavailable and returns an error 'err.func.403': 'Operation not supported Operation not supported for this can (C1A)'.

hvac-settings

Base url:

```
/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/
hvac-settings
```

Sample return:

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "dateTime": "2020-12-24T20:00:00.000Z",
      "mode": "scheduled",
      "schedules": [
        {
          "id": 1,
          "activated": false
        },
        {
          "id": 2,
          "activated": true,
          "wednesday": { "readyAtTime": "T15:15Z" },
          "friday": { "readyAtTime": "T15:15Z" }
        }
      ]
    }
  }
}
```

(continues on next page)

(continued from previous page)

```
    "id": 3,  
    "activated": false  
  },  
  {  
    "id": 4,  
    "activated": false  
  },  
  {  
    "id": 5,  
    "activated": false  
  }  
]  
}  
}
```

location

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/location

Sample return:

```
{  
  "data": {  
    "type": "Car",  
    "id": "VF1AAAAA555777999",  
    "attributes": {  
      "gpsLatitude": 48.1234567,  
      "gpsLongitude": 11.1234567,  
      "lastUpdateTime": "2020-02-18T16:58:38Z"  
    }  
  }  
}
```

```
{  
  "data": {  
    "type": "Car",  
    "id": "VF1AAAAA555777999",  
    "attributes": {  
      "gpsDirection": null,  
      "gpsLatitude": 48.1234567,  
      "gpsLongitude": 11.1234567,  
      "lastUpdateTime": "2020-02-18T16:58:38Z"  
    }  
  }  
}
```

lock-status

Base url:

```
/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/  
lock-status
```

Sample return:

```
{  
  "data": {  
    "type": "Car",  
    "id": "VF1AAAAA555777999",  
    "attributes": {  
      "lockStatus": "locked",  
      "doorStatusRearLeft": "closed",  
      "doorStatusRearRight": "closed",  
      "doorStatusDriver": "closed",  
      "doorStatusPassenger": "closed",  
      "hatchStatus": "closed",  
      "lastUpdateTime": "2022-02-02T13:51:13Z"  
    }  
  }  
}
```

```
{  
  "data": {  
    "type": "Car",  
    "id": "VF1AAAAA555777999",  
    "attributes": {  
      "lockStatus": "unlocked",  
      "doorStatusRearLeft": "closed",  
      "doorStatusRearRight": "closed",  
      "doorStatusDriver": "closed",  
      "doorStatusPassenger": "closed",  
      "hatchStatus": "closed",  
      "lastUpdateTime": "2022-02-02T13:51:13Z"  
    }  
  }  
}
```

notification-settings

Base url:

```
/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/  
notification-settings
```

Sample return:

Sample data is not yet available for this endpoint.

Please check the [Contributor Guide](#) to provide a sample.

pressure

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/pressure

Sample return:

```
{
  "data": {
    "type": "Car",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "flPressure": 2460,
      "frPressure": 2730,
      "rlPressure": 2790,
      "rrPressure": 2790,
      "flStatus": 0,
      "frStatus": 0,
      "rlStatus": 0,
      "rrStatus": 0
    }
  }
}
```

res-state

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/res-state

Sample return:

```
{
  "data": {
    "type": "ResState",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "details": "Stopped, ready for RES",
      "code": "10"
    }
  }
}
```

```
{
  "data": {
    "type": "ResState",
    "id": "VF1AAAAA555777999",
    "attributes": {
      "details": "Running",
      "code": "42"
    }
  }
}
```

10.5.3 Vehicle KCM data endpoints

ev/settings

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kcm/v1/vehicles/{vin}/ev/settings

Sample return:

Empty:

```
{
  "lastSettingsUpdateTimestamp": "2025-04-15T12:06:32.266Z",
  "delegatedActivated": false,
  "chargeModeRq": "SCHEDULED",
  "chargeTimeStart": "00:00",
  "chargeDuration": 9000,
  "preconditioningTemperature": 20.0,
  "preconditioningHeatedStrgWheel": false,
  "preconditioningHeatedRightSeat": false,
  "preconditioningHeatedLeftSeat": false,
  "programs": []
}
```

Single:

