Renault API

epenet

CONTENTS

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-conditionning 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 (>= 3.7.1)

API USAGE

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

```
$ pip install renault-api
```

```python
import aiohttp
import asyncio

from renault_api.renault_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())
```

FOUR

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 renault-api[cli]
```

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

```
$ renault-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

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 as a result of interacting with Renault's API using this project.

EIGHT

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.

9.1 CLI Usage

9.1.1 renault-api

Main entry point for the Renault CLI.

```
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.

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

charge

Charge functionality.

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

history

Display charge history.

```
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.

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

Options

--set <set>

> Target charge mode (schedule_mode/always/always_schedule)

schedule

Display or update charge schedules.

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

activate

Activate charging schedule {ID}.

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

Arguments

ID

> Required argument

deactivate

Deactivate charging schedule {ID}.

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

Arguments

ID

> Required argument

set

Update charging schedule {ID}.

```
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.

--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.

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

sessions

Display charge sessions.

```
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

start

Start charge.

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

stop

Stop charge.

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

contracts

Display vehicle contracts.

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

http

Raw HTTP.

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

get

Process HTTP GET request on endpoint.

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

Arguments

ENDPOINT
> Required argument

post

Process HTTP POST request on endpoint.

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

Arguments

ENDPOINT
> Required argument

JSON_BODY
> Required argument

post-file

Process HTTP POST request on endpoint.

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

Arguments

ENDPOINT

> Required argument

JSON_BODY

> Required argument

hvac

HVAC functionality.

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

cancel

Cancel air conditioning.

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

history

Display air conditioning history.

```
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.

```
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.

```
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'.

login

Login to Renault.

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

Options

--user <user>

--password <password>

reset

Clear all credentials/settings from the credential store.

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

set

Store specified settings into credential store.

```
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.

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

status

Display vehicle status.

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

vehicle

Display vehicle details.

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

vehicles

Display list of vehicles.

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

9.2 Renault high-level API Reference

- *Session*
- *Client*
- *Account*
- *Vehicle*

9.2.1 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** (*aiohttp.client.ClientSession*) –
> > - **locale** (*Optional[str]*) –
> > - **country** (*Optional[str]*) –
> > - **locale_details** (*Optional[Dict[str, str]]*) –
> > - **credential_store** (*Optional[renault_api.credential_store.CredentialStore]*) –
> >
> > **Return type** None
>
> **async get_account_vehicles**(*account_id*)
>
> > GET to /accounts/{account_id}/vehicles.
> >
> > > **Parameters account_id** (*str*) –
> > >
> > > **Return type** *renault_api.kamereon.models.KamereonVehiclesResponse*
>
> **async get_person**()
>
> > GET to /persons/{person_id}.
> >
> > > **Return type** *renault_api.kamereon.models.KamereonPersonResponse*
>
> **async get_vehicle_contracts**(*account_id*, *vin*)
>
> > GET to /v{endpoint_version}/cars/{vin}/contracts.
> >
> > > **Parameters**
> > >
> > > - **account_id** (*str*) –

> > - **vin** (*str*) –

> **Return type** *renault_api.kamereon.models.KamereonVehicleContractsResponse*

async **get_vehicle_data**(*account_id*, *vin*, *endpoint*, *params=None*, ***, *adapter_type='kca'*)

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

> > **Parameters**

> > > - **account_id** (*str*) –

> > > - **vin** (*str*) –

> > > - **endpoint** (*str*) –

> > > - **params** (*Optional[Dict[str, str]]*) –

> > > - **adapter_type** (*str*) –

> > **Return type** *renault_api.kamereon.models.KamereonVehicleDataResponse*

async **get_vehicle_details**(*account_id*, *vin*)

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

> > **Parameters**

> > > - **account_id** (*str*) –

> > > - **vin** (*str*) –

> > **Return type** *renault_api.kamereon.models.KamereonVehicleDetailsResponse*

async **http_request**(*method*, *endpoint*, *json=None*)

> GET to specified endpoint.

> > **Parameters**

> > > - **method** (*str*) –

> > > - **endpoint** (*str*) –

> > > - **json** (*Optional[Dict[str, Any]]*) –

> > **Return type** *renault_api.kamereon.models.KamereonResponse*

async **login**(*login_id*, *password*)

> Attempt login on Gigya.

