



CourierIO

ithelp@autofleetcontrol.de

Version 1.0.19, 08.07.2025



Table of Contents

1. Access	3
2. Version History	4
3. Endpoints	5
3.1. ClaimService	5
3.2. CompanyService	7
3.3. VehicleService	14
4. Models	25
4.1. <i>Claim</i>	25
4.2. <i>ClaimCompany</i>	27
4.3. <i>ClaimCosts</i>	27
4.4. <i>ClaimDriver</i>	28
4.5. <i>ClaimInsurance</i>	28
4.6. <i>ClaimStatus</i>	28
4.7. <i>ClaimType</i>	28
4.8. <i>ClaimVehicle</i>	29
4.9. <i>Company</i>	30
4.10. <i>CompanyType</i>	30
4.11. <i>DamagedPart</i>	31
4.12. <i>Driver</i>	31
4.13. <i>ErrorEntry</i>	31
4.14. <i>ErrorResponse</i>	32
4.15. <i>Insurance</i>	32
4.16. <i>Leasing</i>	32
4.17. <i>Salutation</i>	33
4.18. <i>SyncStatus</i>	33
4.19. <i>Title</i>	33
4.20. <i>Vehicle</i>	33

Abstract

*CourierIO is a robust API solution designed to streamline data exchange between Customer and AFC systems using RESTful communication. It supports the exchange of three main objects: **Companies**, **Vehicles**, and **Claims**. Depending on the object type, it offers typical CRUD operations: listing, creating, updating, and deleting.*

The request processing undergoes the following stages:

1. Authentication

The first step to connect to the API is performing authentication. CourierIO relies on the standard OpenID protocol. The customer will receive the Client ID and Client Secret from AFC. Using this information, the customer must obtain an access token using the Client Credentials grant type by following these steps:

a. Token Request: The customer sends a POST request to the OP's token endpoint with the following parameters:

- `grant_type=client_credentials`
- `client_id=<client_id>`
- `client_secret=<client_secret>`

b. Token Response: The OP responds with a JSON payload containing the access token:

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

Using the obtained token the customer can call the API endpoints. If he is not properly authorized, the API returns an HTTP 401 code.

2. Data Validation

Next, the system proceeds with validating the submitted data. If there are any errors, it returns an HTTP 400 code with an `ErrorResponse` payload containing a list of `ErrorEntry` items detailing the validation errors.

3. Authorization

When the submitted object is valid, the system checks for the required data and performs authorization to ensure it is available and the customer has the necessary access rights. This step can result in an HTTP 403 code if the user lacks permissions or an HTTP 404 code if the resource was not found.

4. Operation Execution

Finally, the service attempts to perform the requested operation. If everything goes well and the operation is successful, the customer receives a response with an HTTP 200 code and an optional payload.



Chapter 1. Access

The API is available in two environments: sandbox and production.

The sandbox environment is a publicly accessible space designed for experimentation and testing. It allows customers to explore and interact with the API without affecting live operations.

The base URL to access the sandbox environment is:

<https://sandbox.autofleetcontrol.net/courierio>

The base URL to access the production environment is:

<https://apps.autofleetcontrol.net/courierio>

To authenticate and obtain an access token, please use the URLs below in conjunction with the Client ID and Client Secret provided by AFC.

- **OAuth** for development environment:

- AuthorizationUrl: <https://auth-dev.autofleetcontrol.net/realms/AFC/protocol/openid-connect/auth>
- TokenUrl: <https://auth-dev.autofleetcontrol.net/realms/AFC/protocol/openid-connect/token>

- **OAuth** for production environment:

- AuthorizationUrl: <https://auth.autofleetcontrol.net/realms/AFC/protocol/openid-connect/auth>
- TokenUrl: <https://auth.autofleetcontrol.net/realms/AFC/protocol/openid-connect/token>

Chapter 2. Version History

Version Number	Date	Description of Changes
v1.0.19	08.07.2025	- Add missing error codes to endpoint in swagger.
v1.0.18	24.06.2025	- Add support for fine-grained access control to the API endpoints.
v1.0.17	23.06.2025	- Add claim cost fields to the Claim object
v1.0.16	18.06.2025	- Update landing page URL.
v1.0.15	13.06.2025	- Add externalId to the Company object. Update landing page URL.
v1.0.14	21, 2025	- Handle empty salutation and title fields in the Driver object.
v1.0.13	Jun 10, 2025	- Add vehicleType field to the Vehicle object.
v1.0.12	May 27, 2025	- Add motorType field to the Vehicle object.
v1.0.11	May 21, 2025	- Handle empty salutation and title fields in the Driver object.
v1.0.10	May 9, 2025	- Update claim type and claim art enumerations
v1.0.9	Apr 30, 2025	- Allow operation in a stand-alone mode without connection to backend
v1.0.6	Mar 27, 2025	- Refactor to use GRUM IDs
v1.0.5	Feb 25, 2025	- Improve validations
v1.0.4	Feb 20, 2025	- Add claim endpoints
v1.0.3	Feb 8, 2025	- Fix date format in the Vehicle object
v1.0.2	Feb 6, 2025	- Fix search by license plate
v1.0.1	Dec 13, 2024	- Fix leasing date nullability - Allow 0 self deductible in insurance
v1.0.0	Nov 20, 2024	- Initial release of the CourierIO API with Companies and Vehicles endpoints.

