1. Lease API Overview

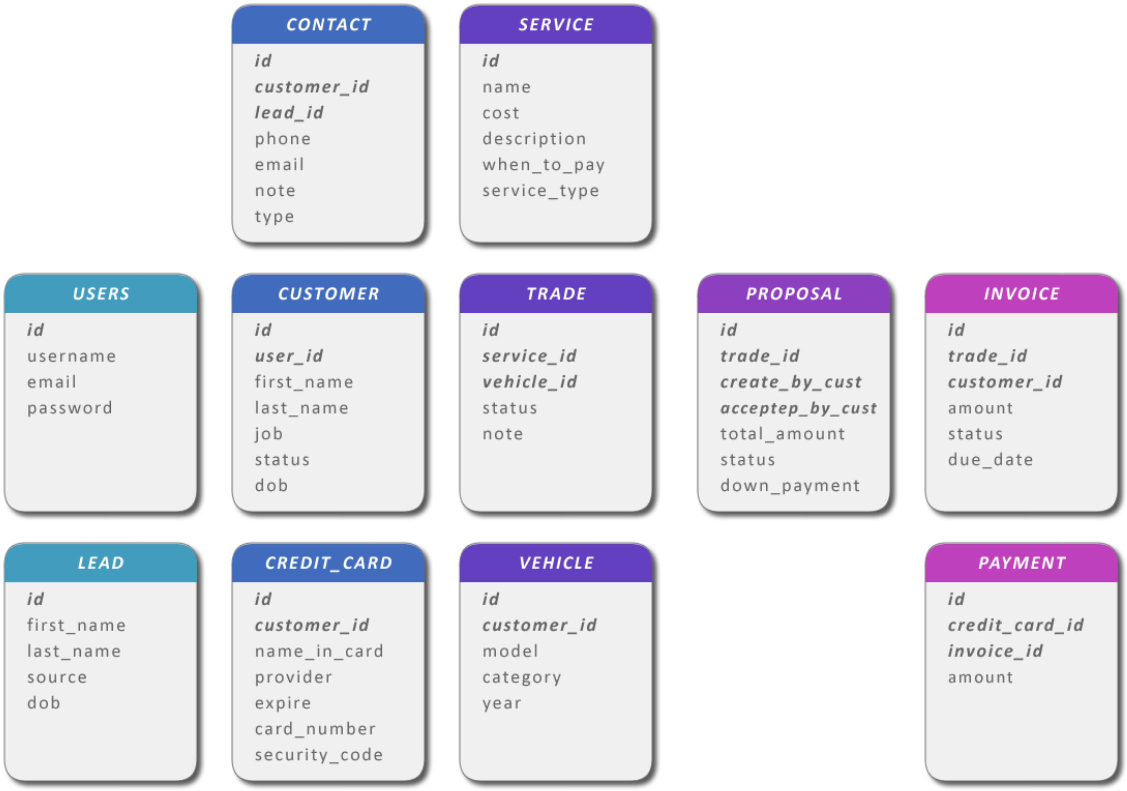
The Lease API is built for a lease company that focus on selling or leasing vehicles. The company doesn't own any vehicle but provides a platform for customers to offer their own vehicles to sell or lease and other clients to shop around.

The API is designer with Django REST API version 4.0.6, uses token authentication and is fully tested with Pytest 7.1.2 having over 300 tests for the models and the API.

Model General Design

The model consist of 11 tables. Each User can log in into the system and he/she will have a related Customer if the user is non administrative. Each Customer can store Credit Card and Contact information. Contacts can be also created for leads which can only be accessed by administrative users.

A Customer can also create a Vehicle for which he can latter create a Trade using one of the available Services (lease or sell, although you may create other services). The Customer owning the Vehicle can latter make a Proposal for that Trade and other Customer may accept that Proposal or create a new one.



If the Trade is for a Service type sell, an Invoice is created after a proposal has been accepted with the amount defined in the Service. If the Trade is created for a Service type lease (which is expected to be the most common), an Invoice is created when the first proposal is submitted. This occurs before the acceptance because the vehicle's owner is being charge for the exposure.

Finally, the customers are supposed make a Payment with a Credit Card before the due_date indicated in the Invoice. This completes the cycle around all the tables.

2. User

The User table uses the default User Django model. First you need to create a user so you can authenticate and access most of the API services.