```
{
  "lastSettingsUpdateTimestamp": "2025-04-24T12:17:12.247Z",
  "delegatedActivated": false,
  "chargeModeRq": "SCHEDULED",
  "chargeTimeStart": "00:00",
  "chargeDuration": 9000,
  "preconditioningTemperature": 20.0,
  "preconditioningHeatedStrgWheel": false,
  "preconditioningHeatedRightSeat": false,
  "preconditioningHeatedLeftSeat": false,
  "programs": [
    {
      "programActivationStatus": true,
      "programType": "CHARGE_AND_PRECONDITIONING",
      "programDepartureTime": "02:00",
      "programActivationMonday": true,
      "programActivationTuesday": true,
      "programActivationWednesday": true,
      "programActivationThursday": true,
      "programActivationFriday": true,
      "programActivationSaturday": true,
      "programActivationSunday": true
    }
  ]
}
```

Multi:

```
{
  "lastSettingsUpdateTimestamp": "2025-12-30T17:03:18.644879Z",
```

(continues on next page)

```
"delegatedActivated": false,
"chargeModeRq": "SCHEDULED",
"chargeTimeStart": "01:55",
"chargeDuration": 300,
"preconditioningTemperature": 19,
"preconditioningHeatedStrgWheel": true,
"preconditioningHeatedRightSeat": false,
"preconditioningHeatedLeftSeat": false,
"programs": [
  {
    "programActivationStatus": true,
    "programType": "CHARGE",
    "programDepartureTime": "07:30",
    "programActivationMonday": true,
    "programActivationTuesday": true,
    "programActivationWednesday": false,
    "programActivationThursday": false,
    "programActivationFriday": true,
    "programActivationSaturday": false,
    "programActivationSunday": false
  },
  {
    "programActivationStatus": true,
    "programType": "PRECONDITIONING",
    "programDepartureTime": "08:30",
    "programActivationMonday": false,
    "programActivationTuesday": false,
    "programActivationWednesday": false,
    "programActivationThursday": true,
    "programActivationFriday": false,
    "programActivationSaturday": false,
    "programActivationSunday": false
  },
  {
    "programActivationStatus": true,
    "programType": "CHARGE_AND_PRECONDITIONING",
    "programDepartureTime": "10:50",
    "programActivationMonday": false,
    "programActivationTuesday": false,
    "programActivationWednesday": true,
    "programActivationThursday": false,
    "programActivationFriday": false,
    "programActivationSaturday": true,
    "programActivationSunday": false
  }
]
```

ev/soc-levels

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kcm/v1/vehicles/{vin}/ev/soc-level

Sample return:

```
{
  "lastEnergyUpdateTimestamp": "2025-04-18T06:51:09Z",
  "socMin": 20,
  "socTarget": 80
}
```

10.5.4 Action endpoints

actions/charge-mode

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/actions/charge-mode

Sample payload:

Use instant charging:

```
{
  "data": {
    "type": "ChargeMode",
    "attributes": {"action": "always_charging"}
  }
}
```

Use scheduled charging:

```
{
  "data": {
    "type": "ChargeMode",
    "attributes": {"action": "schedule_mode"}
  }
}
```

Please check the [Contributor Guide](#) to provide extra samples.

i Note

All vehicles seem to use *always_charging* and *schedule_mode*.

On older vehicles, such as Zoe40 (model code X101VE):

This matches the `vehicle_data.charge-mode` return values: `always_charging` and `schedule_mode`.

On newer vehicles, such as Zoe50 (model code X102VE):

This DOES NOT match the `vehicle_data.charge-mode` return values which are: `always` and `scheduled`.

actions/charge-schedule

Base url:

```
/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v2/cars/{vin}/actions/charge-schedule
```

Sample payload:

Sample payload is not yet available for this endpoint.

Please check the [Contributor Guide](#) to provide a sample.

actions/charging-start

Base url:

