

# comdirect REST API – Schnittstellenspezifikation

Version April 2020

## Änderungslog

Version	Änderung
November 2019	Initiale Version erstellt
November 2019 - Update 1	<ul style="list-style-type: none"><li>- Präzisierung einiger Beschreibungen</li><li>- Korrektur von Beispielen in Kapitel 9.2</li></ul>
Februar 2020	<ul style="list-style-type: none"><li>- Korrektur Response Instrument-Schnittstelle (Kapitel 6.1.1), orderDimensions-Schnittstelle (Kapitel 7.1.1)</li><li>- Ergänzung Resource DOCUMENTS (Kapitel 9)</li><li>- Verschiebung der Beispiele von Kapitel 9 zu 10</li></ul>
Februar 2020 – Update 1	<ul style="list-style-type: none"><li>- Ergänzung von Informationen zum Header bei den DOCUMENTS-Schnittstellen</li></ul>
März 2020	<ul style="list-style-type: none"><li>- Korrektur der Filterparameter in Kapitel 4.1.3</li><li>- Korrektur des Dimension-Objekts in Kapitel 7.2.1, dadurch Verschiebung der nachfolgenden Objekte um eine Nummer</li><li>- neuer Queryparameter beim Abruf Orderbuch (Kapitel 7.1.2)</li><li>- drei neue Queryparameter beim Abruf eines Instruments (Kapitel 6.1.1)</li><li>- Neues API: Kapitel Resource REPORTS hinzugefügt (Kapitel 10)</li><li>- Verschiebung der Beispiele von Kapitel 10 zu 11</li></ul>
April 2020	<ul style="list-style-type: none"><li>- Ergänzung photoTAN-Push als neues TAN-Verfahren</li><li>- Versionsupdate der Schnittstelle 4.1.1 Abruf AccountBalances alle Konten (keine inhaltlichen Änderungen)</li></ul>

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und grundlegende Prinzipien.....</b>	<b>6</b>
1.1	URIs.....	6
1.2	HTTP-Header .....	6
1.2.1	Custom-Header .....	6
1.2.2	Client Request-Id.....	7
1.3	Projektionen .....	7
1.4	Fehlermeldungen .....	8
<b>2</b>	<b>So geht es los.....</b>	<b>10</b>
2.1	OAuth2 Resource Owner Password Credentials Flow .....	10
2.2	Abruf Session-Status.....	11
2.2.1	Session-Objekt .....	12
2.3	Anlage Validierung einer Session-TAN.....	12
2.4	Aktivierung einer Session-TAN .....	14
2.5	CD Secondary Flow .....	16
<b>3</b>	<b>So geht es weiter .....</b>	<b>17</b>
3.1	Weitere Token-Flows .....	17
3.1.1	Refresh-Token-Flow.....	17
3.1.2	Revoke-Token .....	19
3.2	Transaktionen freigeben .....	19
3.3	Header sämtlicher weiterer Schnittstellen.....	20
3.4	Standard REST-Objekte .....	20
3.4.1	AmountValue.....	20
3.4.2	EnumText .....	21
3.4.3	AmountString.....	21
3.4.4	PercentageString.....	21
3.4.5	CurrencyString .....	21
3.4.6	Datumstypen .....	21
<b>4</b>	<b>Resource ACCOUNT .....</b>	<b>22</b>
4.1	REST-API Ressourcen.....	22
4.1.1	Abruf AccountBalances alle Konten.....	22
4.1.2	Abruf AccountBalances .....	23
4.1.3	Abruf Kontoumsätze.....	23
4.2	REST-Objekte .....	24

4.2.1 Account .....	24
4.2.2 AccountBalance .....	25
4.2.3 AccountTransaction.....	25
<b>5 Resource DEPOT .....</b>	<b>27</b>
5.1 REST-API Ressourcen.....	27
5.1.1 Abruf Depots .....	27
5.1.2 Abruf Depotbestand und/oder Positionen .....	28
5.1.3 Abruf einer Position des Depots.....	29
5.1.4 Abruf Depotumsätze.....	29
5.2 REST-Objekte .....	30
5.2.1 Depot .....	30
5.2.2 DepotPosition .....	31
5.2.3 DepotTransaction .....	32
5.2.4 Price .....	33
<b>6 Resource INSTRUMENT .....</b>	<b>33</b>
6.1 REST-API Ressourcen.....	33
6.1.1 Abruf Instrument.....	34
6.2 REST-Objekte .....	34
6.2.1 Instrument .....	34
6.2.2 StaticData .....	35
6.2.3 DerivativeData .....	36
6.2.4 Rating .....	37
6.2.5 FundDistribution .....	37
<b>7 Resource ORDER .....</b>	<b>38</b>
7.1 REST-API Ressourcen.....	38
7.1.1 Abruf OrderDimensionen .....	39
7.1.2 Abruf Orders (Orderbuch) .....	40
7.1.3 Abruf Order (Einzelorder) .....	41
7.1.4 Anlage Prevalidation Orderanlage .....	42
7.1.5 Anlage Validation Orderanlage .....	42
7.1.6 Anlage Ex-Ante Kostenausweis für eine Orderanlage .....	43
7.1.7 Anlage Order .....	44
7.1.8 Anlage Prevalidation Orderänderung .....	44
7.1.9 Anlage Validation Orderänderung und Orderlöschung .....	45
7.1.10 Anlage Ex-Ante Kostenausweis für eine Orderänderung .....	46
7.1.11 Änderung der Order .....	47

7.1.12	Löschung der Order .....	48
7.2	REST-Objekte .....	48
7.2.1	Dimensions .....	48
7.2.2	Venue .....	48
7.2.3	Order .....	50
7.2.4	Execution .....	53
7.2.5	CostIndicationExAnte .....	54
7.2.6	FXRateEUR .....	58
7.2.7	Inducement .....	58
7.2.8	CostGroup .....	58
7.2.9	CostEntry .....	60
7.2.10	TotalCostBlock .....	61
7.2.11	TotalCostEntry .....	62
7.2.12	TotalHoldingCostBlock .....	63
7.2.13	TotalHoldingCostEntry .....	64
<b>8</b>	<b>Resource QUOTE .....</b>	<b>66</b>
8.1	REST-API Ressourcen .....	66
8.1.1	Anlage Validierung Quote Request-Initialisierung .....	66
8.1.2	Änderung Validierung Quote Request-Initialisierung mit TAN .....	67
8.1.3	Anlage Quote Request .....	68
8.1.4	Orderanlage der Quote-Order .....	68
8.2	REST-Objekte .....	69
8.2.1	Quote .....	69
<b>9</b>	<b>Resource DOCUMENTS .....</b>	<b>69</b>
9.1	REST-API Ressourcen .....	69
9.1.1	Abruf PostBox .....	69
9.1.2	Abruf eines Dokuments .....	70
9.1.3	Abruf Dokument-Vorschaltseite .....	71
9.2	REST-Objekte .....	71
9.2.1	Document .....	71
9.2.2	DocumentMetadata .....	71
<b>10</b>	<b>Resource REPORTS .....</b>	<b>72</b>
10.1	REST-API Ressourcen .....	72
10.1.1	Abruf Salden sämtlicher comdirect-Produkte .....	72
10.2	REST-Objekte .....	73
10.2.1	ProductBalance .....	73

10.2.2	BalanceAggregation.....	74
10.2.3	CardBalance .....	74
10.2.4	Card .....	74
10.2.5	VisaCardImage .....	75
10.2.6	InstallmentLoanBalance.....	75
10.2.7	InstallmentLoan.....	76
10.2.8	FixedTermSavings .....	76
10.2.9	FixedTermSavingsType .....	77
<b>11</b>	<b>Beispiele .....</b>	<b>78</b>
11.1	API-Aufrufe zur Ausführung des Livetradings .....	78
11.2	Beispiele für die Orderanlage.....	82
11.2.1	Market Order .....	82
11.2.2	Tagesgültige Limit Order.....	83
11.2.3	Tagesgültige Stop Limit Order .....	83
11.2.4	Trailing Stop Market Verkaufsoorder mit absolutem Abstand .....	83
11.2.5	Trailing Stop Limit Verkaufsoorder mit relativem Abstand.....	84
11.2.6	Kombinationsorder des Typs One Cancels Other (OCO) .....	84
11.2.7	Kombinationsorder des Typs Next Order .....	85

# 1 Einleitung und grundlegende Prinzipien

Mit dem comdirect REST API möchten wir Ihnen die Möglichkeit geben, sich eigene Applikationen für Banking und Brokerage programmieren zu können, die sich an Ihren Bedürfnissen orientieren. Dieses Dokument beschreibt die zur Verfügung gestellten APIs mit den jeweiligen Request- und Response-Objekten.

Im ersten Kapitel finden Sie grundlegende Prinzipien des comdirect REST API. Es beschreibt u.a. den grundsätzlichen Aufbau der API-Urls, die Inhalte des Headers und den Aufbau von Fehlermeldungen.

Das zweite Kapitel führt Sie Schritt für Schritt durch die Authentifizierung. Am Ende des Kapitels werden Sie in der Lage sein, die weiteren fachlichen APIs aufzurufen. Wie Sie dabei vorgehen, steht in Kapitel 3, während ab Kapitel 4 die fachlichen APIs konkret beschrieben werden.

Im letzten Kapitel sind typische Abruffolgen von APIs zusammengestellt. Außerdem werden für die verschiedenen Order-Typen Request-Beispiele beschrieben.

Viel Erfolg bei der Integration des comdirect REST API in Ihre Applikationen!

## 1.1 URIs

Die URI zur Identifikation von Ressourcen folgt einem festen Schema. Im folgenden Beispiel wird eine Primärressource „order“ (Wertpapierorder) adressiert:

`https://api.comdirect.de/api/brokerage/v1/orders/{orderId}`

`https://api.comdirect.de/api/brokerage/v1/orders/{orderId}`

```

\_____ / \_____\_____\_____\_____ / \_____ /
|           |           |           |           |
URL-Präfix   Modul Version Resource  Id

```

Teil	Bedeutung
URL-Päfix	Alle URIs beginnen mit dem Präfix <a href="https://api.comdirect.de/api/">https://api.comdirect.de/api/</a> .
Modul	Das Modul kennzeichnet die fachliche Verortung der Ressource.
Version	Die Version beginnt immer mit einem kleinen „v“, gefolgt von einer Ordinalzahl - der MAJOR Version der konkreten REST API. Die Version steht unmittelbar vor der Ressource.
Resource	Die Ressource(n), die durch die URI identifiziert wird/werden. Die Identifizierung einer einzelnen Ressource erfolgt durch die Angabe eines entsprechenden Identifiers. Ressourcen werden grundsätzlich klein geschrieben und im Plural verwendet.
Id	Angabe einer Id, die eine konkrete Resource benennt. Als Id wird üblicherweise eine UUID (UniqueUserId) verwendet.

## 1.2 HTTP-Header

### 1.2.1 Custom-Header

Header	Format	Bedeutung
X-HTTP-Method-Override	String	Mit diesem Header können Clients, die nicht alle HTTP-Methoden unterstützen, die eigentliche HTTP-Methode überschreiben. Mögliche Werte sind "PUT", "PATCH" und "DELETE". Dieser Header ist optional.
x-http-request-info	JSON-Objekt	Enthält allgemeine Informationen, die zunächst unabhängig vom fachlichen API mitgeliefert werden. Das umfasst zurzeit nur die Client Request-Id. Dieser Header ist ein Pflichtfeld.

### 1.2.2 Client Request-Id

Clients müssen im HTTP-Header „x-http-request-info“ eine Client Request-Id übertragen. Die Client Request-Id besteht aus zwei Teilen:

Teil	Format	Bedeutung
Session-Id	String Max. 32-stellig.  Es sind die Hexadezimalzeichen erlaubt.	Die Session-Id repräsentiert eine Benutzersitzung. Der Client muss vor dem ersten Request eine entsprechende Session-Id erzeugen und in allen folgenden Requests mitsenden. Die Definition einer Benutzersitzung ist abhängig vom Client. Ein Client in Form einer Mobile App startet die Benutzersitzung mit dem Start der App und beendet diese mit dem Schließen der App.
Request-Id	Zahl (ohne Vorzeichen) 9-stellig	Die Request-Id macht den Request innerhalb der Benutzersitzung eindeutig und kann z.B. aus einem Timestamp im Format <i>HHmmssSSS</i> bestehen.

**Beispiel für eine Client Request-Id:**

```
{
  "clientRequestId": {
    "sessionId": "550e8400e29b11d4a716446655440000",
    "requestId": "140113250"
  }
}
```

Die Client Request-Id kann zur Sicherstellung der Idempotenz verwendet werden.

### 1.3 Projektionen

Mit Hilfe von Projektionen kann die Anzeige von Informationen einer Ressource eingeschränkt werden. Die Projektion erfolgt über Query-Parameter:

Parameter	Format	Bedeutung
with-attr	String	Komma-separierte Liste von Attributen, die in der Repräsentation der Ressource enthalten sein soll.
without-attr	String	Komma-separierte Liste von Attributen, die <u>nicht</u> in der Repräsentation der Ressource enthalten sein soll.

Die einzelnen APIs definieren jeweils, für welche Attribute einer Ressource diese Parameter beachtet werden.

## 1.4 Fehlermeldungen

Fehler, Warnungen oder Informationen für den API-Nutzer werden im Header des jeweiligen Responses in Form einer BusinessMessage zurückgegeben. Diese ist im Header-Feld „x-http-response-info“ enthalten. BusinessMessages transportieren Meldungen, die im Normalfall im Client-Programm zur Anzeige gebracht werden sollen.

Bei einem Fehler (severity = ERROR) wird darüber hinaus der normale Response-Body durch eine BusinessMessage ersetzt.

### Beispiel Header (Auszug):

```
x-http-response-info
{
  "messages": [
    {
      "severity": "INFO",
      "key": "hinweis_basisinformationsblatt_vorhanden",
      "message": "Hinweis: Das Basisinformationsblatt ist vorhanden.",
      "args": {},
      "origin": []
    },
    {
      "severity": "ERROR",
      "key": "fehler-erforderliche-tgf-fehlt",
      "message": "Für das von Ihnen gewünschte Wertpapier ist eine Vereinbarung zu Finanztermingeschäften erforderlich. Das entsprechende Formular finden Sie unter comdirect.de/formulare",
      "args": {},
      "origin": []
    }
  ]
}
```

### Beispiel Body

```
{
  "code": "request.object.invalid",
  "messages": [
    {
      "severity": "INFO",
      "key": "hinweis_basisinformationsblatt_vorhanden",
      "message": "Hinweis: Das Basisinformationsblatt ist vorhanden.",
      "args": {},
      "origin": []
    },
    {
      "severity": "ERROR",
      "key": "fehler-erforderliche-tgf-fehlt",

```



```

        "message": "Für das von Ihnen gewünschte Wertpapier ist eine
Vereinbarung zu Finanztermingeschäften erforderlich. Das entsprechende
Formular finden Sie unter comdirect.de/formulare",
        "args": {},
        "origin": []
    }
}

```

Der „code“ im Response-Body ermöglicht eine genauere Differenzierung der Fehlersituation, sofern der HTTP-Statuscode nicht genau genug ist; z.B. können die Standard-HTTP-Statuscodes nicht zwischen den Situationen `IllegalArgument` und `IllegalState` unterscheiden, sodass in beiden Fällen üblicherweise 422 (Unprocessable Entity) geliefert wird und die genauere Kennzeichnung des Fehlers in einem eigenen Error-Code erfolgen muss.

Es wird empfohlen, den Ablauf im aufrufenden Programm aufgrund der Fehler-Codes zu steuern und die Messages lediglich zur Anzeige zu verwenden.

Weitere Informationen können durch zusätzliche Attribute in der Fehler-Response zurückgegeben werden.

Feld	Format	Beschreibung
severity	String	Die severity gibt an, ob es sich bei der Message um eine Fehlermeldung ("ERROR"), eine Warnung ("WARN") oder einen Hinweis ("INFO") handelt.
key	String	Der key identifiziert die Message eindeutig. Hiermit kann ein Client die Message auf einen anzuzeigenden Text abbilden (z.B. sprachabhängig).
message	String	Die message enthält einen Default-Text, der angezeigt werden kann, wenn das Client-Programm nicht über die Fähigkeit verfügt, Message-Keys selbst auf Texte abzubilden oder wenn für einen Key kein Text gefunden wird. In der message wurden bereits evtl. Platzhalter durch die entsprechenden Argumente (aus dem Feld args) ersetzt.
args	Object (Map)	Das Attribut args liefert im Fall von parametrisierten message(s) eine Map der Argumente, die in die message integriert sind. Beispiel: args : {"handelswaehrung": "EUR", "mindestbetrag": "41.50"}. Es gibt keine Festlegungen über die Ausprägungen des Keys, dieser kann ggf. nur eine fortlaufende Nummer sein.
origin	String	origin gibt an, dass sich die Meldung auf ein bestimmtes Attribut des Request-Objekts oder eine Menge solcher Attribute bezieht. Das ist insbesondere für Validierungs-Messages sinnvoll, da dann die Anzeige der Meldung mit dem dazugehörigen Eingabefeld verknüpft werden kann. Beispiel: {"message": "Limit muss > 0 sein", "origin": ["ausfuehrungsLimit"]}. In den meisten Fällen wird in origin nur

		ein Input-Feld stehen. Bei Cross-Validations können das aber auch mehrere sein. Bei allgemeinen Verarbeitungsfehlern ist origin leer (null).
--	--	--

**Anmerkung:**

Die Fehlermeldungen werden bis auf Weiteres in deutscher Sprache transportiert.

## 2 So geht es los

Um auf das API zugreifen zu können, müssen Sie sich einen OAuth2 Authentifikations-Token (<https://oauth.net/2/>) ausstellen lassen. Aufgrund der im API zur Verfügung stehenden Daten wird für die Erstellung des Tokens ein zweiter Faktor (TAN) benötigt.

Der gesamte Prozess besteht aus dem Aufruf von fünf Schnittstellen, die im Folgenden beschrieben werden. Die einzugebende TAN wird zusätzlich als Session-TAN verwendet, so dass alle folgenden Transaktionen der Session ohne weitere TAN-Eingabe ausgeführt werden können.