2.1. Register User

POST https://www.domain.com/api/register/

Example request:

```
{
  "password": "Secret123*",
  "password2": "Secret123*",
  "first_name": "TestFirstName",
  "last_name": "TestLastName",
  "username": "testusername",
  "email": "testemail@gmail.com",
  "job": "TestJob",
  "notes": "TestNote",
  "dob": 1970-03-04,
}
```

Example response:

```
{
  "response": "successfully registered a new user.",
  "user_id": 1916,
  "email": "TestEmail@gmail.com",
  "username": "testusername2",
  "customer_id": 1811,
  "token": "5e01356d9fcd4de865b6be357a6c401b24a4a82d"
}
```

Please notice that a token is returned upon successful user registration. This token will be necessary for authentication of future requests.

2.2. Login User

POST https://www.domain.com/api/login/

A user that's already registered needs to provide his username and password to get a new token generated.

Example request:

```
{
  "username": "testusername",
  "password": "Secret123*",
}
```

Example response:

```
{
  "response": "successful login"
  "username": "testusername",
  "token": "5e01356d9fcd4de865b6be357a6c401b24a4a82d",
}
```

2.3. Get User Details

GET https://www.domain.com/api/user/{user_id} Authorization: Token 5e01...

To get user details, the user needs to provide the Authorization Header with his Token. Each regular user can only get his own details. Staff and superusers can get any user details

Example request:

```
{

}
```

Example response:

```
{ "id": 1916
  "username": "testusername",
  "email": "TestEmail@gmail.com",
  "first_name": "TestFirstName",
  "last_name": "TestLastName",
  "is_active": true,
  "customer_id": 1811}
```

2.4. Update User

PUT https://www.domain.com/api/user/{user_id} Authorization: Token 5e01...

To update the user details, the user needs to provide the Authorization Header with his Token. Each regular user can only update his own details. Staff and superusers can get any user details.

Example request:

```
{
  "first_name": "Garret",
  "last_name": "Osborn",
  "username": "garret499",
  "email": "garret499@gmail.com",
  "is_active": 1,
}
```

Example response:

```
{ "id": 1811,
  "username": "garret499",
  "email": "garret499@gmail.com",
  "first_name": "Garret",
  "last_name": "Osborn",
  "is_active": true,
  "customer_id": 1811}
```

2.5. Update User Password

PUT https://www.domain.com/api/password_update/{user_id} Authorization: Token 5e01...

A regular user can update his own password. Staff and superusers can update any password.

Example request:

```
{
  "password ": "NewPassword123*"
}
```

Example response:

```
{
  "response": "Password Updated",
}
```

2.6. Get User List

GET https://www.domain.com/api/users/ Authorization: Token 5e01...

Only staff and superusers can get the list of all users.

Example request:

```
{

}
```

Example response:

```
[
  {
    "id": 1806,
    "username": "april599",
    "email": "april599@gmail.com",
    "first_name": "April",
    "last_name": "Smith",
    "is_active": true,
    "customer_id": 1796
  },
  {
    "id": 1807,
    "username": "brandon686",
    "email": "brandon686@gmail.com",
    "first_name": "Brandon",
    "last_name": "Bell",
    "is_active": false,
    "customer_id": 1804
  }, …
]
```

2.7. Delete User

DELETE https://www.domain.com/api/user/{user_id} Authorization: Token 5e01...

A regular user can delete his own user. Staff and superusers can delete any user. Deleting a user will also delete the related customer if any.

Example request:

```
{

}
```

Example response:

```
  {
    "response": "Remove Completed",
  }
```

2.8. Add User

POST https://www.domain.com/api/users/ Authorization: Token 5e01...

Only the superuser is allowed to create users using this method. It’s suitable for adding users of any kind: regular, staff or superuser. Regular users are supposed to create their users by registering.

Example request:

```
{
  "password": "Secret123*",
  "is_superuser": false,
  "first_name": "Lisa",
  "last_name": "Thompson",
  "username": "lisa950",
  "email": "lisa950@gmail.com",
  "is_active": 1,
}
```

Example response:

```
{
  "id": 1917,
  "last_login": null,
  "is_superuser": false,
  "username": "lisa950",
  "first_name": "Lisa",
  "last_name": "Thompson",
  "email": "lisa950@gmail.com",
  "is_staff": false,
  "is_active": true,
  "date_joined": "2022-08-31T16:27:13.527005-04:00",
  "groups": [],
  "user_permissions": []
}
```

Please notice that when registering the request must specify the fields job, dob and notes which are not specified when adding. That’s because those three fields are actually stored in the Customer table. When you create via register, a Customer is also created. On the contrary, creating a user via add only creates a record in the User table.