```
/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/actions/charging-start
```

Sample payload:

Start charge:

```
{
  "data": {
    "type": "ChargingStart",
    "attributes": {"action": "start"}
  }
}
```

Stop charge:

```
{
  "data": {
    "type": "ChargingStart",
    "attributes": {"action": "stop"}
  }
}
```

Please check the [Contributor Guide](#) to provide extra samples.

actions/horn-lights

Base url:

```
/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/actions/horn-lights
```

Sample payload:

Start horn:

```
{
  "data": {
    "type": "HornLights",
    "attributes": {"action": "start", "target": "horn"}
  }
}
```

Start lights:

```
{
  "data": {
    "type": "HornLights",
    "attributes": {"action": "start", "target": "lights"}
  }
}
```

actions/hvac-start

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/actions/hvac-start

Sample payload:

Sample payload is not yet available for this endpoint.

Please check the [Contributor Guide](#) to provide a sample.

Note

On Zoe50 (model code X102VE):

Payload {'action': 'cancel'} to stop HVAC does not create errors but has no effect on the vehicle (Renault side limitation).

actions/hvac-schedule

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v2/cars/{vin}/actions/hvac-schedule

Sample payload:

Sample payload is not yet available for this endpoint.

Please check the [Contributor Guide](#) to provide a sample.

actions/refresh-location

Base url:

/commerce/v1/accounts/{account_id}/kamereon/kca/car-adapter/v1/cars/{vin}/actions/refresh-location

Sample payload:

Refresh location:

```
{
  "data": {
```

(continues on next page)

(continued from previous page)

```
    "type": "RefreshLocation",  
  }  
}
```

10.5.5 KCM Action endpoints

charge/pause-resume

Base url:

```
/commerce/v1/accounts/{account_id}/kamereon/kcm/v1/vehicles/{vin}/charge/  
pause-resume
```

Sample return:

```
{  
  "data": {  
    "type": "ChargePauseResume",  
    "id": "guid",  
    "attributes": { "action": "resume" }  
  }  
}
```

```
{  
  "data": {  
    "type": "ChargePauseResume",  
    "id": "guid",  
    "attributes": { "action": "pause" }  
  }  
}
```

10.6 Contributor Guide

Thank you for your interest in improving this project. This project is open-source under the [MIT license](#) and welcomes contributions in the form of bug reports, feature requests, and pull requests.

Here is a list of important resources for contributors:

- [Source Code](#)
- [Documentation](#)
- [Issue Tracker](#)
- [Code of Conduct](#)

10.6.1 How to report a bug

Report bugs on the [Issue Tracker](#).

When filing an issue, make sure to answer these questions:

- Which operating system and Python version are you using?
- Which version of this project are you using?
- What did you do?

- What did you expect to see?
- What did you see instead?

The best way to get your bug fixed is to provide a test case, and/or steps to reproduce the issue.

10.6.2 How to request a feature

Request features on the [Issue Tracker](#).

10.6.3 Providing samples

Providing samples for more vehicles helps us improve the library. If you have a vehicle for which we do not have the specifications yet, or if you have found a new feature for an existing vehicle, then please send us the samples.

- Generate a trace.
- Add the json file to the correct `test/fixtures/kamereon` subfolder.
- Ensure that `vin` starts with `VFIAAAA` _(we do not want the real VIN)_
- Ensure that `vehicleDetails.vin` also starts with `VFIAAAA` _(we do not want the real VIN)_
- Ensure that `vehicleDetails.registrationNumber` starts with `REG-` _(we do not want the real registration number)_
- Ensure that `vehicleDetails.radioCode` is equal to `1234` _(we do not want the real radio code)_
- Ensure that the json file passes pre-commit (can be parsed online via <https://codebeautify.org/jsonviewer>)
- Create a pull request

10.6.4 How to set up your development environment

You need Python 3.10+ and the following tools:

- Poetry
- Nox
- nox-poetry

WARNING: due to an open issue with Poetry, we recommend that you use the 1.0.10 version. You can install it with the command:

```
$ pipx install poetry==1.0.10
```

Install the package with development requirements:

```
$ poetry install --extras "cli"
```

You can now run an interactive Python session, or the command-line interface:

```
$ poetry run python
$ poetry run renault-api
```

10.6.5 How to test the project

Run the full test suite:

```
$ nox
```

List the available Nox sessions:

```
$ nox --list-sessions
```

You can also run a specific Nox session. For example, invoke the unit test suite like this:

```
$ nox --session=tests
```

Unit tests are located in the `tests` directory, and are written using the `pytest` testing framework.

10.6.6 How to submit changes

Open a [pull request](#) to submit changes to this project.

Your pull request needs to meet the following guidelines for acceptance:

- The Nox test suite must pass without errors and warnings.
- Include unit tests. This project maintains 100% code coverage.
- If your changes add functionality, update the documentation accordingly.

Feel free to submit early, though—we can always iterate on this.

To run linting and code formatting checks before committing your change, you can install `pre-commit` as a Git hook by running the following command:

```
$ nox --session=pre-commit -- install
```

It is recommended to open an issue before starting work on anything. This will allow a chance to talk it over with the owners and validate your approach.

10.7 Contributor Covenant Code of Conduct

10.7.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

10.7.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment

- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

10.7.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

10.7.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

10.7.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at mail@tbd.com. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

10.7.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

10.7.7 Attribution

This Code of Conduct is adapted from the [Contributor Covenant](https://www.contributor-covenant.org/version/2/0/code_of_conduct.html), version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by [Mozilla's code of conduct enforcement ladder](#).

For answers to common questions about this code of conduct, see the FAQ at <https://www.contributor-covenant.org/faq>. Translations are available at <https://www.contributor-covenant.org/translations>.

10.8 MIT License

Copyright © 2020 epenet

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The software is provided “as is”, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

PYTHON MODULE INDEX

r

renault_api.gigya, 56
renault_api.gigya.models, 57
renault_api.kamereon, 39
renault_api.kamereon.models, 42
renault_api.renault_account, 33
renault_api.renault_client, 32
renault_api.renault_session, 30
renault_api.renault_vehicle, 34

Symbols

- account
 - renault-api command line option, 21
 - renault-api-set command line option, 29
- at
 - renault-api-charge-start command line option, 25
 - renault-api-hvac-start command line option, 28
- debug
 - renault-api command line option, 21
- friday
 - renault-api-charge-schedule-set command line option, 23
- from
 - renault-api-charge-history command line option, 22
 - renault-api-charge-sessions command line option, 24
 - renault-api-hvac-history command line option, 27
 - renault-api-hvac-sessions command line option, 28
- json
 - renault-api command line option, 21
- locale
 - renault-api command line option, 21
 - renault-api-set command line option, 29
- log
 - renault-api command line option, 21
- min
 - renault-api-charge-soclevels-set command line option, 25
- monday
 - renault-api-charge-schedule-set command line option, 23
- password
 - renault-api-login command line option, 28
- period
 - renault-api-charge-history command line option, 22
 - renault-api-hvac-history command line option, 27