> > **Parameters**

> > > - **login_id** (*str*) –

> > > - **password** (*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** *renault_api.kamereon.models.KamereonVehicleDataResponse*

9.2.2 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** (*Optional[*renault_api.renault_session.RenaultSession*]*) –
>
> - **websession** (*Optional[aiohttp.client.ClientSession]*) –
>
> - **locale** (*Optional[str]*) –
>
> - **country** (*Optional[str]*) –
>
> - **locale_details** (*Optional[Dict[str, str]]*) –
>
> - **credential_store** (*Optional[renault_api.credential_store.CredentialStore]*) –
>
> **Return type** None

async get_api_account(*account_id*)

> Get account proxy for specified account id.
>
> **Parameters account_id** (*str*) –
>
> **Return type** *renault_api.renault_account.RenaultAccount*

async get_api_accounts()

> Get account proxies.
>
> **Return type** *List*[*renault_api.renault_account.RenaultAccount*]

async get_person()

> GET to /persons/{person_id}.
>
> **Return type** *renault_api.kamereon.models.KamereonPersonResponse*

property session: renault_api.renault_session.RenaultSession

> Get session provider.

9.2.3 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** (*Optional[*renault_api.renault_session.RenaultSession*]*) –

- **websession** (*Optional[aiohttp.client.ClientSession]*) –

- **locale** (*Optional[str]*) –

- **country** (*Optional[str]*) –

- **locale_details** (*Optional[Dict[str, str]]*) –

- **credential_store**   (*Optional[renault_api.credential_store.*
  CredentialStore]) –

> **Return type** None

property account_id: str

> Get account id.

async get_api_vehicle(*vin*)

> Get vehicle proxy for specified vin.
>
> > **Parameters vin** (*str*) –
> >
> > **Return type** *renault_api.renault_vehicle.RenaultVehicle*

async get_api_vehicles()

> Get vehicle proxies.
>
> > **Return type** *List*[*renault_api.renault_vehicle.RenaultVehicle*]

async get_vehicles()

> GET to /accounts/{account_id}/vehicles.
>
> > **Return type** *renault_api.kamereon.models.KamereonVehiclesResponse*

property session: *renault_api.renault_session.RenaultSession*

> Get session provider.

9.2.4 Vehicle

Client for Renault API.

class renault_api.renault_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** (*Optional[*renault_api.renault_session.RenaultSession*]*) –
>
> - **websession** (*Optional[aiohttp.client.ClientSession]*) –
>
> - **locale** (*Optional[str]*) –
>
> - **country** (*Optional[str]*) –

- **locale_details** (*Optional[Dict[str, str]]*) –

- **credential_store** (*Optional[renault_api.credential_store.CredentialStore]*) –

- **vehicle_details** (*Optional[*[renault_api.kamereon.models.KamereonVehicleDetails](#)*]*) –

- **car_adapter** (*Optional[*[renault_api.kamereon.models.KamereonVehicleCarAdapterData](#)*]*) –

> **Return type** None

property account_id: str

> Get account id.

async get_battery_status()

> Get vehicle battery status.
>
> > **Return type** *[renault_api.kamereon.models.KamereonVehicleBatteryStatusData](#)*

async get_car_adapter()

> Get vehicle car adapter details.
>
> > **Return type** *[renault_api.kamereon.models.KamereonVehicleCarAdapterData](#)*

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

> Get vehicle charge history.
>
> > **Parameters**
> >
> > - **start** (*datetime.datetime*) –
> >
> > - **end** (*datetime.datetime*) –
> >
> > - **period** (*str*) –
> >
> > **Return type** *[renault_api.kamereon.models.KamereonVehicleChargeHistoryData](#)*

async get_charge_mode()

> Get vehicle charge mode.
>
> > **Return type** *[renault_api.kamereon.models.KamereonVehicleChargeModeData](#)*

async get_charges(*start*, *end*)

> Get vehicle charges.
>
> > **Parameters**
> >
> > - **start** (*datetime.datetime*) –
> >
> > - **end** (*datetime.datetime*) –
> >
> > **Return type** *[renault_api.kamereon.models.KamereonVehicleChargesData](#)*

async get_charging_settings()

> Get vehicle charging settings.
>
> > **Return type** *[renault_api.kamereon.models.KamereonVehicleChargingSettingsData](#)*

async get_cockpit()

> Get vehicle cockpit.
>
> > **Return type** *[renault_api.kamereon.models.KamereonVehicleCockpitData](#)*

async **get_contracts**()

> Get vehicle contracts.
>
> > **Return type** *List*[*renault_api.kamereon.models.KamereonVehicleContract*]

async **get_details**()

