

API-Dokumentation

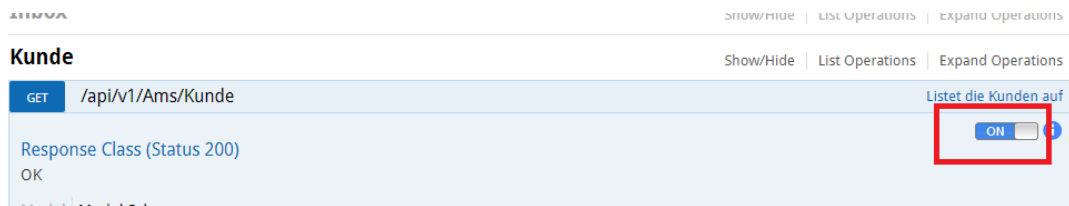
Stand 09.02.2022



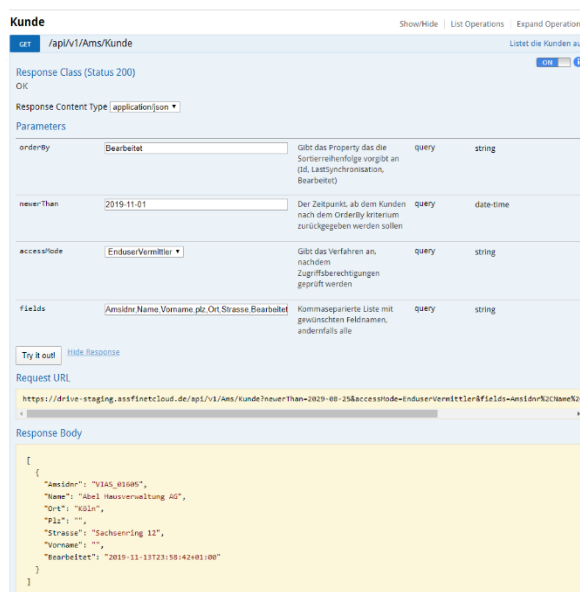
Thema: Kunden-/Vertrags-/Schaden-Daten abfragen

Daten über die API abfragen

- Voraussetzung für eine Abfrage mit der API ist, dass man an der assfinet Cloud aktuell angemeldet ist:



- Die unterschiedlichen Systeme können per API-Oberfläche erreicht werden über:
 - Demo: <https://drive-demo.assfinetcloud.de/swagger/ui/index>
 - UAT: <https://drive-uat.assfinetcloud.de/swagger/ui/index>
 - Staging: <https://drive-staging.assfinetcloud.de/swagger/ui/index>
 - Produktiv: <https://drive-azure.assfinet.de/swagger/ui/index>
- Alle meine Kunden abfragen. ACHTUNG: Es sollten nur beim Projektstart einmal alle Kunden abgerufen werden. Anschließend sollten möglichst nur Differenzabfragen verwendet werden. Wenn man z.B. täglich Nachts die Differenzabfragen macht, kann jeweils nach den Änderungen seit der vorigen Abfrage angefragt werden. Parameter OrderBy und NewerThan müssen dazu übergeben werden
 - GET /api/v1/Ams/Kunde
 - Es werden nur gefüllte Felder zurück gegeben
 - Als Parameter z.B.
 - orderBy Text nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
 - byDescending Boolean wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
 - newerThan Datum nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit).
Format: JJJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
Wenn angegeben, filtert die Kunden auf diejenigen, die innerhalb der angegebenen Tage Geburtstag haben, leer -> nicht filtern
 - birthdayWithinDays Zahl Wie viele Kunden aus der sortierten Liste sollen für die Ausgabe übersprungen werden? Wird zum Paging/Blättern benötigt (z.B. erste 50 Kunden, Zweite 50 Kunden, ...)
 - skip Zahl leer -> 0 = keinen überspringen
 - take Zahl Wie viele Kunden aus der sortierten Liste sollen zurückgegeben werden? leer -> ersten 50 Kunden
Maximal 5000 Datensätze gleichzeitig. Sonst mit Paging über Skip=5000 die nächsten holen
„take=5000“ holt die ersten 5000 Datensätze ab
„take=5000&skip=5000“ holt die nächsten 5000 Datensätze ab
„take=5000&skip=10000“ holt die nachfolgenden Datensätze ab u.s.w.
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden.
Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle



- Kunden-Abfrage mitgeschnitten:
 - REQUEST:
GET /api/v1/Ams/Kunde HTTP/1.1
 - RESPONSE:

```
{["Adrid":"VIAS_51605","Amsidnr":"VIAS_01605","Anrede":"Sehr geehrter Herr
\\","Arbeitgeber":"","Bank1":"","Bearbeiter":"JK","Beruf":"","Berufsstatus":"","Bic1":"","Blz1":"","Branche":"","EmailGeschaeftlich":"Jokoch60_0@gmail.com","EmailPriv
at":"Jokoch60@gmail.com","Familienstand":"","Fax":"0221-3887791-0","Geburtsname":"","Geburtsort":"","Geburtsjahr":"1967-05-
21T00:00:00","Geschlecht":"","Handelsregisternummer":"","HRB_100573726","Iban1":"","InhaberGF":"","Hans_Abel","InterneNr":"","10035","Internet":"www.abel-
hausverwaltung.de","IsHistorie":false,"Kinder":0,"Klasse":"","Konto1":"","Kontobezeichnung":"","Kontoinhaber":"","Kundenname":"Abel Hausverwaltung AG,
","KundeSeit":"2003-01-01T00:00:00","Land":"D","Mandant":"Cloud
Versicherungsmakler","MandantAmsidnr":"VIAS_00CLOUD","Mitarbeiterzahl":3,"Mobiltelefon":"0170-1256789-0","Name":"Abel Hausverwaltung
AG","Name1":"","Name2":"","Ort":"","Köln":"","PersonalausweisBehoerde":"","PersonalausweisNr":"","Plz":"","Rechtsform":"","Einzelunternehm","Sozialversicherungsnummer
":"","Staatsangehoerigkeit":"","Steueridentifikationsnummer":"","Steuerklasse":"","Steuernummer":"","Sta / 4573472612","Strasse":"","Sachsenring
12","TelefonGeschaeftlich":"0221-3887791-0","TelefonPrivat":"0221-3887791","Titel":"","Vorname":"","Zuordnung":"PRIVAT","Id":"","f8b6abb2-eb9a-4f99-88d2-
a26a1dfff9ce","Bearbeitet":"2019-11-13T23:58:42+01:00","Dirty":false,"License":"","DEMO_657661","LastSynchronisation":"2019-11-13T12:49:28.6283075Z"]}]
```

- Ruft einen bestimmten Kunden ab
 - GET /api/v1/Ams/Kunde/{id}
 - Es werden nur die gefüllten Felder zurück gegeben
 - für Änderungen an Kunden-/Vertrag-Daten sollten diese zuvor mit accessMode=Admin gelesen worden sein, damit der vollständige Datensatz gespeichert wird. Sonst werden die fehlenden Felder vom Programm ergänzt. Der Weg über Adminzugriff wird dabei aber empfohlen. Der Admin-Zugriff benötigt keine „Freischaltungen“
 - Schlagwörter/TAGS zum Kunden werden an seiner Adresse gespeichert in Feld-Schlagwoerter: api/v1/Ams/Kunde/{id}/Adresse
 - Als Parameter z.B.

Parameter	Typ	Bedeutung
id	Text	Id des Kunden
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
fields	Text	Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Alle meine Kunden-GUID abfragen, welche gelöscht wurden
 - GET /api/v1/Ams/Kunde/Deleted
 - Als Parameter z.B.

Parameter	Typ	Bedeutung
deletedNewerThan	Datum	Der Zeitpunkt ab dem Kunden gelöscht wurden, Format: JJJJ-MM-DD, leer -> alle anzeigen
accessMode	Text	Gibt das Verfahren an, nachdem Zugriffsberechtigungen geprüft werden. Derzeit nur Ergebnisse bei Admin
 - Rückgabe

Parameter	Typ	Bedeutung
id	Guid	Id des Kunden
Geloescht	Datum	Datum an dem der Datensatz gelöscht wurde
Amsidnr	Text	AmsidNr, Wert ist teilweise ausgenullt
DeletedReason	Zahl	Grund für diesen Datensatz 0=ReplicationDeleted //Datensatz wurde durch Replication gelöscht
- Fragt zu einem Kunden die Beziehungen ab
 - GET /api/v1/Ams/Kunde/{id}/Beziehung
 - Als Parameter z.B.

Parameter	Typ	Bedeutung
id	Guid	Id des Kunden
accessMode	Text	Gibt das Verfahren an, nachdem Zugriffsberechtigungen geprüft werden. Derzeit nur Ergebnisse bei Admin
 - Rückgabe: Adressbeziehungen mit jeweils der Amspk des Beziehungs-Kunden
- Kunde über Namen suchen
 - GET /api/v1/Search
 - Als Parameter z.B.