Chapter 3. Endpoints

3.1. ClaimService

3.1.1. getClaims

GET /v1/claims

Get all claims

Description

Parameters

Query Parameters

Name	Description	Required	Default	Pattern
limit	The maximum number of claims to return on one page.	-	100	
page	The page number to retrieve.	-	0	
companyId	The ID of the company to filter claims by.	-	null	
vehicleLicensePlate	The license plate of the vehicle to filter claims by.	-	null	
claimStatus	Status of the claim	-	null	
claimType	Possible values: `1` - Haftpflicht, `2` - Vollkasko, `3` - Teilkasko, `5` - Drittschaden, `6` - Eigenschaden, `7` - Eigendeckung, `8` - nicht vorhanden, `10` - Panne, `99` - Vorbehalt, `101` - Teilschuld-Drittschaden, `107` - Leasingrückläufer	-	null	
claimDateFrom	Claim Date From in format YYYY-MM-DD	-	null	
claimDateTo	Claim Date To in format YYYY-MM-DD	-	null	
createdDateFrom	Created Date From in format YYYY-MM-DD	-	null	
createdDateTo	Created Date To in format YYYY-MM-DD	-	null	

Return Type

array[[Claim](#)]

Content Type

- application/json
- /

Responses

Table 1. HTTP Response Codes

Code	Message	Datatype
404	Company not found	<<>>
500	Internal server error	ErrorResponse
401	The user is not authorized	<<>>
200	Success	List[Claim]

Samples

Server Responses:

Code 200 response body:

```
[
  {
    "id": 9007199254740991,
    "claimDateTime": "2025-03-05T10:29:20",
    "createdDateTime": "2025-03-05T10:29:20",
    "typeOfTravel": "BUSINESS",
    "claimAddress": "Niederwern",
    "claimDescription": "string",
    "claimFault": "unknown",
    "claimCause": 68,
    "claimKind": 45,
    "claimType": "1",
    "location": 1,
    "claimStatus": "IN_PROGRESS",
    "lastChangeDateTimeStamp": "2025-03-05T10:29:20",
    "claimClosureDate": "2025-03-05",
    "customerPortalLink": "string",
    "thirdParty": true,
    "insurance": {
      "claimComprehensiveInsuranceNumber": "SD1 5552 6944 37 T01",
      "claimLiabilityInsuranceNumber": "string",
      "reserveComprehensiveInsurance": 0,
      "reserveLiabilityInsurance": 0
    }
  },
]
```

```

"vehicle": {
  "id": 9007199254740991,
  "externalId": "string",
  "licensePlate": "string",
  "registrationDate": "2025-03-05",
  "mileage": 9007199254740991,
  "damagedParts": [
    {
      "text": "string",
      "group": "string"
    }
  ],
  "damageDescription": "string",
  "vehicleIdentificationNumber": "string"
},
"holderCompany": {
  "id": 9007199254740991,
  "name": "string"
},
"driver": {
  "lastName": "string",
  "firstName": "string"
},
"claimCosts": {
  "customerPaymentNet": 0,
  "customerPaymentTax": 0,
  "customerPaymentGross": 0,
  "insurancePaymentNet": 0,
  "insurancePaymentTax": 0,
  "insurancePaymentGross": 0,
  "opposingInsurancePaymentNet": 0,
  "opposingInsurancePaymentTax": 0,
  "opposingInsurancePaymentGross": 0
}
}
]

```

3.2. CompanyService

3.2.1. createCompany

POST /v1/companies

Create a new customer company

Description

Parameters

Body Parameter

Name	Description	Required	Default	Pattern
Company	<i>Company</i>	X		

Return Type

Company

Content Type

- /
- application/json

Responses

Table 2. HTTP Response Codes

Code	Message	Datatype
404	Not Found	[String]
200	Success	<i>Company</i>
401	The user is not authorized	<<>>
500	Internal server error	<i>ErrorResponse</i>
403	Access to the requested resource is forbidden	<i>ErrorResponse</i>
400	Bad request, adjust before retrying	<i>ErrorResponse</i>