> Get vehicle details.
>
> > **Return type** *renault_api.kamereon.models.KamereonVehicleDetails*

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

> Get vehicle hvac history.
>
> > **Parameters**
> >
> > - **start** (`datetime.datetime`) –
> > - **end** (`datetime.datetime`) –
> > - **period** (`str`) –
> >
> > **Return type** *renault_api.kamereon.models.KamereonVehicleHvacHistoryData*

async **get_hvac_sessions**(*start*, *end*)

> Get vehicle hvac sessions.
>
> > **Parameters**
> >
> > - **start** (`datetime.datetime`) –
> > - **end** (`datetime.datetime`) –
> >
> > **Return type** *renault_api.kamereon.models.KamereonVehicleHvacSessionsData*

async **get_hvac_settings**()

> Get vehicle hvac settings (schedule+mode).
>
> > **Return type** *renault_api.kamereon.models.KamereonVehicleHvacSettingsData*

async **get_hvac_status**()

> Get vehicle hvac status.
>
> > **Return type** *renault_api.kamereon.models.KamereonVehicleHvacStatusData*

async **get_location**()

> Get vehicle location.
>
> > **Return type** *renault_api.kamereon.models.KamereonVehicleLocationData*

async **get_lock_status**()

> Get vehicle lock status.
>
> > **Return type** *renault_api.kamereon.models.KamereonVehicleLockStatusData*

async **get_notification_settings**()

> Get vehicle notification settings.
>
> > **Return type** *renault_api.kamereon.models.KamereonVehicleNotificationSettingsData*

async **get_res_state**()

> Get vehicle res state.
>
> > **Return type** *renault_api.kamereon.models.KamereonVehicleResStateData*

async **has_contract_for_endpoint**(*endpoint*)

> Check if vehicle has contract for endpoint.
>
> > **Parameters** **endpoint** (`str`) –
> >
> > **Return type** bool

property **session:** [*renault_api.renault_session.RenaultSession*](#)

> Get session.

async **set_ac_start**(*temperature*, *when=None*)

> Start vehicle ac.
>
> > **Parameters**
> >
> > - **temperature** (`float`) –
> > - **when** (`Optional[datetime.datetime]`) –
> >
> > **Return type** [*renault_api.kamereon.models.KamereonVehicleHvacStartActionData*](#)

async **set_ac_stop**()

> Stop vehicle ac.
>
> > **Return type** [*renault_api.kamereon.models.KamereonVehicleHvacStartActionData*](#)

async **set_charge_mode**(*charge_mode*)

> Set vehicle charge mode.
>
> > **Parameters** **charge_mode** (`str`) –
> >
> > **Return type** [*renault_api.kamereon.models.KamereonVehicleChargeModeActionData*](#)

async **set_charge_schedules**(*schedules*)

> Set vehicle charge schedules.
>
> > **Parameters** **schedules** (`List[`[renault_api.kamereon.models.ChargeSchedule](#)`]`) –
> >
> > **Return type** [*renault_api.kamereon.models.KamereonVehicleChargeScheduleActionData*](#)

async **set_charge_start**()

> Start vehicle charge.
>
> > **Return type** [*renault_api.kamereon.models.KamereonVehicleChargingStartActionData*](#)

async **set_charge_stop**()

> Start vehicle charge.
>
> > **Return type** [*renault_api.kamereon.models.KamereonVehicleChargingStartActionData*](#)

async **set_hvac_schedules**(*schedules*)

> Set vehicle charge schedules.
>
> > **Parameters** **schedules** (`List[`[renault_api.kamereon.models.HvacSchedule](#)`]`) –
> >
> > **Return type** [*renault_api.kamereon.models.KamereonVehicleHvacScheduleActionData*](#)

async **supports_endpoint**(*endpoint*)

> Check if vehicle supports endpoint.
>
> > **Parameters** **endpoint** (`str`) –
> >
> > **Return type** bool

property vin: str

> Get vin.

async warn_on_method(*method*)

> Log a warning if the method requires it.
>
> > **Parameters method** (*str*) –
> >
> > **Return type** None

9.3 Kamereon low-level API Reference

- *Core*
- *Data models*

9.3.1 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** (*aiohttp.client.ClientSession*) –
> > - **root_url** (*str*) –
> > - **api_key** (*str*) –
> > - **gigya_jwt** (*str*) –
> > - **country** (*str*) –
> > - **account_id** (*str*) –
> >
> > **Return type** *renault_api.kamereon.models.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**(*websession*, *root_url*, *api_key*, *gigya_jwt*, *country*, *person_id*)

>GET to /persons/{person_id}.