2.9. User Search

GET https://www.domain.com/api/users/search/?username=lis Authorization: Token 5e01...

Only the staff and superuser are allowed to search users. Search can be made by *username*, *first_name* and/or *last_name*. Note that the response indicates customer_id = 0 if there is the user has no related customer

Example request:

```
{

}
```

Example response:

```
[{
    "id": 1917,
    "username": "lisa950",
    "email": "lisa950@gmail.com",
    "first_name": "Lisa",
    "last_name": "Thompson",
    "is_active": true,
    "customer_id": 0
}]
```

3. Customer

The customer tables contains extra data for the non-administrative users such as job, notes and dob. Every customer must have a related user but administrative users don't have a customer.

3.1. Create Customer

Customers can't be directly added using the API. To create a customer you need to register a user and the related customer will also be created.

3.2. Get Customer Details

GET https://www.domain.com/api/customer/{customer_id} Authorization: Token 5e01...

Each regular user can only get his own customer details. Staff and superusers can get any customer details.

Example request:

```
{

}
```

Example response:

```
{ "id": 1747
  "user": 1830,
  "first_name": "Adam",
  "last_name": "Dillon",
  "job": "some job",
  "notes": "Some notes",
  "dob": "1991-04-28",
  "age": 31}
```

3.3. Update Customer

PUT https://www.domain.com/api/customer/{customer_id} Authorization: Token 5e01...

Each regular user can only update his own customer details. Staff and superusers can get any user customer details.

Example request:

```
{ "first_name": "Adam",
  "last_name": "Dillon",
  "job": "some job",
  "notes": "Some notes",
  "dob": "1991-04-28"}
```

Example response:

```
{ "id": 1747
  "user": 1830,
  "first_name": "Adam",
  "last_name": "Dillon",
  "job": "some job",
  "notes": "Some notes",
  "dob": "1991-04-28",
  "age": 31}
```

3.4. Get Customer List

GET https://www.domain.com/api/customers/ Authorization: Token 5e01…

Only staff and superusers can get the list of all customers.

Example request:

```
{

}
```

Example response:

```
[{ "id": 1747
  "user": 1830,
  "first_name": "Adam",
  "last_name": "Dillon",
  "job": "some job",
  "notes": "Some notes",
  "dob": "1991-04-28",
  "age": 31}…]
```

3.5. Delete Customer

DELETE https://www.domain.com/api/customer/{customer_id} Authorization: Token 5e01…

A regular user can delete his own customer. Staff and superusers can delete any customer. Deleting a customer will also delete the related user if any.

Example request:

```
{

}
```

Example response:

```
  {
    "response": "Remove Completed",
  }
```

3.6. Customer Search

GET https://www.domain.com/api/users/search/?first_name=ada Authorization: Token 5e01…

Only the staff and superuser are allowed to search customer using the fields: *first_name*, *last_name*, *job* and *age*.

Example request:

```
{

}
```

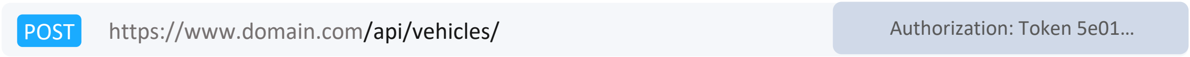
Example response:

```
{ "id": 1747
  "user": 1830,
  "first_name": "Adam",
  "last_name": "Dillon",
  "job": "some job",
  "notes": "Some notes",
  "dob": "1991-04-28",
  "age": 31}
```

4. Vehicle

A customer may create any numbers of vehicles for himself in the system. Those vehicles can be latter used in trades. Staff and superuser can add vehicles for any customer.

4.1. Add Vehicle



POST https://www.domain.com/api/vehicles/ Authorization: Token 5e01...

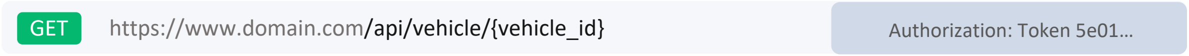
Example request:

```
{ "customer": 1,
  "make_model": "MINI Roadster",
  "make": "Toyota",
  "model": "F150 Super Cab",
  "category": "Van/MiniVan",
  "machine_make_model": "Volvo G780B",
  "machine_make": "Case IH",
  "machine_model": "EH 5000 AC - 2",
  "machine_category": "Wheel Loader",
  "year": 2005,
  "machine_year": 2001}
```

Example response:

```
{ "id": 1,
  "customer": 1,
  "make": "Toyota",
  "model": "F150 Super Cab",
  "category": "Van/MiniVan",
  "machine_make_model": "Volvo G780B",
  "machine_make": "Case IH",
  "machine_model": "EH 5000 AC - 2",
  "machine_category": "Wheel Loader",
  "year": 2005,
  "machine_year": 2001}
```

4.2. Get Vehicle Details



GET https://www.domain.com/api/vehicle/{vehicle_id} Authorization: Token 5e01...