- saturday
 - renault-api-charge-schedule-set command line option, 24
- set
 - renault-api-charge-mode command line option, 22
- sunday
 - renault-api-charge-schedule-set command line option, 24
- target
 - renault-api-charge-soclevels-set command line option, 25
- temperature
 - renault-api-hvac-start command line option, 28
- thursday
 - renault-api-charge-schedule-set command line option, 23
- to
 - renault-api-charge-history command line option, 22
 - renault-api-charge-sessions command line option, 24
 - renault-api-hvac-history command line option, 27
 - renault-api-hvac-sessions command line option, 28
- tuesday
 - renault-api-charge-schedule-set command line option, 23
- user
 - renault-api-login command line option, 28
- version
 - renault-api command line option, 21
- vin
 - renault-api command line option, 21
 - renault-api-set command line option, 29
- wednesday
 - renault-api-charge-schedule-set command line option, 23

A

account_id(*renault_api.renault_account.RenaultAccount* property), 33
 account_id(*renault_api.renault_vehicle.RenaultVehicle* property), 34

C

ChargeDaySchedule (class in *renault_api.kamereon.models*), 42
 ChargeSchedule (class in *renault_api.kamereon.models*), 43
 controls_action_via_kcm() (*renault_api.kamereon.models.KamereonVehicleCarAdapterDetails* method), 47
 controls_action_via_kcm() (*renault_api.kamereon.models.KamereonVehicleDetails* method), 50

E

ENDPOINT

renault-api-http-get command line option, 26
 renault-api-http-post command line option, 26
 renault-api-http-post-file command line option, 27

EndpointDefinition (class in *renault_api.kamereon.models*), 43

F

for_json() (*renault_api.kamereon.models.ChargeDaySchedule* method), 42
 for_json() (*renault_api.kamereon.models.ChargeSchedule* method), 43
 for_json() (*renault_api.kamereon.models.HvacDaySchedule* method), 43
 for_json() (*renault_api.kamereon.models.HvacSchedule* method), 44

G

get_account_info() (in module *renault_api.gigya*), 56
 get_account_url() (in module *renault_api.kamereon*), 39
 get_account_vehicles() (in module *renault_api.kamereon*), 39
 get_account_vehicles() (*renault_api.renault_session.RenaultSession* method), 30
 get_api_account() (*renault_api.renault_client.RenaultClient* method), 32
 get_api_accounts() (*renault_api.renault_client.RenaultClient* method), 32

get_api_vehicle() (*renault_api.renault_account.RenaultAccount* method), 33
 get_api_vehicles() (*renault_api.renault_account.RenaultAccount* method), 33
 get_asset() (*renault_api.kamereon.models.KamereonVehicleDetails* method), 51
 get_attributes() (*renault_api.kamereon.models.KamereonVehicleDataResponse* method), 50
 get_battery_soc() (*renault_api.renault_vehicle.RenaultVehicle* method), 34
 get_battery_status() (*renault_api.renault_vehicle.RenaultVehicle* method), 34
 get_brand_label() (*renault_api.kamereon.models.KamereonVehicleDetails* method), 51
 get_car_adapter() (*renault_api.renault_vehicle.RenaultVehicle* method), 34
 get_car_adapter_url() (in module *renault_api.kamereon*), 39
 get_charge_history() (*renault_api.renault_vehicle.RenaultVehicle* method), 34
 get_charge_mode() (*renault_api.renault_vehicle.RenaultVehicle* method), 35
 get_charge_schedule() (*renault_api.renault_vehicle.RenaultVehicle* method), 35
 get_charges() (*renault_api.renault_vehicle.RenaultVehicle* method), 35
 get_charging_settings() (*renault_api.renault_vehicle.RenaultVehicle* method), 35
 get_charging_status() (*renault_api.kamereon.models.KamereonVehicleBatteryStatusData* method), 46
 get_cockpit() (*renault_api.renault_vehicle.RenaultVehicle* method), 35
 get_commerce_url() (in module *renault_api.kamereon*), 39
 get_contracts() (*renault_api.renault_vehicle.RenaultVehicle* method), 35
 get_contracts_url() (in module *renault_api.kamereon*), 39
 get_details() (*renault_api.renault_vehicle.RenaultVehicle* method), 35
 get_end_time() (*renault_api.kamereon.models.ChargeDaySchedule*