>>> **Parameters**

- **websession** (*aiohttp.client.ClientSession*) –

- **root_url** (*str*) –

- **api_key** (*str*) –

- **gigya_jwt** (*str*) –

- **country** (*str*) –

- **person_id** (*str*) –

>>> **Return type** *renault_api.kamereon.models.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

renault_api.kamereon.**get_required_contracts**(*endpoint*)

>Get the required contracts for the specified endpoint.

>>> **Parameters** **endpoint** (*str*) –

>>> **Return type** str

async renault_api.kamereon.**get_vehicle_contracts**(*websession*, *root_url*, *api_key*, *gigya_jwt*, *country*, *locale*, *account_id*, *vin*)

> GET to /accounts/{accountId}/vehicles/{vin}/contracts.
>
> > **Parameters**
> >
> > - **websession** (*aiohttp.client.ClientSession*) –
> > - **root_url** (*str*) –
> > - **api_key** (*str*) –
> > - **gigya_jwt** (*str*) –
> > - **country** (*str*) –
> > - **locale** (*str*) –
> > - **account_id** (*str*) –
> > - **vin** (*str*) –
> >
> > **Return type** *renault_api.kamereon.models.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** (*aiohttp.client.ClientSession*) –
> > - **root_url** (*str*) –
> > - **api_key** (*str*) –
> > - **gigya_jwt** (*str*) –
> > - **country** (*str*) –
> > - **account_id** (*str*) –
> > - **vin** (*str*) –
> > - **endpoint** (*str*) –
> > - **endpoint_version** (*Optional[int]*) –
> > - **params** (*Optional[Dict[str, str]]*) –
> > - **adapter_type** (*str*) –
> >
> > **Return type** *renault_api.kamereon.models.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** (*aiohttp.client.ClientSession*) –
> > - **root_url** (*str*) –
> > - **api_key** (*str*) –
> > - **gigya_jwt** (*str*) –

- **country** (*str*) –
- **account_id** (*str*) –
- **vin** (*str*) –

> **Return type** *renault_api.kamereon.models.KamereonVehicleDetailsResponse*

renault_api.kamereon.**has_required_contracts**(*contracts*, *endpoint*)

> Check if vehicle has contract for endpoint.
>
> > **Parameters**
> >
> > - **contracts** (*List[renault_api.kamereon.models.KamereonVehicleContract]*) –
> > - **endpoint** (*str*) –
> >
> > **Return type** bool

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** (*aiohttp.client.ClientSession*) –
> > - **method** (*str*) –
> > - **url** (*str*) –
> > - **api_key** (*str*) –
> > - **gigya_jwt** (*str*) –
> > - **params** (*Dict[str, str]*) –
> > - **json** (*Optional[Dict[str, Any]]*) –
> > - **schema** (*Optional[marshmallow.schema.Schema]*) –
> > - **wrap_array_in** (*Optional[str]*) –
> >
> > **Return type** *renault_api.kamereon.models.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** (*aiohttp.client.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** (*Optional[int]*) –
- **data_type** (*Optional[Dict[str, Any]]*) –
- **adapter_type** (*str*) –

Return type *renault_api.kamereon.models.KamereonVehicleDataResponse*

9.3.2 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** (*Optional[str]*) –
> - **duration** (*Optional[int]*) –
>
> **Return type** None

for_json()

Create dict for json.

> **Return type** *Dict*[str, *Any*]

get_end_time()

Get end time.

> **Return type** *Optional*[str]

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** (*Optional[int]*) –
> - **activated** (*Optional[bool]*) –
> - **monday** (*Optional[*renault_api.kamereon.models.ChargeDaySchedule*]*) –
> - **tuesday** (*Optional[*renault_api.kamereon.models.ChargeDaySchedule*]*) –
> - **wednesday** (*Optional[*renault_api.kamereon.models.ChargeDaySchedule*]*) –
> - **thursday** (*Optional[*renault_api.kamereon.models.ChargeDaySchedule*]*) –
> - **friday** (*Optional[*renault_api.kamereon.models.ChargeDaySchedule*]*) –
> - **saturday** (*Optional[*renault_api.kamereon.models.ChargeDaySchedule*]*) –
> - **sunday** (*Optional[*renault_api.kamereon.models.ChargeDaySchedule*]*) –
>
> **Return type** None

> `for_json()`
>> Create dict for json.
>>
>>> **Return type** *Dict*[str, *Any*]