Each regular user can only get details for his own vehicles. Staff and superusers can get details for any vehicle.

Example request:

```
{

}
```

Example response:

```
{ "id": 1,
  "customer": 1,
  "make": "Toyota",
  "model": "F150 Super Cab",
  "category": "Van/MiniVan",
  "machine_make_model": "Volvo G780B",
  "machine_make": "Case IH",
  "machine_model": "EH 5000 AC - 2",
  "machine_category": "Wheel Loader",
  "year": 2005,
  "machine_year": 2001}
```

4.3. Update Vehicle

PUT https://www.domain.com/api/vehicle/{vehicle_id} Authorization: Token 5e01...

Each regular user can only update his own vehicle details. Staff and superusers can get any user vehicle details.

Example request:

```
{ "customer": 1,
  "make_model": "MINI Roadster",
  "make": "Toyota",
  "model": "F150 Super Cab",
  "category": "Van/MiniVan",
  "machine_make_model": "Volvo G780B",
  "machine_make": "Case IH",
  "machine_model": "EH 5000 AC - 2",
  "machine_category": "Wheel Loader",
  "year": 2005,
  "machine_year": 2001}
```

Example response:

```
{ "id": 1,
  "customer": 1,
  "make": "Toyota",
  "model": "F150 Super Cab",
  "category": "Van/MiniVan",
  "machine_make_model": "Volvo G780B",
  "machine_make": "Case IH",
  "machine_model": "EH 5000 AC - 2",
  "machine_category": "Wheel Loader",
  "year": 2005,
  "machine_year": 2001}
```

4.4. Get Vehicle List

GET https://www.domain.com/api/vehicles/ Authorization: Token 5e01...

A regular user can get the list of his own vehicles. Staff and superusers can get the list of all available vehicles.

Example request:

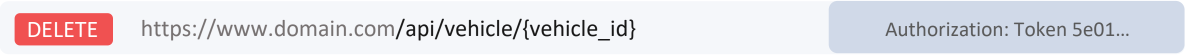
```
{

}
```

Example response:

```
[{ "id": 1,
  "customer": 1,
  "make": "Toyota",
  "model": "F150 Super Cab",
  "category": "Van/MiniVan",
  "machine_make_model": "Volvo G780B",
  "machine_make": "Case IH",
  "machine_model": "EH 5000 AC - 2",
  "machine_category": "Wheel Loader",
  "year": 2005,
  "machine_year": 2001}…]
```

4.5. Delete Vehicle



DELETE https://www.domain.com/api/vehicle/{vehicle_id}

Authorization: Token 5e01...

A regular user can delete his own vehicle. Staff and superusers can delete any vehicle. Deleting a vehicle doesn't affect the customer or user but deleting a customer or user will also delete his vehicles.

Example request:

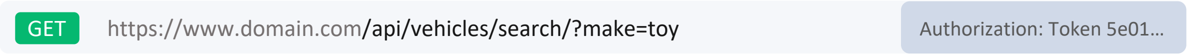
```
{

}
```

Example response:

```
{
  "response": "Remove Completed",
}
```

4.6. Vehicle Search



GET https://www.domain.com/api/vehicles/search/?make=toy

Authorization: Token 5e01...

Only the staff and superuser are allowed to search vehicles using the fields: *make_model*, *make*, *model*, *category*, *machine_make_model*, *machine_make*, *machine_model*, *machine_category*, *year and machine_year*.

Example request:

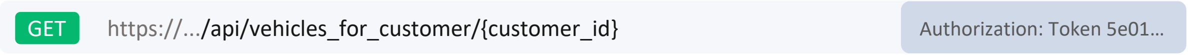
```
{

}
```

Example response:

```
[{ "id": 1,
  "customer": 1,
  "make": "Toyota",
  "model": "F150 Super Cab",
  "category": "Van/MiniVan",
  "machine_make_model": "Volvo G780B",
  "machine_make": "Case IH",
  "machine_model": "EH 5000 AC - 2",
  "machine_category": "Wheel Loader",
  "year": 2005,
  "machine_year": 2001}…]
```

4.7. Vehicles for a Customer



GET https://.../api/vehicles_for_customer/{customer_id}

Authorization: Token 5e01...