### 2.1 OAuth2 Resource Owner Password Credentials Flow

POST <https://api.comdirect.de/oauth/token>

**Beschreibung:**

Im ersten Schritt wird ein OAuth2 Authentifikations-Token erzeugt, mit dem in der Folge die Schnittstellen zur Erstellung der Session-TAN aufgerufen werden können.

#### REQUEST (= Generierung eines Access- und Refresh-Tokens)

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-
Body	client_id	String	zugewiesene client_id
	client_secret	String	zugewiesenes client_secret
	grant_type	String	password
	username	String	Ihre achtstellige Zugangsnummer
	password	String	Ihre sechsstellige PIN

**Header:**

```
Accept:application/json
Content-Type:application/x-www-form-urlencoded
```

**Body:**

```
client_id:"zugewiesene client_id"
client_secret:"zugewiesenes client_secret"
username:"Zugangsnummer"
password:"PIN"
grant_type:password
```

**RESPONSE (= Access- und Refresh-Token)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
String		access_token	Authentifikations-Token
String		token_type	Art des Tokens (Bearer)
String		refresh_token	Refresh-Token
String		expires_in	Angabe der Tokengültigkeit in Sekunden (599)
String		scope	Benennt die Zugriffsrechte (TWO_FACTOR)
String		kdnr	Ihre Kundennummer
String		bpid	Interne Identifikationsnummer
String		kontaktId	Interne Identifikationsnummer

**Beispiel Body:**

```
{
  "access_token": "1234567890_Access-Token_NEU_34567890",
  "token_type": "bearer",
  "refresh_token": "1234567890_Refresh-Token_NEU_4567890",
  "expires_in": 599,
  "scope": "TWO_FACTOR",
  "kdnr": "1234567890",
  "bpid": "1234567",
  "kontaktId": "1234567890"
}
```

**Bitte beachten:**

Der Access-Token muss im Header der weiteren Requests mitgegeben werden:

Authorization: "Bearer 1234567890\_\_Access-Token\_\_1234567890"

**2.2 Abruf Session-Status**GET *URL-Präfix*/session/clients/{clientId}/v1/sessions

**Beschreibung:** Um eine Session-TAN anfordern zu können, muss zunächst das Session-Objekt angefordert werden.

**REQUEST (= Abruf des Session-Status)**

Parameter Type	Parameter Name	Data Type	Description
Path	clientId	String	Literal „user“

**Beispiel Header:**

```

Accept:application/json
Authorization:Bearer 1234567890__Access-Token__1234567890
x-http-request-info:
    {"clientRequestId":{"sessionId":"123_beliebige_ID_fuer_Session_12","
    requestId":"123456789"}}
Content-Type:"application/json"

```

**RESPONSE (= Status der Session-TAN)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Session			JSON-Objekt SessionTAN

**HTTP Statuscodes:**

- 200 - OK
- 422 - UNPROCESSABLE ENTITY

**2.2.1 Session-Objekt**

Parameter	Datentyp	Schreib- barkeit	Beschreibung
identifizier	String(<=40)	read-only	UUID der Session
sessionTanActive	Boolean	read-only	Gibt an, ob eine aktive Session-TAN vorliegt (true)
activated2FA	Boolean	read-only	Gibt an, ob eine 2-Faktor-Authentifizierung vorliegt (true)

**2.3 Anlage Validierung einer Session-TAN**POST *URL-Präfix*/session/clients/{clientId}/v1/sessions/{sessionId}/validate

**Beschreibung:** Für das nun bekannte Session-Objekt wird in diesem Schritt eine TAN-Challenge angefordert.

**Bitte beachten:**

Das Abrufen von fünf TAN-Challenges ohne zwischenzeitliche Entwertung einer korrekten TAN führt zur Sperrung des Onlinebanking-Zugangs.

**REQUEST (= Anlage Validierung einer Session-TAN)**

Parameter Type	Parameter Name	Data Type	Description
Path	clientId	String	Literal „user“
	sessionId	String	„identifizier“ aus dem Session-Objekt
Body		Session	Session-Objekt, in dem der identifizier den Wert der sessionId hat und sessionTanActive und activated2FA den Wert „true“ erhält.

**Beispiel Header:**

```
Accept:application/json
Authorization:Bearer 1234567890__Access-Token__1234567890
x-http-request-info:
    {"clientId":{"sessionId":"123_beliebige_ID_fuer_Session_12","requestId":"123456789"}}
Content-Type:application/json
```

**Beispiel Body:**

```
{
    "identifizier": "12345__identifizier_der_session__1234",
    "sessionTanActive": true,
    "activated2FA": true
}
```

**RESPONSE (= Im Falle einer inaktiven Session-TAN wird eine TAN-Challenge erzeugt)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Session			JSON-Objekt Session

**Beispiel Response-Header (Auszug):**

```
x-once-authentication-info:
    {
        "id":"7654321",
        "typ":"M_TAN",
        "challenge":"+49-160-99XXXX",
        "availableTypes":["P_TAN","M_TAN"]
    }
```

Im Response-Header werden im "x-once-authentication-info"-Feld folgende Daten zurückgegeben:

- id: die ID der TAN-Challenge; diese ist in jedem Fall beim Aufruf der anschließenden Schnittstelle zur Aktivierung der Session-TAN zu übergeben

- typ: der TAN-Typ; mögliche Ausprägungen sind M\_TAN, P\_TAN, P\_TAN\_PUSH; ohne weitere Angaben im Request-Header dieser Schnittstelle wird immer eine TAN Ihres favorisierten TAN-Verfahrens erzeugt
- challenge: je TAN-Verfahren werden verschiedene Angaben gemacht; bei P\_TAN wird die photoTAN-Grafik im PNG-Format (Base64-codiert) übergeben, bei MTAN die Telefonnummer, an die die mobileTAN gesendet wurde. Im Falle von P\_TAN\_PUSH entfällt die Challenge.
  - x-once-authentication-info Response Header im Falle P\_TAN:  
 {"id":"123456","typ":"P\_TAN","challenge":"Base64Code","availableTypes":["verfügbare TAN-Verfahren"]}
  - x-once-authentication-info Response Header im Falle M\_TAN:  
 {"id":"123456","typ":"M\_TAN","challenge":"+49-1234-567XXXX","availableTypes":["verfügbare TAN-Verfahren"]}
- availableTypes: alle Ihre aktivierten TAN-Verfahren

Um auf ein anderes TAN-Verfahren zu wechseln (Beispiel: P\_TAN), wird diese Validation-Schnittstelle erneut aufgerufen. Dabei muss im Header folgende Informationen hinzugefügt werden:

```
x-once-authentication-info: {"typ":"P_TAN"}
```

#### HTTP Statuscodes:

- 201 - CREATED
- 422 - UNPROCESSABLE ENTITY

## 2.4 Aktivierung einer Session-TAN

PATCH *URL-Präfix*/session/clients/{clientId}/v1/sessions/{sessionId}

**Beschreibung:** Die Schnittstelle dient zur Aktivierung der Session-TAN.

#### REQUEST (= Aktivierung der Session-TAN)

Parameter Type	Parameter Name	Data Type	Description
Path	clientId	String	Literal „user“
	sessionId	String	„identifizier“ aus dem Session-Objekt
Body		Session	Session-Objekt, in dem der identifizier den Wert der sessionId hat und sessionTanActive und activated2FA den Wert „true“ erhält.

#### Beispiel Header:

```
Accept:application/json
Authorization:Bearer 1234567890__Access-Token__1234567890
```

```

x-http-request-info:
  {"clientRequestId":{"sessionId":"123_beliebige_ID_fuer_Session_12","requestId":"123456789"}}
Content-Type:application/json,
x-once-authentication-info:{"id":"7654321"}, //id der Challenge
x-once-authentication:123456 //die eigentliche TAN

```

Die „id“ wird dem Response-Header der Validation-Schnittstelle entnommen und im Header dieser Schnittstelle im „x-once-authentication-info“-Parameter übergeben.

Über den Header-Parameter „x-once-authentication“ wird die aus der photoTAN-Grafik oder der mobileTAN-SMS ermittelte TAN in der Schnittstelle angegeben.

Wenn das TAN-Verfahren photoTAN-Push genutzt wird, erfolgt die Freigabe der TAN in der comdirect photoTAN-App. Eine TAN-Eingabe im Header der Schnittstelle ist damit nicht mehr erforderlich. Der Parameter wird deshalb in diesem Falle nicht im Header benötigt.

#### Beispiel Body:

```

{
  "identifizier" : "12345__identifizier_der_session__1234",
  "sessionTanActive": true,
  "activated2FA": true
}

```

#### Bitte beachten:

Nach drei falschen TAN-Eingaben wird der Zugang zum Onlinebanking gesperrt. Nach zwei Fehleingaben über das API kann der Fehlerzähler über eine korrekte TAN-Eingabe auf der comdirect Website wieder zurückgesetzt werden.

#### RESPONSE (= Session TAN wurde aktiviert, Zwei-Faktor-Authentifizierung wurde durchgeführt)

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Session			JSON-Objekt Session

#### HTTP Statuscodes:

- 200 - OK
- 422 - UNPROCESSABLE ENTITY

#### Bitte beachten:

- Nach Aktivierung der Session TAN für eine Session wird bei TAN-pflichtigen Prozessen auf die Übergabe einer TAN verzichtet (vergleiche hierzu Kapitel 3).
- Eine Session-TAN bleibt so lange gültig, bis das letzte Access/Refresh-Token seine Gültigkeit verliert.

## 2.5 CD Secondary Flow

POST <https://api.comdirect.de/oauth/token>

### Beschreibung:

Der comdirect-spezifische OAuth2 Authentication Flow „cd\_secondary“ ist ein Mix aus dem „client-credentials“-Flow und dem „resource-owner-password-credentials“-Flow. Über diesen Flow lassen Sie sich im letzten Schritt einen Access-Token ausstellen, der die Berechtigungen für die Banking- und Brokerage-Schnittstellen hat.

### REQUEST (= Generierung eines Access- und Refresh-Tokens)

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-
Body	client_id	String	zugewiesene client_id
	client_secret	String	zugewiesenes client_secret
	grant_type	String	cd_secondary
	token	String	gültiger Access-Token

### Header:

```
Accept:application/json
Content-Type:application/x-www-form-urlencoded
```

### Beispiel Body:

```
client_id:"zugewiesene client_id"
client_secret:"zugewiesenes client_secret"
grant_type:cd_secondary
token:1234567890__Access-Token__1234567890
```

### RESPONSE (= Access- und Refresh-Token)

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
String		access_token	Authentifikations-Token
String		token_type	Art des Tokens (Bearer)
String		refresh_token	Refresh-Token
String		expires_in	Angabe der Tokengültigkeit in Sekunden (599)



String		scope	Benennt die Zugriffsrechte (BANKING_RO, BROKERAGE_RW, SESSION_RW)
String		kdnr	Ihre Kundennummer
String		bpid	Interne Identifikationsnummer
String		kontaktId	Interne Identifikationsnummer

**Body:**

```
{
  "access_token": "1234567890__Access-Token__1234567890",
  "token_type": "bearer",
  "refresh_token": "1234567890_Refresh-Token__1234567890",
  "expires_in": 599,
  "scope": "BANKING_RO BROKERAGE_RW SESSION_RW",
  "kdnr": "1234567890",
  "bpid": "1234567",
  "kontaktId": "1234567890"
}
```

### 3 So geht es weiter

Mit dem Access-Token haben Sie Zugriff auf die REST-API Ressourcen. Der Access-Token hat üblicherweise eine Gültigkeit von 10 Minuten. Nach Ablauf dieser Zeitspanne muss die Anwendung mittels des Refresh-Tokens erneut einen Access- /Refresh-Token anfordern (Kapitel 3.1.1 Refresh-Token-Flow).

Sie haben sicherzustellen, dass evtl. implementierte Mechanismen zur automatischen Aktualisierung des Access-Tokens beendet werden, sobald die Anwendung beendet wird. Dadurch erlischt die Gültigkeit von Access/Refresh-Token und der Session-TAN. Ein API-Zugriff ist dann erst nach einem erneuten Login möglich.

Die nun folgenden Unterkapitel sind wichtig für die grundlegende Verwendung des APIs. Sie werden allerdings nur in speziellen Situationen benötigt, weshalb sie eine Art Glossar darstellen, auf das im weiteren Verlauf des Dokuments verwiesen wird.

#### 3.1 Weitere Token-Flows

##### 3.1.1 Refresh-Token-Flow

POST <https://api.comdirect.de/oauth/token>

**Anmerkung:**

Der Refresh-Token-Flow dient zur Erneuerung des Access-Tokens. Ein erfolgreicher Refresh-Token Request hat zur Folge, dass ein neuer Access- sowie ein neuer Refresh-Token ausgestellt werden. Sofern kein erfolgreicher Refresh-Token Request innerhalb der Lebensdauer des Refresh-Tokens getätigt wurde, muss eine erneute Authentifizierung gemäß Kapitel 2 erfolgen.

#### REQUEST (= Generierung eines neuen Access- und Refresh-Tokens)

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-
Body	client_id	String	zugewiesene client_id
	client_secret	String	zugewiesenes client_secret
	grant_type	String	refresh_token
	refresh_token	String	Refresh-Token

#### Header:

```
Accept:application/json
Content-Type:application/x-www-form-urlencoded
```

#### Beispiel Body:

```
client_id:"zugewiesene client_id"
client_secret:"zugewiesenes client_secret"
grant_type:refresh_token
refresh_token:1234567890_Refresh-Token__1234567890
```

#### RESPONSE (= Access- und Refresh-Token)

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
String		access_token	Authentifikations-Token
String		token_type	Art des Tokens (Bearer)
String		refresh_token	Refresh-Token
String		expires_in	Angabe der Tokengültigkeit in Sekunden (599)
String		scope	Benennt die Zugriffsrechte

#### Beispiel Body:

```
{
  "access_token": "1234567890_Access-Token_NEU_34567890",
  "token_type": "bearer",
  "refresh_token": "1234567890_Refresh-Token_NEU_4567890",
  "expires_in": 599,
  "scope": "BANKING_RO BROKERAGE_RW SESSION_RW"
```

}

**Bitte beachten:**

- Eine Session-TAN bleibt auch nach Abruf eines Refresh-Tokens gültig.
- Ein Refresh-Token muss nicht nach jedem Request angefragt werden, sondern nur, wenn der aktuell gültige Access-Token zeitlich abläuft

**3.1.2 Revoke-Token**

DELETE https://api.comdirect.de/oauth/revoke

**Anmerkung:**

Mittels des Revoke-Token-Flows besteht die Möglichkeit, einen Access Token und den dazugehörigen Refresh-Token zu invalidieren. Im Rahmen des erfolgreichen Requests wird der Status 204 zurückgeliefert.

**REQUEST (= Löschen des aktuellen Access- und Refresh-Tokens)**

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-

**Beispiel Header:**

```
Accept:application/json
Content-Type:application/x-www-form-urlencoded
Authorization:Bearer 1234567890__Access-Token__1234567890
```

**RESPONSE (= leer)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description

Im Erfolgsfall werden Access- und Refresh-Token sowie die Session-TAN ungültig.

**3.2 Transaktionen freigeben**

Bereits bei der Authentifizierung wurde eine Session-TAN erzeugt, die bei Aufrufen von Transaktions-Schnittstellen zur Freigabe der Transaktionen verwendet werden kann.

Das Freigeben von Transaktionen erfolgt über zwei Schnittstellen: Validation + Execution.

In der Validation-Schnittstelle wird die durchzuführende Transaktion übergeben und geprüft. Sind alle Daten vorhanden, wird eine TAN-Challenge erzeugt. Der TAN-Typ ist aufgrund der existierenden Session-TAN „TAN\_FREI“.

Beispiel aus dem Response-Header einer Validierungs-Schnittstelle:

```
x-once-authentication-info: {
  "id": "7654321",
  "typ": "TAN_FREI",
  "availableTypes": ["M_TAN", "P_TAN"]
}
```

Beim Aufruf der eigentlichen Transaktions-Schnittstelle (Execution) muss im Header der Requests das Feld „x-once-authentication-info“ angegeben werden. Es transportiert die in der Validation mitgeteilte Challenge-ID an das comdirect-System zurück. Aufgrund der bereits vorhandenen TAN (Session-TAN) ist die Angabe des Feldes "x-once-authentication" nicht notwendig.

**Beispiel des Headers (Auszug):**

```
x-once-authentication-info:{"id":"7654321"} //id der Challenge
```

### 3.3 Header sämtlicher weiterer Schnittstellen

Die Header der in den nächsten Kapiteln beschriebenen Schnittstellen sind vom Aufbau her immer identisch. Nur bei den Transaktions-Schnittstellen werden die vier anzugebenden Felder wie oben beschrieben ergänzt.

**Beispiel Header:**

```
Accept:application/json
Content-Type:application/json
Authorization:Bearer 1234567890__Access-Token__1234567890
x-http-request-info:
  {"clientRequestId":{"sessionId":"123_beliebige_ID_fuer_Session_12",
    requestId:"123456789"}}
```

### 3.4 Standard REST-Objekte

Im Folgenden sind Standard REST-Objekte definiert, die in den verschiedenen fachlichen Objekten wiederverwendet werden.

#### 3.4.1 AmountValue

Parameter	Datentyp	Schreibbarkeit	Beschreibung
value	\$AmountString	read-only	Nominale in entsprechender Einheit
unit	String(3)	read-only	Währungen gemäß ISO-4217 (EUR, USD, GBP, ...) oder Sonderbezeichner für andere Einheiten: <ul style="list-style-type: none"> <li>XXX: Stücke</li> <li>XXC: Prozent</li> </ul>

Parameter	Datentyp	Schreibbarkeit	Beschreibung
			<ul style="list-style-type: none"> <li>XXM: Promille</li> <li>XXP: Punkte</li> <li>XXU: Unbekannt inkl. Sonstige (nicht spezifiziert)</li> </ul>

### 3.4.2 EnumText

Parameter	Datentyp	Schreibbarkeit	Beschreibung
key	String(<=40)	read-only	Eindeutiger Schlüsselwert für einen Enumerationstypen, z.B. \$uuld
text	String(<=65)	read-only	Anzeigetext in deutscher Sprache.