Parameter	Typ	Bedeutung
searchTerm	Text	Text nach dem Objekte gefiltert werden. Groß-/Kleinschreibung ist egal. Text kann auch Textteil am Anfang/Mitte/Ende sein (Ausnahmen bei FullText... : nur Wortanfang und mindestens 2 Buchstaben). Nachfolgende Zeichen werden wie Leerzeichen behandelt: ;,:-_*+~`?={}) /!&%\$%\$ " ' ^ < >
objects	Text,Text	Die Objekte die durchsucht werden sollen (ansonsten alle und langsamer), Kunde,Vertrag,Schaden,Vorgang,Brief,Vermittler,All
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
skip	Zahl	Wie viele Datensätze aus der sortierten Liste sollen für die Ausgabe übersprungen werden? Wird zum Paging/Blättern benötigt (z.B. erste 10 Datensätze, Zweite 10 Datensätze, ...), leer -> 0 = keinen überspringen
take	Zahl	Wie viele Datensätze aus der sortierten Liste sollen zurückgegeben werden? leer -> ersten 10 Datensätze Maximal 5000 Datensätze gleichzeitig. Sonst mit Paging über Skip=5000 die nächsten holen „take=5000“ holt die ersten 5000 Datensätze ab „take=5000&skip=5000“ holt die nächsten 5000 Datensätze ab „take=5000&skip=10000“ holt die nachfolgenden Datensätze ab u.s.w.
parentAmsidnr	Text	Amsidnr des Vaterdatensatz (beim Vertrag/Schaden/Termin). So kann z.B. gezielt nur innerhalb der Verträge eines Kunden gesucht werden (oder Schäden eines Vertrages)
algorithmns	Text,Text	Die Suchalgorithmen die verwendet werden sollen (ansonsten alle und langsamer) z.B. KundenVerbindung,KundeAddress <ul style="list-style-type: none"> • SchadenNummer Suche nach Schadennummer (Feld Schäden.SchadNr normalisiert [ohne Sonderzeichen] und nicht normalisiert) • TerminNummer Suche nach Vorgangsnummer (Feld Termine.Nummer) • VSNR Suche nach Versicherungsscheinnummer (Feld Verträge.VNR normalisiert [ohne Sonderzeichen wie Striche+Leerzeichen+Doppelpunkt] und nicht normalisiert) • KundenVerbindung Suche in der Verbindung des Kunden ab Wortanfang (Feld Kunde.Matchcode) • FullTextAddress Suche in Adressdaten ab Wortanfang (Kundenname, Strasse, Plz, Ort) des Kunden (Feld Kunde.Vname + Str + Plz + Ort) • FullTextName Suche im Drive Vorgangsnamen (Feld Processes.Name) • FullTextRisiko Suche im Vertrags Risiko aber ab Wortanfang (Feld Verträge.Risiko) • FullTextTerminBetreff Suche im Vorgangsbetreff (Feld Processes.Grund) • FullTextTitel Suche im Dokumententitel (Feld Briefe.Titel) • FullTextVorgang Suche im Schaden Vorgang ab Wortanfang (Feld Schäden.Vorgang) • FullTextSuchname Suche im Vermittler Suchname aber ab Wortanfang (Feld Vermittler.VCode) • VermittlerNummer Suche in Vermittler Nummer (Feld Vermittler.VCode) • InterneNummer Suche nach InterneNummer (Feld Verträge.Freifeld) • KundeAddress Suche in Adressdaten (Kundenname, Strasse, Plz, Ort) des Kunden (Feld Kunde.Vname + Str + Plz + Ort) • ProcessName Suche im Drive Vorgangsnamen (Feld Processes.Name) • VertragRisiko Suche im Vertrags Risiko (Feld Verträge.Risiko) • TerminBetreff Suche im Vorgangsbetreff (Feld Processes.Betreff)

- BriefTitel Suche im Dokumententitel (Feld Briefe.Titel)
- SchadenVorgang Suche im Schaden Vorgang (Feld Schäden.Vorgang)
- VermittlerSuchname Suche im Vermittler Suchname (Feld Vermittler.VName)
- All = SchadenNummer | TerminNummer | VSNR | KundenVerbindung | KundeAddress | ProcessName | VertragRisiko | TerminBetreff | BriefTitel | SchadenVorgang | VermittlerSuchname | VermittlerNummer | InterneNummer,
- AllFullText = SchadenNummer | TerminNummer | VSNR | KundenVerbindung | FullTextAddress | FullTextName | FullTextRisiko | FullTextTerminBetreff | FullTextTitel | FullTextVorgang | FullTextSuchname | VermittlerNummer | InterneNummer

Search Show/Hide List Operations Expand Operations

GET /api/v1/Search Durchsucht freigeschaltete Objekte nach einem String

Response Class (Status 200)

OK

Parameters

Parameter	Value	Description	Parameter Type	Data Type
searchTerm	Hausverw	String nach dem Objekte gefiltert werden	query	string
objects	Kunde	Die Objekte die durchsucht werden sollen (ansonsten alle)	query	Array(string)
algorithms	FullTextAddress	Die Suchalgorithmen die verwendet werden sollen (ansonsten alle)	query	Array(string)
accessMode	EnduserVermittler	Gibt das Verfahren an, nachdem Zugriffsberechtigungen geprüft werden	query	string

Request URL

https://drive-staging.assfinetcloud.de/api/v1/Search?searchTerm=hausverw&objects=kunde&algorithms=fullTextAddress&accessMode=enduserVermittler

Response Body

```
{
  "Id": "f8bde001-e50a-4f99-8802-a26e10ff9fce",
  "rev": "Abel Hausverwaltung AG, Sachsenring 12 Köln",
  "Object": "Kunde",
  "Algorithms": "FullTextAddress"
}
```

Response Code

200

- Ruft die Adresse eines Kunden ab (eigene Adresse). Gibt auch die Kunden-Schlagwörter in Feld-Schlagwoerter
 - GET /api/v1/Ams/Kunde/{id}/Adresse
 - Als Parameter z.B.
 - Id Text Id des Kunden
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Ruft die Referenzadresse eines Kunden ab (z.B. Hausmeister/Geschäftsführer/Rechnungsanschrift)
 - GET /api/v1/Ams/Kunde/{id}/ReferenzAdresse
 - Als Parameter z.B.
 - Id Text Id des Kunden
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - Verwendung Text Filter auf die Verwendung der LinkAdressen, mehrere Kommagetrennt angeben
 - objects Text,Text Wenn angegeben, erlaubt zusätzlich das Auflisten von Adressen der Verträgen und oder Schäden des Kunden, Kunde,Vertrag,Schaden,Vorgang,Brief,Vermittler,All
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Ruft die Verträge eines Kunden ab
 - GET /api/v1/Ams/Kunde/{id}/Vertrag
 - Als Parameter z.B.
 - Id Text Id des Kunden
 - orderBy Text nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
 - byDescending Boolean wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
 - newerThan Datum nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit). Format: JJJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
 - birthdayWithinDays Zahl Wenn angegeben, filtert die Kunden auf diejenigen, die innerhalb der angegebenen Tage Geburtstag haben, leer -> nicht filtern
 - skip Zahl Wie viele Kunden aus der sortierten Liste sollen für die Ausgabe übersprungen werden? Wird zum Paging/Blättern benötigt (z.B. erste 50 Kunden, Zweite 50 Kunden, ...), leer -> 0 = keinen überspringen
 - take Zahl Wie viele Kunden aus der sortierten Liste sollen zurückgegeben werden? leer -> ersten 50 Kunden
 - sparte Text „take=5000“ holt die ersten 5000 Datensätze ab
 - status Text „take=5000&skip=5000“ holt die nächsten 5000 Datensätze ab
 - tags Text „take=5000&skip=10000“ holt die nachfolgenden Datensätze ab. u.s.w.
 - accessMode Text Gibt optional eine Sparte an, auf die gefiltert wird (z.B. KFZ), leer -> alle
 - fields Text Gibt optional einen VertragStatus an, auf die gefiltert wird (z.B. Eigen,Fremd,Anfrage,Angebot, Fremtteil,Gesperrt,Info,Ruhend,Storno,Antrag,Police), leer -> alle. Feld: ComputedStatus. Dabei wird EIGEN aufgeteilt: mit Dokument: POLICE, mit Antragsdatum: ANTRAG, sonst EIGEN.
 - tags Text Schlagworte auf die gefiltert wird, leer -> alle
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Ruft den Kunden eines Vertrages ab
 - GET /api/v1/Ams/Vertrag/{id}/Kunde

- Als Parameter z.B.

Id	Text	Id des Vertrages
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
fields	Text	Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Alle meine Vertrag-GUID abfragen, welche gelöscht wurden
 - GET /api/v1/Ams/Vertrag/Deleted
 - Als Parameter z.B.

deletedNewerThan	Datum	Der Zeitpunkt ab dem Verträge gelöscht wurden, Format: JJJJ-MM-DD, leer -> alle anzeigen
accessMode	Text	Gibt das Verfahren an, nachdem Zugriffsberechtigungen geprüft werden. Derzeit nur Ergebnisse bei Admin
 - Rückgabe

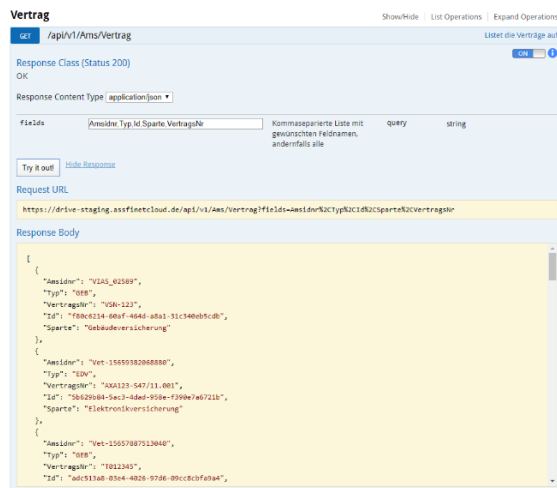
Id	Guid	Id des Vertrages
Geloescht	Datum	Datum an dem der Datensatz gelöscht wurde
AmsidNr	Text	AmsidNr, Wert ist teilweise ausgegült
DeletedReason	Zahl	Grund für diesen Datensatz 0=ReplicationDeleted //Datensatz wurde durch Replication gelöscht
- Listet die Termine/Aufgaben/Vorgänge/Wiedervorlagen eines Kunden
 - GET /api/v1/Ams/Kunde/{id}/Termin
 - Als Parameter z.B.

Id	Text	Id des Kunden
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
kategorie	Text	Art des Termins: 'Aufgabe', 'Termin', 'Vorgang', 'Wiedervorlage', 'Kontaktnotiz'
fields	Text	Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Listet die Briefe eines Kunden
 - GET /api/v1/Ams/Kunde/{id}/Brief
 - Als Parameter z.B.

Id	Text	Id des Kunden
orderBy	Text	nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
byDescending	Boolean	wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
newerThan	Datum	nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit). Format: JJJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
objects	Text,Text	Wenn angegeben, erlaubt zusätzlich das Auflisten von Briefen der Verträgen und oder Schäden des Kunden Kunde,Vertrag,Schaden, leer -> Kunden
onlyUnread	Boolean	Gibt nur die ungelesene Briefe zurück, true oder false, leer -> false
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
fields	Text	Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Erstellt eine Download URL zu diesem Brief
 - POST /api/v1/Ams/Kunde/{id}/Brief/{briefId}/Download
 - Als Parameter z.B.