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

> Kamereon vehicle hvac schedule for day.
>
>> **Parameters**
>>
>>> • **raw_data** (`Dict[str, Any]`) –
>>>
>>> • **readyAtTime** (`Optional[str]`) –
>>
>> **Return type** None
>
> `for_json()`
>> Create dict for json.
>>
>>> **Return type** *Dict*[str, *Optional*[str]]

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** (`Optional[int]`) –
>>>
>>> • **activated** (`Optional[bool]`) –
>>>
>>> • **monday** (`Optional[`renault_api.kamereon.models.HvacDaySchedule`]`) –
>>>
>>> • **tuesday** (`Optional[`renault_api.kamereon.models.HvacDaySchedule`]`) –
>>>
>>> • **wednesday** (`Optional[`renault_api.kamereon.models.HvacDaySchedule`]`) –
>>>
>>> • **thursday** (`Optional[`renault_api.kamereon.models.HvacDaySchedule`]`) –
>>>
>>> • **friday** (`Optional[`renault_api.kamereon.models.HvacDaySchedule`]`) –
>>>
>>> • **saturday** (`Optional[`renault_api.kamereon.models.HvacDaySchedule`]`) –
>>>
>>> • **sunday** (`Optional[`renault_api.kamereon.models.HvacDaySchedule`]`) –
>>
>> **Return type** 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** (`Optional[str]`) –
>>>
>>> • **accountType** (`Optional[str]`) –
>>>
>>> • **accountStatus** (`Optional[str]`) –

> **Return type** 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** (*Optional[List[*renault_api.kamereon.models. KamereonResponseError*]]*) –
> - **accounts** (*Optional[List[*renault_api.kamereon.models. KamereonPersonAccount*]]*) –
>
> **Return type** None

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

Kamereon response.

> **Parameters**
>
> - **raw_data** (*Dict[str, Any]*) –
> - **errors** (*Optional[List[*renault_api.kamereon.models. KamereonResponseError*]]*) –
>
> **Return type** 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** (*Optional[str]*) –
> - **errorMessage** (*Optional[str]*) –
>
> **Return type** None

> **get_error_details**()
>
> Extract the error details sometimes hidden inside nested JSON.
>
> > **Return type** *Optional*[str]

> **raise_for_error_code**()
>
> Raise exception from response error.
>
> > **Return type** None

class renault_api.kamereon.models.**KamereonVehicleBatteryStatusData**(*raw_data*, *timestamp*,
batteryLevel,
batteryTemperature,
batteryAutonomy,
batteryCapacity,
batteryAvailableEnergy,
plugStatus, *chargingStatus*,
chargingRemainingTime,
chargingInstantaneous-
Power)

Kamereon vehicle battery-status data.

> **Parameters**
>
> - **raw_data** (*Dict[str, Any]*) –
> - **timestamp** (*Optional[str]*) –
> - **batteryLevel** (*Optional[int]*) –
> - **batteryTemperature** (*Optional[int]*) –
> - **batteryAutonomy** (*Optional[int]*) –
> - **batteryCapacity** (*Optional[int]*) –
> - **batteryAvailableEnergy** (*Optional[int]*) –
> - **plugStatus** (*Optional[int]*) –
> - **chargingStatus** (*Optional[float]*) –
> - **chargingRemainingTime** (*Optional[int]*) –
> - **chargingInstantaneousPower** (*Optional[float]*) –
>
> **Return type** None

> **get_charging_status()**
>
> > Return charging status.
> >
> > > **Return type** *Optional*[renault_api.kamereon.enums.ChargeState]

> **get_plug_status()**
>
> > Return plug status.
> >
> > > **Return type** *Optional*[renault_api.kamereon.enums.PlugState]

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** (`Optional[str]`) –
- **vehicleId** (`Optional[int]`) –
- **batteryCode** (`Optional[str]`) –
- **brand** (`Optional[str]`) –
- **canGeneration** (`Optional[str]`) –
- **carGateway** (`Optional[str]`) –
- **deliveryCountry** (`Optional[str]`) –
- **deliveryDate** (`Optional[str]`) –
- **energy** (`Optional[str]`) –
- **engineType** (`Optional[str]`) –
- **familyCode** (`Optional[str]`) –
- **firstRegistrationDate** (`Optional[str]`) –
- **gearbox** (`Optional[str]`) –
- **modelCode** (`Optional[str]`) –
- **modelCodeDetail** (`Optional[str]`) –
- **modelName** (`Optional[str]`) –
- **radioType** (`Optional[str]`) –
- **region** (`Optional[str]`) –
- **registrationCountry** (`Optional[str]`) –
- **registrationNumber** (`Optional[str]`) –
- **tcuCode** (`Optional[str]`) –
- **versionCode** (`Optional[str]`) –
- **privacyMode** (`Optional[str]`) –
- **privacyModeUpdateDate** (`Optional[str]`) –
- **svtFlag** (`Optional[bool]`) –
- **svtBlockFlag** (`Optional[bool]`) –

Return type 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

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

    Kamereon vehicle data charge-history attributes.