### 3.4.3 AmountString

Der Datentyp AmountString [Vorkommastellen+Nachkommastellen] repräsentiert einen Decimal-Wert (z.B. Java-Type BigDecimal). Das ist ein Datentyp, der in den JSON-Objekten anderer REST-Abfragen verwendet wird. Hierbei handelt es sich um einen JSON-String, der das folgende Format hat:

- Ggf. führendes '-'-Zeichen (kein führendes '+'-Zeichen und keine führenden Blanks!)
- Dezimal-Ziffern (ohne führende Nullen, es sei denn, der Vorkommaanteil ist nur '0')
- Bei Bedarf ein Dezimalpunkt '.' und Nachkommastellen (mindestens eine)
- Es werden keine Tausender-Trennzeichen oder Blanks verwendet

Bei der Verwendung dieses Typs kann die Anzahl von Vor- und Nachkomma-Stellen mit angegeben werden, z.B.: `$AmountString[13+2]`. Wird auf die Angabe der Klammer mit Vorkommastellen und Nachkommastellen verzichtet, wird die maximal mögliche Anzahl an Vorkomma- und Nachkommastellen zurückgegeben.

### 3.4.4 PercentageString

Dies ist ein AmountString, der einen Prozentwert (zwischen 0 und 100) angibt. Default 3 Vorkommastellen, Dezimalstellen 3: `PercentageString = AmountString[3+3]`, Bei Bedarf kann eine andere Anzahl Nachkommastellen explizit angegeben werden: `PercentageString[5] = AmountString[3+5]`. Müssen Prozentwerte größer als 100% angegeben werden, können auch die Vorkommastellen explizit angegeben werden, z.B.: `PercentageString[5+2]`.

### 3.4.5 CurrencyString

Der Typ CurrencyString repräsentiert den ISO-Code einer Währung. Das ist ein Datentyp, der in den JSON-Objekten anderer REST-Abfragen verwendet wird. Hierbei handelt es sich um einen JSON-String, der aus drei Buchstaben gemäß ISO-4217 besteht. Beispiel: EUR.

### 3.4.6 Datumstypen

Datentypen, die in den JSON-Objekten anderer REST-Abfragen verwendet werden. Hierbei handelt es sich jeweils um einen JSON-String, der das folgende mit ISO-8601 kompatible Format hat:

- \$DateString - "YYYY-MM-DD"
- \$DateTimeString - "YYYY-MM-DDThh:mm:ss+zz"
- \$TimestampString - "YYYY-MM-DDThh:mm:ss,ffffff+zz"

Dabei bedeutet:

- YYYY - Jahreszahl vierstellig
- MM - Monatszahl zweistellig (01 .. 12)
- DD - Tageszahl zweistellig (01 .. 31) [Achtung: Bei Valuta-Daten kann der 30.02. eines Jahres auftreten, auch wenn das kein echtes Datum ist!]
- T - "T" (Trenner zwischen Datum und Uhrzeit)
- hh - Stunde (00 .. 23), immer zweistellig
- mm - Minute (00 .. 59), immer zweistellig
- ss - Sekunde (00 .. 59), immer zweistellig
- ffffff - Sekundenbruchteile (bis zu sechs Stellen)
- zz - Zeitzone (01 für MEZ, 02 für MESZ)

## 4 Resource ACCOUNT

Mit den Schnittstellen der Resource Account können Sie auf die Salden und Umsätze Ihrer Konten zugreifen. Die in einigen URLs benötigte „accountId“ können Sie dem Response der Schnittstelle „/banking/clients/user/v1/accounts/balances“ entnehmen.

### 4.1 REST-API Ressourcen

Meth.	URI (Endpunkt)	Bemerkung
GET	/banking/clients/clientId/v2/accounts/balances	Abruf Kontoinformation einschließlich Cash-Saldo und Buying Power zu allen Konten
GET	/banking/v2/accounts/{accountId}/balances	Abruf Kontoinformation einschließlich Cash-Saldo und Buying Power
GET	/banking/v1/accounts/{accountId}/transactions	Abruf einer Liste von Kontoumsätzen für ein konkretes Konto

#### 4.1.1 Abruf AccountBalances alle Konten

GET /banking/clients/{clientId}/v2/accounts/balances

**REQUEST (= Abruf Kontostände aller Konten)**

Parameter Type	Parameter Name	Data Type	Description
Path	clientId	String	Literal „user“
Body	-	-	-

**Anmerkung:**

Mittels der Angabe von "clientId" können Sie alle Ihre Konten abrufen.

**Query-Paramter:**

- „without-attr=account“: es wird das Befüllen des Account-Objektes unterdrückt

### RESPONSE (= Salden aller Konten)

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
paging		index	Index erstes Konto
		matches	Anzahl Konten
values	AccountBalance[]	...	Liste der REST-Objekte AccountBalance in JSON

### HTTP Statuscodes:

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

#### 4.1.2 Abruf AccountBalances

GET /banking/v2/accounts/{accountId}/balances

### REQUEST (= Abruf Kontostand eines Kontos)

Parameter Type	Parameter Name	Data Type	Description
Path	accountId	String	Kontold
Body	-	-	-

### RESPONSE (= Saldo eines Kontos)

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
AccountBalance	Account	accountId	Liefert die Konto-Stammdaten zu diesem Konto zurück

### HTTP Statuscodes:

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

#### 4.1.3 Abruf Kontoumsätze

GET /banking/v1/accounts/{accountId}/transactions

### REQUEST (= Abruf aller Umsätze eines Kontos)

Parameter Type	Parameter Name	Data Type	Description
Path	accountId	String	Kontold

**Filter-Parameter:**

- transactionState: BOTH (Default); BOOKED; NOTBOOKED

**Query-Parameter:**

- „with-attr=account“: es wird das Account-Objekt in den aggregierten Informationen hinzugefügt

**RESPONSE (= Alle Umsätze eines einzelnen Kontos)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
paging		index	Index erster Kontoumsatz
		matches	Anzahl Kontoumsätze
aggregated	\$AccountTransactionAggregate		REST-Objekt AccountTransactionAggregate
values	\$AccountTransaction[]		Liste der REST-Objekte AccountTransaction

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND

**4.2 REST-Objekte****4.2.1 Account**

Parameter	Datentyp	Schreibbarkeit	Beschreibung		
accountId	String(<=40)	read-only	Kontold (uuld)		
account DisplayId	String(12)	read-only	Kontonummer		
currency	\$CurrencyString	read-only	Konto-Währung		
clientId	String(<=40)	read-only	Kundennummer		
accountType	\$EnumText	read-only	Kontotyp: Typ als Key, Value beinhaltet die Produktart in Englisch bzw. in der entsprechenden Sprache:		
			API accountType key	API accountType value	Beschreibung
			FX	Foreign Currency Account	Fremdwährungs- konto
			OF	Options- & Futures Trading Account	Options- & Futures-Konto



Parameter	Datentyp	Schreibbarkeit	Beschreibung
			CA Checking Account Girokonto
			DAS Direct Access Savings-Plus Account Tagesgeld-Plus Konto
			CFD Contract for Difference Account Contract for Difference Konto
			SA Settlement Account Tagesgeld-/ Verrechnungskonto
			LLA Lombard Loan Account Wertpapier-Kreditkonto
iban	String(<=34) + NULL	read-only	IBAN, sofern verfügbar
creditLimit	\$AmountValue + NULL	read-only	Kreditlinie, sofern verfügbar

#### 4.2.2 AccountBalance

Parameter	Datentyp	Schreibbarkeit	Beschreibung
account	\$Account	read-only	Die Konto-Stammdaten zu diesem Konto
accountId	String(<=40)	read-only	Kontoidentifikation (uuld)
balance	\$AmountValue	read-only	Aktueller Kontostand (Saldo)
balanceEUR	\$AmountValue	read-only	Aktueller Kontostand (Saldo) in EUR
availableCashAmount	\$AmountValue	read-only	Aktueller Kontostand+Kreditlinie-Summe aller bereits disponierten, aber nicht gebuchten Geldbeträge: =maximaler Verfügungsrahmen
availableCashAmountEUR	\$AmountValue	read-only	Aktueller Kontostand+Kreditlinie-Summe aller bereits disponierten, aber nicht gebuchten Geldbeträge in EUR: =maximaler Verfügungsrahmen in EUR

#### 4.2.3 AccountTransaction

Parameter	Datentyp	Schreibbarkeit	Beschreibung
bookingStatus	String	read-only	Status der Kontotransaktion: <ul style="list-style-type: none"> <li>BOOKED</li> </ul>

Parameter	Datentyp	Schreibbarkeit	Beschreibung																												
			<ul style="list-style-type: none"><li>NOTBOOKED</li></ul>																												
bookingDate	\$DateString+NULL	read-only	Buchungstag des Umsatzes im Format YYYY-MM-DD																												
amount	\$AmountValue	read-only	Umsatzwert																												
remitter	\$AccountInformation	read-only	Enthält die Kontoverbindungsinformationen über den Inhabernamen, IBAN und BIC des Auftraggebers.																												
debtor	\$AccountInformation	read-only	Enthält die Kontoverbindungsinformationen über den Inhabernamen, IBAN und BIC des Zahlungspflichtigen.																												
creditor	\$AccountInformation	read-only	Enthält die Kontoverbindungsinformationen über den Inhabernamen, IBAN und BIC des Zahlungsempfängers.																												
reference	String	read-only	Eine eindeutige Referenznummer für diesen Umsatz.																												
endToEndReference	String	read-only	Gibt die End-to-End-Referenz des Umsatzes zurueck, falls es sich um eine Lastschrift handelt.																												
valutaDate	String	read-only	Valuta Datum des Umsatzes - es muss sich nicht um ein valides Datum handeln (z.B. 30.02.2019)																												
directDebitCreditorId	String	read-only	Gibt die Gläubigeridentifikationsnummer des Umsatzes zurueck, falls es sich um eine Lastschrift handelt.																												
directDebitMandateId	String	read-only	Gibt die Mandatsreferenz des Umsatzes zurueck, falls es sich um eine Lastschrift handelt.																												
transactionType	\$EnumText	read-only	Die konkrete Bezeichnung des Geschäftsvorgangs (Kategorie): <table><tr><th>transactionType</th><th>Übersetzung</th></tr><tr><td>Sparplan</td><td>Saving Plan</td></tr><tr><td>Wertpapier</td><td>Securities</td></tr><tr><td>Geldanlage</td><td>Investment Saving</td></tr><tr><td>Bankgebühren</td><td>Bank fees</td></tr><tr><td>Sonstiges</td><td>Miscellaneous</td></tr><tr><td>Bar</td><td>Cash</td></tr><tr><td>Zinsen / Dividenden</td><td>Interest / Dividends</td></tr><tr><td>Devisen</td><td>Currency Exchange</td></tr><tr><td>Sorno</td><td>Cancellation</td></tr><tr><td>Scheck</td><td>Cheque</td></tr><tr><td>Lastschrift</td><td>Direct Debit</td></tr><tr><td>Überweisung</td><td>Transfer</td></tr><tr><td>Kartenverfügung</td><td>Card transaction</td></tr></table>	transactionType	Übersetzung	Sparplan	Saving Plan	Wertpapier	Securities	Geldanlage	Investment Saving	Bankgebühren	Bank fees	Sonstiges	Miscellaneous	Bar	Cash	Zinsen / Dividenden	Interest / Dividends	Devisen	Currency Exchange	Sorno	Cancellation	Scheck	Cheque	Lastschrift	Direct Debit	Überweisung	Transfer	Kartenverfügung	Card transaction
transactionType	Übersetzung																														
Sparplan	Saving Plan																														
Wertpapier	Securities																														
Geldanlage	Investment Saving																														
Bankgebühren	Bank fees																														
Sonstiges	Miscellaneous																														
Bar	Cash																														
Zinsen / Dividenden	Interest / Dividends																														
Devisen	Currency Exchange																														
Sorno	Cancellation																														
Scheck	Cheque																														
Lastschrift	Direct Debit																														
Überweisung	Transfer																														
Kartenverfügung	Card transaction																														

Parameter	Datentyp	Schreibbarkeit	Beschreibung	
			Sorten (Kasse)	Foreign Currency exchange
			Geldautomat	ATM Withdrawal
			Geldanlage	Savings
			Dauerauftrag	Standing Order
remittanceInfo	String	read-only	Der Buchungstext dieses Umsatzes. Der Buchungstext kann mehrere Zeilen enthalten, die jeweils 35 Zeichen lang sind. Bei gebuchten Umsätzen werden den einzelnen Zeilen Zeilennummern vorangestellt.	
newTransaction	Boolean	read-only	Gibt an, ob dieser Umsatz bereits vom Kunden im Web gesehen wurde.	

## 5 Resource DEPOT

Mit den Schnittstellen der Resource Depot können Sie Ihre Depotsalden, Depotpositionen und die Depotumsätze abfragen. Die in den URLs benötigte „depotId“ können Sie dem Response der Schnittstelle „/brokerage/clients/clientId/v3/depots“ entnehmen.

### 5.1 REST-API Ressourcen

Meth.	URI (Endpunkt)	Bemerkung
GET	/brokerage/clients/clientId/v3/depots	Abruf Depot
GET	/brokerage/v3/depots/{depotId}/positions	Abruf der Depotpositionen
GET	/brokerage/v3/depots/{depotId}/positions/{positionId}	Abruf einer einzelnen Depotposition
GET	/brokerage/v3/depots/{depotId}/transactions	Abruf einer Liste von Depotumsätzen für ein konkretes Depot

#### 5.1.1 Abruf Depots

GET /brokerage/clients/{clientId}/v3/depots

##### REQUEST (= Abruf aller Depots)

Parameter Type	Parameter Name	Data Type	Description
Path	clientId	String	Literal „user“
Body	-	-	-

##### Anmerkung:

Mittels der Angabe von clientId können Sie Ihr Depot abrufen.

##### RESPONSE (= Alle Depots)

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
paging		index	Index erstes Depot
		matches	Anzahl Depots, i.d.R. nur ein Depot, sofern gefunden
values	Depot[]	...	Liste der REST-Objekte Depot in JSON

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

### 5.1.2 Abruf Depotbestand und/oder Positionen

GET /brokerage/v3/depots/{depotId}/positions

**Request (= Abruf aller Positionen eines Depots)**

Parameter Type	Parameter Name	Data Type	Description
Path	depotId	String	Depot, dessen Positionen abgerufen werden sollen
Body	-	-	-

**Filter-Parameter:**

- instrumentId: filtert die Positionen auf eine WKN oder ISIN und instrumentUUID

**Query-Parameter:**

- „with-attr=instrument“: es wird das Befüllen der Instrument-Objekte angefordert
- „without-attr=depot“: es wird das Befüllen des Depot-Objektes unterdrückt
- „without-attr=positions“: es wird das Befüllen des Positions-Objektes unterdrückt

Folgende Query-Parameter sind kombinierbar:

**Matrix der Query-Parameter**

V: möglich

X: unmöglich

GET /depots/{depotId}/positions	with-attr=instrument	without-attr=depot	without-attr=positions
with-attr=instrument		V	X
without-attr=depot			V
without-attr=positions			

**Response (= Alle Positionen eines einzelnen Depots)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
paging		index	Index erster Depotposition
		matches	Anzahl Depotpositionen
aggregated	DepotAggregation		Rest-Objekt DepotAggregation

Objects	Nested Objects	Keys	Description
values	DepotPosition[]	...	Liste der REST-Objekte Depotpositionen in JSON

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

**5.1.3 Abruf einer Position des Depots**

GET /brokerage/v3/depots/{depotId}/positions/{positionId}

**Request (= Abruf einer Position des Depots)**

Parameter Type	Parameter Name	Data Type	Description
Path	depotId	String	DepotId
Path	positionId	String	PositionId
Body	-	-	-

**Query-Parameter:**

- „with-attr=instrument“: es wird das Befüllen des Attributs 'instrument' bei der Depotposition veranlasst
- „without-attr=depot“: es wird die Lieferung des Depot-Objekts unterdrückt
- „without-attr=positions“: es wird die Lieferung des Position-Objekts unterdrückt

**Response (= selektierte Position eines Depots)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
values	DepotPosition	...	REST-Objekt DepotPosition

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

**5.1.4 Abruf Depotumsätze**

GET /brokerage/v3/depots/{depotId}/transactions

**Request (= Abruf aller Umsätze eines Depots)**

Parameter Type	Parameter Name	Data Type	Description
Path	depotId	String	Depot, dessen Umsätze abgerufen werden sollen
Body	-	-	-

**Filter-Parameter:**

- WKN
- ISIN
- instrumentId
- max-bookingDate: Format JJJJ – MM - TT
- transactionDirection: IN; OUT
- transactionType: BUY; SELL; TRANSFER\_IN; TRANSFER\_OUT
- bookingStatus: BOOKED; NOTBOOKED; BOTH
- min-transactionValue
- max-transactionValue

**Query-Parameter:**

- „without-attr=instrument“ es wird das Befüllen des Attributs 'instrument' in den Depottransaktionen unterdrückt

**Response (= Alle Umsätze eines einzelnen Depots)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
paging		index	Index erster Depotumsatz
		matches	Anzahl Depotumsätze
values	DepotTransaction[]	...	Liste der REST-Objekte DepotTransaction in JSON

**HTTP Statuscodes:**

- 200 – OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

**5.2 REST-Objekte****5.2.1 Depot**

Parameter	Datentyp	Schreibbarkeit	Beschreibung
depotId	String(<=40)	read-only	Depotidentifikation (UUID)

Parameter	Datentyp	Schreibbarkeit	Beschreibung
depotDisplayId	String(7)	read-only	Depotnummer
clientId	String(<=40)	read-only	Zugangskennung des Kunden, der dem Depot eindeutig zugeordnet ist
defaultSettlementAccountId	String(<=40)	read-only	Default Verrechnungskontonummer, welche dem Depot eindeutig zugeordnet ist
settlementAccountIds[]	String(<=40)[] + NULL	read-only	Liste von weiterer Verrechnungskontennummern, welche dem Depot zugeordnet sind

### 5.2.2 DepotPosition

Parameter	Datentyp	Schreibbarkeit	Beschreibung
depotId	String(<=40)	read-only	Depotidentifikation (UUID)
positionId	String(<=40)	read-only	Positionsnummer im Depot, wird als UUID generiert
wkn	String(6)	read-only	WKN
custodyType	String(3)	read-only	Verwahrart
quantity	\$AmountValue	read-only	Stückanzahl oder Nominalbetrag im Falle einer Prozentnotiz
availableQuantity	\$AmountValue	read-only	Verfügbare Stückanzahl oder Nominalbetrag im Falle einer Prozentnotiz
currentPrice	\$Price	read-only	Kurswert, sofern verfügbar
purchasePrice	\$AmountValue + NULL	read-only	Anschaffungskurs, sofern verfügbar
prevDayPrice	\$Price + NULL	read-only	Kurs Vortag, sofern verfügbar
currentValue	\$AmountValue	read-only	Positionswert zu aktuellen Preisen
purchaseValue	\$AmountValue + NULL	read-only	Durchschnittlicher Anschaffungswert Position
profitLossPurchaseAbs	\$AmountValue + NULL	read-only	Profit/Loss zum Anschaffungswert absolut, sofern verfügbar
profitLossPurchaseRel	\$PercentageString + NULL	read-only	Profit/Loss zum Anschaffungswert absolut, sofern verfügbar
profitLossPrevDayAbs	\$AmountValue + NULL	read-only	Profit/Loss zum Vortageskurs absolut, sofern verfügbar
profitLossPrevDayRel	\$PercentageString + NULL	read-only	Profit/Loss zum Vortageskurs in Prozent, sofern verfügbar

Parameter	Datentyp	Schreibbarkeit	Beschreibung
instrument	\$Instrument	read-only	Informationen zum Instrument der Depotposition

**Anmerkung:**

1. Quantity: Menge wird als Dezimalbetrag und Einheit (Stückzahl, Prozent, Währung, ...) angegeben
2. availableQuantity: Anzahl der tatsächlich verfügbaren und handelbaren Stücke, d.h. exklusive eventuell geblockter Mitarbeiteraktien oder abdisponierter Stücke.