Id	Text	Id des Kunden
briefId	Text	Id des Briefes
showInline	Boolean	True, um zu versuchen, den Inhalt im Browser darzustellen; False, um das Speichern des Downloads zu erzwingen
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
- Alle meine Verträge abfragen. ACHTUNG: Es sollten nur beim Projektstart einmal alle Verträge abgerufen werden.
 Anschließend sollten möglichst nur Differenzabfragen verwendet werden. Wenn man z.B. täglich Nachts die Differenzabfragen macht, kann jeweils nach den Änderungen seit der vorigen Abfrage angefragt werden: Parameter OrderBy und NewerThan müssen dazu übergeben werden
 - GET /api/v1/Ams/Vertrag
 - Als Parameter z.B.

orderBy	Text	nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
byDescending	Boolean	wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
newerThan	Datum	nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit). Format: JJJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
skip	Zahl	Wie viele Verträge aus der sortierten Liste sollen für die Ausgabe übersprungen werden? Wird zum Paging/Blättern benötigt (z.B. erste 50 Verträge, Zweite 50 Verträge, ...) leer -> 0 = keinen überspringen
take	Zahl	Wie viele Verträge aus der sortierten Liste sollen zurückgegeben werden? leer -> ersten 50 Verträge Maximal 5000 Datensätze gleichzeitig. Sonst mit Paging über Skip=5000 die nächsten holen „take=5000“ holt die ersten 5000 Datensätze ab „take=5000&skip=5000“ holt die nächsten 5000 Datensätze ab „take=5000&skip=10000“ holt die nachfolgenden Datensätze ab u.s.w.
sparte	Text	Gibt optional eine Sparte an, auf die gefiltert wird (z.B. KFZ), leer -> alle
status	Text	Gibt optional einen VertragStatus an, auf die gefiltert wird (z.B. Eigen,Anfrage,Angebot,Fremd, Fremdteil,Gesperrt,Info,Ruhend,Storno,Antrag,Police), leer -> alle. Feld: ComputedStatus. Dabei wird EIGEN aufgeteilt: mit Dokument: POLICE, mit Antragsdatum: ANTRAG, sonst EIGEN
ablaufNewerThan	Datum	Gibt optional ein Ablaufdatum an, um Verträge mit Ablaufdatum kleiner als diesem Wert zu filtern, Format: JJJJ-MM-DD
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
fields	Text	Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle



- Abfrage mitgeschnitten mit fields-Parameter:

- REQUEST:

GET /api/v1/Ams/Vertrag?fields=AmsIdNr%2CVertragsbeginn%2CVertragsablauf%2CId%2CSparte%2CVertragsNr HTTP/1.1

- RESPONSE:

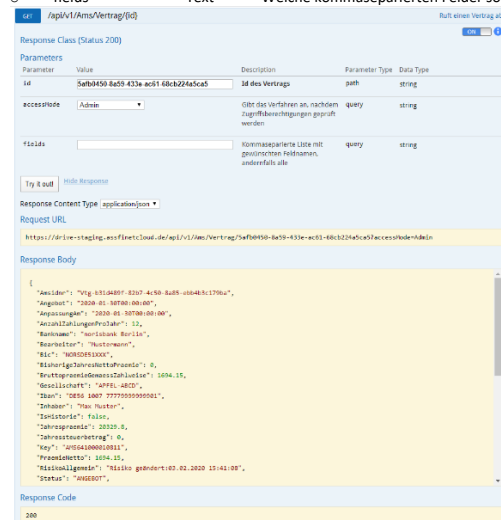
```
[{"AmsIdnr": "VIAS_02589", "Vertragsablauf": "2020-02-02T00:00:00", "Vertragsbeginn": "2019-02-01T00:00:00", "VertragsNr": "VSN-123", "Id": "f80c6214-60af-464d-a8a1-31c340eb5cdb", "Sparte": "Gebäudeversicherung"}, {"AmsIdnr": "Vet-15659382068880", "Vertragsablauf": "2020-08-01T00:00:00", "Vertragsbeginn": "2019-08-01T00:00:00", "VertragsNr": "AXA123-547/11.001", "Id": "5b629b84-5ac3-4dad-958e-f390e7a6721b", "Sparte": "Elektronikversicherung"}, {"AmsIdnr": "Vet-15657887513040", "Vertragsablauf": "2019-08-14T00:00:00", "Vertragsbeginn": "2019-08-14T00:00:00", "VertragsNr": "T012345", "Id": "adc513a8-03e4-4026-97d6-09cc8cfa9a4", "Sparte": "Gebäudeversicherung"}, {"AmsIdnr": "VIAS_02582", "VertragsNr": "47-11.123 3", "Id": "2fe87798-59d0-4b9a-993b-f9e0156687de", "Sparte": "Hausrat-Versicherung"}, {"AmsIdnr": "VIAS_02581", "VertragsNr": "47-11-123 2", "Id": "2c52e851-8fa6-4488-a6c0-f094ff85337e", "Sparte": "Glasversicherung"}, {"AmsIdnr": "VIAS_02202", "Vertragsablauf": "2014-01-01T00:00:00", "Vertragsbeginn": "2012-01-01T00:00:00", "VertragsNr": "20131120183017", "Id": "6c34d89f-0729-4435-8ed0-004cb72a1666", "Sparte": "KFZ-Versicherung"}]
```

- einen bestimmten Vertrag abfragen

wenn man die Vertrags-ID schon kennt (z.B. über GetSearchResults), kann der Vertrag direkt abgefragt werden:

- GET /api/v1/Ams/Vertrag/{id}
- Es werden nur die gefüllten Felder zurück gegeben
- bei ?accessMode=Admin werden sämtliche Felder zurückgegeben, welche für ein Update benötigt werden
- Schlagwörter zum Vertrag werden in Feld-Schlagwoerter gespeichert (| -getrennt)
- Als Parameter z.B.

- Id Text Id des Vertrages
- accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
- fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle



- Listet die Termine/Aufgaben/Vorgänge/Wiedervorlagen eines Vertrages

- GET /api/v1/Ams/Vertrag/{id}/Termin

- Als Parameter z.B.

- Id Text Id des Vertrages
- accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
- kategorie Text Art des Termines: 'Aufgabe', 'Termin', 'Vorgang', 'Wiedervorlage', 'Kontaktnotiz'
- fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle

- zu einem bestimmten Vertrag die Spartendaten abfragen
Die Spartendaten zu einem Vertrag können über eine weitere Abfrage ermittelt werden. Je nach Spartendefinitionen stehen die gesuchten Sparten-Felder in den durchnummerierten API-Feldern.
 - GET /api/v1/Ams/Vertrag/{id}/Sparte?accessMode=Admin"; //mit ADMIN alle Felder lesen
 - Als Parameter z.B.
 - Id Text Id des Vertrages
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle

GET /api/v1/Ams/Vertrag/{id}/Sparte

Response Class (Status 200)

Parameters

Parameter	Value	Description	Parameter Type	Data Type
Id	5ef0b459-8a59-433e-ac11-68cb22465ca5	Die Id des Vertrages	path	string
accessMode	Admin	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden	query	string
fields		Kommaseparierte Liste mit gewünschten Feldern; andersfalls alle	query	string

Request URL

https://drive-staging.asf.techcloud.de/api/v1/Ams/Vertrag/5ef0b459-8a59-433e-ac11-68cb22465ca5?accessMode=Admin

Response Body

```
{
  "Id": "5ef0b459-8a59-433e-ac11-68cb22465ca5",
  "bearbeitet": "2009-02-04T14:17:09+01:00",
  "lastSync": "2009-02-04T14:17:09+01:00",
  "key": "ams-0000000028",
  "type": "PKV",
  "group": "Kfz",
  "license": "G0010_1",
  "lastSynchronisation": "2009-02-19T09:25:15-2913887",
  "PKV001": "nummer_Kfz",
  "PKV002": "0",
  "PKV003": "1973-05-23T00:00:00",
  "PKV004": "0",
  "PKV005": "0",
  "PKV006": "0",
  "PKV007": "0",
  "PKV008": "0",
  "PKV009": "0",
  "PKV010": "0",
  "PKV011": "0",
  "PKV012": "0"
}
```