Samples

Request body:

```
{
  "externalId": "EXT-34237",
  "type": "CUSTOMER",
  "name": "Kreative Werke",
  "street": "Kunststraße 789",
  "postalCode": "22750",
  "city": "Hamburg"
}
```

Server Responses:

Code 200 response body:

```
{
  "id": 678,
  "externalId": "EXT-34237",
```

```

    "type": "CUSTOMER",
    "name": "Kreative Werke",
    "street": "Kunststraße 789",
    "postalCode": "22750",
    "city": "Hamburg",
    "isConcern": false
  }

```

Code 400 response body:

```

{
  "success": false,
  "timestamp": "2025-03-05T12:47:37.02762933",
  "error": "Validation errors: street Company address must not be empty, city City is mandatory, postalCode Postal code is mandatory",
  "validationErrors": [
    {
      "fieldName": "street",
      "invalidValue": "null",
      "errorMessage": "Company address must not be empty"
    },
    {
      "fieldName": "city",
      "invalidValue": "null",
      "errorMessage": "City is mandatory"
    },
    {
      "fieldName": "postalCode",
      "invalidValue": "null",
      "errorMessage": "Postal code is mandatory"
    }
  ],
  "requestId": "d0141e5e-1484-40db-b361-665af36ee1fa"
}

```

3.2.2. deleteCompany

DELETE /v1/companies/{id}

Deactivate an existing company with ID

Description

Parameters

Path Parameters

Name	Description	Required	Default	Pattern
id	The ID of the company to delete.	X	null	

Return Type

[Object]

Content Type

- /
- application/json

Responses

Table 3. HTTP Response Codes

Code	Message	Datatype
404	Not Found	[String]
401	The user is not authorized	<<>>
500	Internal server error	<i>ErrorResponse</i>
403	Access to the requested resource is forbidden	<i>ErrorResponse</i>
200	Success	[Object]

Samples

3.2.3. getCompanies

GET /v1/companies

Get all companies

Description

Parameters

Query Parameters

Name	Description	Required	Default	Pattern
companyType	Specifies the type of company to filter the results by.	X	null	
limit	The maximum number of companies to return on one page.	-	100	
page	The page number to retrieve.	-	0	

Name	Description	Required	Default	Pattern
companyId	Filters the results to include only the specified root company ID and its subsidiary companies.	-	null	
levels	Limits the depth of the hierarchy for the subsidiary companies.	-	null	

Return Type

array[[Company](#)]

Content Type

- application/json

Responses

Table 4. HTTP Response Codes

Code	Message	Datatype
404	Company not found	<<>>
401	The user is not authorized	<<>>
500	Internal server error	ErrorResponse
403	Access to the requested resource is forbidden	ErrorResponse
200	Success	List[Company]

Samples

Server Responses:

Code 200 response body:

```
[
  {
    "id": 501559,
    "externalId": "EXT-34235",
    "type": "CUSTOMER",
    "name": "Vera GmbH",
    "street": "Holzstraße 10",
    "postalCode": "22469",
    "city": "Hamburg",
    "isConcern": false
  },
  {
    "id": 501621,
    "externalId": "EXT-34236",
    "parentCompanyId": 501559,
```

```

    "type": "CUSTOMER",
    "name": "Spedition Abc GmbH",
    "street": "Bergstraße 393",
    "postalCode": "22223",
    "city": "Hamburg",
    "isConcern": false
  }
]

```

3.2.4. getCompanyById

GET /v1/companies/{id}

Get a company by ID

Description

Parameters

Path Parameters

Name	Description	Required	Default	Pattern
id	The ID of the company to retrieve.	X	null	

Return Type

[Company](#)

Content Type

- application/json

Responses

Table 5. HTTP Response Codes

Code	Message	Datatype
404	Company not found	<<>>
200	Success	Company
401	The user is not authorized	<<>>
500	Internal server error	ErrorResponse
403	Access to the requested resource is forbidden	ErrorResponse

Samples

Server Responses:

Code 200 response body:

```
[
  {
    "id": 678,
    "externalId": "EXT-34237",
    "parentCompanyId": 345,
    "type": "CUSTOMER",
    "name": "Kreative Werke",
    "street": "Kunststraße 789",
    "postalCode": "22750",
    "city": "Hamburg"
    "isConcern": false
  }
]
```

3.2.5. updateCompany

PUT /v1/companies

Update an existing customer company with ID

Description

Parameters

Body Parameter

Name	Description	Required	Default	Pattern
Company	Company	X		

Return Type

[Company](#)