**5.2.3 DepotTransaction**

Parameter	Datentyp	Schreibbarkeit	Beschreibung												
transactionId	String(<=40)	read-only	Depotumsatz (UUID)												
bookingStatus	String(<=10)	read-only	Status der Depottransaktion: <ul style="list-style-type: none"><li>BOOKED</li><li>NOTBOOKED</li></ul>												
bookingDate	\$DateString + NULL	read-only	Buchungstag des Umsatzes												
businessDate	\$DateString	read-only	Geschäftstag des Umsatzes												
quantity	\$AmountValue	read-only	Stückanzahl oder Nominalbetrag im Falle einer Prozentnotiz												
instrumentId	String(<=40)	read-only	instrumentId als UUID												
instrument	\$Instrument + NULL	read-only	Informationen zum Instrument der Depotposition												
executionPrice	\$AmountValue	read-only	Ausführungskurs												
transactionValue	\$AmountValue	read-only	Umsatzwert												
transactionDirection	String(<=3)	read-only	ENUM {IN/OUT}												
transactionType	\$EnumText	read-only	<table><tr><th>transactionType</th><th>Text deutsch</th><th>Übersetzung en_us</th></tr><tr><td>SELL</td><td>Verkauf</td><td>Sell</td></tr><tr><td>OTHER</td><td>Sonstige</td><td>Other</td></tr><tr><td>BUY</td><td>Kauf</td><td>Buy</td></tr></table>	transactionType	Text deutsch	Übersetzung en_us	SELL	Verkauf	Sell	OTHER	Sonstige	Other	BUY	Kauf	Buy
transactionType	Text deutsch	Übersetzung en_us													
SELL	Verkauf	Sell													
OTHER	Sonstige	Other													
BUY	Kauf	Buy													



Parameter	Datentyp	Schreibbarkeit	Beschreibung		
			TRANSFER_IN	Depot-übertrag eingehend	Incoming securities account transfer
			TRANSFER_OUT	Depot-übertrag ausgehend	Outgoing securities account transfer

### 5.2.4 Price

Parameter	Datentyp	Schreibbarkeit	Beschreibung
price	\$AmountValue	read-only	Kurs, d.h. Quote/Taxe (unverbindlicher oder verbindlicher Quote) oder festgestellter (umsatzloser oder umsatzbehafteter) Preis
type	String(<=5)	read-only	<p>LAST: Last Price (letzter festgestellter Kurs im elektronischen oder Auktionshandel)</p> <p>BID: Bid (höchstes Kaufgebot im elektronischen Handel bzw. Kauftaxe des Market Makers oder Emittenten)</p> <p>ASK: Ask (niedrigstes Verkaufsangebot im elektronischen Handel bzw. Verkaufstaxe des Market Makers oder Emittenten)</p> <p>MID: Mittelkurs zwischen Bid &amp; Ask</p>
quantity	\$AmountValue + NULL		Stückanzahl oder Nominalbetrag im Falle einer Prozentnotiz
priceDateTime	\$DateTimeString	read-only	<p>Datums- und Zeitstempel bis auf Sekundenangabe in Format</p> <p>YYYY-MM-DDThh:mm:ss+zz</p>

## 6 Resource INSTRUMENT

Mit der Schnittstelle der Resource Instrument können Sie sich zu einer WKN, ISIN oder einem Symbol Wertpapierinformationen zurückgeben lassen.

### 6.1 REST-API Ressourcen

Meth.	URI (Endpunkt)	Bemerkung
GET	/brokerage/v1/instruments/{instrumentId}	Abruf der Informationen zu einem Instrument (Wertpapier)

### 6.1.1 Abruf Instrument

GET /brokerage/v1/instruments/{instrumentId}

#### REQUEST (= Abruf der Instrumenten-Informationen)

Parameter Type	Parameter Name	Data Type	Description
Path	instrumentId	String	instrumentId: entweder wkn ODER isin ODER mnemonic
Body	-	-	-

#### Anmerkung:

instrumentId kann sein:

- wkn (6-stellig)
- isin (12-stellig)
- mnemonic (Symbol):

Bei der Schreibweise bitte auf Klein/Großbuchstaben achten.

#### Query-Paramter:

- „with-attr=orderDimensions“: es wird das OrderDimension-Objekt befüllt
- „with-attr=fundDistribution“: es wird das FundDistribution-Objekt befüllt, wenn es sich bei dem Wertpapier um einen Fonds handelt
- „with-attr=derivativeData“: es wird das DerivativeData-Objekt befüllt, wenn es sich bei dem Wertpapier um ein Derivat handelt
- „without-attr=staticData“: gibt das StaticData -Objekt nicht zurück

#### RESPONSE (= Alle angeforderten Instrument-Stammdaten)

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
paging		index	Index erstes Attribut
		matches	Anzahl Attribute
values	Instrument[]		REST-Objekt Instrument in JSON; es wird immer nur ein Listenelement zurückgegeben

#### HTTP Statuscodes:

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

## 6.2 REST-Objekte

### 6.2.1 Instrument

Parameter	Datentyp	Schreibbarkeit	Beschreibung
instrumentId	String(<=40) + NULL	read-only	InstrumentId (UUID), eindeutige Kennung eines Instrumentes (Wertpapier, Derivat etc.)
wkn	String(6)	read-only	WKN
mnemonic	String(<=5) + NULL	read-only	Wertpapierkürzel
isin	String(12)	read-only	ISIN
name	String(<=60)	read-only	Name des Instruments
shortName	String(<=25)	read-only	Kurzbezeichnung des Instrumentes
staticData	\$StaticData + NULL	read-only	Statische Daten des Wertpapiers, u.a. Notierung, Instrument-Typ
orderDimensions	\$Dimensions + NULL	read-only	Liste der Handelsplätze inklusive der Attribute (Dimensions)
fundsDistribution	\$FundDistribution + NULL	read-only	Zusätzliche Fondsdaten, wenn das Instrument ein Fonds ist
derivativeData	\$DerivativeData + NULL	read-only	Zusätzliche Daten eines Derivats

### 6.2.2 StaticData

Parameter	Datentyp	Schreibbarkeit	Beschreibung
notation	String(3)	read-only	Notierungs-Einheit des Wertpapiers (ENUM): <ul style="list-style-type: none"> <li>• XXX (deutsch: STK)</li> <li>• XXC (deutsch: PRZ)</li> <li>• XXM (deutsch: PRM)</li> <li>• XXP (deutsch: PKT)</li> <li>• XXU (deutsch: UNB, SON)</li> </ul>
currency	\$Currency String	read-only	Depotwährung des Wertpapiers, z.B. bei Anleihen, Renten- und offenen Immobilienfonds; zusätzlich zu der ISO 4217-Währungscodierung sind folgende Werte möglich: <ul style="list-style-type: none"> <li>• XXX (deutsch: STK)</li> <li>• XXP (deutsch: PKT)</li> <li>• XXU (deutsch: UNB, SON)</li> </ul>
instrumentType	String(<=30)	read-only	Wertpapierart als ENUM: <ul style="list-style-type: none"> <li>• SHARE (deutsch: AKTIE)</li> <li>• BONDS (deutsch: ANLEIHE)</li> <li>• SUBSCRIPTION_RIGHT (deutsch: BEZUGSRECHT)</li> <li>• ETF (deutsch: ETF)</li> </ul>

Parameter	Datentyp	Schreibbarkeit	Beschreibung
			<ul style="list-style-type: none"> <li>• PROFIT_PART_CERTIFICATE (deutsch: GENUSSSCHEIN)</li> <li>• FUND (deutsch: FONDS)</li> <li>• WARRANT (deutsch: OPTIONSSCHEIN)</li> <li>• CERTIFICATE (deutsch: ZERTIFIKAT)</li> <li>• NOT_AVAILABLE (deutsch: NICHT_VERFUEGBAR)</li> </ul>
priipsRelevant	Boolean	read-only	Kennzeichen, welches anzeigt, inwieweit die PRIIPs-Verordnung für das Wertpapier relevant ist
kidAvailable	Boolean	read-only	Kennzeichen, das vom Emittenten ein Basisinformationsblatt vorliegt. Vor Orderanlage soll in diesem Fall ein statischer Hinweis angezeigt werden.
shippingWaiverRequired	Boolean	read-only	Kennzeichen bei Fonds, welches angibt, dass bei Einstellung einer Kauforder Frontend-seitig (z.B. per Checkbox) eine explizite Verzichtserklärung auf die Zusendung von Fonds-Verkaufsunterlagen abzugeben ist und ein entsprechender Hinweis anzugeben ist. Ohne explizite Verzichtserklärung ist die Orderanlage im Frontend zu verhindern.
fundRedemptionLimited	Boolean	read-only	Kennzeichen für das Vorliegen eines offenen Immobilienfonds. Im Prozess der Prevalidierung der Einstellung einer Kauforder soll in diesem Fall ein statischer Hinweis mit Bezug auf die beschränkten Rücknahmemöglichkeiten angezeigt werden.

### 6.2.3 DerivativeData

DerivativeData enthält zusätzliche Daten eines derivativen Instruments.

Parameter	Datentyp	Schreibbarkeit	Beschreibung
underlyingInstrument	\$Instrument + NULL	read-only	eindeutige ID des Basiswertes
underlyingPrice	\$Price + NULL	read-only	Kurs des Underlyings
certificateType	String + NULL	read-only	Typ des Zertifikates: <ul style="list-style-type: none"> <li>• Hebel</li> <li>• Index</li> <li>• Basket</li> <li>• Hedge-Fonds-Zertifikat</li> <li>• Discount</li> <li>• Aktienanleihe</li> <li>• Bandbreite</li> <li>• Outperformance</li> <li>• Express</li> </ul>

Parameter	Datentyp	Schreibbarkeit	Beschreibung												
			<ul style="list-style-type: none"><li>Bonus</li><li>Kapitalschutz</li></ul>												
rating	\$Rating + NULL	read-only	Rating												
strikePrice	\$AmountValue	read-only	Basispreis des Underlyings												
leverage	String + NULL	read-only	Hebel des Derivats												
multiplier	String + NULL	read-only	Bezugsverhältnis des Underlyings												
expiryDate	\$DateString + NULL	read-only	Laufzeitende (bspw. eines Derivates)												
yieldPA	\$PercentageString + NULL	read-only	Rendite p.a.												
remainingTermInYears	\$AmountString + NULL	read-only	Restlaufzeit (expiryDate-today)												
nominalRate	\$PercentageString + NULL	read-only	Nominalzins												
warrantType	String + NULL	read-only	Typ des Optionsscheins <ul style="list-style-type: none"><li>Call</li><li>Put</li></ul>												
maturityDate	\$DateString + NULL	read-only	Fälligkeitsdatum (betrifft vorrangig Anleihen)												
interestPaymentDate	\$DateString + NULL	read-only	Datum der Zins-/Kuponzahlung einer Anleihe												
interestPaymentInterval	String + NULL	read-only	Intervall der Zins-/Kuponzahlung einer Anleihe.  Enum mit folgenden Ausprägungen: <table><tr><th>Bedeutung</th><th>Übersetzung</th></tr><tr><td>ANNUALLY</td><td>jährlich</td></tr><tr><td>SEMIANNUALLY</td><td>halbjährlich</td></tr><tr><td>QUARTERLY</td><td>vierteljährlich</td></tr><tr><td>MONTHLY</td><td>monatlich</td></tr><tr><td>OTHER</td><td>andere</td></tr></table>	Bedeutung	Übersetzung	ANNUALLY	jährlich	SEMIANNUALLY	halbjährlich	QUARTERLY	vierteljährlich	MONTHLY	monatlich	OTHER	andere
Bedeutung	Übersetzung														
ANNUALLY	jährlich														
SEMIANNUALLY	halbjährlich														
QUARTERLY	vierteljährlich														
MONTHLY	monatlich														
OTHER	andere														
underlyingInstrument	\$Instrument + NULL	read-only	eindeutige ID des Basiswertes												

## 6.2.4 Rating

Parameter	Datentyp	Schreibbarkeit	Beschreibung
morningstar	String+NULL	read-only	Rating (Fonds)
moodys	String+NULL	read-only	Rating (Anleihe)

## 6.2.5 FundDistribution

Parameter	Datentyp	Schreibbarkeit	Beschreibung
currency	\$CurrencyString	read-only	ISO 4217-WährungsCODIERUNG
regularIssueSurcharge	\$PercentageString	read-only	regulärer Ausgabeaufschlag
discountIssueSurcharge	\$PercentageString	read-only	Discount Ausgabeaufschlag
reducedIssueSurcharge	\$PercentageString	read-only	reduzierter Ausgabeaufschlag
investmentCategory	String	read-only	Anlagekategorie <ul style="list-style-type: none"> <li>• Aktienfonds</li> <li>• Aktien</li> <li>• Rentenfonds</li> <li>• Renten</li> <li>• Geldmarktfonds</li> <li>• Geldmarkt</li> <li>• Gemischte Fonds</li> <li>• Dachfonds</li> <li>• Immobilienfonds</li> <li>• Alternative Fonds</li> <li>• Strukturierte Fonds</li> <li>• Alternative Investments</li> <li>• sonstige ETF's</li> </ul>
totalExpenseRatio	\$PercentageString	read-only	Laufende Kosten in %
rating	\$Rating + NULL	read-only	Rating

## 7 Resource ORDER

Die Schnittstellen der Resource Order ermöglicht Ihnen die Anlage, Änderung und Streichung von Orders. Außerdem können Sie das Orderbuch und den Status einzelner Orders abfragen sowie sich den Kostenausweis für eine Order anzeigen lassen.

Die bei der Orderanlage benötigte „venueId“ (Handelsplatz-Id) kann dem Response der Schnittstelle „/brokerage/v1/orders/dimensions“ entnommen werden.

### 7.1 REST-API Ressourcen

Meth.	URI (Endpunkt)	Bemerkung
GET	/brokerage/v3/orders/dimensions	Abruf der Handelsplatz- und Orderzusätze für das jeweilige Instrument
GET	/brokerage/depots/{depotId}/v3/orders	Abruf des Orderbuches
GET	/brokerage/v3/orders/{orderId}	Abruf einer bestimmten Order
POST	/brokerage/v3/orders/prevalidation	Vorprüfung der Order
POST	/brokerage/v3/orders/validation	Prüfung einer Orderanlage und Triggerung einer TAN-Challenge im Falle der Nicht-Nutzung einer Session-TAN

Meth.	URI (Endpunkt)	Bemerkung
POST	/brokerage/v3/orders/costindicationexante	Generierung des ex-ante Kostenausweises bei Orderanlage auf Basis der Orderdaten
POST	/brokerage/v3/orders	Orderanlage
POST	/brokerage/v3/orders/{orderId}/prevalidation	Vorprüfung der Orderänderung
POST	/brokerage/v3/orders/{orderId}/validation	Prüfung einer Orderänderung oder Orderlöschung und Triggerung einer TAN-Challenge im Falle der Nicht-Nutzung einer Session-TAN
POST	/brokerage/v3/orders/{orderId}/costindicationexante	Generierung des ex-ante Kostenausweises bei Orderänderung auf Basis der Orderdaten
PATCH	/brokerage/v3/orders/{orderId}	Orderänderung
DELETE	/brokerage/v3/orders/{orderId}	Orderlöschung

### 7.1.1 Abruf OrderDimensionen

GET /brokerage/v3/orders/dimensions

#### Anmerkung:

Die OrderDimensions-Schnittstelle enthält die Zuordnung eines Instrumentes zu den Handelsplätzen, an denen dieses handelbar ist. Die Handelbarkeit wird über eine Map einer jeden venue auf die Handelsplatz-spezifischen Informationen abgebildet. Es sind ebenso die Ordertypen sowie Handelsplatz- und Ordertyp-spezifischen Attribute enthalten.