method), 43
 get_endpoint() (*renault_api.kamereon.models.KamereonVehicleDetails* *method*), 51
 get_endpoint_definition() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 35
 get_endpoints() (*renault_api.kamereon.models.KamereonVehicleDetails* *method*), 51
 get_energy_code() (*renault_api.kamereon.models.KamereonVehicleDetails* *method*), 51
 get_error_details() (*renault_api.kamereon.models.KamereonResponseError* *method*), 45
 get_full_endpoint() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 35
 get_hvac_history() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 36
 get_hvac_sessions() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 36
 get_hvac_settings() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 36
 get_hvac_status() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 36
 get_jwt() (*in module renault_api.gigya*), 56
 get_jwt() (*renault_api.gigya.models.GigyaGetJWTResponse* *method*), 58
 get_location() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 36
 get_lock_status() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 36
 get_model_code() (*renault_api.kamereon.models.KamereonVehicleDetails* *method*), 51
 get_model_endpoint() (*in module renault_api.kamereon.models*), 56
 get_model_endpoints() (*in module renault_api.kamereon.models*), 56
 get_model_label() (*renault_api.kamereon.models.KamereonVehicleDetails* *method*), 51
 get_notification_settings() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 36
 get_person() (*in module renault_api.kamereon*), 40
 get_person() (*renault_api.renault_client.RenaultClient* *method*), 32
 get_person() (*renault_api.renault_session.RenaultSession* *method*), 30
 get_person_id() (*renault_api.gigya.models.GigyaGetAccountInfoResponse* *method*), 57
 get_person_url() (*in module renault_api.kamereon*), 40
 get_picture() (*renault_api.kamereon.models.KamereonVehicleDetails* *method*), 51
 get_plug_status() (*renault_api.kamereon.models.KamereonVehicleBatteryStatusData* *method*), 46
 get_res_state() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 37
 get_session_cookie() (*renault_api.gigya.models.GigyaLoginResponse* *method*), 58
 get_tyre_pressure() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 37
 get_vehicle_contracts() (*in module renault_api.kamereon*), 40
 get_vehicle_contracts() (*renault_api.renault_session.RenaultSession* *method*), 30
 get_vehicle_data() (*in module renault_api.kamereon*), 41
 get_vehicle_data() (*renault_api.renault_session.RenaultSession* *method*), 30
 get_vehicle_details() (*in module renault_api.kamereon*), 41
 get_vehicle_details() (*renault_api.renault_session.RenaultSession* *method*), 31
 get_vehicles() (*renault_api.renault_account.RenaultAccount* *method*), 33
 GigyaGetAccountInfoData (*class in renault_api.gigya.models*), 57
 GigyaGetAccountInfoResponse (*class in renault_api.gigya.models*), 57
 GigyaGetJWTResponse (*class in renault_api.gigya.models*), 58
 GigyaLoginResponse (*class in renault_api.gigya.models*), 58
 GigyaLoginSessionInfo (*class in renault_api.gigya.models*), 58
 GigyaResponse (*class in renault_api.gigya.models*), 58

H

http_get() (*renault_api.renault_vehicle.RenaultVehicle* *method*), 37

- `http_post()` (*renault_api.renault_vehicle.RenaultVehicle* method), 37
- `http_request()` (*renault_api.renault_session.RenaultSession* method), 31
- `HvacDaySchedule` (class in *renault_api.kamereon.models*), 43
- `HvacSchedule` (class in *renault_api.kamereon.models*), 43
- I**
- ID
- `renault-api-charge-schedule-activate` command line option, 23
 - `renault-api-charge-schedule-deactivate` command line option, 23
 - `renault-api-charge-schedule-set` command line option, 24
- J**
- JSON_BODY
- `renault-api-http-post` command line option, 26
 - `renault-api-http-post-file` command line option, 27
- K**
- `KamereonPersonAccount` (class in *renault_api.kamereon.models*), 44