A user can get his own list of vehicles. Staff and superusers can get the list of vehicles for any customer. Examples request and response are same as previous.

5. Lead

Leads are potential customers that haven't registered yet into the system. Leads can be managed only by staff and superusers.

5.1. Add Lead

POST https://www.domain.com/api/leads/ Authorization: Token 5e01...

Example request:

```
{ "first_name": "Taylor",
  "last_name": "Glenn",
  "source": "https://www.facebook.com",
  "notes": "Some notes",
  "dob": 1971-05-15}
```

Example response:

```
{ "id": 1,
  "first_name": "Taylor",
  "last_name": "Glenn",
  "source": "https://www.facebook.com",
  "notes": "Some notes",
  "dob": 1971-05-15}
```

5.2. Get Lead Details

GET https://www.domain.com/api/lead/{lead_id} Authorization: Token 5e01...

Example request:

```
{


}
```

Example response:

```
{ "id": 1,
  "first_name": "Taylor",
  "last_name": "Glenn",
  "source": "https://www.facebook.com",
  "notes": "Some notes",
  "dob": 1971-05-15}
```

5.3. Update Lead

PUT https://www.domain.com/api/lead/{lead_id} Authorization: Token 5e01...

Example request:

```
{ "first_name": "Taylor",
  "last_name": "Glenn",
  "source": "https://www.facebook.com",
  "notes": "Some notes",
  "dob": 1971-05-15}
```

Example response:

```
{ "id": 1,
  "first_name": "Taylor",
  "last_name": "Glenn",
  "source": "https://www.facebook.com",
  "notes": "Some notes",
  "dob": 1971-05-15}
```

5.4. Get Lead List

GET https://www.domain.com/api/leads/ Authorization: Token 5e01…

Example request:

```
{

}
```

Example response:

```
[{ "id": 1,
  "first_name": "Taylor",
  "last_name": "Glenn",
  "source": "https://www.facebook.com",
  "notes": "Some notes",
  "dob": 1971-05-15}…]
```

5.5. Delete Lead

DELETE https://www.domain.com/api/lead/{lead_id} Authorization: Token 5e01…

Example request:

```
{

}
```

Example response:

```
  {
    "response": "Remove Completed",
  }
```

5.6. Lead Search

GET https://www.domain.com/api/leads/search/?first_name=tay Authorization: Token 5e01…

Leads can be searched by: *first_name*, *last_name*, *source* and *notes*.

Example request:

```
{

}
```

Example response:

```
{ "id": 1,
  "first_name": "Taylor",
  "last_name": "Glenn",
  "source": "https://www.facebook.com",
  "notes": "Some notes",
  "dob": 1971-05-15}
```

6. Contact

A customer may create any numbers of contacts for himself in the system. Every contact record may contain a phone or an email. Staff and superusers can create contacts for any customer or lead. Contact type = 1 is for emails and type = 2 is for phones.

6.1. Add Contact

POST https://www.domain.com/api/contacts/ Authorization: Token 5e01…

Any user can add a contact for himself. Staff and superusers can add contacts for any customer or for any lead.

Example request:

```
{ "customer": 1,
  "lead ": "",
  "phone": "",
  "email": "james744@gmail.com",
  "note": "Some note",
  "type": 1,
}
```

Example response:

```
{ "id": 1,
  "customer": 1,
  "lead ": "",
  "phone": "",
  "email": "james744@gmail.com",
  "note": "Some note",
  "type": 1,
}
```

6.2. Get Contact Details

GET https://www.domain.com/api/contact/{contact_id} Authorization: Token 5e01…

Each regular user can only get details for his own contacts. Staff and superusers can get details for any contact.

Example request:

```
{


}
```

Example response:

```
{ "id": 1,
  "customer": 1,
  "lead ": "",
  "phone": "",
  "email": "james744@gmail.com",
  "note": "Some note",
  "type": 1,
}
```

6.3. Update Contact

PUT https://www.domain.com/api/contact/{vehicle_id} Authorization: Token 5e01…

Each regular user can only update his own contact details. Staff and superusers can get any user customer details.