Die im entsprechenden JSON-Objekt definierte Hierarchie ist:

- Handelsplatz
  - Währungen
  - Geschäftsart (wegen Verschachtelung bei Kombinationsorders)
  - Ordertyp
    - Limitzusatz
    - Handelsbeschränkungen/-hinweis
  - Gültigkeitstypen

#### Bitte beachten:

Im Response der Schnittstelle werden im Header ggf. Informationen zu Zielmarktkriterien, wesentlichen Anlegerinformationen und dem Basisinformationsblatt angegeben. Diese Informationen sind vor der Orderaufgabe zu prüfen.

#### REQUEST (= Abruf der Handelbarkeit eines Instruments)

Parameter Type	Parameter Name	Data Type	Description
Body	-	-	-

**Filter-Parameter:**

- instrumentId
- WKN
- ISIN
- mneomic
- venueId: Mit Hilfe der venueId, welche als UUID eingegeben werden muss, kann auf einen Handelsplatz gefiltert werden
- side: Entspricht der Geschäftsart. Filtermöglichkeiten sind BUY oder SELL
- orderType: Entspricht dem Ordertypen (bspw. LIMIT, MARKET oder ONE\_CANCELS\_OTHER)
- type: Mittels EXCHANGE oder OFF kann unterschieden werden, ob nach einem Börsenplatz (EXCHANGE) oder einem LiveTrading Handelsplatz (OFF) gefiltert werden soll

**RESPONSE (= Alle Instrument-/Handelsplatz-Attribute (OrderDimensionen) gemäß Filterung auf das jeweilige Instrument)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
paging		index	Index erstes Attribut
		matches	Anzahl Attribute
values	Dimensions[]		REST-Objekt Dimensions in JSON; es wird nur ein Listenelement zurückgegeben, welches die Liste der Dimensions beinhaltet

**HTTP Statuscodes:**

- 200 - OK
- 422 - UNPROCESSABLE ENTITY

**7.1.2 Abruf Orders (Orderbuch)**

GET /brokerage/depots/{depotId}/v3/orders

**REQUEST (= Abruf aller Orders des angegebenen Depots)**

Parameter Type	Parameter Name	Data Type	Description
Path	depotId	String	DepotId (UUID)
Body	-	-	-

**Filter-Parameter:**

- orderStatus
- venueId
- side
- orderType



**Query-Parameter:**

- „without-attr=executions“: es wird das Befüllen des Executions-Objekts unterdrückt
- „with-attr=instrument“: es wird das Befüllen des Instrument-Objekts angefordert

**RESPONSE (= Alle Orders des Depots eines Kunden)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
paging		index	Index erste Order
		matches	Anzahl Orders
values	Order[]	...	Liste der REST-Objekte Order in JSON

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

**7.1.3 Abruf Order (Einzelorder)**

GET /brokerage/v3/orders/{orderId}

**Anmerkung:**

Liefert im Kontext Depot für eine OrderId die aktuellen Orderinformationen.

**REQUEST (= Abruf einer Order des Depots eines Kunden)**

Parameter Type	Parameter Name	Data Type	Description
Path	orderId	String	OrderId
Body	-	-	-

**RESPONSE (= Spezifizierte Order des Depots)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Order		orderId	OrderId

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND

**Beispiel Abruf Order:**

GET /brokerage/v3/orders/1234\_\_order\_uuid\_\_\_\_1234:

### 7.1.4 Anlage Prevalidation Orderanlage

POST /brokerage/v3/orders/prevalidation

**Anmerkung:**

Bei diesem Aufruf werden alle über das REST-Objekt Order übergebenen Attribute der Order einzeln auf Validität geprüft. Ebenfalls validiert werden Attribut-übergreifende Werte. Die Attribute werden bei dieser Vorvalidierung weder auf Pflichtfeld noch auf Vollständigkeit geprüft; dies erfolgt erst bei der endgültigen Validierung (s.u.).

Die Vorvalidierung dient der Prüfung von Order-Parametern während einer Frontend-Eingabe.

**REQUEST (= Anlage einer Vorvalidierung einer Order)**

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-
Body	Order	Order	JSON-Objekt Order

**RESPONSE (= Spezifizierte Order des Depots)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Order		-	JSON-Objekt Order

**HTTP Statuscodes:**

- 200 - OK
- 422 - UNPROCESSABLE ENTITY

### 7.1.5 Anlage Validation Orderanlage

POST /brokerage/v3/orders/validation

**Anmerkung:**

Bei diesem Aufruf wird das REST-Objekt wie bei der eigentlichen Orderanlage übergeben und validiert. Die Validierung beinhaltet die Prüfung auf Vollständigkeit aller Pflicht-Parameter.

**Bitte beachten:**

Der Validation-Request muss vor der Orderanlage aufgerufen werden, da in der Response eine TAN-Challenge übermittelt wird, die an die eigentliche Orderanlage übergeben werden muss.

**REQUEST (= Anlage einer Order-Validierung)**

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-
Body	Order	Order	JSON-Objekt Order

**RESPONSE (= Spezifizierte Order sowie TAN-Challenge (Header))**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Order		-	JSON-Objekt Order

**Anmerkung:**

Der Response-Header enthält die TAN-Challenge gemäß Kapitel 3.2.

**HTTP Statuscodes:**

- 201 - CREATED
- 422 - UNPROCESSABLE ENTITY

**7.1.6 Anlage Ex-Ante Kostenausweis für eine Orderanlage**

POST /brokerage/v3/orders/costindicationexante

**Anmerkung:**

Generiert für eine neu eingestellte Order den ex-ante Kostenausweis:

Bei diesem Aufruf wird das REST-Objekt Order wie bei der eigentlichen Orderanlage übergeben sowie die für den Kostenausweis erforderlichen Attribute validiert und hinsichtlich Kostenausweis auf Vollständigkeit geprüft. Eine TAN-Eingabe ist hierzu nicht erforderlich. Sofern die Ordervalidierung erfolgreich war, die Kostendaten jedoch nicht ermittelt werden konnten, wird dies gesondert gekennzeichnet und ein Link auf den generischen Kostenausweis zurückgegeben.

**REQUEST (= Anlage des ex-ante Kostenausweises einer neu eingestellten Order)**

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-
Body	Order	Order	JSON-Objekt Order

**RESPONSE (= ex-ante Kostenausweis einer Order)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
CostIndicationExAnte[]		-	Liste JSON-Objekt CostIndicationExAnte

**HTTP Statuscodes:**

- 201 - CREATED
- 422 - UNPROCESSABLE ENTITY

### 7.1.7 Anlage Order

POST /brokerage/v3/orders

**Anmerkung:**

Bei diesem Aufruf wird das REST-Objekt Order mit einer TAN-Challenge übergeben, validiert und eingestellt.

Die Orderanlage unterscheidet gemäß Ordertyp hinsichtlich der Zuordnung zur Einzelorder:

1. Market
2. Limit
3. RfQ
4. Stop Market
5. Stop Limit
6. Trailing Stop
7. Trailing Stop Limit

bzw. Kombinationsorder:

- OCO-Order
- Next-Order.

**REQUEST (= Eingabe einer Order)**

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-
Body	Order	Order	JSON-Objekt Order

**Anmerkung:**

Der Request-Header benötigt die TAN-Challenge aus der Validations-Schnittstelle (siehe Kapitel 3.2).

**RESPONSE (= Spezifizierte Order)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Order		orderId	OrderId

**HTTP Statuscodes:**

- 201 - CREATED
- 422 - UNPROCESSABLE ENTITY

### 7.1.8 Anlage Prevalidation Orderänderung

POST /brokerage/v3/orders/{orderId}/prevalidation

**Anmerkung:**

Bei diesem Aufruf werden alle über das REST-Objekt Order geänderten Attribute der Order einzeln auf Validität geprüft. Ebenfalls validiert werden Attribut-übergreifende Werte. Die Attribute werden bei dieser Vorvalidierung weder auf Pflichtfeld noch auf Vollständigkeit geprüft; dies erfolgt erst bei der endgültigen Validierung (s.u.).

Die Vorvalidierung dient z.B. der Prüfung der Änderung von Order-Parametern während der Frontend-Eingabe.

Folgende Daten können bei einer Orderänderung angepasst werden:

- Limit
- Gültigkeit

#### REQUEST (= Anlage einer Vorvalidierung einer Orderänderung)

Parameter Type	Parameter Name	Data Type	Description
Path	orderId	String	OrderId
Body	Order	Order	JSON-Objekt Order

#### RESPONSE (= Order mit geänderten Parametern)

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Order		-	JSON-Objekt Order

#### HTTP Statuscodes:

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

### 7.1.9 Anlage Validation Orderänderung und Orderlöschung

POST /brokerage/v3/orders/{orderId}/validation

#### Anmerkung:

Validierung einer Orderänderung oder Orderlöschung. Bei Übergabe eines REST-Objektes Order wird dieses wie bei der eigentlichen Orderänderung übergeben und validiert. Die Validierung beinhaltet die Prüfung auf Vollständigkeit aller Pflicht-Parameter.

Werden keine zu prüfenden Order-Parameter übergeben, wird nur der Orderstatus geprüft. Dies kann für die Orderlöschung verwendet werden.

#### Bitte beachten:

Der Validation-Request muss vor der Orderänderung bzw. Orderlöschung aufgerufen werden, da in der Response eine TAN-Challenge übermittelt wird, die an die eigentliche Orderänderung bzw. Orderlöschung übergeben werden muss.

#### 1) Beispiel für die Validierung einer Orderänderung:

**REQUEST (= Anlage der Validierung einer Orderänderung)**

Parameter Type	Parameter Name	Data Type	Description
Path	orderId	String	OrderId
Body	Order	Order	JSON-Objekt Order

**RESPONSE (= Order mit geänderten Parametern sowie TAN-Challenge (Header))**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Order		-	JSON-Objekt Order

**Anmerkung:**

Der Response-Header enthält die TAN-Challenge gemäß Kapitel 3.2.

**HTTP Statuscodes:**

- 201 - CREATED
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

**2) Beispiel für die Validierung einer Orderlöschung:****REQUEST (= Anlage einer Validierung einer Orderlöschung)**

Parameter Type	Parameter Name	Data Type	Description
Path	orderId	String	OrderId
Body		String	Leerer JSON-String: { }

**RESPONSE (= leerer Body sowie TAN-Challenge (Header))**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
---------	----------------	------	-------------

**Anmerkung:**

Der Response-Header enthält die TAN-Challenge gemäß Kapitel 3.2.

**HTTP Statuscodes:**

- 201 - CREATED
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

**7.1.10 Anlage Ex-Ante Kostenausweis für eine Orderänderung**

POST /brokerage/v3/orders/{orderId}/costindicationexante

**Beschreibung:** Generiert für eine geänderte Order den neuen ex-ante Kostenausweis:

Bei diesem Aufruf wird das REST-Objekt Order wie bei der eigentlichen Orderänderung übergeben sowie die für den Kostenausweis erforderlichen Attribute validiert und hinsichtlich Kostenausweis auf Vollständigkeit geprüft. Eine TAN-Eingabe ist hierzu nicht erforderlich. Sofern die Ordervalidierung erfolgreich war, die Kostendaten jedoch nicht ermittelt werden konnten, wird dies gesondert gekennzeichnet und ein Link auf den generischen Kostenausweis angezeigt.

**REQUEST (= Anlage des ex-ante Kostenausweises einer geänderten Order)**

Parameter Type	Parameter Name	Data Type	Description
Path	orderId	String	Ordernummer
Body	Order	Order	JSON-Objekt Order

**RESPONSE (= ex-ante Kostenausweis einer Order)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
CostIndicationExAnte[]		-	Liste JSON-Objekt CostIndicationExAnte

HTTP Statuscodes:

- 201 - CREATED
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

### 7.1.11 Änderung der Order

PATCH /brokerage/v3/orders/{orderId}

**Anmerkung:**

Bei diesem Aufruf wird das REST-Objekt Order mit geänderten Order-Parametern sowie der TAN-Challenge übergeben.

**REQUEST (= Einstellung einer Orderänderung mit TAN)**

Parameter Type	Parameter Name	Data Type	Description
Path	orderId	String	OrderId
Body	Order	Order	JSON-Objekt Order

**Anmerkung:**

Der Request-Header benötigt die TAN-Challenge aus der Validations-Schnittstelle (siehe Kapitel 3.2).

**RESPONSE (= Order mit geänderten Parametern)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
---------	----------------	------	-------------

Order		orderId	OrderId
-------	--	---------	---------

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

**7.1.12 Löschung der Order**

DELETE /brokerage/v3/orders/{orderId}

**Anmerkung:**

Bei diesem Aufruf wird die Order mit Ordnernummer {orderId} gelöscht. Ein REST-Objekt wird nicht übergeben.

**REQUEST (= Einstellung einer Orderlöschung mit TAN)**

Parameter Type	Parameter Name	Data Type	Description
Path	orderId	String	OrderId

**Anmerkung:**

Der Request-Header benötigt die TAN-Challenge aus der Validations-Schnittstelle (siehe Kapitel 3.2).

**RESPONSE (= leer)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
---------	----------------	------	-------------

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE ENTITY

**7.2 REST-Objekte****7.2.1 Dimensions**

Parameter	Datentyp	Schreibbarkeit	Beschreibung
venues	\$Venue[]	read-only	Liste von Handelsplätzen

**7.2.2 Venue**



Parameter	Datentyp	Schreibbarkeit	Beschreibung
venueId	String(<=40)	read-only	Eindeutige IDs eines Handelsplatzes (String(<=40)), an denen das Instrument handelbar ist.
name	String(<=65)	read-only	Name des Handelsplatzes, an denen das Instrument bei comdirect handelbar ist
type	String(<=20)	read-only	{EXCHANGE OFF FUND}
country	String(2)	read-only	Land des Handelsplatzes gem. ISO 3166-2 (DE, US, FR...)
currencies	\$CurrencyString[] + NULL	read-only	Abrechnungswährungen in ISO 4217 (EUR, USD, GBP, ...) bezogen auf Handelsplatz und Instrument (mehrere möglich). Entspricht der Währung des REST-Objektes venue
sides	String(<=4)[]	read-only	Liste der möglichen Geschäftsarten: {BUY, SELL}
validityTypes	String(3)[]	read-only	Liste möglicher Order-Gültigkeitstypen des jeweiligen Handelsplatzes: <ul style="list-style-type: none"> <li>• GFD: Good-for-day (Default)</li> <li>• GTD: Good-til-date (inkl. Ultimo; noch zu klären ist, inwiefern für Ultimo ein separater Gültigkeitstyp angegeben wird)</li> </ul>
orderTypes	Map(ordertypes.name[] -> ordertypes[])	read-only	Liste der pro Handelsplatz (und ggf.pro Instrument) verfügbaren Ordertypen: Map des Ordertyps auf die unterstützten möglichen Limitzusätze, Handelsbeschränkungen.
orderTypes[].name	String(<=30)[]	read-only	Namen der möglichen Ordertypen: {MARKET, LIMIT, QUOTE, STOP_MARKET, STOP_LIMIT, TRAILING_STOP_MARKET, TRAILING_STOP_LIMIT, ONE_CANCELS_OTHER, NEXT_ORDER}
orderTypes[].limitExtensions	String(<=3)[] + NULL	read-only	Namen der möglichen Limitzusätze gemäß REST-Objekt ordertype: <ul style="list-style-type: none"> <li>• FOK: Fill-or-Kill</li> <li>• IOC: Immediate-or-Cancel</li> <li>• AON: All-or-None</li> </ul>
orderTypes[].tradingRestrictions	String(<=3)[] + NULL	read-only	Namen der möglichen Handelsbeschränkungen gemäß REST-Objekt ordertype: <ul style="list-style-type: none"> <li>• OAO: Opening Auction Only</li> <li>• AO: Auction Only</li> <li>• CAO: Closing Auction Only</li> </ul>

### 7.2.3 Order

Parameter	Datentyp	Schreibbarkeit	Beschreibung
depotId	String(<=40)	initial, mandatory	Eindeutige DepotId (UUID)
settlementAccountId	String(<=40) + NULL	initial, optional	Referenz Verrechnungskonto zum Depot, sofern abweichend vom dem Depot direkt zugeordneten Verrechnungskonto (UUID)
orderId	String(<=40)	read-only	Eindeutige OrderId orderId (UUID)
creationTimestamp	\$TimestampString	read-only	Datum-/Zeitstempel der Orderanlage in UTC in Format YYYY-MM-DDThh:mm:ss,ffffff+zz
legNumber	Integer	read-only	Für untergeordnete Teile (Legs) einer Order deren Reihenfolge (bei Kombinationsorder)
bestEx	Boolean	initial, optional	Flag zur Indikation einer BestExecution-Order, für die der Handelsplatz automatisch ausgewählt wird und damit bei der Ordereingabe nicht eingegeben werden darf; Defaultwert=FALSE.
orderType	String (<=30)	initial, mandatory	Ordertyp. Teilausgeführte und offene Orders werden in der Liste der Executions-Parameter geführt: {MARKET, LIMIT, QUOTE, STOP_MARKET, STOP_LIMIT, TRAILING_STOP_MARKET, TRAILING_STOP_LIMIT, ONE_CANCELS_OTHER, NEXT_ORDER}
orderStatus	String (<=30)	read-only	Status der Order: <ul style="list-style-type: none"> <li>• PENDING (hat mehrere Bedeutungen: angenommen, in Bearbeitung Änderung, in Bearbeitung Streichung, dezentral erfasst, maschinell erfasst, manuell erfasst, Rückmeldung erwartet, erfasst wartend, oder gekündigt)</li> <li>• OPEN (offen)</li> <li>• EXECUTED (ausgeführt)</li> <li>• SETTLED (abgerechnet)</li> <li>• CANCELLED_USER (gestrichen)</li> <li>• EXPIRED (verfallen)</li> <li>• CANCELLED_SYSTEM (hat zwei Bedeutungen: abgelehnt, oder nicht ausgeführt)</li> <li>• CANCELLED_TRADE (storniert)</li> <li>• UNKNOWN (Gesamtstatus nicht eindeutig)</li> <li>• PARTIALLY_EXECUTED (wenn openQuantity&gt;0 und die 'Summe(quantity) über alle Ausführungen mit executionStatus={EXECUTED SETTLED}' &gt;0</li> </ul>