- `KamereonPersonResponse` (class in *renault_api.kamereon.models*), 44
- `KamereonResponse` (class in *renault_api.kamereon.models*), 44
- `KamereonResponseError` (class in *renault_api.kamereon.models*), 45
- `KamereonVehicleBatterySocActionData` (class in *renault_api.kamereon.models*), 45
- `KamereonVehicleBatterySocData` (class in *renault_api.kamereon.models*), 45
- `KamereonVehicleBatteryStatusData` (class in *renault_api.kamereon.models*), 45
- `KamereonVehicleCarAdapterData` (class in *renault_api.kamereon.models*), 46
- `KamereonVehicleChargeHistoryData` (class in *renault_api.kamereon.models*), 48
- `KamereonVehicleChargeModeActionData` (class in *renault_api.kamereon.models*), 48
- `KamereonVehicleChargeModeData` (class in *renault_api.kamereon.models*), 48
- `KamereonVehicleChargeScheduleActionData` (class in *renault_api.kamereon.models*), 48
- `KamereonVehicleChargesData` (class in *renault_api.kamereon.models*), 48
- `KamereonVehicleChargingSettingsData` (class in *renault_api.kamereon.models*), 48
- `KamereonVehicleChargingStartActionData` (class in *renault_api.kamereon.models*), 49
- `KamereonVehicleCockpitData` (class in *renault_api.kamereon.models*), 49
- `KamereonVehicleContract` (class in *renault_api.kamereon.models*), 49
- `KamereonVehicleContractsResponse` (class in *renault_api.kamereon.models*), 49
- `KamereonVehicleData` (class in *renault_api.kamereon.models*), 49
- `KamereonVehicleDataAttributes` (class in *renault_api.kamereon.models*), 50
- `KamereonVehicleDataResponse` (class in *renault_api.kamereon.models*), 50
- `KamereonVehicleDetails` (class in *renault_api.kamereon.models*), 50
- `KamereonVehicleDetailsGroup` (class in *renault_api.kamereon.models*), 52
- `KamereonVehicleDetailsResponse` (class in *renault_api.kamereon.models*), 52
- `KamereonVehicleHvacHistoryData` (class in *renault_api.kamereon.models*), 53
- `KamereonVehicleHvacModeActionData` (class in *renault_api.kamereon.models*), 53
- `KamereonVehicleHvacScheduleActionData` (class in *renault_api.kamereon.models*), 53
- `KamereonVehicleHvacSessionsData` (class in *renault_api.kamereon.models*), 53
- `KamereonVehicleHvacSettingsData` (class in *renault_api.kamereon.models*), 53
- `KamereonVehicleHvacStartActionData` (class in *renault_api.kamereon.models*), 53
- `KamereonVehicleHvacStatusData` (class in *renault_api.kamereon.models*), 53
- `KamereonVehicleLocationData` (class in *renault_api.kamereon.models*), 54
- `KamereonVehicleLockStatusData` (class in *renault_api.kamereon.models*), 54
- `KamereonVehicleNotificationSettingsData` (class in *renault_api.kamereon.models*), 54
- `KamereonVehicleResStateData` (class in *renault_api.kamereon.models*), 55
- `KamereonVehiclesLink` (class in *renault_api.kamereon.models*), 55
- `KamereonVehiclesResponse` (class in *renault_api.kamereon.models*), 55
- `KamereonVehicleTyrePressureData` (class in *renault_api.kamereon.models*), 55
- L**
- `login()` (in module *renault_api.gigya*), 56
- `login()` (*renault_api.renault_session.RenaultSession* method), 31

login_token(*renault_api.renault_session.RenaultSession* property), 31

M

module

- renault_api.gigya*, 56
- renault_api.gigya.models*, 57
- renault_api.kamereon*, 39
- renault_api.kamereon.models*, 42
- renault_api.renault_account*, 33
- renault_api.renault_client*, 32
- renault_api.renault_session*, 30
- renault_api.renault_vehicle*, 34

R

raise_for_error_code() (*renault_api.gigya.models.GigyaResponse* method), 58

raise_for_error_code() (*renault_api.kamereon.models.KamereonResponse* method), 44

raise_for_error_code() (*renault_api.kamereon.models.KamereonResponseError* method), 45

renault_api.gigya module, 56

renault_api.gigya.models module, 57

renault_api.kamereon module, 39

renault_api.kamereon.models module, 42

renault_api.renault_account module, 33

renault_api.renault_client module, 32

