

Integracja Dealera poprzez API

Dealer ma do dyspozycji REST API, pozwalające zautomatyzować niektóre czynności takie jak dodawanie samochodów.

Środowisko produkcyjne

Adres url serwera izzylease: `https://api.izzylease.pl`

Autoryzacja

Do autoryzacji *Dealer* potrzebuje `client_id` oraz `client_secret`, które może uzyskać od administratora systemu

IzzyLease. Powinny one być przechowywane i przesyłane w sposób bezpieczny aby uniemożliwić podmiotom trzecim wykonywanie operacji w imieniu *Dealera*.

Protokół używany do autoryzacji zapytań do serwera to *OAuth2* z *Client Credentials* flow. *OAuth2* używany jest do

pozyskania tokena (`access token`), który później używany jest do wykonywania każdego requestu.

Pobieranie tokena `access token`

Poniżej opisany request pozwana na pobranie `access tokena` wraz z informacją kiedy wygasa.

Gdy token wygaśnie należy

ponowić zapytanie o nowy `access token`.

Aby pobrać token wykonaj poniższe zapytanie. Pamiętaj, aby zastąpić `<client_id>` i `<client_secret>`.

```
curl --request POST \  
  --url https://api.izzylease.pl/oauth/token \  
  --header 'content-type: application/x-www-form-urlencoded' \  
  --data client_id=<client_id> \  
  --data client_secret=<client_secret> \  
  --data grant_type=client_credentials
```

Odpowiedź:

```
{
  "access_token": "some_access_token",
  "token_type": "bearer",
  "expires_in": 43199,
  "scope": "dealer_integration",
  "jti": "9dc2d3f4-1a91-4eb8-93b6-51bca4ac6104"
}
```

Użycie tokena

Token powinien zostać użyty przy każdym zapytaniu w nagłówku

```
Authorization: <token_type> <access_token>
```

gdzie `<token_type>` oraz `<access_token>` należy zastąpić wartościami zwróconymi z odpowiedzi zapytania o token.

Zarządzanie samochodami

API pozwala na dodawanie samochodu, jego aktualizację, oraz dodawanie i usuwanie zdjęć samochodu.

Struktura danych samochodu

Pola oznaczone symbolem `*` są wymagane. Pozostałe mogą przyjmować wartość `null`.

Struktura używana jest do dodawania i

aktualizacji samochodu. Powinna być serializowana formatem `json`. Wszystkie daty powinny być serializowane w formacie `ISO`.

```
{
  configurationNumber  string //numer konfiguracji
  vin                  string //numer VIN
  category*            string Enum: [ PASSENGER, DELIVERY ]
  make*                string //marka samochodu
  model*               string //model samochodu
  manufactureYear      integer($int32) //rok produkcji
}
```

```

mileage                integer($int32) //przebieg (w kilometrach)
engineCode*            string //oznaczenie silnika
cubicCapacity*         number //pojemność silnika
acceleration*          number //czas przyspieszenia do 100km/h
fuelType*              string Enum: [ PETROL, DIESEL, ELECTRIC, HYBRID, LPG,
HYDROGEN ]
power                  integer($int32) //moc w KM
transmissionType*      string Enum: [ AUTOMATIC, MANUAL ]
driveWheels*           string Enum: [ FOUR, FRONT, REAR ]
type*                  string Enum: [ SALOON, CONVERTIBLE, SUV, VAN, ESTATE,
HATCHBACK ]
carClass               string Enum: [ ADRENALINE, SWEET, FAMILY, BUSINESS ]
doors                  integer($int32) //liczba drzwi
color*                 string //kolor
availableFrom          string (iso date format) //data od kiedy samochód jest
dostępny
firstRegistrationDate string (iso date format) //data pierwszej rejestracji
description             string //opis
pricing*               {
    listPrice*          string //cena katalogowa
    salesPrice*         string //cena sprzedaży
}
locationId             string($uuid) //identyfikator salonu (może być null)
}

```

Dodawanie samochodu

Przykład CURL

```

curl --request POST \
--url https://api.izzylease.pl/external/cars \
--header 'authorization: <token_type> <access_token>' \
--header 'content-type: application/json' \
--data '{
"configurationNumber": null,
"vin": null,
"category": "PASSENGER",
"make": "Mercedes",
"model": "C class",
"manufactureYear": 2010,
"mileage": 1000,
"engineCode": "???",
"cubicCapacity": 2999,

```

```
"acceleration": 6.3,  
"fuelType": "PETROL",  
"power": 231,  
"transmissionType": "AUTOMATIC",  
"driveWheels": "FOUR",  
"type": "SALOON",  
"carClass": "ADRENALINE",  
"doors": 4,  
"color": "white",  
"availableFrom": null,  
"firstRegistrationDate": null,  
"description": "sample description",  
"pricing": {  
  "listPrice": "100000",  
  "salesPrice": "120000"  
},  
"locationId": null  
}'
```