Content Type

- /
- application/json

Responses

Table 6. HTTP Response Codes

Code	Message	Datatype
404	Not Found	[String]
200	Success	Company
401	The user is not authorized	<<>>
500	Internal server error	ErrorResponse

Code	Message	Datatype
403	Access to the requested resource is forbidden	<i>ErrorResponse</i>
400	Bad request, adjust before retrying	<i>ErrorResponse</i>

Samples

Request body:

```
{
  "id": 678,
  "externalId": "EXT-34237",
  "type": "CUSTOMER",
  "name": "Kreative Werke",
  "street": "Kunststraße 789",
  "postalCode": "22750",
  "city": "Hamburg"
}
```

Server Responses:

Code 200 response body:

```
{
  "id": 678,
  "externalId": "EXT-34237",
  "type": "CUSTOMER",
  "name": "Kreative Werke",
  "street": "Kunststraße 789",
  "postalCode": "22750",
  "city": "Hamburg",
  "isConcern": false
}
```

3.3. VehicleService

3.3.1. createVehicle

POST /v1/vehicles

Create a new vehicle

Description

Parameters

Body Parameter

Name	Description	Required	Default	Pattern
Vehicle	Vehicle	X		

Return Type

[Vehicle](#)

Content Type

- application/json

Responses

Table 7. HTTP Response Codes

Code	Message	Datatype
404	Company not found	<<>>
401	The user is not authorized	<<>>
500	Internal server error	ErrorResponse
403	Access to the requested resource is forbidden	ErrorResponse
400	Bad request, adjust before retrying	ErrorResponse
200	Success	Vehicle

Samples

Request body:

```
{
  "externalId": "ID123",
  "licensePlate": "A-BC 123",
  "holderCompanyId": 123456,
  "manufacturer": "Opel",
  "model": "Opel Insignia 1.6",
  "motorType": "PETROL",
  "vehicleType": "CAR",
  "hsn": "0035",
  "tsn": "AIQ",
  "registrationDate": "2024-08-07",
  "vehicleIdentificationNumber": "2G1WH55K429147256",
  "costCenter": "2632",
  "carSpecificationsEntries": {
    "Assekuradeur VR": "Helvetia Schweizerische "
  },
  "driver": {
    "salutation": "HERR",
    "title": "DR",
```

```

    "lastName": "Bosch",
    "firstName": "Niklas",
    "phoneNumber": "+49 123 456789",
    "mobileNumber": "+49 173 6542321",
    "vip": true,
    "email": "niklas.bosch@example.com"
  },
  "leasing": {
    "leasingCompanyId": 7890,
    "leasingContractNumber": "LC123456",
    "leasingStart": "2023-01-01",
    "leasingEnd": "2025-12-31"
  },
  "insurance": {
    "liabilityInsuranceCompanyId": 654,
    "insuranceContractNumber": "IC987654",
    "comprehensiveInsuranceCompanyId": 987,
    "deductibleComprehensiveInsurance": 500,
    "fullComprehensiveInsuranceCompanyId": 3210,
    "deductibleFullComprehensiveInsurance": 1000
  }
}

```

Server Responses:

Code 200 response body:

```

{
  "id": 123,
  "externalId": "ID123",
  "licensePlate": "A-BC 123",
  "holderCompanyId": 123456,
  "manufacturer": "Opel",
  "model": "Opel Insignia 1.6",
  "motorType": "PETROL",
  "vehicleType": "CAR",
  "hsn": "0035",
  "tsn": "AIQ",
  "registrationDate": "2024-08-07",
  "vehicleIdentificationNumber": "2G1WH55K429147256",
  "costCenter": "2632",
  "carSpecificationsEntries": {
    "Assekurateur VR": "Helvetia Schweizerische "
  },
  "driver": {
    "salutation": "HERR",
    "title": "DR",
    "lastName": "Bosch",
    "firstName": "Niklas",
    "phoneNumber": "+49 123 456789",

```

```

    "mobileNumber": "+49 173 6542321",
    "vip": true,
    "email": "niklas.bosch@example.com"
  },
  "leasing": {
    "leasingCompanyId": 7890,
    "leasingContractNumber": "LC123456",
    "leasingStart": "2023-01-01",
    "leasingEnd": "2025-12-31"
  },
  "insurance": {
    "liabilityInsuranceCompanyId": 654,
    "insuranceContractNumber": "IC987654",
    "comprehensiveInsuranceCompanyId": 987,
    "deductibleComprehensiveInsurance": 500,
    "fullComprehensiveInsuranceCompanyId": 3210,
    "deductibleFullComprehensiveInsurance": 1000
  }
}