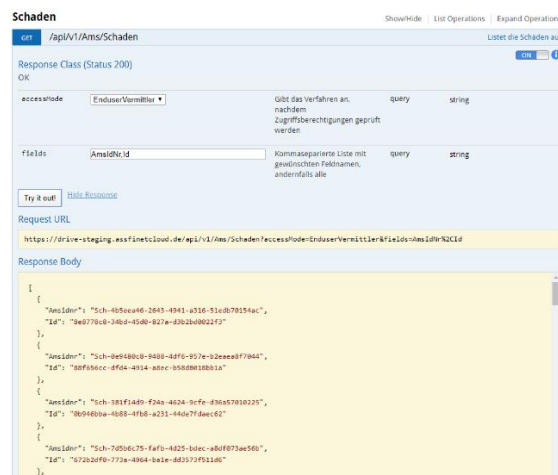
Response Code

200

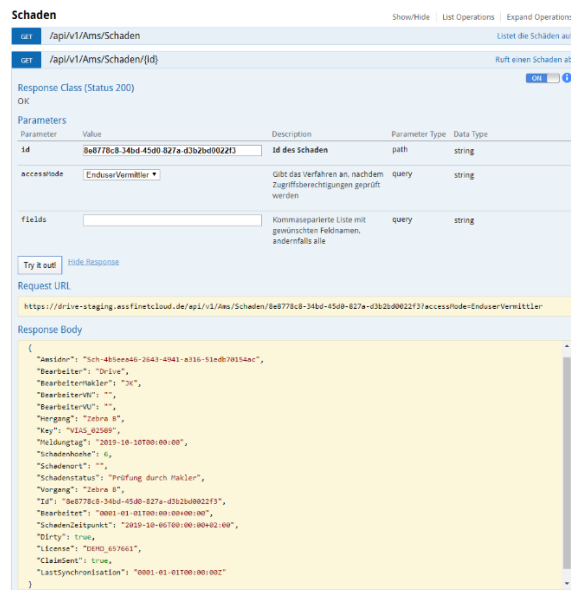
- zu allen Verträgen die Spartendaten abfragen, auch z.B. alle Spartendaten bei denen nach Datum+Uhrzeit-XXX der Datensatz geändert wurde. Je nach Spartendefinitionen stehen die gesuchten Sparten-Felder in den durchnummerierten API-Feldern.
 - GET /api/v1/Ams/Vertrag/Sparte?accessMode=Admin"; //mit ADMIN alle Felder lesen
 - Als Parameter z.B.
 - orderBy Text nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
 - byDescending Boolean wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
 - newerThan Datum nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit).
Format: JJJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
 - skip Zahl Wie viele Kunden aus der sortierten Liste sollen für die Ausgabe übersprungen werden? Wird zum Paging/Blättern benötigt (z.B. erste 50 Kunden, Zweite 50 Kunden, ...), leer -> 0 = keinen überspringen
 - take Zahl Wie viele Kunden aus der sortierten Liste sollen zurückgegeben werden? leer -> ersten 50 Kunden
Maximal 5000 Datensätze gleichzeitig. Sonst mit Paging über Skip=5000 die nächsten holen
„take=5000“ holt die ersten 5000 Datensätze ab
„take=5000&skip=5000“ holt die nächsten 5000 Datensätze ab
„take=5000&skip=10000“ holt die nachfolgenden Datensätze ab u.s.w.
 - sparte Text Gibt optional eine Sparte an, auf die gefiltert wird (z.B. KFZ), leer -> alle
 - ablaufNewerThan Datum Gibt optional ein Ablaufdatum an, um Verträge mit Ablaufdatum kleiner als diesem Wert zu filtern, Format: JJJJ-MM-DD
 - status Text Gibt optional einen Vertragsstatus an, auf die gefiltert wird (z.B. Eigen, Fremd, Anfrage, Angebot, Fremdeitel, Gesperrt, Info, Ruhend, Storno, Antrag, Police). leer -> alle. Feld: ComputedStatus. Dabei wird EIGEN aufgeteilt: mit Dokument: POLICE, mit Antragsdatum: ANTRAG, sonst EIGEN.
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Listet die Briefe eines Vertrages
 - GET /api/v1/Ams/Vertrag/{id}/Brief
 - Als Parameter z.B.
 - Id Text Id des Vertrages
 - orderBy Text nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
 - byDescending Boolean wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
 - newerThan Datum nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit).
Format: JJJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Ruft die Referenzadresse eines Vertrages ab (z.B. Hausmeister/Geschäftsführer/Rechnungsanschrift)
 - GET /api/v1/Ams/Vertrag/{id}/ReferenzAdresse
 - Als Parameter z.B.
 - Id Text Id des Vertrages
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - Verwendung Text Filter auf die Verwendung der LinkAdressen, mehrere Kommagetrennt angeben
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle

- alle meine Schäden abfragen. ACHTUNG: Es sollten nur beim Projektstart einmal alle Schäden abgerufen werden. Anschließend sollten möglichst nur Differenzabfragen verwendet werden. Wenn man z.B. täglich Nachts die Differenzabfragen macht, kann jeweils nach den Änderungen seit der vorigen Abfrage angefragt werden: Parameter OrderBy und NewerThan müssen dazu übergeben werden

- GET /api/v1/Ams/Schaden
- Als Parameter z.B.
 - orderBy Text nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
 - byDescending Boolean wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
 - newerThan Datum nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit).
Format: JJJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
Der Zeitpunkt, vor dem Schäden nach dem OrderBy Kriterium zurückgegeben werden sollen.
Format: JJJJ-MM-DD
 - olderThan Datum Wie viele Schäden aus der sortierten Liste sollen für die Ausgabe übersprungen werden? Wird zum Paging/Blättern benötigt (z.B. erste 50 Schäden, zweite 50 Schäden, ...), leer -> 0 = keinen überspringen
 - skip Zahl Wie viele Schäden aus der sortierten Liste sollen zurückgegeben werden? leer -> ersten 50 Schäden
 - take Zahl Maximal 5000 Datensätze gleichzeitig. Sonst mit Paging über Skip=5000 die nächsten holen
„take=5000“ holt die ersten 5000 Datensätze ab
„take=5000&skip=5000“ holt die nächsten 5000 Datensätze ab
„take=5000&skip=10000“ holt die nachfolgenden Datensätze ab u.s.w.
 - sparte Text Gibt optional eine Sparte an, auf die gefiltert wird (z.B. KFZ), leer -> alle
 - today Datum Wenn "pending" angegeben ist, gibt dies den Betrachtungstag an. Format: JJJJ-MM-DD
 - pending Boolean Gibt optional an, ob auf ausstehende Schäden (true) oder abgeschlossene Schäden (false) gefiltert wird
 - meldungsJahr Zahl Gibt optional eine Jahr als Filter an, in dem der Schaden gemeldet wurde
 - ablaufNewerThan Datum Gibt optional an, ob Schäden an Verträgen mit Ablaufdatum älter als diesen Wert, gefiltert werden, Format: JJJJ-MM-DD
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden.
Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle



- einen bestimmten Schaden abfragen
 - GET /api/v1/Ams/Schaden/{id}
 - Es werden nur die gefüllten Felder zurück gegeben
 - Schlagwörter zum Schaden werden in Feld-Schlagwoerter gespeichert (| -getrennt)
 - Als Parameter z.B.
 - Id Text Id des Schaden
 - accessMode Text Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden.
Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
 - fields Text Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle



- Listet die Termine/Aufgaben/Vorgänge/Wiedervorlagen eines Schadens
 - GET `/api/v1/Ams/Schaden/{id}/Termin`
 - Als Parameter z.B.

Parameter	Typ	Beschreibung
Id	Text	Id des Schadens
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
kategorie	Text	Art des Termins: 'Aufgabe', 'Termin', 'Vorgang', 'Wiedervorlage', 'Kontaktnotiz'
fields	Text	Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- Listet die Briefe eines Schades
 - GET `/api/v1/Ams/Schaden/{id}/Brief`
 - Als Parameter z.B.

Parameter	Typ	Beschreibung
Id	Text	Id des Schadens
orderBy	Text	nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
byDescending	Boolean	wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
newerThan	Datum	nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit). Format: JJJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
fields	Text	Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
- alle meine Referenzadressen abfragen. ACHTUNG: Es sollten nur beim Projektstart einmal alle Referenzadressen abgerufen werden. Anschließend sollten möglichst nur Differenzabfragen verwendet werden. Wenn man z.B. täglich Nachts die Differenzabfragen macht, kann jeweils nach den Änderungen seit der vorigen Abfrage angefragt werden: Parameter OrderBy und NewerThan müssen dazu übergeben werden geändert wurde
 - GET `/api/v1/Ams/Adresse`
 - Als Parameter z.B.

Parameter	Typ	Beschreibung
orderBy	Text	nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
byDescending	Boolean	wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
newerThan	Datum	nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit). Format: JJJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
skip	Zahl	Wie viele Schäden aus der sortierten Liste sollen für die Ausgabe übersprungen werden? Wird zum Paging/Blättern benötigt (z.B. erste 50 Schäden, Zweite 50 Schäden, ...), leer -> 0 = keinen überspringen
take	Zahl	Wie viele Datensätze aus der sortierten Liste sollen zurückgegeben werden? leer -> ersten 50 Datensätze Maximal 5000 Datensätze gleichzeitig. Sonst mit Paging über Skip=5000 die nächsten holen „take=5000“ holt die ersten 5000 Datensätze ab „take=5000&skip=5000“ holt die nächsten 5000 Datensätze ab „take=5000&skip=10000“ holt die nachfolgenden Datensätze ab u.s.w.
accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
fields	Text	Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle
filters	Text	Filter die auf die zurückzugebenden Objekte anzuwenden sind
verwendung	Text	Filter auf die Verwendung der LinkAdressen, mehrere Kommagetrennt angeben

In RefVerwendung wird eine Liste zurück gegeben, wofür diese Referenzadresse genutzt wird (z.B. Geschwister, Ehepartner).

In RefKeyArt wird eine Liste zurückgegeben, wozu diese Referenzadresse gehört (z.B. Kunde, Vertrag, Schaden, Gesell)

- alle meine Briefe abfragen. ACHTUNG: Es sollten nur beim Projektstart einmal alle Briefe abgerufen werden. Anschließend sollten möglichst nur Differenzabfragen verwendet werden. Wenn man z.B. täglich Nachts die Differenzabfragen macht, kann jeweils nach den Änderungen seit der vorigen Abfrage angefragt werden: Parameter OrderBy und NewerThan müssen dazu übergeben werden
Die Zuordnung der Briefe kann anschließend über `api/v1/Ams/Brief/{id}/Principals` ermittelt werden (Kunde/Vertrag/Schaden/Vermittler/Termin/Adresse)

- GET /api/v1/Ams/Brief
- Als Parameter z.B.

○ orderBy	Text	nach welchem Feld werden die Ergebnisse sortiert? Feldname, leer -> Id
○ byDescending	Boolean	wird aufsteigend oder absteigend sortiert? true oder false, leer -> true = absteigend
○ newerThan	Datum	nur dann nach neuem Datum gegen 'orderBy' gefiltert, falls bei 'orderBy' übergeben wurde 'LastSynchronisation' oder 'Bearbeitet' (z.B. größer Feld Bearbeitet in Weltzeit). Format: JJJ-MM-DDThh:mm:ss bezogen auf Weltzeit. Sonst ist dieser Parameter egal.
○ skip	Zahl	Wie viele Schäden aus der sortierten Liste sollen für die Ausgabe übersprungen werden? Wird zum Paging/Blättern benötigt (z.B. erste 50 Schäden, Zweite 50 Schäden, ...), leer -> 0 = keinen überspringen
○ take	Zahl	Wie viele Schäden aus der sortierten Liste sollen zurückgegeben werden? leer -> ersten 50 Schäden Maximal 5000 Datensätze gleichzeitig. Sonst mit Paging über Skip=5000 die nächsten holen „take=5000“ holt die ersten 5000 Datensätze ab „take=5000&skip=5000“ holt die nächsten 5000 Datensätze ab „take=5000&skip=10000“ holt die nachfolgenden Datensätze ab u.s.w.
○ folderFilter	Text	Gibt nur Briefe aus diesen Ordnern zurück. Syntax: "%Angebot%" or "Police%"
○ onlyUnread	Boolean	Gibt nur die ungelesenen Briefe zurück
○ accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler
○ fields	Text	Welche kommaseparierten Felder sollen zurückgegeben werden? leer -> alle

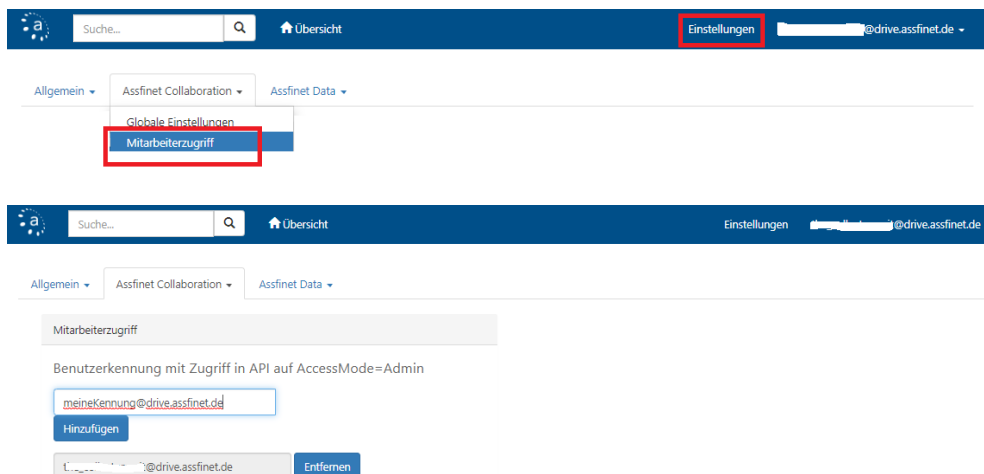
Die Zuordnung (Kunde/Vertrag/Schaden/ Vermittler/Termin/Adresse) der Briefe kann über Principals ermittelt werden

- GET /api/v1/Ams/ Brief/{id}/Principals
- Als Parameter z.B.