Odpowiedź zawiera identyfikator stworzonego samochodu:

```
{  
  "id": "e1f15f90-786b-447f-9eab-f804b0633c27"  
}
```

Aktualizacja samochodu

```
curl --request PUT \  
  --url https://api.izzylease.pl/external/cars/<car_id> \  
  --header 'authorization: <token_type> <access_token>' \  
  --header 'content-type: application/json' \  
  --data '{  
    "configurationNumber": null,  
    "vin": null,  
    "category": "PASSENGER",  
    "make": "Mercedes",  
    "model": "C class",  
    "manufactureYear": 2010,  
    "mileage": 1000,  
    "engineCode": "???",  
    "cubicCapacity": 2999,  
    "acceleration": 6.3,  
    "fuelType": "PETROL",
```

```
"power": 231,  
"transmissionType": "AUTOMATIC",  
"driveWheels": "FOUR",  
"type": "SALOON",  
"carClass": "ADRENALINE",  
"doors": 4,  
"color": "white",  
"availableFrom": null,  
"firstRegistrationDate": null,  
"description": "sample description",  
"pricing": {  
  "listPrice": "100000",  
  "salesPrice": "120000"  
},  
"locationId": null  
}'
```

Dodawanie zdjęcia samochodu

Multipart request z obrazkiem w parcie nazwanym `image`. Konieczne jest wysłanie `content-type` obrazka.

```
curl --request POST \  
  --url https://api.izzylease.pl/external/cars/<car_id>/images \  
  --header 'authorization: <token_type> <access_token>' \  
  --header 'content-type: multipart/form-data' \  
  --form 'image=@<your_file_path>;type=image/png'
```

Response zawiera identyfikator utworzonego obrazka:

```
{  
  "imageId": "61d6c108-8f44-4844-9daf-afd0f39b8f55"  
}
```

Usuwanie zdjęcia samochodu

```
curl --request DELETE \  
  --url https://api.izzylease.pl/external/cars/<car_id>/images/<image_id> \  
  --header 'authorization: <token_type> <access_token>'
```

Usuwanie samochodu

System pozwala na usunięcie samochodu z dwóch powodów:

- `DELETED` - usunięty przez dealera (domyślny)
- `SOLD` - sprzedany przez dealera

Powód powinien być przesłany w *opcjonalnym* body requestu. W przypadku nie podania body zostanie ustawiony powód domyślny.

Przykładowe body:

```
{  
  "reason": "DELETED" //DELETED, SOLD  
}
```

Przykład CURL

```
curl --request DELETE \  
  --url https://api.izzylease.pl/external/cars/<car_id> \  
  --header 'authorization: <token_type> <access_token>' \  
  --header 'content-type: application/json' \  
  --data '{ "reason": "DELETED" }'
```

Generowanie raportów IzzyCheck

API pozwala na inicjalizację generowania raportu, pobranie szczegółów procesu generowania oraz pobranie raportu jako pdf, html lub json.

Generowanie raportów odbywa się asynchronicznie, co oznacza konieczność zainicjowanie generowania raportu a następnie oczekiwanie na jego wygenerowanie. Pobieranie szczegółów raportu pozwala sprawdzić status generowania. Status zakończony pozwala na pobranie raportu.

Parametr `vinIdentification` pozwala na włączenie / wyłączenie identyfikacji wyposażenia pojazdu po VINie (zapytanie płatne). Parametry `licensePlateNumber` oraz `registrationDate` są opcjonalne. Gdy nie zostaną podane raport nie będzie zawierał identyfikacji z CEPIK.

Aby wygenerować uproszczoną wycenę należy podać obiekt `estimation` z parametrami

wejściowymi.

Właściwość `mileageInKm` jest opcjonalna - jeśli nie zostanie podana to podejmiemy próbę pobrania danych z CEPIK.

Zainicjowanie generowania raportu

Przykład CURL

```
curl --request POST \
  --url https://api.izzylease.pl/external/car/identification/process \
  --header 'authorization: <token_type> <access_token>' \
  --header 'content-type: application/json' \
  --data '{
    "vin": <your_vin>,
    "licensePlateNumber": <your_license_plate_number>,
    "registrationDate": <your_registration_date_in_iso>
    "vinIdentification": true,
    "estimation": {
      "mileageInKm": 12000,
      "productionYear": 2010,
    }
  }'
```

Odpowiedź:

Zawiera identyfikator raportu potrzebny do kolejnych operacji.

```
{
  "id": "string" // uuid
}
```

Pobranie szczegółów generacji raportu

Inicjalizacja zwraca `id` (identyfikator raportu). Za jego pomocą można pobrać szczegóły generowanego raportu.

Przykład CURL

```
curl --request GET \
```

```
--url https://api.izzylease.pl/external/car/identification/process/{id} \  
--header 'authorization: <token_type> <access_token>' \  
--header 'content-type: application/json'
```

Odpowiedź:

```
{  
  "id": "string", // uuid  
  "state": "Processing", // enum = "Processing", "Failed", "Completed"  
  "startingTime": "2021-12-08T19:01:09.428Z", // ISO format  
  "vin": "string",  
  "licensePlateNumber": "string",  
  "registrationDate": "string"  
}
```

Pobranie raportu

Raport w stanie **Completed** można pobrać.