```

Code 400 response body:

```

{
  "success": false,
  "timestamp": "2025-03-05T12:43:09.734938196",
  "error": "Validation errors: id must be null, driver.salutation Salutation must be empty or have one of the following values: 'DAMEN_HERREN', 'FRAU' oder 'HERR'., driver.title Titel must be empty or have one of the following values: 'DR', 'PROF' oder 'PROF_DR'.",
  "validationErrors": [
    {
      "fieldName": "id",
      "invalidValue": "123",
      "errorMessage": "must be null"
    },
    {
      "fieldName": "driver.salutation",
      "invalidValue": "string",
      "errorMessage": "Salutation must be empty or have one of the following values: 'DAMEN_HERREN', 'FRAU' oder 'HERR'."
    },
    {
      "fieldName": "driver.title",
      "invalidValue": "string",
      "errorMessage": "Titel must be empty or have one of the following values: 'DR', 'PROF' oder 'PROF_DR'."
    }
  ],
  "requestId": "de05b787-d740-408f-9186-a1bc93fb6660"
}

```

```
}
```

3.3.2. deleteVehicle

DELETE /v1/vehicles/{vehicleId}

Deactivate an existing vehicle with ID

Description

Parameters

Path Parameters

Name	Description	Required	Default	Pattern
vehicleId	The ID of the vehicle to delete.	X	null	

Return Type

[Object]

Content Type

- application/json
- /

Responses

Table 8. HTTP Response Codes

Code	Message	Datatype
404	Company not found	<<>>
401	The user is not authorized	<<>>
500	Internal server error	ErrorResponse
403	Access to the requested resource is forbidden	ErrorResponse
200	Success	[Object]

Samples

3.3.3. getVehicle

GET /v1/vehicles/{vehicleId}

Get a vehicle by ID

Description

Parameters

Path Parameters

Name	Description	Required	Default	Pattern
vehicleId	The ID of the vehicle to retrieve.	X	null	

Return Type

Vehicle

Content Type

- application/json
- /

Responses

Table 9. HTTP Response Codes

Code	Message	Datatype
404	Vehicle not found	<<>>
401	The user is not authorized	<<>>
500	Internal server error	<i>ErrorResponse</i>
200	Success	<i>Vehicle</i>

Samples

3.3.4. getVehicles

GET /v1/vehicles

Get all vehicles

Description

Parameters

Query Parameters

Name	Description	Required	Default	Pattern
limit	The maximum number of claims to return on one page.	-	100	
page	The page number to retrieve.	-	0	
companyId	The ID of the company to filter vehicles by.	-	null	

Return Type

array[[Vehicle](#)]

Content Type

- application/json
- /

Responses

Table 10. HTTP Response Codes

Code	Message	Datatype
404	Company not found	<<>>
401	The user is not authorized	<<>>
500	Internal server error	ErrorResponse
200	Success	List[Vehicle]

Samples

Server Responses:

Code 200 response body:

```
[
  {
    "id": 123,
    "externalId": "ID123",
    "licensePlate": "A-BC 123",
    "holderCompanyId": 123456,
    "manufacturer": "Opel",
    "model": "Opel Insignia 1.6",
    "motorType": "PETROL",
    "vehicleType": "CAR",
    "hsn": "0035",
    "tsn": "AIQ",
    "registrationDate": "2024-08-07",
    "vehicleIdentificationNumber": "2G1WH55K429147256",
    "costCenter": "2632",
    "carSpecificationsEntries": {
      "Assekuradeur VR": "Helvetia Schweizerische "
    },
    "driver": {
      "salutation": "HERR",
      "title": "DR",
      "lastName": "Bosch",
      "firstName": "Niklas",
      "phoneNumber": "+49 123 456789",
```

```

    "mobileNumber": "+49 173 6542321",
    "vip": true,
    "email": "niklas.bosch@example.com"
  },
  "leasing": {
    "leasingCompanyId": 7890,
    "leasingContractNumber": "LC123456",
    "leasingStart": "2023-01-01",
    "leasingEnd": "2025-12-31"
  },
  "insurance": {
    "liabilityInsuranceCompanyId": 654,
    "insuranceContractNumber": "IC987654",
    "comprehensiveInsuranceCompanyId": 987,
    "deductibleComprehensiveInsurance": 500,
    "fullComprehensiveInsuranceCompanyId": 3210,
    "deductibleFullComprehensiveInsurance": 1000
  }
}
]

```

3.3.5. updateVehicle

PUT /v1/vehicles

Update an existing vehicle with ID

Description

Parameters

Body Parameter

Name	Description	Required	Default	Pattern
Vehicle	Vehicle	X		

Return Type

[Vehicle](#)