        Parameters `raw_data` (`Dict[str, Any]`) –

        Return type None

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

    Kamereon vehicle action data charge-mode attributes.

        Parameters `raw_data` (`Dict[str, Any]`) –

        Return type None

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

    Kamereon vehicle data charge-mode attributes.

        Parameters

            • `raw_data` (`Dict[str, Any]`) –

            • `chargeMode` (`Optional[str]`) –

        Return type None

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

    Kamereon vehicle action data charge-schedule attributes.

        Parameters `raw_data` (`Dict[str, Any]`) –

        Return type None

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

    Kamereon vehicle data charges attributes.

        Parameters `raw_data` (`Dict[str, Any]`) –

        Return type None

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

    Kamereon vehicle data charging-settings attributes.

        Parameters

            • `raw_data` (`Dict[str, Any]`) –

            • `mode` (`Optional[str]`) –

- **schedules** (*Optional[List[*renault_api.kamereon.models.ChargeSchedule*]])*) –

    Return type 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]*) –

　　　　Return type None

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

　　Kamereon vehicle data cockpit attributes.

　　　　Parameters

- **raw_data** (*Dict[str, Any]*) –
- **fuelAutonomy** (*Optional[float]*) –
- **fuelQuantity** (*Optional[float]*) –
- **totalMileage** (*Optional[float]*) –

　　　　Return type 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** (*Optional[str]*) –
- **contractId** (*Optional[str]*) –
- **code** (*Optional[str]*) –
- **group** (*Optional[str]*) –
- **durationMonths** (*Optional[int]*) –
- **startDate** (*Optional[str]*) –
- **endDate** (*Optional[str]*) –
- **status** (*Optional[str]*) –
- **statusLabel** (*Optional[str]*) –
- **description** (*Optional[str]*) –

　　　　Return type 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** (`Optional[List[`renault_api.kamereon.models. KamereonResponseError`]]`) –
> >
> > - **contractList** (`Optional[List[`renault_api.kamereon.models. KamereonVehicleContract`]]`) –
> >
> > **Return type** None

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

> Kamereon vehicle data.
>
> > **Parameters**
> >
> > - **raw_data** (`Dict[str, Any]`) –
> >
> > - **type** (`Optional[str]`) –
> >
> > - **id** (`Optional[str]`) –
> >
> > - **attributes** (`Optional[Dict[str, Any]]`) –
> >
> > **Return type** None

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

> Kamereon vehicle data attributes.
>
> > **Parameters raw_data** (`Dict[str, Any]`) –
> >
> > **Return type** None

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** (`Optional[List[`renault_api.kamereon.models. KamereonResponseError`]]`) –
> >
> > - **data** (`Optional[`renault_api.kamereon.models.KamereonVehicleData`]`) –
> >
> > **Return type** None

> **get_attributes**(*schema*)
>
> > Return jwt token.
> >
> > > **Parameters schema** (`marshmallow.schema.Schema`) –
> > >
> > > **Return type** *Optional*[*renault_api.kamereon.models.KamereonVehicleDataAttributes*]

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

> Kamereon vehicle details.
>
> > **Parameters**

- **raw_data** (*Dict[str, Any]*) –

- **vin** (*Optional[str]*) –

- **registrationNumber** (*Optional[str]*) –

- **radioCode** (*Optional[str]*) –

- **brand** (*Optional[*renault_api.kamereon.models.KamereonVehicleDetailsGroup*]*) –

- **model** (*Optional[*renault_api.kamereon.models.KamereonVehicleDetailsGroup*]*) –

- **energy** (*Optional[*renault_api.kamereon.models.KamereonVehicleDetailsGroup*]*) –

- **engineEnergyType** (*Optional[str]*) –

> **Return type** None

controls_action_via_kcm(*action*)

> Return True if model uses endpoint via kcm.
>
> > **Parameters** **action** (*str*) –
> >
> > **Return type** bool

get_brand_label()

> Return vehicle model label.
>
> > **Return type** *Optional*[str]

get_energy_code()

> Return vehicle energy code.
>
> > **Return type** *Optional*[str]

get_model_code()

> Return vehicle model code.
>
> > **Return type** *Optional*[str]

get_model_label()

> Return vehicle model label.
>
> > **Return type** *Optional*[str]

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

warns_on_method(*method*)

> Return warning message if model trigger a warning on the method call.