○ id	Text	Id des Briefes
○ accessMode	Text	Gibt das Verfahren an, nach dem Zugriffsberechtigungen geprüft werden. Enduser/Vermittler/EnduserVermittler/Admin, leer -> EnduserVermittler

API-Mitarbeiterzugriff über AccessMode

- bei Lesezugriffen auf Kunde/Vertrag/Vermittler werden nur dann sämtliche Felder zurück gegeben, wenn der Zugriff mit dem Parameter `accessMode=Admin` gemacht wird
- für Änderungen an Kunden-/Vertrag-Daten sollten diese zuvor mit `accessMode=Admin` gelesen worden sein, damit der vollständige Datensatz gespeichert wird. Sonst werden die fehlenden Felder vom Programm ergänzt. Der Weg über Adminzugriff wird dabei aber empfohlen. Der Admin-Zugriff benötigt keine „Freischaltungen“
- API-Admin-Erlaubnis kann für die eigene Lizenz eingestellt werden in der Collaboration-Administration:
 - Mit „Hinzufügen“ kann die darüber eingegebene Benutzerkennung für den API-Admin-Zugriff gespeichert werden
 - Mit „Entfernen“ wird die daneben stehende Benutzerkennung wieder für den API-Admin-Zugriff entfernt



Programmbeispiel

- für die API wird über OAuth2 ein BearerToken ermittelt über ClientId+ClientSecret und Benutzername+Passwort. Dieses Token ist einige Minuten gültig (meist 20 Minuten), und sollte in der Zeit wiederverwendet werden. Wenn es ausgelaufen ist, so wird über RefreshToken ein neues geholt.
- ein Beispiel in C#

```
static string bearerToken;
static DateTime bearerExpiresTimeUTC;
static string refreshToken;

public static async Task ApiKundeLesen()
{
    HttpClient httpClient = new HttpClient();
    Uri baseUriAuth, baseUriApi;
    string userName, password, license;
    GetServerUrl(out baseUriAuth, out baseUriApi, out userName, out password, out license); // meine Servereinstellungen
    (bearerToken, bearerExpiresTimeUTC, refreshToken) = await GetBearerToken(httpClient, baseUriAuth, userName, password, bearerToken, bearerExpiresTimeUTC, refreshToken); // Bearer-Token ermitteln, und Ablaufzeit des BearerToken, und RefreshToken
    string apiPath = "api/v1/Ams/Kunde"; //alle meine freigegebenen Kunden abfragen
    var requestData = new HttpRequestMessage
    {
        Method = HttpMethod.Get,
        RequestUri = new Uri(baseUriApi, apiPath),
    };
    requestData.Headers.TryAddWithoutValidation("Authorization", String.Format("Bearer {0}", bearerToken));

    var results = await httpClient.SendAsync(requestData);
    var apiErgebnis = await results.Content.ReadAsStringAsync(); //Ergebnis der API-Abfrage erhalten
}
```

```

        if (resulta.StatusCode != HttpStatusCode.OK)
            throw new Exception("Fehler: " + apiErgebnis);
        var myKundenList = Newtonsoft.Json.JsonConvert.DeserializeObject<List<KundeModel>>(apiErgebnis);

        var myKundenListId = myKundenList.Select(x => x.Id).ToList();
    }

    // meine Server-Einstellungen
    public static void GetServerUrl(out Uri baseUriAuth, out Uri baseUriApi, out string userName, out string passwort, out string license)
    {
        license = "meineLizenz 999999"; // hier Ihre Lizenznummer eintragen siehe api/v1/User/Licenses
        baseUriAuth = new Uri("https://account-staging.assfinet.de/"); baseUriApi = new Uri("https://drive-staging.assfinetcloud.de/"); //STAGING
        userName = "IhrBenutzername"; passwort = "IhrPasswort"; //Ihr Username+Password

        // baseUriAuth = new Uri("https://account-uat.assfinetcloud.de/"); baseUriApi = new Uri("https://drive-uat.assfinetcloud.de/"); //UAT
        //userName = "IhrBenutzername"; passwort = "IhrPasswort"; //Ihr Username+Password

        // baseUriAuth = new Uri("https://account-demo.assfinetcloud.de/"); baseUriApi = new Uri("https://drive-demo.assfinetcloud.de/"); //DEMO
        //userName = "IhrBenutzername"; passwort = "IhrPasswort"; //Ihr Username+Password

        //baseUriAuth = new Uri("https://account.assfinetcloud.de/"); baseUriApi = new Uri("https://drive-azure.assfinet.de/"); //PROD
        //userName = "IhrBenutzername"; passwort = "IhrPasswort"; //Ihr Username+Password
    }

    // Bearer-Token und Ablaufzeit des BearerToken und RefreshToken als Rückgabewerte zurückgeben
    public static async Task<(string, DateTime, string)> GetBearerToken(HttpClient httpClient, Uri baseUriAuth, string userName, string passwort, string bearerToken, DateTime
    bearerExpiresTimeUTC, string refreshToken)
    {
        var restGueiltig = bearerExpiresTimeUTC - DateTime.UtcNow; //wenn das BearerToken nicht mehr lange gültig ist, dann über RefreshToken erneuern
        if (!String.IsNullOrEmpty(bearerToken) && !String.IsNullOrEmpty(refreshToken) && restGueiltig.TotalSeconds > 20)
        {
            //BearerToken ist noch gültig, also nehme es
        }
        else
        {
            if (!String.IsNullOrEmpty(bearerToken) && !String.IsNullOrEmpty(refreshToken))
            {
                //über RefreshToken ein neues BearerToken holen
                string grantStr = String.Format("grant_type=refresh_token&refresh_token={0}&scope=%20", Uri.EscapeDataString(refreshToken));
                (bearerToken, bearerExpiresTimeUTC, refreshToken) = await GetBearerTokenCall(httpClient, baseUriAuth, grantStr);
            }
            if (String.IsNullOrEmpty(bearerToken) || String.IsNullOrEmpty(refreshToken))
            {
                //neues BearerToken+RefreshToken ermitteln
                string grantStr = String.Format("grant_type=password&username={0}&password={1}&scope=%20", Uri.EscapeDataString(userName),
                Uri.EscapeDataString(passwort));
                (bearerToken, bearerExpiresTimeUTC, refreshToken) = await GetBearerTokenCall(httpClient, baseUriAuth, grantStr);
            }
        }
        return (bearerToken, bearerExpiresTimeUTC, refreshToken);
    }

    public static async Task<(string, DateTime, string)> GetBearerTokenCall(HttpClient httpClient, Uri baseUriAuth, string grantStr)
    {
        var meineClientId = "ihreClientId"; var meineClientIdSecret = "ihreClientSecret"; //Ihre ClientId und ClientIdSecret für Api
        var encodedPair = System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetBytes(String.Format("{0}:{1}", meineClientId, meineClientIdSecret)));

        var requestToken = new HttpRequestMessage
        {
            Method = HttpMethod.Post,
            RequestUri = new Uri(baseUriAuth, "oauth/token"),
            Content = new StringContent(grantStr)
        };

        requestToken.Content.Headers.ContentType = new MediaTypeWithQualityHeaderValue("application/x-www-form-urlencoded") { CharSet = "UTF-8" };
        requestToken.Headers.TryAddWithoutValidation("Authorization", String.Format("Basic {0}", encodedPair));

        var bearerResult = await httpClient.SendAsync(requestToken);
        if (bearerResult.StatusCode != HttpStatusCode.OK)
        {
            bearerToken = null; //wenn Fehler (z.B. RefreshToken abgelaufen), dann hier zurücksetzen damit anschließend ein neues ermittelt wird
            bearerExpiresTimeUTC = DateTime.MinValue;
            refreshToken = null;
        }
        else
        {
            var bearerData = await bearerResult.Content.ReadAsStringAsync();
            bearerToken = JObject.Parse(bearerData)["access_token"].ToString(); //Bearer-Token erhalten
            var expiresin = int.Parse(JObject.Parse(bearerData)["expires_in"].ToString()); //Bearer-Token läuft in wieviel Sekunden ab
            bearerExpiresTimeUTC = DateTime.UtcNow.AddSeconds(expiresin);
            refreshToken = JObject.Parse(bearerData)["refresh_token"].ToString(); //Refresh-Token erhalten
        }
        return (bearerToken, bearerExpiresTimeUTC, refreshToken);
    }
}