Content Type

- application/json

Responses

Table 11. HTTP Response Codes

Code	Message	Datatype
404	Company not found	<<>>

Code	Message	Datatype
401	The user is not authorized	<<>>
500	Internal server error	<i>ErrorResponse</i>
400	Bad request, adjust before retrying	<i>ErrorResponse</i>
200	Success	<i>Vehicle</i>

Samples

Request body:

```
{
  "id": 123,
  "externalId": "ID123",
  "licensePlate": "A-BC 123",
  "holderCompanyId": 123456,
  "manufacturer": "Opel",
  "model": "Opel Insignia 1.6",
  "motorType": "PETROL",
  "vehicleType": "CAR",
  "hsn": "0035",
  "tsn": "AIQ",
  "registrationDate": "2024-08-07",
  "vehicleIdentificationNumber": "2G1WH55K429147256",
  "costCenter": "2632",
  "carSpecificationsEntries": {
    "Assekurateur VR": "Helvetia Schweizerische "
  },
  "driver": {
    "salutation": "HERR",
    "title": "DR",
    "lastName": "Bosch",
    "firstName": "Niklas",
    "phoneNumber": "+49 123 456789",
    "mobileNumber": "+49 173 6542321",
    "vip": true,
    "email": "niklas.bosch@example.com"
  },
  "leasing": {
    "leasingCompanyId": 7890,
    "leasingContractNumber": "LC123456",
    "leasingStart": "2023-01-01",
    "leasingEnd": "2025-12-31"
  },
  "insurance": {
    "liabilityInsuranceCompanyId": 654,
    "insuranceContractNumber": "IC987654",
    "comprehensiveInsuranceCompanyId": 987,
    "deductibleComprehensiveInsurance": 500,
  }
}
```

```
    "fullComprehensiveInsuranceCompanyId": 3210,  
    "deductibleFullComprehensiveInsurance": 1000  
  }  
}
```

Server Responses:

Code 200 response body:

```
{  
  "id": 123,  
  "externalId": "ID123",  
  "licensePlate": "A-BC 123",  
  "holderCompanyId": 123456,  
  "manufacturer": "Opel",  
  "model": "Opel Insignia 1.6",  
  "motorType": "PETROL",  
  "vehicleType": "CAR",  
  "hsn": "0035",  
  "tsn": "AIQ",  
  "registrationDate": "2024-08-07",  
  "vehicleIdentificationNumber": "2G1WH55K429147256",  
  "costCenter": "2632",  
  "carSpecificationsEntries": {  
    "Assekurateur VR": "Helvetia Schweizerische "  
  },  
  "driver": {  
    "salutation": "HERR",  
    "title": "DR",  
    "lastName": "Bosch",  
    "firstName": "Niklas",  
    "phoneNumber": "+49 123 456789",  
    "mobileNumber": "+49 173 6542321",  
    "vip": true,  
    "email": "niklas.bosch@example.com"  
  },  
  "leasing": {  
    "leasingCompanyId": 7890,  
    "leasingContractNumber": "LC123456",  
    "leasingStart": "2023-01-01",  
    "leasingEnd": "2025-12-31"  
  },  
  "insurance": {  
    "liabilityInsuranceCompanyId": 654,  
    "insuranceContractNumber": "IC987654",  
    "comprehensiveInsuranceCompanyId": 987,  
    "deductibleComprehensiveInsurance": 500,  
    "fullComprehensiveInsuranceCompanyId": 3210,  
    "deductibleFullComprehensiveInsurance": 1000  
  }  
}
```

AFEC

Chapter 4. Models

4.1. Claim

Field Name	Required	Nullable	Type	Description	Format
id			Long		int64
claimDateTime			Date		date-time
createdDateTime			Date		date-time
typeOfTravel			String	Type of travel. Possible values: * `UNKNOWN` - Unbekannt * `BUSINESS` - Dienstlich * `PRIVATE` - Privat	
claimAddress			String		
claimDescription			String		
claimFault			String	ClaimFault Possible values: * `OWN` - Eigen * `THIRD_PARTY` - Fremd * `SUBJECT_TO_CONDITION` - Vorbehalt * `UNKNOWN` - Unbekannt	
claimCause			Integer	Cause of the claim. Possible values: * `81` - Panne Sonstiges * `83` - Leasingrückläufer Sonstiges * `84` - Leasingrückläufer - Steinschlag * `101` - Steinschlag * `102` - Einbruch/Diebstahl * `103` - Hagel * `104` - Sturm * `105` - Blitz * `106` - Feuer * `107` - Überschwemmung * `108` - Marder * `109` - Wild * `110` - Glätte * `111` - Vandalismus * `112` - Gegenstand auf der Fahrbahn * `113` - Unachtsamkeit * `116` - Externe Ursache * `124` - Tierschlag * `125` - Geröll * `127` - Unbekannte Ursache	int32