			<ul style="list-style-type: none"> <li>• WAITING (nur Ordertype NEXT_ORDER: NEXT-Teil)</li> </ul>
subOrders	Order[] + NULL	initial, mand. bei Komb.-O.	Teile dieser Order, z.B. bei Kombinationsorders OCO Combi, NEO (Next Order) mit verschiedenen Orderlegs
side	String(<=4)	initial, mandatory	Geschäftsart: {BUY, SELL}
instrumentId	String (<=40)	initial, mandatory	WKN, ISIN oder eine uuld; bei Eingabe einer WKN wird als instrumentId eine WKN zurückgegeben, bei Eingabe einer ISIN entsprechend eine ISIN, bei Eingabe einer uuld entsprechend eine uuld
quoteTicketId	String(<=40)	initial, mand. bei RfQ- Order	Quote Ticket-ID als Referenz auf die bereits vor Quote-Anforderung eingegebene und noch nicht validierte TAN
quoteId	String(<=40)	initial, mand. bei RfQ- Order	Quote-ID als Referenz für den auf den Quote-Request erhaltenen Quote des Handelsplatzes (Emittent oder Börse)
venueId	String (<=40) + NULL	initial, mandatory bei Nicht- BestEx- Orders	uuld des Handelsplatzes bzw. –partners: Pflicht, sofern bestEx=FALSE
quantity	\$AmountValue	initial, mandatory	Stückanzahl oder Nominalbetrag im Falle einer Prozentnotiz
limitExtension	String(<=3) + NULL	initial, optional	Orderzusatz: <ul style="list-style-type: none"> <li>• FOK: Fill-or-Kill</li> <li>• IOC: Immediate-or-Cancel</li> <li>• AON: All-or-None</li> </ul>
tradingRestriction	\$String(<=3) + NULL	initial, optional	Handelsbeschränkung: <ul style="list-style-type: none"> <li>• OAO: Opening Auction Only</li> <li>• AO: Auction Only</li> <li>• CAO: Closing Auction Only</li> </ul>

limit	\$AmountValue + NULL	editable	Limit der Order, leer im Falle einer Market Order, Stop Market-, Trailing Stop Market- Order oder einer entsprechenden Suborder
triggerLimit	\$Amount Value + NULL	editable	Trigger Limit: Stop Limit, bei dem eine Stop Order getriggert wird (Stop, TLS, OCO)
trailingLimitDistAbs	\$AmountString + NULL	editable, optional	Abstand Trigger Limit der Trailing Stop-Order zum aktuellen Kurs absolut
trailingLimitDistRel	PercentageString + NULL	editable, optional	Abstand Trigger Limit der Trailing Stop-Order zum aktuellen Kurs in Prozent
validityType	String(3) +NULL	editable, optional	Typ Ordergültigkeit: <ul style="list-style-type: none"> <li>GFD: Good-for-day (Default)</li> <li>GTD: Good-til-date (inkl. Ultimo, noch zu klären ist, inwiefern für Ultimo ein separater Gültigkeitstyp angegeben wird)</li> </ul>
validity	\$DateString + NULL	editable, optional	Datum Ordergültigkeit in Format YYYY-MM-DD; erforderlich bei validityType=GTD
openQuantity	\$AmountValue	read-only	noch nicht ausgeführte Stückanzahl oder Nominalbetrag im Falle von Teilausführungen
cancelledQuantity	\$AmountValue	read-only	Stückanzahl oder Nominalbetrag aller gestrichenen Orderteile oder Ausführungen inklusive aller gestrichener Teilausführungen
executedQuantity	\$AmountValue	read-only	kumulierte ausgeführte Stückanzahl oder Nominalbetrag im Falle mehrerer Teilausführungen
expectedValue	\$Amount Value + NULL	read-only	Erwarteter ausmachender Betrag der Limitorder
executions	\$Execution[]	read-only	Liste von Ausführungen zu der Order

**Anmerkung:**

Der Orderstatus wird **im Falle von Teilausführungen** wie folgt gesetzt:

open Quantity	canceledQuantity	Summe (quantity) [alle Executions mit executionStatus={EXECUTED}]	Neuer orderStatus ...	... nach Transaktion

		SETTLED}		
> 0	> 0	> 0	PARTIALLY_EXECUTED	<p>Stornierung einer oder mehrerer, aber nicht aller Teilausführungen nach einer oder mehreren Teilausführungen einer Order mit restlichem offenen Orderteil (unwahrscheinlicher Fall).</p> <p>Stornierung einer oder mehrerer, aber nicht aller Teilausführungen mit Ersatz, d.h. neuem offenen Orderteil, nach einer oder mehreren Teilausführungen einer Order ohne offenen Orderteil (unwahrscheinlicher Fall).</p>
> 0	> 0	= 0	OPEN	<p>Stornierung einer bzw. aller Teilausführungen nach einer oder mehreren Teilausführungen einer Order mit restlichem offenen Orderteil (unwahrscheinlicher Fall).</p> <p>Stornierung einer oder aller Teilausführungen mit Ersatz, d.h. neuem offenen Orderteil, nach einer oder mehreren Teilausführungen einer Order ohne offenen Orderteil (unwahrscheinlicher Fall).</p>
> 0	= 0	> 0	PARTIALLY_EXECUTED	Eine oder mehrere Teilausführungen einer Order mit restlichem offenen Orderteil.
= 0	= 0	> 0	EXECUTED	Vollausführung einer Order nach einer oder mehreren Teilausführungen.
= 0	> 0	= 0	CANCELLED_TRADE	Stornierung einer oder mehrerer Teilausführungen nach einer bzw. mehrerer Teilausführungen ohne restlichen offenen Orderteil (Vollausführung).
= 0	> 0	> 0	CANCELLED_USER CANCELLED_SYSTEM EXPIRED CANCELLED_TRADE	<p>Benutzerseitige Löschung des restlichen Orderteiles nach einer oder mehreren Teilausführungen einer Order.</p> <p>Systemseitige Löschung des restlichen Orderteiles nach einer oder mehreren Teilausführungen einer Order.</p> <p>Verfall des restlichen Orderteiles nach einer mehreren Teilausführungen einer Order.</p> <p>Stornierung einer, aber nicht aller Teilausführungen nach einer oder mehreren Teilausführungen einer Order (unwahrscheinlicher Fall).</p>

Aus Conveniencegründen und zur Kontrolle besteht der Parameter `executedQuantity`, der der Summe der Stückzahlen bzw. Nominalbeträge aller Teilausführungen mit `executionStatus={SETTLED|EXECUTED}` entsprechen muss.

Die Summe aus `openQuantity`, `cancelledQuantity` und `Summe(quantity)` [alle Executions mit `executionStatus={EXECUTED|SETTLED}` (=executedQuantity)] muss `Order.quantity` zzgl. der mit Ersatz stornierten Nominalbeträge ergeben.

## 7.2.4 Execution

Parameter	Datentyp	Schreibbarkeit	Beschreibung
executionId	String(<=40)	read-only	Ausführungs-ID (uuld)
executionNumber	Integer	read-only	Rang (zeitlich) der Ausführung einer Order
executedQuantity	\$AmountValue	read-only	Ausgeführte Stückanzahl bzw. Nominalbetrag
executionPrice	\$AmountValue	read-only	Ausführungspreis
executionTimestamp	\$TimestampString	read-only	Datum-/Zeitstempel der Orderanlage in UTC in Format (MiFID II) YYYY-MM-DDThh:mm:ss,ffffff+zz

### 7.2.5 CostIndicationExAnte

Die Schnittstelle costindicationexante enthält die vor Abgabe einer Order ausgewiesenen Kosteninformationen (Ex ante-Kostenausweis MiFID II gemäß Website).

#### • Anmerkung:

- Der Bezug der Felder zum Ex Ante-Kostenausweis wird in der Spalte Beschreibung über die Nummer der Felder angegeben. Grundsätzlich werden insgesamt sieben Informationsbereiche (Kostenblöcke) gebildet:
  - Transaktionsdaten
  - Kosten des Wertpapierkaufes (Kaufkosten)
  - Kosten während der Haltedauer pro Jahr (Haltekosten)
  - Kosten des Wertpapierverkaufes (Verkaufskosten)
  - Erläuterung zu den Gesamtkosten
  - Gesamtkosten im Detail (aggregierte Werte)
  - Gesamtkosten im Zeitverlauf
- Bei einem Verkauf werden die Kostenblöcke Kaufkosten, Haltekosten und Gesamtkosten im Zeitverlauf nicht angezeigt.
- Statische Texte und Labels, welche sich unmittelbar aus Enumerationen der Rückgabewerte ergeben, werden über den Kostenausweis nicht übertragen, z.B. Regulatorische Texte, Labels "WKN", "Kurswert", Verkauf (side="SELL").
- In den Spalten „Anzeige-Systematik“ sowie „Anzeige der ...“ ist die Systematik der Anzeige der Parameter im Anzeigetext im Detail erläutert.
- Statische Anzeige für Abkürzungen in der Fuß-Zeile: "[E] Eigene Entgelte, [F] Fremde Entgelte, [P] Produktkosten".
- Die Schreibbarkeit des Objekts ist immer „read-only“.

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
depotId	String(<=40)	Depotnummer (uuld)		
calculation Successful	Boolean	Ergebnis der Ermittlung des Kostenausweis; wenn =FALSE enthält linkCosts einen Link auf einen generischen Kostenausweis		
name	String(<=60)	1: Instrumentname analog Instrument.name	immer	<name>
wkn	String(6)	2: WKN analog Instrument.wkn	immer	"WKN: "<wkn>
side	String(<=4)	3: Geschäftsart analog Order.side: • BUY • SELL	side=BUY side=SELL	"Kauf: " "Verkauf: "
quantity	\$AmountValue	4: Stückzahl analog Order.quantity	quantity.unit=XXX	<quantity.value> "Stück zu " <limit.amount> <tradingCurrency> <quantity.value> " " <tradingCurrency> " zu " <limit.amount> " %"
limit	\$Amount Value + NULL	4: Limit analog Order.limit mit Handelswährung	quantity.unit=XXC	
expectedValue	\$Amount Value	5: Erwarteter ausmachender Betrag der Order (Nettokosten): in Handelswährung	immer	"Kurswert: " <expectedValue.value> " " <expected.unit>
venueName	String(<=65)	8: Ausführungsplatz als Name für die Anzeige	immer	"Ausführungsplatz: " <venueName>
settlementCurrency	\$CurrencyString	12: Settlement currency/ Abrechnungswährung analog Account.currency	immer	"Abrechnung in " <settlementCurrency>
tradingCurrency	\$CurrencyString	12: Trading Currency/ Handelswährung		
reportingCurrency	\$CurrencyString	12: Reporting Currency/ Bilanzwährung		
fxRate	\$FXRateEUR + NULL	13: Devisenkurs Abrechnungswährung zu FX:	Handelswährung<>EUR AND side=BUY Handelswährung<>EUR AND side=SELL	"Devisenkurs 1 EUR   G: " <fxRate.bid.value> <tradingCurrency>

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
		Ankauf und Verkauf (beidseitig für Umrechnung in Abhängigkeit BUY/SELL)		"  B: " <fxRate.ask.value> <tradingCurrency> "Devisenkurs 1 EUR   G: " <fxRate.bid.value <tradingCurrency>
expectedSettlementCosts	\$Amount Value + NULL	erwartete Orderkosten für den Kunden gemäß Wertpapierabrechnung		"Von den unten aufgelisteten Kosten werden Ihnen über die Wertpapierabrechnung voraussichtlich folgende Orderkosten abgerechnet"
purchaseCosts	\$CostGroup + NULL	18, 19, Liste 20-23: Kostengruppe Typ K	Im Kostenblock Kaufkosten sollen einzelne Entgelte angezeigt werden, die der Kunde beim Erwerb des Produktes zu zahlen hat. Der Kostenblock soll immer bei einem Kauf (side=BUY) angezeigt werden. Kaufkosten werden in der Kostengruppe CostGroup.type=K (Kaufkosten) geliefert.	Überschrift=CostGroup.label: Kosten des Wertpapierkaufes
holdingCosts	\$CostGroup + NULL	25, 26, Liste 27-30: Kostengruppe Typ H	Im Kostenblock Haltekosten sollen einzelne Entgelte angezeigt werden, die der Kunde während der Haltedauer des Produktes zu zahlen hat. Der Kostenblock soll immer bei einem Kauf (side=BUY) angezeigt werden. Haltekosten-Entgelte werden in der Kostengruppe CostGroup.type=H (Haltekosten) geliefert.	Überschrift=CostGroup.label: Kosten während der Haltedauer (pro Jahr)
salesCosts	\$CostGroup	31, 32, Liste 33-36: Kostengruppe Typ V	Im Kostenblock Verkaufskosten sollen einzelne Entgelte angezeigt werden, die der Kunde beim Verkauf der Position zu zahlen hat. Der Kostenblock soll sowohl bei Käufen als auch bei Verkäufen immer angezeigt werden (side=BUY oder SELL).	Überschrift=CostGroup.label: Kosten des Wertpapierverkaufes



Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
			Verkaufs-kostenentgelte werden in der Kostengruppe CostGroup.type=V (Verkaufskosten) geliefert.	
holdingPeriod	\$AmountString + NULL	14: Haltedauer in Jahren, wird angezeigt bei Kauf	side=BUY	<p>"Die Gesamtkosten eines Wertpapiergeschäftes setzen sich aus den jeweiligen Kosten des Wertpapierkaufes, während der Haltedauer und des Wertpapierverkaufes zusammen. Bei einer angenommenen Haltedauer von " &lt;holdingPeriod&gt; "Jahren belaufen sich die voraussichtlichen Gesamtkosten auf " &lt;totalCostsAbs&gt; "und reduzieren die Rendite um durchschnittlich " &lt;totalCostsRel&gt; "% pro Jahr. Die Zusammensetzung dieser Kosten ist im Folgenden näher erläutert. Die tatsächlichen Gesamtkosten können, z.B. bei kürzerer Haltedauer, davon abweichen."</p> <p>"Beim Wertpapierverkauf belaufen sich die voraussichtlichen Kosten auf " &lt;totalCostsAbs&gt; "und reduzieren die Rendite um " &lt;totalCostsRel&gt; "% . Die Zusammensetzung dieser Kosten ist im Folgenden näher erläutert. Die tatsächlichen Gesamtkosten können davon abweichen."</p>
totalCostsAbs	\$AmountValue	15: Betrag der Gesamtkosten absolut	side=SELL	
totalCostsRel	\$PercentageString	16: Betrag der Gesamtkosten relativ zum Investment		
totalCostsDetail	\$TotalCostBlock	37-45, 60: Liste von Kostenblöcken mit Aufteilung: 1. E: eigene Dienstleistungskosten 2. F: fremde Dienstleistungskosten 3. P: Produktkosten	side=BUY  side=SELL	Überschrift: <b>"Gesamtkosten im Detail (einschließlich durchschnittlicher Kosten pro Jahr)"</b>  Überschrift:

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
				<b>"Kosten im Detail"</b>
totalHoldingCosts	\$TotalHoldingCostBlock	46-51: Liste von Kostenblöcken im Zeitablauf	side=BUY  side=SELL	Überschrift: <b>"Gesamtkosten im Zeitablauf und Auswirkung auf die Rendite"</b>  Der Kostenblock wird nicht angezeigt.
linkCosts	String(<=200)	HTTP-Link zu weiteren Kostenangaben		
linkKid	String(<=200)	HTTP-Link zum Produktinformationsblatt		

## 7.2.6 FXRateEUR

Devisenkurs für 1 Einheit Abrechnungswährung (1 EUR):

Parameter	Datentyp	Schreib-barkeit	Beschreibung
bid	\$AmountValue	read-only	Geldkurs der Abrechnungswährung zur Devise, z.B. 1,1752 USD (1 EUR)
ask	\$AmountValue	read-only	Briefkurs der Abrechnungswährung zur Devise, z.B. 1,1757 USD (1 EUR)

## 7.2.7 Inducement

Parameter	Datentyp	Beschreibung
amount	\$AmountValue	Betrag der Zuwendung
estimated	Boolean	TRUE, wenn der Betrag ein Schätzwert ist

## 7.2.8 CostGroup

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
type	String(1)	Kostengruppe: 1. K: Kosten Wertpapier-kauf 2. H: Kosten Haltedauer im Jahr		

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
		3. V: Kosten Wertpapier-verkauf		
label	String(<=100)	Kostengruppenname für die Anzeige im Kostenausweis - fester String: <ul style="list-style-type: none"> <li>"Kosten des Wertpapier-kaufes"</li> <li>"Kosten der Haltedauer (pro Jahr)"</li> <li>"Kosten des Wertpapier-verkaufes "</li> </ul>	immer bei type=K immer bei type=H immer bei type=V	Überschrift: <label>="Kosten des Wertpapierkaufes" <label>="Kosten der Haltedauer (pro Jahr)" <label>="Kosten des Wertpapierverkaufes"
sum	\$Amount Value + NULL	Summe der Kostengruppe in Handelswährung (quantity.amount.unit): <ul style="list-style-type: none"> <li>18: Summe der Kosten in Handelswährung für Gebühren in Kostengruppe Wertpapierkauf</li> <li>25: Summe der Kosten in Handelswährung für Gebühren in Kostengruppe Haltedauer</li> <li>31: Summe der Kosten in Handelswährung für Gebühren in Kostengruppe Wertpapierverkauf</li> </ul>	Handelswährung=EUR Handelswährung<>EUR und keine Entgelte in der Kostengruppe=type (K,H,V) geliefert oder die Summe der (Kauf-,Halten-Verkauf-)Kostenentgelte in Handelswährung=0 oder das Element nicht geliefert wird Handelswährung<>0 und Summe der (Kauf-,Halte-,Verkauf-) Kostenentgelte in Handelswährung<>0	Keine Anzeige der Position 18: "-" 25: "-" 31: "-" 18: <sum.value> <sum.unit> 25: <sum.value> <sum.unit> 31: <sum.value> <sum.unit>
sumReporting Currency	\$Amount Value	Summe der Kostengruppe in Bilanzwährung: <ul style="list-style-type: none"> <li>19: Summe der Kosten in Bilanzwährung für Gebühren in Kostengruppe Wertpapierkauf</li> <li>26: Summe der Kosten in Bilanzwährung für Gebühren in Kostengruppe Haltedauer</li> <li>32: Summe der Kosten in Bilanzwährung für Gebühren in Kostengruppe Wertpapierverkauf</li> </ul>	Wenn keine Entgelte in der Kostengruppe=type (K,H,V) geliefert oder die Summe der (Kauf-,Halten-,Verkauf-) Kostenentgelte in Bilanzwährung=0 oder das Element nicht geliefert wird Wenn Summe der (Kauf-,Halte-,Verkauf-) Kostenentgelte in Bilanzwährung<>0	19: "-" 26: "-" 32: "-"