Example request:

Example request:

```
{ "customer": 1,
  "lead ": "",
  "phone": "",
  "email": "james744@gmail.com",
  "note": "Some note",
  "type": 1}
```

Example response:

```
{ "id": 1,
  "customer": 1,
  "lead ": "",
  "phone": "",
  "email": "james744@gmail.com",
  "note": "Some note",
  "type": 1}
```

6.4. Get Contact List

GET https://www.domain.com/api/contacts/ Authorization: Token 5e01…

A regular user can get the list of his own contacts. Staff and superusers can get the list of all available contacts.

Example request:

```
{

}
```

Example response:

```
[{"id": 1,
  "customer": 1,
  "lead ": "",
  "phone": "",
  "email": "james744@gmail.com",
  "note": "Some note",
  "type": 1}…]
```

6.5. Delete Contact

DELETE https://www.domain.com/api/contact/{contact_id} Authorization: Token 5e01…

A regular user can delete his own contact. Staff and superusers can delete any contact for a customer or lead. When deleting a customer or a lead, its related contacts also get deleted.

Example request:

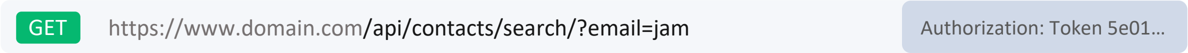
```
{

}
```

Example response:

```
  {
    "response": "Remove Completed",
  }
```

6.6. Contact Search



GET https://www.domain.com/api/contacts/search/?email=jam Authorization: Token 5e01…

Only the staff and superuser are allowed to search contacts using the fields: *phone*, *email*, *note* and *type*.

Example request:

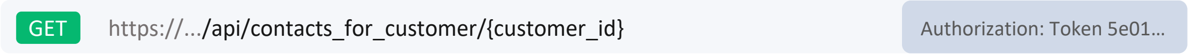
```
{

}
```

Example response:

```
[{"id": 1,
  "customer": 1,
  "lead ": "",
  "phone": "",
  "email": "james744@gmail.com",
  "note": "Some note",
  "type": 1}…]
```

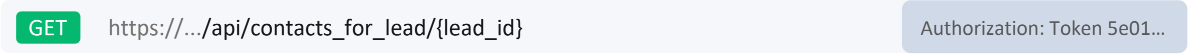
6.7. Contacts for a Customer



GET https://.../api/contacts_for_customer/{customer_id} Authorization: Token 5e01…

Returns all available contacts for a given customer id. Return is similar to previous. Every customer is able to obtain the contacts for his customer. Staff and superusers can obtain contacts for any customer.

6.8. Contacts for a Lead



GET https://.../api/contacts_for_lead/{lead_id} Authorization: Token 5e01…

Returns all available contacts for a given lead id. Only staff and superusers can obtain contacts for any lead.

Example request:

```
{

}
```

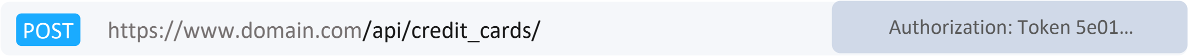
Example response:

```
[{"id": 1,
  "customer": "",
  "lead ": 1,
  "phone": "",
  "email": "james744@gmail.com",
  "note": "Some note",
  "type": 1}…]
```

7. Credit Card

A customer may create any numbers of credit cards for himself. Those credit cards can be latter used for payments. Staff and superuser can handle credits cards for any customer.

7.1. Add Credit Card



POST https://www.domain.com/api/credit_cards/

Authorization: Token 5e01...

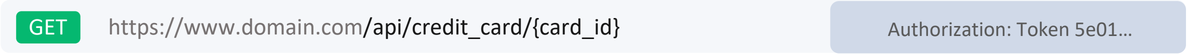
Example request:

```
{ "customer": 1,
  "name_in_card ": "josephroberts",
  "provider": "Visa",
  "expire month": 8,
  "expire_year": 2027,
  "security_code": "892",
  "card_number": "352929361782272"
}
```

Example response:

```
{ "id": 1,
  "customer": 1,
  "name_in_card ": "josephroberts",
  "provider": "Visa",
  "expire month": 8,
  "expire_year": 2027,
  "security_code": "892",
  "card_number": "352929361782272"
}
```

7.2. Get Card Details



GET https://www.domain.com/api/credit_card/{card_id}

Authorization: Token 5e01...

Each regular user can only get details for his own cards. Staff and superusers can get details for any card.

Example request:

```
{

}
```

Example response:

```
{ "id": 1,
  "customer": 1,
  "name_in_card ": "josephroberts",
  "provider": "Visa",
  "expire month": 8,
  "expire_year": 2027,
  "security_code": "892",
  "card_number": "352929361782272"
}
```

7.3. Update Card

PUT https://www.domain.com/api/credit_card/{card_id} Authorization: Token 5e01…

Each regular user can only update his own credit cards details. Staff and superusers can get any user card details.