Field Name	Required	Nullable	Type	Description	Format
claimKind			Integer	Claim kind. Possible values: * `45` - Panne Sonstiges * `46` - Panne Reifen * `50` - Leasingrückläufer * `101` - Totaldiebstahl * `102` - Teildiebstahl * `103` - Zusammenstoß * `104` - Auffahrunfall * `105` - Streifkollision * `106` - Schleuderunfall * `107` - Rangierschaden * `108` - Parkschaden * `109` - Glasschaden WSS * `110` - Sachbeschädigung * `111` - Betriebs-/Brems-/Bruchschaden * `112` - Be-und Entladen * `113` - Elementarschaden * `115` - Feststellungsschaden * `116` - Glasschaden Sonstiges * `117` - Schlüsselverlust * `120` - Tierschaden	int32
claimType			<i>ClaimType</i>		1, 2, 3, 5, 6, 7, 8, 10, 99, 101, 107,
location			Integer	Location category. Possible values: * `1` - Autobahn * `2` - Str. innerhalb geschl. Ortschaften * `3` - Str. außerhalb geschl. Ortschaften * `4` - Firmengelände / eigen * `7` - Tiefgarage / Parkhaus * `8` - Werkstatt * `6` - Parkplatz * `9` - Privatgelände * `10` - Zone 30 * `11` - Bundesstrasse * `12` - Landstrasse * `13` - Tankstelle * `14` - Autozug * `15` - unbekannt * `16` - Firmengelände / fremd	int32
claimStatus			<i>ClaimStatus</i>		IN_PROGRESS, CLOSED,
lastChangeDateTimestamp			Date		date-time

Field Name	Required	Nullable	Type	Description	Format
claimClosureDate			date		date
customerPortalLink			String		
thirdParty			Boolean		
insurance			<i>ClaimInsurance</i>		
vehicle			<i>ClaimVehicle</i>		
holderCompany			<i>ClaimCompany</i>		
driver			<i>ClaimDriver</i>		
claimCosts			<i>ClaimCosts</i>		

4.2. ClaimCompany

Field Name	Required	Nullable	Type	Description	Format
id			Long		int64
name			String		

4.3. ClaimCosts

Field Name	Required	Nullable	Type	Description	Format
customerPaymentNet			BigDecimal		
customerPaymentTax			BigDecimal		
customerPaymentGross			BigDecimal		
insurancePaymentNet			BigDecimal		
insurancePaymentTax			BigDecimal		
insurancePaymentGross			BigDecimal		
opposingInsurancePaymentNet			BigDecimal		

Field Name	Required	Nullable	Type	Description	Format
opposingInsurancePaymentTax			BigDecimal		
opposingInsurancePaymentGross			BigDecimal		

4.4. ClaimDriver

Field Name	Required	Nullable	Type	Description	Format
lastName			String		
firstName			String		

4.5. ClaimInsurance

Field Name	Required	Nullable	Type	Description	Format
claimComprehensiveInsuranceNumber			String		
claimLiabilityInsuranceNumber			String		
reserveComprehensiveInsurance			BigDecimal		
reserveLiabilityInsurance			BigDecimal		

4.6. ClaimStatus

Claim status. Possible values: * `IN\_PROGRESS` - in Bearbeitung

- `CLOSED` - abgeschlossen

Enum Values
IN_PROGRESS
CLOSED

4.7. ClaimType

Claim type. Possible values: * `1` - Haftpflicht

- `2` - Vollkasko
- `3` - Teilkasko
- `5` - Drittschaden
- `6` - Eigenschaden
- `7` - Eigendeckung
- `8` - nicht vorhanden
- `10` - Panne
- `99` - Vorbehalt
- `101` - Teilschuld-Drittschaden
- `107` - Leasingrückläufer

Enum Values
1
2
3
5
6
7
8
10
99
101
107

4.8. *ClaimVehicle*

Field Name	Required	Nullable	Type	Description	Format
id			Long		int64
externalId			String		
licensePlate			String		
registrationDate			date		date
mileage			Long		int64
damagedParts			List of <i>DamagedPart</i>		

Field Name	Required	Nullable	Type	Description	Format
damageDescription			String		
vehicleIdentificationNumber			String		

4.9. Company

Field Name	Required	Nullable	Type	Description	Format
id			Long	A unique internal identifier for the company	int64
externalId			String		
parentCompanyId			Long	The unique identifier of the parent company, if applicable	int64
type	X		[String]		Enum: CUSTOMER, INSURANCE, LEASING,
name	X		String	The name of the company	
street	X		String	The street address of the company	
postalCode	X		String	The postal code of the company's address	
city	X		String	The city where the company is located	
isConcern			Boolean	Indicates whether the company can have sub-companies	