>> **Parameters** **method** (`str`) –

>> **Return type** *Optional*[str]

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

> Kamereon vehicle details group data.

>> **Parameters**
>> - **raw_data** (`Dict[str, Any]`) –
>> - **code** (`Optional[str]`) –
>> - **label** (`Optional[str]`) –
>> - **group** (`Optional[str]`) –

>> **Return type** None

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

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

>> **Parameters**
>> - **raw_data** (`Dict[str, Any]`) –
>> - **vin** (`Optional[str]`) –
>> - **registrationNumber** (`Optional[str]`) –
>> - **radioCode** (`Optional[str]`) –
>> - **brand** (`Optional[`renault_api.kamereon.models.KamereonVehicleDetailsGroup`]`) –
>> - **model** (`Optional[`renault_api.kamereon.models.KamereonVehicleDetailsGroup`]`) –
>> - **energy** (`Optional[`renault_api.kamereon.models. KamereonVehicleDetailsGroup`]`) –
>> - **engineEnergyType** (`Optional[str]`) –
>> - **errors** (`Optional[List[`renault_api.kamereon.models. KamereonResponseError`]]`) –

>> **Return type** None

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

    Kamereon vehicle data hvac-history attributes.

        Parameters raw_data (`Dict[str, Any]`) –

        Return type None

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

    Kamereon vehicle action data hvac-mode attributes.

        Parameters raw_data (`Dict[str, Any]`) –

        Return type None

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

    Kamereon vehicle action data hvac-schedule attributes.

        Parameters raw_data (`Dict[str, Any]`) –

        Return type None

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

    Kamereon vehicle data hvac-sessions attributes.

        Parameters raw_data (`Dict[str, Any]`) –

        Return type None

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** (`Optional[str]`) –

            • **schedules** (`Optional[List[`renault_api.kamereon.models.HvacSchedule`]]`) –

        Return type None

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

    Kamereon vehicle action data hvac-start attributes.

        Parameters raw_data (`Dict[str, Any]`) –

        Return type None

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

    Kamereon vehicle data hvac-status attributes.

        Parameters

            • **raw_data** (`Dict[str, Any]`) –

            • **lastUpdateTime** (`Optional[str]`) –

            • **externalTemperature** (`Optional[float]`) –

            • **hvacStatus** (`Optional[str]`) –

            • **nextHvacStartDate** (`Optional[str]`) –

> * **socThreshold** (*Optional[float]*) –

> **Return type** None

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

> Kamereon vehicle data location attributes.

> > **Parameters**

> > > * **raw_data** (*Dict[str, Any]*) –

> > > * **lastUpdateTime** (*Optional[str]*) –

> > > * **gpsLatitude** (*Optional[float]*) –

> > > * **gpsLongitude** (*Optional[float]*) –

> > **Return type** 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** (*Optional[str]*) –

> > > * **doorStatusRearLeft** (*Optional[str]*) –

> > > * **doorStatusRearRight** (*Optional[str]*) –

> > > * **doorStatusDriver** (*Optional[str]*) –

> > > * **doorStatusPassenger** (*Optional[str]*) –

> > > * **hatchStatus** (*Optional[str]*) –

> > > * **lastUpdateTime** (*Optional[str]*) –

> > **Return type** None

class renault_api.kamereon.models.**KamereonVehicleNotificationSettingsData**(*raw_data*)

> Kamereon vehicle data notification-settings attributes.

> > **Parameters** **raw_data** (*Dict[str, Any]*) –

> > **Return type** None

class renault_api.kamereon.models.**KamereonVehicleResStateData**(*raw_data*, *details*, *code*)

> Kamereon vehicle data res-set attributes.

> > **Parameters**

> > > * **raw_data** (*Dict[str, Any]*) –

> > > * **details** (*Optional[str]*) –

> > > * **code** (*Optional[str]*) –

> > **Return type** None

class renault_api.kamereon.models.**KamereonVehiclesLink**(*raw_data*, *vin*, *vehicleDetails*)

    Kamereon vehicles link data.