Example request:

```
{ "customer": 1,
  "make_model": "MINI Roadster",
  "make": "Toyota",
  "model": "F150 Super Cab",
  "category": "Van/MiniVan",
  "machine_make_model": "Volvo G780B",
  "machine_make": "Case IH",
  "machine_model": "EH 5000 AC - 2",
  "machine_category": "Wheel Loader",
  "year": 2005,
  "machine_year": 2001}
```

Example response:

```
{ "id": 1,
  "customer": 1,
  "name_in_card ": "josephroberts",
  "provider": "Visa",
  "expire month": 8,
  "expire_year": 2027,
  "security_code": "892",
  "card_number": "352929361782272"
}
```

7.4. Get Card List

GET https://www.domain.com/api/credit_cards/ Authorization: Token 5e01…

A regular user can get the list of his own credit cards. Staff and superusers can get the list of all available cards.

Example request:

```
{ "customer": 1,
  "name_in_card ": "josephroberts",
  "provider": "Visa",
  "expire month": 8,
  "expire_year": 2027,
  "security_code": "892",
  "card_number": "352929361782272"
}
```

Example response:

```
[{ "id": 1,
  "customer": 1,
  "name_in_card ": "josephroberts",
  "provider": "Visa",
  "expire month": 8,
  "expire_year": 2027,
  "security_code": "892",
  "card_number": "352929361782272"
}…]
```

7.5. Delete Credit Card

DELETE https://www.domain.com/api/credit_card/{card_id} Authorization: Token 5e01...

A regular user can delete his own credit card. Staff and superusers can delete any card. When deleting a user or customer, their related credit cards will also get deleted.

Example request:

```
{

}
```

Example response:

```
{
  "response": "Remove Completed",
}
```

7.6. Credit Card Search

GET https://.../api/credit_cards/search/?name_in_car=jos Authorization: Token 5e01...

Only the staff and superuser are allowed to search credit cards using the fields: *name_in_card*, *expire_month*, *expire_year* and *last_four_digits*.

Example request:

```
{

}
```

Example response:

```
[{ "id": 1,
  "customer": 1,
  "name_in_card ": "josephroberts",
  "provider": "Visa",
  "expire month": 8,
  "expire_year": 2027,
  "security_code": "892",
  "card_number": "352929361782272"
}…]
```

7.7. Credit Cards for a Customer

GET https://.../api/credit_cards_for_customer/{customer_id} Authorization: Token 5e01...

A user can get his own list of credit cards. Staff and superusers can get the list of cards for any customer. Examples request and response are same as previous.

8. Service

The system has two basic services which are lease and sale. Extra services can be added in this table. Services can only be created by staff and superusers. Regular users can only get or list services.

Service allow specifying if the invoice is to be created on the first trade (like in lease using *when_to_pay* = 1) or when any proposal is approved (like in sale using *when_to_pay* = 2). The variable *service_type* can be set to 1 for type Lease or 2 for type Sale.

8.1. Add Service

POST https://www.domain.com/api/services/ Authorization: Token 5e01...

Only staff and superusers can add services.

Example request:

```
{ "name": "Lease",
  "cost ": 923.54,
  "when_to_pay": 2,
  "service_type": 1,
  "description": "some desc"}
```

Example response:

```
{ "id": 1,
  "name": "Lease",
  "cost ": 923.54,
  "when_to_pay": 2,
  "service_type": 1,
  "description": "some desc"}
```

8.2. Get Service Details

GET https://www.domain.com/api/service/{service_id} Authorization: Token 5e01...

Any registered user can get a service details.

Example request:

```
{

}
```

Example response:

```
{ "id": 1,
  "name": "Lease",
  "cost ": 923.54,
  "when_to_pay": 2,
  "service_type": 1,
  "description": "some desc"}
```

8.3. Update Service

PUT https://www.domain.com/api/service/{service_id} Authorization: Token 5e01...

Only staff and superusers can edit services.

Example request:

```
{ "name": "Lease",
  "cost ": 923.54,
  "when_to_pay": 2,
  "service_type": 1,
  "description": "some desc"}
```

Example response:

```
{ "id": 1,
  "name": "Lease",
  "cost ": 923.54,
  "when_to_pay": 2,
  "service_type": 1,
  "description": "some desc"}
```

8.4. Get Service List

GET https://www.domain.com/api/services/ Authorization: Token 5e01...