Endpoint do pobrania to: **GET**

<https://api.izzylease.pl/external/car/identification/process/{id}/reports/initial>.

Aby wybrać format zwracanych danych należy podać oczekiwany format w nagłówku **Accept**.

Dostępne formaty ty:

- **application/json**
- **application/pdf**
- **text/html**

Pobranie raportu w formacie pdf

Przykład CURL

```
curl --request GET \  
  --url  
https://api.izzylease.pl/external/car/identification/process/{id}/reports/initia  
l \  
  --header 'authorization: <token_type> <access_token>' \  
  --header 'content-type: application/json' \  
  --header 'accept: application/pdf'
```


Pobranie raportu w formacie html

Przykład CURL

```
curl --request GET \
  --url
https://api.izzylease.pl/external/car/identification/process/{id}/reports/initia
l \
  --header 'authorization: <token_type> <access_token>' \
  --header 'content-type: application/json' \
  --header 'accept: text/html'
```

Pobranie raportu w formacie json

Przykład CURL

```
curl --request GET \
  --url
https://api.izzylease.pl/external/car/identification/process/{id}/reports/initia
l \
  --header 'authorization: <token_type> <access_token>' \
  --header 'content-type: application/json' \
  --header 'accept: application/json'
```

Przykładowa odpowiedź

```
{
  "registrationInformation": {
    "vin": "XXX",
    "licensePlateNumber": "XXX",
    "firstRegistrationDate": "2012-03-07",
    "registrationAbroad": true,
    "registrationVoivodship": "mazowieckie",
    "registrationStatus": "Zarejestrowany",
    "inspectionStatus": "aktualne",
    "ocInsuranceStatus": "aktualna",
    "ocInsuranceExpiryDate": "2022-10-16",
    "ownerCount": 1,
    "ownerType": "osoba fizyczna",
```

```
"mileage": {
  "value": 167385,
  "unit": "Kilometers",
  "inKilometers": 167385
},
"ownerCountAllTime": 2
},
"basicInformation": {
  "make": "MAZDA",
  "model": "2",
  "carType": "samochód osobowy",
  "carSubType": "hatchback",
  "manufactureYear": 2011,
  "modelYear": 2013,
  "madeDate": null,
  "lienRegistry": false
},
"technicalSpecification": {
  "cubicCapacity": 1349,
  "power": 55,
  "grossVehicleWeight": 997,
  "allowedGrossVehicleWeight": 1485,
  "axesNumber": 2,
  "seatsCount": 5,
  "seatingSeatsCount": 5,
  "engineDescription": null,
  "fuelType": "BENZYNA",
  "color": null,
  "transmission": "Ręczny/Standardowy",
  "drive": "Napęd na przednie koła",
  "doorsCount": 4
},
"carValuation": {
  "estimatedValue": null,
  "estimatedValueError": null
},
"carFeatures": {
  "basic": [],
  "additional": []
},
"vehicleHistory": [
  {
    "eventType": "PeriodicVehicleInspection",
    "date": "2019-07-22",
    "expires": "2020-07-22",
```

```

    "skpNumber": "XXX",
    "odometerReading": {
      "value": 146961,
      "unit": "Kilometers",
      "inKilometers": 146961
    },
    "result": "pozytywny"
  },
  {
    "eventType": "ChangedOwner",
    "date": "2019-08-18",
    "type": "osoba fizyczna",
    "registrationVoivodship": "mazowieckie"
  },
  {
    "eventType": "PeriodicVehicleInspection",
    "date": "2020-06-10",
    "expires": null,
    "skpNumber": "XXX",
    "odometerReading": {
      "value": 152265,
      "unit": "Kilometers",
      "inKilometers": 152265
    },
    "result": "negatywny"
  },
  {
    "eventType": "PeriodicVehicleInspection",
    "date": "2021-06-10",
    "expires": "2022-06-10",
    "skpNumber": "XXX",
    "odometerReading": {
      "value": 167385,
      "unit": "Kilometers",
      "inKilometers": 167385
    },
    "result": "pozytywny"
  }
]
}

```

Jedyną wartością słownikową jest `eventType` w `vehicleHistory`. Każdy z możliwych typów ma inną strukturę danych:

- `VehicleTheft`

- date
- VehicleRecovery
 - date
- ChangedOwner
 - date
 - type
 - registrationVoivodship
- VehicleSale
 - date
- EndOfTemporaryRegistration
 - date
- ReRegistration
 - date
- PeriodicVehicleInspection
 - date
 - expires
 - skipNumber
 - odometerReading
 - result
- AdditionalVehicleInspection
 - date
 - type
 - skipNumber
 - odometerReading
 - result
- OdometerReadingRoadsideInspection
 - date
 - type
 - odometerReading