```

- **Klasse-KundeModel**

```

using System;
using System.Collections.Generic;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace ApiTest
{
    public partial class KundeModel
    {
        public Guid Id { get; set; }
        public DateTime? Bearbeitet { get; set; }
        /// <summary>
        /// Gibt an ob dieser Datensatz lokal geändert und noch nicht von AMS bestätigt wurde
        /// </summary>
        public bool Dirty { get; set; }
        /// <summary>
        /// Die Lizenz der der Kunde angehört
        /// </summary>
        public string License { get; set; }
        public DateTime LastSynchronisation { get; private set; }
        [JsonExtensionData]
        public Dictionary<string, JToken> AdditionalData { get; set; }
        public string Adrid { get; set; }
        public DateTime? Aenderung { get; set; }
        public string Aenderungsgrund { get; set; }
        public string Amsidnr { get; set; } // Kwangsfeld
        public string Anrede { get; set; }
        public string Arbeitgeber { get; set; }
        public decimal? AusbildungskostenMonatlich { get; set; }
        public DateTime? AustrittsdatumArbeitgeber { get; set; }
        public string Bank1 { get; set; }
        public string Bank2 { get; set; }
        public string Bearbeiter { get; set; }
        public string BelegteSparten { get; set; }
        public string Beruf { get; set; }
        public string Berufstatus { get; set; }
        public string Besitzverhältnis { get; set; }
        public string Bic1 { get; set; }
        public string Bic2 { get; set; }
        public string Bilanzstichtag { get; set; }
        public string Bic1 { get; set; }
        public string Bic2 { get; set; }
        public string Branche { get; set; }
        public decimal? Bruttojahresmietwert { get; set; }
        public decimal? Bruttojahreumsatz { get; set; }
        public string Bundesland { get; set; }
        public string Debitor { get; set; }
        public int? Einkommen { get; set; }
        public DateTime? EintrittsdatumArbeitgeber { get; set; }
        public string EmailGeschäftlich { get; set; }
        public string EmailPrivat { get; set; }
        public DateTime? Erldwg7 { get; set; }
        public decimal? ErwerbsminderungsrenteTeil { get; set; }
        public decimal? ErwerbsminderungsrenteVoll { get; set; }
        public string Familienstand { get; set; }
        public string Fax { get; set; }
        public string Finanzamt { get; set; }
        public string Geburtsname { get; set; }
        public string Geburtsort { get; set; }
        public DateTime? Geburtstag { get; set; }
        public string Geschlecht { get; set; }
        public decimal? GesetzlicheAltersrenteMonatlich { get; set; }
        public decimal? GesetzlicheBUrenteMonatlich { get; set; }
        public decimal? GesetzlicheHinterbliebenenRenteMonatlich { get; set; }
        public bool? GesetzlichRenteversicherung { get; set; }
        public bool? GesetzlichRentenversicherung { get; set; }
        public DateTime? GewunschterRentenbeginn { get; set; }
    }
}

```

```

        public DateTime? Gruendung { get; set; }
        public decimal? Grundsueck { get; set; }
        public DateTime? GueltigAb { get; set; }
        public DateTime? GueltigBis { get; set; }
        public string Handelsregisternummer { get; set; }
        public bool? Hinterbliebenensente { get; set; }
        public string HistorieZuKunde { get; set; }
        public string Iban1 { get; set; }
        public string Iban2 { get; set; }
        public decimal? Inflationsrate { get; set; }
        public string Info_01 { get; set; }
        public string InhaberGF { get; set; }
        public string InternetNr { get; set; }
        public string Internet { get; set; }
        public bool? IsHistorie { get; set; } // Ewangsfield, false
        public bool? IsZukunft { get; set; }
        public decimal? Kapitalmarktzins { get; set; }
        public int? Kinder { get; set; }
        public string Kindergeldnummer { get; set; }
        public decimal? KinderLautSteuerkarte { get; set; }
        public bool? Kirchensteuer { get; set; }
        public string Klasse { get; set; }
        public string Kontol { get; set; }
        public string Konto2 { get; set; }
        public string Kontobezeichnung1 { get; set; }
        public string Kontobezeichnung2 { get; set; }
        public string Kontoinhaber1 { get; set; }
        public string Kontoinhaber2 { get; set; }
        public decimal? Kundeninkassokonto { get; set; } // Ewangsfield
        public string Kundenname { get; set; }
        public string KundennamePhonetisch { get; set; }
        public string Kundenverbindung { get; set; }
        public DateTime? KundeSeit { get; set; }
        public string LagerBetriebsgrundtuecks { get; set; }
        public string Land { get; set; }
        public DateTime? LetzterBesuch { get; set; }
        public decimal? LohnGehaltsSumme { get; set; }
        public DateTime? MahntoppBis { get; set; }
        public DateTime? MaklervertragBis { get; set; }
        public DateTime? MaklervertragDatum { get; set; }
        public string Mandant { get; set; } // Freigabe für welchen Vermittler? Feld "Mandant" in api/v1/Ams/Vermittler
        public string MandantAusdrung { get; set; } // Freigabe für welchen Vermittler? Feld "MandantAmsidnr" in api/v1/Ams/Vermittler
        public decimal? MietenNebenkostenMonatlich { get; set; }
        public decimal? Mietsteigerung { get; set; }
        public int? Mitarbeiterzahl { get; set; }
        public string Mobiltelefon { get; set; }
        public string Name { get; set; } // Ewangsfield
        public string Name1 { get; set; }
        public string Name2 { get; set; }
        public string Name3 { get; set; }
        public decimal? NichtSelbstaendigeArbeitBruttoMonatlich { get; set; }
        public decimal? NichtSelbstaendigeArbeitNettoMonatlich { get; set; }
        public string Ort { get; set; }
        public string OrtPhonetisch { get; set; }
        public string Partner { get; set; }
        public DateTime? PersonalausweisAusstellungsdatum { get; set; }
        public string PersonalausweisBehoerde { get; set; }
        public DateTime? PersonalausweisGueltigBis { get; set; }
        public string PersonalausweisStr { get; set; }
        public string Personalnummer { get; set; }
        public string Plz { get; set; }
        public DateTime? PraemienrechnungAb { get; set; }
        public DateTime? PraemienrechnungBis { get; set; }
        public int? Provision { get; set; }
        /// <summary>
        /// Rechnungsversand via Mail ausschließen?
        /// </summary>
        public bool? RechnungsMail { get; set; }
        public string Rechtsform { get; set; }
        public string Repraesentant { get; set; }
        public string S21_F1 { get; set; }
        public string S21_F10 { get; set; }
        public string S21_F2 { get; set; }
        public string S21_F3 { get; set; }
        public string S21_F4 { get; set; }
        public string S21_F5 { get; set; }
        public string S21_F6 { get; set; }
        public string S21_F7 { get; set; }
        public string S21_F8 { get; set; }
        public string S21_F9 { get; set; }
        public string S22_F1 { get; set; }
        public string S22_F10 { get; set; }
        public string S22_F2 { get; set; }
        public string S22_F3 { get; set; }
        public string S22_F4 { get; set; }
        public string S22_F5 { get; set; }
        public string S22_F6 { get; set; }
        public string S22_F7 { get; set; }
        public string S22_F8 { get; set; }
        public string S22_F9 { get; set; } //KUNDE hat keine Schlagwörter/TAGS, sondern seine ADRESSE
        public decimal? SelbstaendigeArbeitNettoMonatlich { get; set; }
        public DateTime? Sepamandat { get; set; }
        public string Sepamandatarferenz { get; set; }
        public decimal? SonstigeEinkuenfteMonatlich { get; set; }
        public decimal? SonstigeEinkuenfteNettoMonatlich { get; set; }
        public decimal? SonstigeLaufendeKostenMonatlich { get; set; }
        public string Sozialversicherungsnummer { get; set; }
        public string Staatsangehoerigkeit { get; set; }
        public string Status { get; set; }
        public string SteuerIdentifikationsnummer { get; set; }
        public string Steuerklasse { get; set; }
        public string Steuernummer { get; set; }
        public DateTime? StichtagBruttoJahresmietwert { get; set; }
        public DateTime? StichtagBruttoJahresumsatz { get; set; }
        public DateTime? StichtagLohnGehaltsSumme { get; set; }
        public DateTime? StichtagSubunternehmerAuftragswert { get; set; }
        public DateTime? StichtagSummeEigeneBauvorhaben { get; set; }
        public DateTime? StichtagWarenMaterialwertProJahr { get; set; }
        public string Strasse { get; set; }
        public string StrassePhonetisch { get; set; }
        public decimal? SubunternehmerAuftragswert { get; set; }
        public decimal? SummeEigeneBauvorhaben { get; set; }
        public DateTime? TaetigAufBetriebsgrundtueckSeit { get; set; }
        public string TelefonGeschaeflich { get; set; }
        public string TelefonPrivat { get; set; }
        public string Titel { get; set; }
        public bool? Trockenbett { get; set; }
        public bool? Ueberschwemmungsgebiet { get; set; }
        public bool? Unfriedet { get; set; }
        public int? Umsatz { get; set; }
        public string UmsatzJahr { get; set; }
        public decimal? UnterhaltszahlungenMonatlich { get; set; }
        public string USTRIMr { get; set; }
        public DateTime? Verstorben { get; set; }
        public string Vm1 { get; set; } // Freigabe für welchen Vermittler? Feld "Code" in api/v1/Ams/Vermittler
        public string Vm10 { get; set; }
        public string Vm2 { get; set; }
        public string Vm3 { get; set; }
        public string Vm4 { get; set; }
        public string Vm5 { get; set; }
        public string Vm6 { get; set; }
        public string Vm7 { get; set; }
        public string Vm8 { get; set; }
        public string Vm9 { get; set; }
        public string Vnummer { get; set; }
        public string Vorname { get; set; }
        public decimal? WarenMaterialwertProJahr { get; set; }
        public bool? Wasserschutzgebiet { get; set; }
        public string Zuordnung { get; set; } // Ewangsfield, 'PRIVAT', 'FIRMA'
        private HashSet<string> columns;
        public bool ShouldSerialize(string name) => columns?.Contains(name) ?? true;
    }
}

```

- Klasse-VertragModel

```

using System;
using System.Collections.Generic;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