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
				19: <sumReportingCurrency.value> <sumReportingCurrency.unit> 26: <sumReportingCurrency.value> <sumReportingCurrency.unit> 32: <sumReportingCurrency.value> <sumReportingCurrency.unit>
costs	\$CostEntry[] + Null	Liste der Kosten pro Kostengruppe	Es werden keine Entgelte in der Kostengruppe (K,H,V) geliefert oder die Kostengruppe selbst nicht geliefert Wiederholungsgruppe gemäß CostEntry, Zeilen 1..n, wobei n=Anzahl Entgelte in der Kostengruppe K,H,V	"Es fallen keine Kosten an."  Zeileneinträge gemäß CostEntry für Kostengruppe K,H,V

### 7.2.9 CostEntry

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
type	String(1)	Entgeltart: <ul style="list-style-type: none"> <li>E: eigene Dienst-leistungen</li> <li>F: fremde Dienst-leistungen</li> <li>P: Produk-tionskosten</li> </ul>		
label	String(<=100)	Entgeltbezeich-nung für die Anzeige im Kostenhinweis: <ul style="list-style-type: none"> <li>20: Kaufkosten</li> <li>27: Haltekosten</li> <li>33: Verkauf-kosten</li> </ul>	immer bei CostGroup.type=K immer bei CostGroup.type=H immer bei CostGroup.type=V	20: <label> "[" <type> "]" 27: <label> "[" <type> "]" 33: <label> "[" <type> "]"
amount	\$Amount Value + NULL	Engelt in Handelswährung: <ul style="list-style-type: none"> <li>21: Kosten für Kostengruppe Wertpapierkauf</li> </ul>	Handelswährung=EUR Handelswährung<> EUR	Die Position wird nicht angezeigt. 21: <amount.value> <amount.unit> 28: <amount.value> <amount.unit>

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
		<ul style="list-style-type: none"> <li>28: Kosten für Kostengruppe Haltedauer</li> <li>34: Kosten für Kostengruppe Wertpapierverkauf</li> </ul>		34: <amount.value> <amount.unit>
amount Reporting Currency	\$Amount Value	Entgelt in Bilanzwährung: <ul style="list-style-type: none"> <li>22: Kosten für Kostengruppe Wertpapierkauf</li> <li>29: Kosten für Kostengruppe Haltedauer</li> <li>35: Kosten für Kostengruppe Wertpapier-verkauf</li> </ul>	immer	22: <amountReportingCurrency.value> <amountReportingCurrency.unit> 29: <amountReportingCurrency.value> <amountReportingCurrency.unit> 35: <amountReportingCurrency.value> <amountReportingCurrency.unit>
inducement	\$Inducement + NULL	Zuwendungs-betrag; wenn isEstimated=TRUE wird der Zuwendungs-betrag mit ca. angegeben: <ul style="list-style-type: none"> <li>23: Zuwendungen für Kosten-gruppe Wertpapier-kauf</li> <li>30: Zuwendungen für Kosten-gruppe Haltedauer</li> <li>36: Zuwendungen für Kosten-gruppe Wertpapier-verkauf</li> </ul>	CostGroup.type=K,H,V: Wenn das Element <inducement> zu dem Entgelt geliefert wird und inducement.amount.value<>0 und inducement.estimated=FALSE Sonst  CostGroup.type=K,V: Wenn das Element <inducement> zu dem Entgelt geliefert wird und inducement.amount.value<>0 und inducement.estimated=TRUE Sonst	23,30,36: "Von Dritten erhält die Bank eine Zahlung von " <inducement.amount.value> " " <inducement.amount.unit> Die Position wird nicht angezeigt.  23,26: "Von Dritten erhält die Bank eine Zahlung von ca. "  <inducement.amount.value> " " <inducement.amount.unit> Die Position wird nicht angezeigt.

## 7.2.10 TotalCostBlock

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
serviceCosts	\$TotalCostEntry	37-39: Gesamtkosten-einträge für E: eigene Dienstleistungs-kosten	siehe TotalCostEntry	37-39: Zeileneinträge gemäß TotalCostEntry für TotalCostEntry.type=E
serviceInducement	\$AmountValue	60: Gesamte Zuwendungen zu eigenen Dienstleistungs-kosten in Bilanzwährung	immer anzeigen, auch bei Zuwendung-Betrag 0 oder 0,00	"Davon erhält die Bank von Dritten " <serviceInducement.value> <serviceInducement.unit>
externalCosts	\$TotalCostEntry	40-42: Gesamtkosten-einträge für F: fremde Dienstleistungs-kosten	siehe TotalCostEntry	40-42: Zeileneinträge gemäß TotalCostEntry für TotalCostEntry.type=F
productCosts	\$TotalCostEntry	43-45: Gesamtkosteneinträge für P: Produktkosten	siehe TotalCostEntry	43-45: Zeileneinträge gemäß TotalCostEntry für TotalCostEntry.type=P

### 7.2.11 TotalCostEntry

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
type	String(1)	Gesamtkostenart: <ul style="list-style-type: none"> <li>E: eigene Dienstleistungs-kosten</li> <li>F: fremde Dienstleistungs-kosten</li> <li>P: Produktkosten</li> </ul>	Der Kostenblock Gesamtkosten soll sowohl bei Käufen als auch bei Verkäufen angezeigt werden. Folgende Kostenelemente sind IMMER im Kostenblock anzuzeigen: <i>Zeile 1 – Dienstleistungs-kosten der Bank</i> Bei Dienstleistungs-kosten der Bank kann eine Zuwendung ausgewiesen werden. Der Zuwendungstext mit dem Zuwendungs-betrag soll unter dem Kostenelement angezeigt werden. <i>Zeile 2 – Dienstleistungs-kosten fremd</i> <i>Zeile 3 - Produktkosten</i>	

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
			<i>Zeile 4 - Summe der Zuwendungen, die zu keinem der o.g. Kostenelemente im Bezug steht, d.h. der Zuwendungsbetrag gehört zu dem Kostenblock allgemein, soll separat in der vierten Zeile angezeigt werden.</i>	
label	String(65)	Anzeige-Text Gesamtkosten: <ul style="list-style-type: none"> <li>37: "- Dienstleistungs-kosten der Bank"</li> <li>40: "- Dienstleistungs-kosten fremd"</li> <li>43: "- Produktkosten (nach Zahlung von Dritten)"</li> </ul>	immer bei type=E immer bei type=F immer bei type=P	37: <label>="- Dienstleistungs-kosten der Bank" 40: <label>="- Dienstleistungs-kosten fremd" 43: <label>="- Produktkosten (nach Zahlung von Dritten)"
amount	\$AmountValue	Engelt in Bilanzwährung: <ul style="list-style-type: none"> <li>38: Gesamtbetrag für eigene Entgelte in Bilanzwährung</li> <li>41: Gesamtbetrag für fremde Entgelte in Bilanzwährung</li> <li>44: Gesamtbetrag für Produktkosten in Bilanzwährung</li> </ul>	immer anzeigen, auch bei Gesamtbetrag (eigene Entgelte, fremde Entgelte, Produktkosten) 0 oder 0,00	38: <amount.value> <amount.unit> 41: <amount.value> <amount.unit> 44: <amount.value> <amount.unit>
averageReturnPA	\$PercentageString + NULL	Durchschnittlicher Wert pro Jahr: <ul style="list-style-type: none"> <li>39: durchschnitt-licher Betrag pro Jahr für Eigenkosten Bank</li> <li>42: durchschnitt-licher Betrag pro Jahr für Fremdkosten Bank</li> <li>45: durchschnitt-licher Betrag pro Jahr für Produktkosten Bank</li> </ul>	side=BUY: immer anzeigen, auch bei dem Durchschnitts-wert 0 oder 0,00  side=SELL	39: <averageReturnPA> " %" 42: <averageReturnPA> " %" 45: <averageReturnPA> " %"  Die Position wird nicht angezeigt.

## 7.2.12 TotalHoldingCostBlock

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
year1	\$TotalHoldingCostEntry	46-47: Gesamtkosten im Zeitablauf, 1. Jahr	siehe TotalHoldingCostEntry	Zeileneinträge gemäß Gesamtkosten im Zeitablauf, 1. Jahr
year2	\$TotalHoldingCostEntry	48-49: Gesamtkosten im Zeitablauf, 2. Jahr	siehe TotalHoldingCostEntry	Zeileneinträge gemäß Gesamtkosten im Zeitablauf, 1. Jahr
sales	\$TotalHoldingCostEntry	50-51: Gesamtkosten im Zeitablauf, Jahr der Veräußerung	siehe TotalHoldingCostEntry	Zeileneinträge gemäß Gesamtkosten im Zeitablauf, Jahr der Veräußerung

### 7.2.13 TotalHoldingCostEntry

Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
type	String(30)	Gesamtkostenart im Zeitablauf: <ul style="list-style-type: none"> <li>IM_ERSTEN_JAHR</li> <li>IM_ZWEITEN_JAHR</li> <li>IM_JAHR_DER_VERAUESSERUNG</li> </ul>	Der Kostenblock soll nur bei Käufen angezeigt werden. Es sollen immer drei Kostenelemente, die Summen der Entgelte im zeitlichen Ablauf darstellen, angezeigt werden: <i>Zeile 1 - Kosten im ersten Jahr der Haltedauer</i> <i>Zeile 2 - Kosten ab dem zweiten Jahr der Haltedauer</i> <i>Zeile 3 - Kosten im Jahr der Veräußerung</i>	
amount	\$AmountValue	Engelt in Bilanzwährung: <ul style="list-style-type: none"> <li>46: Summe der Entgelte im 1. Jahr der Haltedauer</li> <li>48: Summe der Entgelte im 2. Jahr der Haltedauer</li> <li>50: Summe der Entgelte im Jahr der Veräußerung</li> </ul>	immer bei type=IM_ERSTEN_JAHR, auch anzeigen wenn der Betrag 0 oder 0,00 ist  immer bei type=IM_ZWEITEN_JAHR, auch anzeigen wenn der Betrag 0 oder 0,00 ist	Zeile 1: Linksbündig: " - Im 1. Jahr (Kosten Wertpapier-kauf und 1. Jahr Haltedauer) " Rechts-bündig: 46: <amount.value> <amount.unit>  Zeile 2:



Parameter	Datentyp	Beschreibung	Anzeige-Systematik	Anzeige der statischen (in " ") Layer und dynamischen Inhalte (in <...>)
			immer bei type=IM_JAHR_DER_VERAEUSSERUNG, auch anzeigen wenn der Betrag 0 oder 0,00 ist	Linksbündig: " - Ab dem 2. Jahr während der Haltedauer (pro Jahr)" Rechts-bündig: 48: <amount.value> <amount.unit>  Zeile 3: Linksbündig: " - Im Jahr des Wertpapierverkaufes (zusätzliche zu den Kosten für Haltedauer) " Rechts-bündig: 50: <amount.value> <amount.unit>
averageReturnPA	\$PercentageString	Durchschnittlicher Wert pro Jahr <ul style="list-style-type: none"> <li>47: durchschnittlicher Renditewert im 1. Jahr der Haltedauer</li> <li>49: durchschnittlicher Renditewert im 2. Jahr der Haltedauer</li> <li>51: durchschnittlicher Renditewert im Jahr der veräußerung</li> </ul>	immer bei type=IM_ERSTEN_JAHR, auch anzeigen wenn der Betrag 0 oder 0,00 ist  immer bei type=IM_ZWEITEN_JAHR, auch anzeigen wenn der Betrag 0 oder 0,00 ist  immer bei type=IM_JAHR_DER_VERAEUSSERUNG, auch anzeigen wenn der Betrag 0 oder 0,00 ist	Zeile 1: Rechts-bündig: 47: <averageReturnPA> " %" Zeile 2:Rechts-bündig: 49: <averageReturnPA> " %" Zeile 3: Rechts-bündig: 51: <averageReturnPA> " %"

## 8 Resource QUOTE

Die Schnittstellen der Resource Quote werden für das Livetrading benötigt und bereiten die Abfrage eines Quotes vor bzw. führen diese aus.

### 8.1 REST-API Ressourcen

Meth.	URI (Endpunkt)	Bemerkung
POST	/brokerage/v3/quoteticket	Validierung der Anlage eines Quote-Request und Anfrage einer TAN-Challenge.
PATCH	/brokerage/v3/quoteticket/{quoteTicketId}	Validierung der Anlage eines Quote-Request und TAN-Eingabe.
POST	/brokerage/v3/quotes	Erstellt einen Quote-Request zur Anfrage eines Quote.

#### Anmerkung:

Beim Quote-Handel ist es erforderlich, die TAN-Challenge vor dem Absenden des Quote-Requests an den Handelsplatz (Emittent im außerbörslichen Direkthandel) einzustellen. Der Quote-Request ist geknüpft an ein Instrument, einen Handelsplatz, eine Geschäftsart Kauf/Verkauf und eine Menge. Der auf Basis des Quote-Request beantwortete Quote liefert einen i.d.R. unverbindlichen Preis für die angefragte Menge, eine Gültigkeit in Sekunden sowie eine Quote ID, auf die sich die Anlage einer Order später beziehen muss. Bei der späteren Anlage der Order müssen die Parameter der Order mit denen des Quote-Request übereinstimmen.

Um die TAN-Eingabe vor Einstellung des Quote-Requests abzuschließen, wird eine Subressource /orders/quoteticket verwendet, welche im Folgenden ebenfalls verwendet wird und nur der Validierung des späteren Quote-Request dient.

#### 8.1.1 Anlage Validierung Quote Request-Initialisierung

POST /brokerage/v3/quoteticket

#### Anmerkung:

Bei diesem Aufruf wird ein Order-Objekt mit den für den Quote-Request erforderlichen Angaben wie bei der eigentlichen Quote Request-Einstellung übergeben und validiert. Die Validierung beinhaltet die Prüfung auf Vollständigkeit aller Pflicht-Parameter und die Speicherung der Quote-Parameter. Die TAN-Challenge wird im Response-Header übermittelt.

Zur Referenzierung der nachfolgenden Quote-Initialisierung wird eine QuoteTicketId zurückgegeben.

#### REQUEST (= Anlage der Validierung der Quote Request-Initialisierung)

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-
Body	Order	Order	JSON-Objekt Order

#### Anmerkung:

Folgende Felder des REST-Objektes Order werden im Rahmen der Quote-Initialisierung minimal gefüllt:

- depotId
- side (optional)
- instrumentId
- venue
- quantity

#### RESPONSE (= QuoteTAN-Challenge (Header))

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Order	-		JSON-Objekt Order mit quoteTicketId

#### Anmerkung:

Der Response-Header enthält die TAN-Challenge gemäß Kapitel 3.2.

#### HTTP Statuscodes:

- 201 - CREATED
- 422 - UNPROCESSABLE ENTITY

### 8.1.2 Änderung Validierung Quote Request-Initialisierung mit TAN

PATCH /brokerage/v3/orders/quoteticket/{quoteTicketId}

#### Anmerkung:

Bei diesem Aufruf werden die TAN-Challenge und die quoteTicketId als Referenz auf die mittels Quote Request generierte TAN-Challenge übergeben.

Die Änderung bezieht sich ausschließlich auf die Übertragung der TAN-Challenge im Header.

#### REQUEST (= Anlage der Validierung der Quote Request-Initialisierung)

Parameter Type	Parameter Name	Data Type	Description
Path	quoteTicketId	-	quoteTicketId, für die die TAN-Challenge übergeben wird
Body (leer)	-	-	-

#### Anmerkung:

Der Request-Header benötigt die TAN-Challenge aus der Validations-Schnittstelle (siehe Kapitel 3.2).

#### RESPONSE

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
---------	----------------	------	-------------

#### HTTP Statuscodes:

- 204 - NO CONTENT
- 422 - UNPROCESSABLE ENTITY

### 8.1.3 Anlage Quote Request

POST /brokerage/v3/quotes

**Anmerkung:**

Bei diesem Aufruf wird der Quote-Request mit Referenz auf die quoteTicketId übergeben.

**REQUEST (= Eingabe eines Quote Requests)**

Parameter Type	Parameter Name	Data Type	Description
Path	-	-	-
Body	Order	Order	JSON-Objekt Order

**Anmerkung:**

Folgende Felder des REST-Objektes Order werden im Rahmen eines Quote-Request minimal gefüllt:

- depotId
- orderType („QUOTE“)
- side
- instrumentId
- venue
- quantity

**RESPONSE (= Ein- oder beidseitiger Quote des Handelsplatzes mit jeweiliger Quantity und Gültigkeit)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
Quote		quoteld	Quoteld

**HTTP Statuscodes:**

- 200 - OK
- 422 - UNPROCESSABLE ENTITY

### 8.1.4 Orderanlage der Quote-Order

Nachdem der Quote angefordert wurde muss nun gemäß der Beschreibung in Kapitel 7.1.5 zuerst eine Order-Validierung aufgerufen werden, bevor die Orderanlage ausgeführt werden kann.

POST /brokerage/v3/orders/validation

**Anmerkung:**

Bei diesem Aufruf wird das REST-Objekt Order mit Spezifizierung des Ordertyp "Quote" und mit Referenz auf quoteTicketId in Analogie zu allen anderen Ordertypen übergeben und eine formale Validierung durchgeführt. Dieser Schritt kann mit dem nachfolgenden Schritt der Orderanlage direkt hintereinander ausgeführt werden, da die TAN-Challenge bereits zuvor (PATCH /v3/orders/quoteticket/{uuld}) vor dem Quote Request angefragt, validiert und entwertet worden ist.