renault_api.renault_session module, 30

renault_api.renault_vehicle module, 34

renault-api command line option

- account, 21
- debug, 21
- json, 21
- locale, 21
- log, 21
- version, 21
- vin, 21

renault-api-charge-history command line option

- from, 22
- period, 22
- to, 22

renault-api-charge-mode command line option

- set, 22

renault-api-charge-schedule-activate command line option

- ID, 23

renault-api-charge-schedule-deactivate command line option

- ID, 23

renault-api-charge-schedule-set command line option

- friday, 23
- monday, 23
- saturday, 24
- sunday, 24
- thursday, 23
- tuesday, 23
- wednesday, 23
- ID, 24

renault-api-charge-sessions command line option

- from, 24
- to, 24

renault-api-charge-soclevels-set command line option

- min, 25
- target, 25

renault-api-charge-start command line option

- at, 25

renault-api-http-get command line option

- ENDPOINT, 26

renault-api-http-post command line option

- ENDPOINT, 26
- JSON_BODY, 26

renault-api-http-post-file command line option

- ENDPOINT, 27
- JSON_BODY, 27

renault-api-hvac-history command line option

- from, 27
- period, 27
- to, 27

renault-api-hvac-sessions command line option

- from, 28
- to, 28

renault-api-hvac-start command line option

- at, 28
- temperature, 28

renault-api-login command line option

- password, 28
- user, 28

renault-api-set command line option

- account, 29

- locale, 29
--vin, 29
- RenaultAccount (class in *renault_api.renault_account*), 33
- RenaultClient (class in *renault_api.renault_client*), 32
- RenaultSession (class in *renault_api.renault_session*), 30
- RenaultVehicle (class in *renault_api.renault_vehicle*), 34
- reports_charge_session_durations_in_minutes() (*renault_api.kamereon.models.KamereonVehicleDetails* method), 52
- reports_charging_power_in_watts() (*renault_api.kamereon.models.KamereonVehicleCarAdapterData* method), 47
- reports_charging_power_in_watts() (*renault_api.kamereon.models.KamereonVehicleDetails* method), 52
- request() (in module *renault_api.gigya*), 57
- request() (in module *renault_api.kamereon*), 41
- ## S
- session (*renault_api.renault_account.RenaultAccount* property), 33
- session (*renault_api.renault_client.RenaultClient* property), 33
- session (*renault_api.renault_vehicle.RenaultVehicle* property), 37
- set_ac_start() (*renault_api.renault_vehicle.RenaultVehicle* method), 37
- set_ac_stop() (*renault_api.renault_vehicle.RenaultVehicle* method), 37
- set_battery_soc() (*renault_api.renault_vehicle.RenaultVehicle* method), 37
- set_charge_mode() (*renault_api.renault_vehicle.RenaultVehicle* method), 38
- set_charge_schedules() (*renault_api.renault_vehicle.RenaultVehicle* method), 38
- set_charge_start() (*renault_api.renault_vehicle.RenaultVehicle* method), 38
- set_charge_stop() (*renault_api.renault_vehicle.RenaultVehicle* method), 38
- set_hvac_schedules() (*renault_api.renault_vehicle.RenaultVehicle* method), 38
- set_login_token() (*renault_api.renault_session.RenaultSession* method), 31
- set_vehicle_action() (in module *renault_api.kamereon*), 42
- set_vehicle_action() (*renault_api.renault_session.RenaultSession* method), 32
- supports_endpoint() (*renault_api.kamereon.models.KamereonVehicleDetails* method), 52
- supports_endpoint() (*renault_api.renault_vehicle.RenaultVehicle* method), 38
- ## U
- update() (*renault_api.kamereon.models.KamereonVehicleChargingSettingsData* method), 48
- update() (*renault_api.kamereon.models.KamereonVehicleHvacSettingsData* method), 53
- uses_electricity() (*renault_api.kamereon.models.KamereonVehicleCarAdapterData* method), 47
- uses_electricity() (*renault_api.kamereon.models.KamereonVehicleDetails* method), 52
- uses_fuel() (*renault_api.kamereon.models.KamereonVehicleCarAdapterData* method), 47
- uses_fuel() (*renault_api.kamereon.models.KamereonVehicleDetails* method), 52
- ## V
- vin (*renault_api.renault_vehicle.RenaultVehicle* property), 38