```

```

namespace ApiTest
{

```

```

public partial class VertragModel
{
    public Guid Id { get; set; }
    public DateTime? Bearbeitet { get; set; }
    public string Sparte { get; set; }
    public string Details { get; set; }
    public string Vu { get; set; }
    public int VuNummer { get; set; }
    public string ComputedStatus { get; set; } //errechnetes Feld aus Status: dabei wird EIGEN aufgeteilt: mit Dokument: POLICE, mit Antragsdatum: ANTRAG, sonst EIGEN
    /// <summary>
    /// Gibt an ob dieser Datensatz lokal geändert und noch nicht von AMS bestätigt wurde
    /// </summary>
    public bool Dirty { get; set; }
    /// <summary>
    /// Die Lizenz der der Vertrag angehört
    /// </summary>
    public string License { get; set; }
    public DateTime LastSynchronisation { get; private set; }
    [JsonExtensionData]
    public IDictionary<string, JToken> AdditionalData { get; set; }
    public bool? Abbuchung { get; set; }
    public bool? AbrechnenAuerVertrag { get; set; } // Zwangsfield
    public bool? AbrechnenueberEinzelpositionen { get; set; } // Zwangsfield
    public string Abschlussvermittler { get; set; }
    public string AmsIdnr { get; set; }
    public DateTime? Angebot { get; set; } // Zwangsfield, meine eindeutige interne Vertragsnummer
    public DateTime? AnpassungAm { get; set; }
    public DateTime? Antrag { get; set; }
    public int? AnzahlZahlungenProJahr { get; set; }
    public int? AnzCourtzahlung { get; set; }
    public decimal? ApBetrag { get; set; }
    public DateTime? ApBuchungsdatum { get; set; }
    public string ApInfo { get; set; }
    public decimal? Appro1 { get; set; }
    public decimal? Appro10 { get; set; }
    public decimal? Appro2 { get; set; }
    public decimal? Appro3 { get; set; }
    public decimal? Appro4 { get; set; }
    public decimal? Appro5 { get; set; }
    public decimal? Appro6 { get; set; }
    public decimal? Appro7 { get; set; }
    public decimal? Appro8 { get; set; }
    public decimal? Appro9 { get; set; }
    public decimal? Apzv { get; set; }
    public decimal? ApZvpro1 { get; set; }
    public decimal? ApZvpro10 { get; set; }
    public decimal? ApZvpro2 { get; set; }
    public decimal? ApZvpro3 { get; set; }
    public decimal? ApZvpro4 { get; set; }
    public decimal? ApZvpro5 { get; set; }
    public decimal? ApZvpro6 { get; set; }
    public decimal? ApZvpro7 { get; set; }
    public decimal? ApZvpro8 { get; set; }
    public decimal? ApZvpro9 { get; set; }
    public string Bankname { get; set; }
    public string Bearbeiter { get; set; }
    public string Berater { get; set; }
    public bool? BeratungErforderlich { get; set; }
    public string Beratungsgebiet { get; set; }
    public string Beschreibung { get; set; }
    public DateTime? Bestaetigung { get; set; }
    public bool? Beteiligungsverhaeltnis { get; set; }
    public string Betreuungsvertragsnr { get; set; }
    public string Bic { get; set; }
    public decimal? BisherigeJahresNettopraemie { get; set; }
    public string B12 { get; set; }
    public decimal? Bonusbetrag { get; set; }
    public DateTime? Bonuseingang { get; set; }
    public decimal? BruttopraemieGemeassZahlweise { get; set; }
    public bool? BruttoVertrag { get; set; }
    public decimal? BuSumme { get; set; }
    public string Ccxid { get; set; }
    public string Ccximport { get; set; }
    public DateTime? CourtageAb { get; set; }
    public string CourtageSeignageGrosserFormel { get; set; }
    public DateTime? CourtageBis { get; set; }
    public decimal? CourtageFormelBetrag { get; set; }
    public decimal? CourtageProzent { get; set; }
    public string CourtageZahlweise { get; set; }
    public DateTime? DeckungBis { get; set; }
    public DateTime? DeckungVon { get; set; }
    public DateTime? Defzeit { get; set; }
    public DateTime? Dokument { get; set; }
    public decimal? Dyvm1 { get; set; }
    public decimal? Dyvm10 { get; set; }
    public decimal? Dyvm2 { get; set; }
    public decimal? Dyvm3 { get; set; }
    public decimal? Dyvm4 { get; set; }
    public decimal? Dyvm5 { get; set; }
    public decimal? Dyvm6 { get; set; }
    public decimal? Dyvm7 { get; set; }
    public decimal? Dyvm8 { get; set; }
    public decimal? Dyvm9 { get; set; }
    public decimal? EffektivCourtageProzent { get; set; }
    public decimal? EigeneAnteilMitversicherung { get; set; }
    public bool? EinzelpraemienErfassen { get; set; }
    public int? Erloeskonto { get; set; }
    public string Extid { get; set; }
    public DateTime? Faellig { get; set; }
    public DateTime? FestwertPraemieVom { get; set; }
    public decimal? GebuehrNetto { get; set; }
    public string GekuendigtVon { get; set; }
    public string Gesellschaft { get; set; } // Zwangsfield, Gesellschaft des Vertrages, muss vorkommen in Feld "Matchcode" in api/v1/Ams/Gesellschaft.
    // Must fit to table Gesellschaft with field Matchcode
    public string HaftungMakler { get; set; }
    public string HaftungVm1 { get; set; }
    public string HaftungVm10 { get; set; }
    public string HaftungVm2 { get; set; }
    public string HaftungVm3 { get; set; }
    public string HaftungVm4 { get; set; }
    public string HaftungVm5 { get; set; }
    public string HaftungVm6 { get; set; }
    public string HaftungVm7 { get; set; }
    public string HaftungVm8 { get; set; }
    public string HaftungVm9 { get; set; }
    public bool? HatFestwertBeteiligung { get; set; }
    public DateTime? Historiendatum { get; set; }
    public string Historiengrund { get; set; }
    public string HistorieZuVertrag { get; set; }
    public decimal? HonorarVm1 { get; set; }
    public decimal? HonorarVm10 { get; set; }
    public decimal? HonorarVm2 { get; set; }
    public decimal? HonorarVm3 { get; set; }
    public decimal? HonorarVm4 { get; set; }
    public decimal? HonorarVm5 { get; set; }
    public decimal? HonorarVm6 { get; set; }
    public decimal? HonorarVm7 { get; set; }
    public decimal? HonorarVm8 { get; set; }
    public decimal? HonorarVm9 { get; set; }
    public string Iban { get; set; }
    public string IndexanpassungAbrechnung { get; set; }
    public DateTime? IndexanpassungAm { get; set; }
    public decimal? IndexanpassungNjpAlt { get; set; }
    public decimal? IndexanpassungNjpNeu { get; set; }
    public string inhaber { get; set; }
    public bool? Inkasso { get; set; }
    public decimal? Inkassomaklergeb { get; set; }
    public bool? InkassoVm { get; set; }
    public string InterneNr { get; set; }
    public DateTime? Invitationsdatum { get; set; }
    public bool? IsHistorie { get; set; } // Zwangsfield
    public decimal? Jahrescourtage { get; set; }
    public decimal? Jahrespraemie { get; set; }
    public decimal? JahresueberBetrag { get; set; }
    public string Key { get; set; } // Zwangsfield, Vater - Datensatz, also zugehöriger Kunde.AmsIdnr
    public string KeyAngebot { get; set; }
    public string KeyAntrag { get; set; }
    public string KeyBestat { get; set; }
    public string KeyGekuend { get; set; }
    public string KeyNachtrag { get; set; }
    public string KeyPolice { get; set; }
    public string Kommentar { get; set; }
    public string Konto { get; set; }
    public string Kontobezeichnung { get; set; }
    public int? Kostenstelle { get; set; }
    public DateTime? Kuendigung { get; set; }
    public DateTime? LetzteAnfrage { get; set; }
}

```

```

    public DateTime? LetzteFaelligkeit { get; set; }
    public decimal? LetzteMettopraemie { get; set; }
    public decimal? LetzteProvision { get; set; }
    public DateTime? LetzterNachtrag { get; set; }
    public DateTime? MahnstoppBis { get; set; }
    public bool? MaklerPushorVerssteuerAb { get; set; }
    public string Matchcode { get; set; }
    public decimal? MindestJahresMettopraemie { get; set; }
    public string NachtragNr { get; set; }
    public decimal? Praxismetto { get; set; }
    public string Produkt { get; set; }
    public string Produktgruppe { get; set; }
    public string Produktgruppeid { get; set; }
    public DateTime? Provisionsreferenzdatum { get; set; }
    public decimal? Rabatt { get; set; }
    /// <summary>
    /// </summary>
    /// Rechnungsversand via Mail ausschließen?
    /// </summary>
    public bool? RechnungsMail { get; set; }
    public bool? RGDrukstop { get; set; }
    public string RisikoAllgemein { get; set; }
    public string RisikoOrt { get; set; }
    public string RisikoPlz { get; set; }
    public string RisikoStrasse { get; set; }
    public string Risikotraeger { get; set; }
    public string RVKumulnr { get; set; }
    public decimal? RVRelKumulsomme { get; set; }
    public decimal? RVRelSumme { get; set; }
    public string S2_01 { get; set; }
    public string S2_02 { get; set; }
    public string S2_03 { get; set; }
    public string S2_04 { get; set; }
    public string S2_05 { get; set; }
    public string S2_06 { get; set; }
    public string S2_07 { get; set; }
    public string S2_08 { get; set; }
    public string S2_09 { get; set; }
    public string S2_10 { get; set; }
    public string S2_11 { get; set; }
    public string S2_12 { get; set; }
    public string S2_13 { get; set; }
    public bool? Sammelvertrag { get; set; } // Zwangsfield
    public string Sammelvertragsnr { get; set; }
    public string Schlagwoerter { get; set; } // Schlagwörter/TAGS zum Vertrag
    public string SchwebeZuVertrag { get; set; }
    public DateTime? SEPAEinzugsmandat { get; set; }
    public string SEPA Mandatsreferenz { get; set; }
    public bool? Sicherungsschein { get; set; }
    public bool? SpartenberechnungAus { get; set; }
    public DateTime? SperdatumVUAbrechnung { get; set; }
    public string Status { get; set; } // Zwangsfield, Status des Vertrages,
    // "ANFRAGE", "ANGEBOT", "EIGEN", "FREMD", "FREMDTEIL", "GESPERRT", "RUHEND", "STORNO", "INFO"

    public string Statusgrund { get; set; }
    public decimal? Steuer { get; set; }
    public DateTime? Storno { get; set; }
    public decimal? Stornoges { get; set; }
    public int? Stornohaft { get; set; }
    public decimal? StornoVm1 { get; set; }
    public decimal? StornoVm10 { get; set; }
    public decimal? StornoVm2 { get; set; }
    public decimal? StornoVm3 { get; set; }
    public decimal? StornoVm4 { get; set; }
    public decimal? StornoVm5 { get; set; }
    public decimal? StornoVm6 { get; set; }
    public decimal? StornoVm7 { get; set; }
    public decimal? StornoVm8 { get; set; }
    public decimal? StornoVm9 { get; set; }
    public string Typ { get; set; } // Zwangsfield, Sparte des Vertrages, muss vorkommen in Tabelle-Sparte mit Feld-Sparte.
    // Feld "Spartenkuerzel" in api/v1/Ams/Sparte. Must fit to table sparte with field sparte

    public decimal? Unterjaehrlichkeitszuschlag { get; set; }
    public string Vermittlerkonto { get; set; }
    public DateTime? Vertragsablauf { get; set; } // Zwangsfield, wann ist Vertragsablauf
    public DateTime? Vertragsbeginn { get; set; } // Zwangsfield, wann ist Vertragsbeginn
    public string Vertragscheck { get; set; }
    public string VertragsNr { get; set; }
    public string VertragsnrAusdruck { get; set; }
    public string Vm1 { get; set; }
    public string Vm10 { get; set; }
    public string Vm2 { get; set; }
    public string Vm3 { get; set; }
    public string Vm4 { get; set; }
    public string Vm5 { get; set; }
    public string Vm6 { get; set; }
    public string Vm7 { get; set; }
    public string Vm8 { get; set; }
    public string Vm9 { get; set; }
    public decimal? Vmpro1 { get; set; }
    public decimal? Vmpro10 { get; set; }
    public decimal? Vmpro2 { get; set; }
    public decimal? Vmpro3 { get; set; }
    public decimal? Vmpro4 { get; set; }
    public decimal? Vmpro5 { get; set; }
    public decimal? Vmpro6 { get; set; }
    public decimal? Vmpro7 { get; set; }
    public decimal? Vmpro8 { get; set; }
    public decimal? Vmpro9 { get; set; }
    public int? VorschaedenBetrag { get; set; }
    public int? VorschaedenZahl { get; set; }
    public string VorvertragNr { get; set; }
    public string VorvertragVU { get; set; }
    public string Waehrung { get; set; }
    public DateTime? Wiedervorlage { get; set; }
    public string Zahlweise { get; set; }
    public int? Zeichnungsjahr { get; set; }
    public string Zuordnung { get; set; }
    private HashSet<string> columns;
    public bool ShouldSerialize(string name) => columns?.Contains(name) ?? true;
}
}