Any authenticated user can get the list of services.

Example request:

```
{

}
```

Example response:

```
[{ "id": 1,
  "name": "Lease",
  "cost ": 923.54,
  "when_to_pay": 2,
  "service_type": 1,
  "description": "some desc"}…]
```

8.5. Delete Service

DELETE https://www.domain.com/api/service/{service_id} Authorization: Token 5e01...

Only staff and superadmins can delete services.

Example request:

```
{

}
```

Example response:

```
{
  "response": "Remove Completed",
}
```

8.6. Service Search

GET https://.../api/services/search/?name=lea Authorization: Token 5e01...

Is possible to search services using the fields: *name*, *cost*, *when_to_pay*, and *service_type*.

Example request:

```
{


}
```

Example response:

```
[{ "id": 1,
  "name": "Lease",
  "cost ": 923.54,
  "when_to_pay": 2,
  "service_type": 1,
  "description": "some desc"}…]
```

9. Trade

A customer may create trades for any of his vehicles but he can't create two trades with the same service/vehicle combination. Staff and superusers also have this limitation, however, they can create trades for any customer. The status field in trades contains 1 for open, 2 for accepted and 3 for canceled.

9.1. Add Trade

POST https://www.domain.com/api/trades/ Authorization: Token 5e01...

A regular user can only add trades for his own vehicles. Staff and Superuser can add trades for any vehicle. All trades are created with status = 1 which means the trade is open.

Example request:

```
{ "service": 1,
  "vehicle": 1,
  "note": "Some note"
}
```

Example response:

```
{ "id": 1,
  "service": 1,
  "vehicle": 1,
  "note": "Some note",
  "status": 1}
```

9.2. Get Trade Details

GET https://www.domain.com/api/trade/{trade_id} Authorization: Token 5e01...

Each regular user can only get details for his own trades. Staff and superusers can get details for any trade.

Example request:

```
{

}
```

Example response:

```
{ "id": 1,
  "service": 1,
  "vehicle": 1,
  "note": "Some note",
  "status": 1}
```

9.3. Update Trade

PUT https://www.domain.com/api/trade/{trade_id} Authorization: Token 5e01…

Each regular user can only update his own trade details. Staff and superusers can get any trade details.

Example request:

```
{ "id": 1,
  "service": 1,
  "vehicle": 1,
  "note": "Some note",
  "status": 1}
```

Example response:

```
{ "id": 1,
  "service": 1,
  "vehicle": 1,
  "note": "Some note",
  "status": 1}
```

9.4. Get Trade List

GET https://www.domain.com/api/trades/ Authorization: Token 5e01…

A regular user can get the list of his own trades. Staff and superusers can get the list of all available vehicles. Buyer (other customers) can get the list of all proposals (which include trades) but can't directly visualize trades.

Example request:

```
{

}
```

Example response:

```
[{ "id": 1,
  "service": 1,
  "vehicle": 1,
  "note": "Some note",
  "status": 1}…]
```

9.5. Delete Trade

DELETE https://www.domain.com/api/trade/{trade _id} Authorization: Token 5e01…

A regular user can delete his own trade. Staff and superusers can delete any trade. Deleting a trade doesn't affect the vehicle or service but will delete all proposals.

Example request:

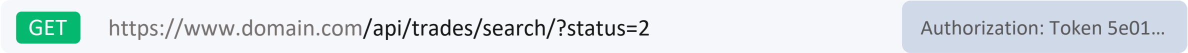
```
{

}
```

Example response:

```
{
  "response": "Remove Completed",
}
```

9.6. Trade Search



GET https://www.domain.com/api/trades/search/?status=2

Authorization: Token 5e01...

Only the staff and superuser are allowed to search trades using the fields: *note* and *status*.

Example request:

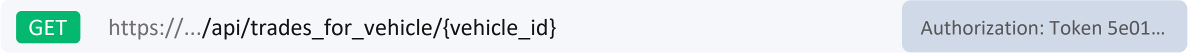
```
{


}
```

Example response:

```
[{ "id": 1,
  "service": 1,
  "vehicle": 1,
  "note": "Some note",
  "status": 2}…]
```

9.7. Trades for a Vehicle



GET https://.../api/trades_for_vehicle/{vehicle_id}

Authorization: Token 5e01...

The user who owns the vehicle can get the list of all trades for a given vehicle. Staff and superusers can get the related trades for any vehicle.