        Parameters

- **raw_data** (*Dict[str, Any]*) –

- **vin** (*Optional[str]*) –

- **vehicleDetails**   (*Optional[*renault_api.kamereon.models.
  KamereonVehicleDetails*]*) –

        Return type None

class renault_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**   (*Optional[List[*renault_api.kamereon.models.
  KamereonResponseError*]]*) –

- **accountId** (*Optional[str]*) –

- **country** (*Optional[str]*) –

- **vehicleLinks**   (*Optional[List[*renault_api.kamereon.models.
  KamereonVehiclesLink*]]*) –

        Return type None

9.4 Gigya low-level authentication API Reference

- *Core*
- *Data models*

9.4.1 Core

Gigya API.

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

    Send POST to /accounts.getAccountInfo.

        Parameters

- **websession** (*aiohttp.client.ClientSession*) –

- **root_url** (*str*) –

- **api_key** (*str*) –

- **login_token** (*str*) –

        Return type *renault_api.gigya.models.GigyaGetAccountInfoResponse*

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

> Send POST to /accounts.getJWT.
>
> > **Parameters**
> >
> > - **websession** (*aiohttp.client.ClientSession*) –
> > - **root_url** (*str*) –
> > - **api_key** (*str*) –
> > - **login_token** (*str*) –
> >
> > **Return type** *renault_api.gigya.models.GigyaGetJWTResponse*

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

> Send POST to /accounts.login.
>
> > **Parameters**
> >
> > - **websession** (*aiohttp.client.ClientSession*) –
> > - **root_url** (*str*) –
> > - **api_key** (*str*) –
> > - **login_id** (*str*) –
> > - **password** (*str*) –
> >
> > **Return type** *renault_api.gigya.models.GigyaLoginResponse*

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

> Send request to Gigya.
>
> > **Parameters**
> >
> > - **websession** (*aiohttp.client.ClientSession*) –
> > - **method** (*str*) –
> > - **url** (*str*) –
> > - **data** (*Dict[str, Any]*) –
> > - **schema** (*marshmallow.schema.Schema*) –
> >
> > **Return type** *renault_api.gigya.models.GigyaResponse*

9.4.2 Data models

Gigya models.

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

> Gigya GetAccountInfo data details.
>
> > **Parameters**
> >
> > - **raw_data** (*Dict[str, Any]*) –
> > - **personId** (*Optional[str]*) –
> >
> > **Return type** 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** (`Optional[str]`) –
> >
> > - **data** (`Optional[`[renault_api.gigya.models.GigyaGetAccountInfoData](#)`]`) –
> >
> > **Return type** 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** (`Optional[str]`) –
> >
> > - **id_token** (`Optional[str]`) –
> >
> > **Return type** 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** (`Optional[str]`) –
> >
> > - **sessionInfo** (`Optional[`[renault_api.gigya.models.GigyaLoginSessionInfo](#)`]`) –
> >
> > **Return type** 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** (*Optional[str]*) –

    Return type None

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

Gigya response.

Parameters

- **raw_data** (*Dict[str, Any]*) –

- **errorCode** (*int*) –

- **errorDetails** (*Optional[str]*) –

    Return type None

raise_for_error_code()

Checks the response information.

    Return type None

9.5 Renault endpoints

> – *actions/hvac-schedule*
>
> • *KCM Action endpoints*

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

9.5.1 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" }
  }
}
```

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:

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

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

(continues on next page)

```
                }
            ]
        }
    }
}
```

location

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

Sample return:

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

```json
{
  "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:

```json
{
  "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.

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"
    }
  }
}
```

9.5.2 Action endpoints

actions/charge-mode

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

Sample payload: Sample payload is not yet available for this endpoint.

Please check the Contributor Guide to provide a sample.

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/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.

9.5.3 KCM Action endpoints

9.6 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

9.6.1 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.

9.6.2 How to request a feature

Request features on the Issue Tracker.

9.6.3 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 *VF1AAAA* _(we do not want the real VIN)_

• Ensure that *vehicleDetails.vin* also starts with *VF1AAAA* _(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

9.6.4 How to set up your development environment

You need Python 3.7+ and the following tools:

• Poetry

• Nox

• nox-poetry

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

```
$ 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
```

9.6.5 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.

9.6.6 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 commiting 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.

9.7 Contributor Covenant Code of Conduct

9.7.1 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.

9.7.2 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

9.7.3 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.

9.7.4 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.

9.7.5 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.

9.7.6 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.

9.7.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, 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.

9.8 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

INDEX