```

- Klasse-VertragSparteModel

```

public partial class VertragSparteModel
{
    public Guid Id { get; set; }
    public DateTimeOffset? Bearbeitet { get; set; }
    public string Amsidnr { get; set; }
    public string Key { get; set; } // Vater - Datensatz, also zugehöriger Vertrag.Amsidnr
    public string Typ { get; set; }
    [JsonExtensionData]
    public IDictionary<string, JToken> AdditionalData { get; set; }
    /// <summary>
    /// Gibt an ob dieser Datensatz lokal geändert und noch nicht von AMS bestätigt wurde
    /// </summary>
    public bool Dirty { get; set; }
    /// <summary>
    /// Die Lizenz der der Vertrag angehört
    /// </summary>
    public string License { get; set; }
    public DateTime LastSynchronisation { get; private set; }
}

```

- Klasse-SchadenModel

```

using System;
using System.Collections.Generic;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace ApiTest
{
    public partial class SchadenModel
    {
        public Guid Id { get; set; }
        public DateTime? Bearbeitet { get; set; }
        public DateTime? SchadenZeitpunkt { get; set; }
        /// <summary>
        /// Gibt an ob dieser Datensatz lokal geändert und noch nicht von AMS bestätigt wurde
        /// </summary>
        public bool Dirty { get; set; }
        public string License { get; set; }
        /// <summary>
        /// Gibt an ob der Schaden an AMS gesendet wurde, oder andersrum, ob er (noch) bearbeitet werden darf
    }
}

```

```

    /// </summary>
    public bool ClaimSent { get; set; }
    public DateTime LastSynchronisation { get; private set; }
    [JsonExtensionData]
    public IDictionary<string, JToken> AdditionalData { get; set; }
    public DateTime? Abschlussstag { get; set; }
    public string Amsidnr { get; set; } // Kwangsfield
    public DateTime? Anlage { get; set; }
    public string Anspruchsteller { get; set; }
    public DateTime? Anzeigehvn { get; set; }
    public DateTime? AnzeigeAnVU { get; set; }
    public string Bearbeiter { get; set; }
    public string BearbeiterMakler { get; set; }
    public string BearbeiterVN { get; set; }
    public string BearbeiterVU { get; set; }
    public string EmailMakler { get; set; }
    public string EmailVn { get; set; }
    public string EmailVU { get; set; }
    public decimal? Franchise { get; set; }
    public string Gesellschaft { get; set; }
    public string Hergang { get; set; }
    public string Info { get; set; }
    public string Key { get; set; } // Kwangsfield, Vater - Datensatz, also zugehöriger Vertrag.Amsidnr
    public int? Kostenstelle { get; set; }
    public DateTime? Meldungtag { get; set; }
    public string Niederlassung { get; set; }
    public string NiederlassungKey { get; set; }
    public string NummerMakler { get; set; }
    public string NummerVN { get; set; }
    public string NummerVU { get; set; }
    public string Objekte { get; set; }
    public string Regulierungsart { get; set; }
    public decimal? Reserve { get; set; }
    public string Risiko { get; set; }
    public string S21_F1 { get; set; }
    public string S21_F10 { get; set; }
    public string S21_F2 { get; set; }
    public string S21_F3 { get; set; }
    public string S21_F4 { get; set; }
    public string S21_F5 { get; set; }
    public string S21_F6 { get; set; }
    public string S21_F7 { get; set; }
    public string S21_F8 { get; set; }
    public string S21_F9 { get; set; }
    public string S22_F1 { get; set; }
    public string S22_F10 { get; set; }
    public string S22_F2 { get; set; }
    public string S22_F3 { get; set; }
    public string S22_F4 { get; set; }
    public string S22_F5 { get; set; }
    public string S22_F6 { get; set; }
    public string S22_F7 { get; set; }
    public string S22_F8 { get; set; }
    public string S22_F9 { get; set; }
    public decimal? Schadenhoehe { get; set; }
    public string Schadenort { get; set; }
    public string Schadenstatus { get; set; }
    public bool? Sichscheln { get; set; }
    public string TelefonMakler { get; set; }
    public string TelefonVN { get; set; }
    public string TelefonVU { get; set; }
    public string Ursache { get; set; }
    public string Vereinbarung { get; set; }
    public string VereinbarungKey { get; set; }
    public string Vermittlerkonto { get; set; }
    public bool? VertellenAufBeteiligte { get; set; }
    public string Vnr { get; set; }
    public string Vorgang { get; set; }
    public decimal? Zahlung { get; set; }
    public int? Zeichnungsjahr { get; set; }
    public string Zuordnung { get; set; }
    public string Schlagwoerter { get; set; } // Schlagwoerter/TAGS zum Schaden
    private HashSet<string> columns;
    public bool ShouldSerialize(string name) => columns.Contains(name) ?? true;
}
}
}

```

- Klasse-AdressModel

```

    public partial class AdresseModel
    {
        public Guid Id { get; set; }
        public DateTime? Bearbeitet { get; set; }
        public AddressDetails PrivateAddress { get; set; }
        public AddressDetails BusinessAddress { get; set; }
        /// <summary>
        /// Gibt an ob dieser Datensatz lokal geändert und noch nicht von AMS bestätigt wurde
        /// </summary>
        public bool Dirty { get; set; }
        /// <summary>
        /// Die Lizenz der der Brief angehört
        /// </summary>
        public string License { get; set; }
        public DateTime LastSynchronisation { get; private set; }
        public int? Abhaengig { get; set; }
        public string Amsidnr { get; set; }
        public string Anrede { get; set; }
        public string Bankname1 { get; set; }
        public string Bankname2 { get; set; }
        public string Bearbeiter { get; set; }
        public string Beruf { get; set; }
        public string Berufsstatus { get; set; }
        public string Bic1 { get; set; }
        public string Bic2 { get; set; }
        public string Bilanzstichtag { get; set; }
        public string Blz1 { get; set; }
        public string Blz2 { get; set; }
        public string Email { get; set; }
        public string Email2 { get; set; }
        public string Familienstand { get; set; }
        public string Fax { get; set; }
        public string Fax2 { get; set; }
        public DateTime? Geburt { get; set; }
        public string Geburtsname { get; set; }
        public string Geburtsort { get; set; }
        public string Geschlecht { get; set; }
        public DateTime? Gruendung { get; set; }
        public string Iban1 { get; set; }
        public string Iban2 { get; set; }
        public string Info { get; set; }
        public string Inhaber1 { get; set; }
        public string Inhaber2 { get; set; }
        public string Internet { get; set; }
        public string Internet2 { get; set; }
        public string Kommentar { get; set; }
        public string Kontaktname { get; set; }
        public string Kontobezeichnung1 { get; set; }
        public string Kontobezeichnung2 { get; set; }
        public string Kontonummer1 { get; set; }
        public string Kontonummer2 { get; set; }
        public string Land { get; set; }
        public string Land2 { get; set; }
        public DateTime? Mahnstop { get; set; }
        public string Mandant { get; set; }
        public string MandantId { get; set; }
        public int? Mitarbeiterzahl { get; set; }
        public string Mobil { get; set; }
        public string Mobil2 { get; set; }
        public string Nachname { get; set; }
        public string Name1 { get; set; }
        public string Name2 { get; set; }
        public string Name22 { get; set; }
        public string Name3 { get; set; }
        public string Name32 { get; set; }
        public string Ort { get; set; }
        public string Ort2 { get; set; }
        public DateTime? PersonalausweisAusstellungsdatum { get; set; }
        public string PersonalausweisBehoerde { get; set; }
        public string PersonalausweisNr { get; set; }
        public string Plz { get; set; }
        public string Plz2 { get; set; }
        public string Rechtsform { get; set; }
        public DateTime? Sepamandat { get; set; }
    }

```

```

public string Sepamandatsreferenz { get; set; }
public string Staatsangehoerigkeit { get; set; }
public string Steueridentifikationsnummer { get; set; }
public string Steuernummer { get; set; }
public string Strasse { get; set; }
public string Strasse2 { get; set; }
public string Telefon1 { get; set; }
public string Telefon12 { get; set; }
public string Telefon2 { get; set; }
public string Telefon22 { get; set; }
public string Telnummern { get; set; }
public string Titel { get; set; }
public string Typ1 { get; set; }
public string Typ2 { get; set; }
public int? Umsatz { get; set; }
public string Ustidnr { get; set; }
public string Vorname { get; set; }
public string Zuordnung { get; set; }
public string Schlagwoerter { get; set; } //Schlagwörter zur Adresse. KUNDE selbst hat keine Schlagwörter
public string[] RefVerwendung { get; set; } // Verwendung aus RefAdresse
private HashSet<string> columns;
public bool ShouldSerialize(string name) => columns?.Contains(name) ?? true;
}

```