4.10. CompanyType

Enum Values
CUSTOMER
INSURANCE
LEASING

4.11. *DamagedPart*

Field Name	Required	Nullable	Type	Description	Format
text			String		
group			String		

4.12. *Driver*

Information about the driver of the vehicle

Field Name	Required	Nullable	Type	Description	Format
salutation			[String]	Salutation of the driver	Enum: DAMEN, _HERREN, FRAU, HERR,
title			[String]	Title of the driver	Enum: DR, PROF, PROF_D R,
lastName			String	Last name of the driver	
firstName			String	First name of the driver	
phoneNumber			String	Phone number of the driver	
mobileNumber			String	Mobile number of the driver	
vip			Boolean	Indicates if the driver is a VIP	
email			String	Email address of the driver	

4.13. *ErrorEntry*

Field Name	Required	Nullable	Type	Description	Format
fieldName			String		
invalidValue			String		
errorMessage			String		

4.14. *ErrorResponse*

Field Name	Required	Nullable	Type	Description	Format
success			Boolean		
timestamp			Date		date-time
error			String		
validationErrors			List of ErrorEntry		
requestId			String		

4.15. *Insurance*

Information about the vehicle's insurance details

Field Name	Required	Nullable	Type	Description	Format
liabilityInsuranceCompanyId	X		Long	ID of the liability insurance company	int64
insuranceContractNumber			String	Contract number of the insurance agreement	
comprehensiveInsuranceCompanyId			Long	ID of the comprehensive insurance company	int64
deductibleComprehensiveInsurance			BigDecimal	Deductible amount for comprehensive insurance	
fullComprehensiveInsuranceCompanyId			Long	ID of the full comprehensive insurance company	int64
deductibleFullComprehensiveInsurance			BigDecimal	Deductible amount for full comprehensive insurance	

4.16. *Leasing*

Information about the vehicle's leasing details

Field Name	Required	Nullable	Type	Description	Format
leasingCompanyId	X		Long	ID of the leasing company	int64
leasingContractNumber			String	Contract number of the leasing agreement	
leasingStart			date	Start date of the leasing contract in format YYYY-MM-DD	date
leasingEnd			date	End date of the leasing contract in format YYYY-MM-DD	date

4.17. *Salutation*

Enum Values
DAMEN_HERREN
FRAU
HERR

4.18. *SyncStatus*

Enum Values
UPDATE_PENDING
DELETE_PENDING
SYNCED

4.19. *Title*

Enum Values
DR
PROF
PROF_DR

4.20. *Vehicle*

Field Name	Required	Nullable	Type	Description	Format
id			Long	Internal vehicle identifier automatically generated during vehicle creation	int64

Field Name	Required	Nullable	Type	Description	Format
externalId			String	Arbitrary external identifier of the vehicle	
licensePlate	X		String	The vehicle's license plate number	
holderCompanyId	X		Long	The ID of the holder company as obtained from the company service	int64
manufacturer			String	The manufacturer of the vehicle	
model			String	The model of the vehicle	
motorType			String	Motor Type. Possible values: * `PETROL` - Benzin * `DIESEL` - Diesel * `NATURAL_GAS` - Gas * `ELECTRIC` - Elektro * `HYBRID` - Hybrid * `HYDROGEN_GAS` - Wasserstoff	
vehicleType			String	Vehicle Type. Possible values: * `CAMPER_VAN` - Wohnmobil * `BUS` - BUS * `TRUCK` - LKW * `MOTORCYCLE` - Motorrad * `VAN` - Transporter * `SPECIAL_PURPOSE_VEHICLE` - Sonderfahrzeug * `TRAILER` - Anhänger * `CAR` - PKW	
hsn			String	The code identifying the manufacturer of the vehicle	
tsn			String	The code identifying the vehicle type of the respective manufacturer	
registrationDate	X		date	Registration date in the format YYYY-MM-DD maximum 80 years in the past and 2 months in the future	date
vehicleIdentificationNumber	X		String	The code comprised of capital letters and numbers that uniquely identifies the vehicle usually 17 character long or shorter for a special vehicle type (e.g.trailer)	
costCenter			String	The cost center associated with the vehicle	
carSpecificationsEntries			Map of [string]	List of key-value pairs containing additional vehicle information	

Field Name	Required	Nullable	Type	Description	Format
driver			<i>Driver</i>		
leasing			<i>Leasing</i>		
insurance			<i>Insurance</i>		

ALFC