Im Order-Objekt wird die quoteTicketId als Referenz auf die TAN, die quoteId als Referenz auf die zuvor im Quote-Objekt bereitgestellten Quote-Daten übergeben. Wie bei jeder anderen Order-Validierung wird erneut eine TAN-Challenge erzeugt, die im nächsten Schritt an die Orderanlage übergeben werden muss.

Im zweiten Schritt erfolgt die Orderanlage gemäß Kapitel 7.1.7.

POST /brokerage/v3/orders

**Anmerkung:**

Bei diesem Aufruf wird der identische Body wie schon bei der Validation verwendet.

Der Request-Header benötigt wie in Kapitel 3.2 beschrieben die TAN-Challenge im Header übergeben.

## 8.2 REST-Objekte

### 8.2.1 Quote

Parameter	Datentyp	Schreibbarkeit	Beschreibung
depotId	String(<=40)	read-only	Depotnummer
instrumentId	String (<=40)	read-only	WKN, ISIN oder eine uuld; bei Eingabe einer WKN wird als instrumentId eine WKN zurückgegeben, bei Eingabe einer ISIN entsprechend eine ISIN, bei Eingabe einer uuld entsprechend eine uuld
venueId	String(<=40)	read-only	Handelsplatz bzw. -partner
side	String(<=4)	read-only	Geschäftsart des Quote: BUY (Ask) oder SELL (Bid)
quantity	\$AmountValue	read-only	Stückzahl, bis zu der der Quote Gültigkeit hat

## 9 Resource DOCUMENTS

Die Schnittstellen der Document-Resource ermöglichen Ihnen den Abruf Ihrer PostBox-Dokumente.

### 9.1 REST-API Ressourcen

Meth.	URI (Endpunkt)	Bemerkung
GET	/messages/clients/{clientId}/v2/documents	Abruf der PostBox-Inhalte
GET	/messages/v2/documents/{documentId}	Dokumentenabruf. MimeType gemäß Suchergebnis
GET	/messages/v2/documents/{documentId}/predocument	Abruf Vorschaltseite zu einem Dokument (falls vorhanden)

#### 9.1.1 Abruf PostBox

GET /messages/clients/{clientId}/v2/documents

**REQUEST (= Abruf einer Dokumentenliste mit den Inhalten der PostBox)**

Parameter Type	Parameter Name	Data Type	Description
Path	clientId	String	Literal "user" oder die UUID des clients

**RESPONSE (= Liefert eine Liste von Metadaten zu Dokumenten)**

Content-Type: application/json

JSON-Model:

Objects	Nested Objects	Keys	Description
paging		index	Index erstes Dokument
		matches	Anzahl Dokumente (gesamt)
values	\$Document []		Liste der Dokumente

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND
- 422 – UNPROCESSABLE

**9.1.2 Abruf eines Dokuments**

GET /messages/v2/documents/{documentId}

**Anmerkung:**

Bitte beachten Sie, dass ein ungelesenes Dokument nach dem Abruf über diese Schnittstelle als gelesen markiert wird.

**REQUEST (= Aufruf eines Dokuments)**

Parameter Type	Parameter Name	Data Type	Description
Path	documentId	String	DokumentId (UUID)

Je nach MIME-Type des angeforderten Dokuments muss der Standardheader (vergleiche Kapitel 3.3) angepasst werden. Das Feld „Accept“ muss dabei den erwarteten MIME-Type erhalten.

**Beispiel-Header (Auszug):**

```
Accept:application/pdf
```

**RESPONSE (= Liefert das angefragte Dokument)**

Content-Type: application/pdf oder text/html (gemäß des im Document-JSON angegebenen mimeType)

**HTTP Statuscodes:**

- 200 – OK

- 404 - NOT FOUND
- 406 – NOT ACCEPTABLE (der Content-Type des Responses entspricht nicht dem, der im Request-Header akzeptiert wird (Angabe im Feld „Accept“))

### 9.1.3 Abruf Dokument-Vorschaltseite

GET /messages/v2/documents/{documentId}/predocument

**REQUEST (= Aufruf zur Anzeige der Dokument-Vorschaltseite)**

Parameter Type	Parameter Name	Data Type	Description
Path	documentId	String	DokumentId (UUID)

Das Header-Feld „Accept“ muss den erwarteten MIME-Type text/html erhalten.

**Beispiel-Header (Auszug):**

```
Accept: text/html
```

**RESPONSE (= Liefert die gewünschte Dokument-Vorschaltseite)**

Content-Type: text/html

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND

## 9.2 REST-Objekte

### 9.2.1 Document

Parameter	Datentyp	Schreibbarkeit	Beschreibung
documentId	String	read-only	UUID des Dokuments
name	String	read-only	Betreff/Titel des Dokuments
dateCreation	\$DateString	read-only	Eingangsdatum/Erstellungsdatum
contentType	String	read-only	MimeType des Dokuments
deletable	Boolean	read-only	TRUE, wenn ein Dokument löschtbar ist
advertisement	Boolean	read-only	TRUE, wenn es sich bei dem Dokument um Werbung handelt
documentMetadata	\$DocumentMetaData	read-only	Metadaten zum Dokument

### 9.2.2 DocumentMetadata

Parameter	Datentyp	Schreibbarkeit	Beschreibung
-----------	----------	----------------	--------------

archived	Boolean	read-only	TRUE, wenn das Dokument in das Archiv verschoben wurde
dateRead	\$DateString+NULL	read-only	Datum, an dem das Dokument gelesen wurde
alreadyRead	Boolean	read-only	TRUE, wenn das Dokument gelesen wurde
predocumentExists	Boolean	read-only	TRUE, wenn für das Dokument eine Vorschaltseite im HTML-Format verfügbar ist

## 10 Resource REPORTS

Die Schnittstellen der Resource Reports liefern die Salden zu Ihren sämtlichen comdirect Produkten inkl. Depotwert und Visa-Kartensaldo.

### 10.1 REST-API Ressourcen

Meth.	URI (Endpunkt)	Bemerkung
GET	/reports/participants/{participantId}/v1/allbalances	Die Schnittstelle liefert zu den gewählten Produkten des Kunden die Salden

#### 10.1.1 Abruf Salden sämtlicher comdirect-Produkte

GET /reports/participants/{participantId}/v1/allbalances

**REQUEST (= Ruft für sämtliche comdirect-Produkte des Kunden die Salden ab)**

Parameter Type	Parameter Name	Data Type	Description
Path	participantId	String	Literal "user" oder die UUID des clients

#### Filter-Parameter

- clientConnectionType: CURRENT\_CLIENT, OTHER\_COMDIRECT
- productType: ACCOUNT, CARD, DEPOT, LOAN, SAVINGS (auch kommaseparierte Auswahl ist möglich)
- targetClientId: Mittels des Filters kann eine Liste der Kundenverbindungs-UUIDs übergeben werden, für die Salden ermittelt werden sollen. Als Einzelwert oder Aufzählung

#### Query-Parameter

- "without-attr=balance.staticdata": verhindert die Ermittlung der Stammdaten in den "Balance"-Objekten

#### RESPONSE (= Liefert eine Liste von Produktsalden)

Content-Type: application/json

JSON-Model:



Objects	Nested Objects	Keys	Description
paging		index	Index erster Saldo
		matches	Anzahl Produktsalden (gesamt)
aggregated	\$BalanceAggregation		REST-Objekt BalanceAggregation
values	\$ProductBalance[]		Liste der REST-Objekte ProductBalance in JSON

**HTTP Statuscodes:**

- 200 - OK
- 404 - NOT FOUND
- 422 - UNPROCESSABLE

**10.2 REST-Objekte****10.2.1 ProductBalance**

Parameter	Datentyp	Schreibbarkeit	Beschreibung												
productId	String	read-only	UUID des Produkts												
productType	String	read-only	<div>ENUM {ACCOUNT, CARD, DEPOT, LOAN, SAVINGS}</div> <table><tr><th>Wert</th><th>Beschreibung</th></tr><tr><td>ACCOUNT</td><td>Girokonto, Verrechnungskonto, Tagesgeld PLUS-Konto, Fremdwährungskonto, CFD Konto, Wertpapierkredit, Options- &amp; Futures-Konto</td></tr><tr><td>CARD</td><td>Visa-Karte</td></tr><tr><td>DEPOT</td><td>Depot</td></tr><tr><td>LOAN</td><td>Ratenkredit</td></tr><tr><td>SAVINGS</td><td>Termingeld</td></tr></table>	Wert	Beschreibung	ACCOUNT	Girokonto, Verrechnungskonto, Tagesgeld PLUS-Konto, Fremdwährungskonto, CFD Konto, Wertpapierkredit, Options- & Futures-Konto	CARD	Visa-Karte	DEPOT	Depot	LOAN	Ratenkredit	SAVINGS	Termingeld
Wert	Beschreibung														
ACCOUNT	Girokonto, Verrechnungskonto, Tagesgeld PLUS-Konto, Fremdwährungskonto, CFD Konto, Wertpapierkredit, Options- & Futures-Konto														
CARD	Visa-Karte														
DEPOT	Depot														
LOAN	Ratenkredit														
SAVINGS	Termingeld														
targetClientId	String	read-only	UUID der Kundenverbindung CURRENT_CLIENT, OTHER_COMDIRECT ist dabei identisch zu clientId												
clientConnectionType	String	read-only	<div>ENUM {CURRENT_CLIENT, OTHER_COMDIRECT}</div> <div>CURRENT_CLIENT = Produkte der aktiven Kundenverbindung</div>												

			OTHER_COMDIRECT = Produkt einer weiteren Kundenverbindung
balance	AccountBalance oder CardBalance oder DepotAggregation oder InstallmentLoanBalance oder FixedTermSavings	read-only	Liefert gemäß des productTypes ein entsprechendes Balance-Objekt zurück

### 10.2.2 BalanceAggregation

Parameter	Datentyp	Schreibbarkeit	Beschreibung
balanceEUR	\$AmountValue	read-only	Enthält die aggregierten Salden der gelieferten Produkte
availableCashAmountEUR	\$AmountValue	read-only	Enthält das verfügbare Kapital über alle gelieferten Produkte

### 10.2.3 CardBalance

Parameter	Datentyp	Schreibbarkeit	Beschreibung
cardId	String	read-only	Kartenidentifikation (UUID)
card	\$Card + NULL	read-only	Stammdaten der Visa-Karte
balance	\$AmountValue	read-only	Aktueller Kontostand (Saldo)
availableCashAmount	\$AmountValue	read-only	Aktueller Kontostand + Kreditlinie - Summe aller bereits disponierten, aber nicht gebuchten Geldbeträge: = maximaler Verfügungsrahmen

### 10.2.4 Card

Parameter	Datentyp	Schreibbarkeit	Beschreibung	
cardId	String(<=40)	read-only	Kartenidentifikation (UUID)	
clientId	String	read-only	Kundennummer (UUID)	
participantId	String	read-only	Vertragsnummer (UUID)	
cardType	\$EnumText	read-only	Kartentyp	
			key	Text
			AMERICAN_EXPRESS	AMEX
			MASTERCARD	Mastercard
			VISA_PREPAID	Visa-Karte (Prepaid-Kreditkarte)

Parameter	Datentyp	Schreibbarkeit	Beschreibung	
			VISA_CREDIT	Visa-Karte (Kreditkarte)
			UNKNOWN	Unbekannt
holderName	String	read-only	Name des Inhabers der Karte	
settlementAccountId	String	read-only	Default Verrechnungskontonummer, welche dem Visa-Kartenkonto eineindeutig zugeordnet ist (UUID). In diesem Falle handelt es sich um das Girokonto	
cardDisplayId	String(16)	read-only	Teilanonymisierte Kartennummer zur Anzeige (Bsp. XXXX XXXX XXXX 1234)	
cardValidity	String + NULL	read-only	Beschreibt die Kartengültigkeit im Format MM/JJ	
cardImage	\$VisaCardImage	read-only	Kartenmotiv	
primaryAccountNumberSuffix	String	read-only	letzte 4 Ziffern der Kreditkartennummer	
cardLimit	\$AmountValue + NULL	read-only	Kartenlimit, sofern verfügbar	
status	String	read-only	Status der Karte. ENUM {ACTIVE, INACTIVE, IN_CHANGE, UNKNOWN}	

### 10.2.5 VisaCardImage

Parameter	Datentyp	Schreibbarkeit	Beschreibung
visaCardImageId	String	read-only	Id des Visa-Karten-Motivs
imageDescription	String	read-only	Name des Visa-Karten-Motivs
imageBaseFilename	String	read-only	Basis des Bilddateinamens, zu ergänzen um Postfix für gewünschte Variante

### 10.2.6 InstallmentLoanBalance

Parameter	Datentyp	Schreibbarkeit	Beschreibung
installmentLoanId	String	read-only	Ratenkreditidentifikation (UUID)
installmentLoan	\$InstallmentLoan + NULL	read-only	Stammdaten des Ratenkredits

balance	\$AmountValue	read-only	Aktueller Saldo des Kreditbetrages in EUR, inklusive Restschuldversicherung-Prämie (wenn die Restschuldversicherung mitfinanziert wurde) und anfallender Zinsen.)
---------	---------------	-----------	---

### 10.2.7 InstallmentLoan

Parameter	Datentyp	Schreibbarkeit	Beschreibung
installmentLoanId	String	read-only	Interne ID des Ratenkredits (UUID)
productDisplayId	String	read-only	Die 10-stellige Kredit- Nummer.
creditAmount	\$AmountValue	read-only	Genehmigter Kreditbetrag bei initialer Einrichtung des Ratenkredits in EUR, inklusive Restschuldversicherung-Prämie (wenn die Restschuldversicherung mitfinanziert wurde) und inklusive anfallender Zinsen.
netCreditAmount	\$AmountValue	read-only	Genehmigter Kreditbetrag bei initialer Einrichtung des Ratenkredits in EUR, vor anfallender Zinsen und Restschuldversicherung-Prämie.
paidOutAmount	\$AmountValue	read-only	Der tatsächlich ausgezahlte Betrag (Zinsen oder Restschuldverschreibung können bereits abgezogen oder hinzugefügt worden sein).
installmentAmount	\$AmountValue	read-only	Die Kreditrate in EUR
contractPeriodInMonths	Integer	read-only	Laufzeit des Ratenkredits
effectiveInterest	\$PercentageString	read-only	effektiver Zinssatz
nominalInterest	\$PercentageString	read-only	nominaler Zinssatz
contractConclusionDate	\$DateString	read-only	Abschlussdatum

### 10.2.8 FixedTermSavings

Parameter	Datentyp	Schreibbarkeit	Beschreibung						
fixedTermSavingsId	String	read-only	Interne ID des Termingeldkontos (UUID)						
savingsAmount	\$AmountValue	read-only	Summe, welche als Termingeld angelegt wird						
interestRate	\$AmountValue	read-only	Gibt den Zinssatz des Termingeldkontos an						
fixedTermSavingsType	\$FixedTermSavingsType	read-only	Art des Termingeldkontos.						
fixedTermSavingsDisplayName	String	read-only	Bezeichnung des Termingeldkontos.						
contractPeriodInMonths	Integer	read-only	<div>Gibt die Laufzeit des Termingeldkontos in Monaten an, je nach productType</div> <table><tr><th>Bezeichnung</th><th>Anlage-dauer</th></tr><tr><td>Festgeldkonto</td><td>1 2 3</td></tr><tr><td>Laufzeitkonto</td><td>120</td></tr></table>	Bezeichnung	Anlage-dauer	Festgeldkonto	1 2 3	Laufzeitkonto	120
Bezeichnung	Anlage-dauer								
Festgeldkonto	1 2 3								
Laufzeitkonto	120								
creationDate	\$DateString	read-only	Datum, zu wann die Anlage erfolgt ist						
expirationDate	\$DateString	read-only	Datum, zu wann die Anlage fällig wird						
prolongationAmount	\$AmountValue	read-only	Summe, welche prolongiert werden soll. Diese Summe kann die Zinsen aus dem vorherigen Anlagezeitraum enthalten, muss sie aber nicht. Ebenso ist eine Aufstockung bzw. Reduktion der Summe denkbar.						
extendable	boolean	read-only	Gibt an, ob das Termingeld prolongierbar ist.						

### 10.2.9 FixedTermSavingsType

Wert	Beschreibung
SHORT_TERM	Festgeldkonto
LONG_TERM	Laufzeitkonto

## 11 Beispiele

### 11.1 API-Aufrufe zur Ausführung des Livetradings

Der komplette Ablauf eines Quote-Request mit TAN-Eingabe sowie anschließender Order sieht wie folgt aus:

1. Anforderung Challenge:

- Request: POST /v3/orders/quoteticket mit Order-Objekt, welches Handelsplatz, InstrumentID, Geschäftsseite, Menge enthält

```
{
  "depotId" : "1234_depot_UUID_1234",
  "orderType" : "QUOTE",
  "side" : "BUY",
  "instrumentId" : "WKN123",
  "quantity" : {"value":"10", "unit": "XXX"},
  "venueId" : "1234_venue_UUID_LIVETRADING_1234"
}
```

- Response: TAN-Challenge und uuld als Referenz auf Quote-Initialisierung mit Order-Objekt  
**Header Auszug:**

```
x-Once-Authentication-Info: {"id":"1212121","typ":"TAN_FREI"}
```

**Body:**

```
{
  "depotId": "1234_depot_UUID_1234",
  "orderType": "QUOTE",
  "side": "BUY",
  "instrumentId": "WKN123",
  "venueId": "1234_venue_UUID_LIVETRADING_1234",
  "quantity": {
    "value": "10",
    "unit": "XXX"
  },
  "quoteTicketId": "1233_quoteTicketId_1234"
}
```

2. TAN-Eingabe:

- Request: PATCH /v3/orders/quoteticket/1233\_quoteTicketId\_1234 mit Eingabe TAN, wobei die uuld die aus der Quote-Initialisierung ist

**Header-Auszug:**

```
x-once-authentication-info: {"id":"1212121"}
```

**Body:**

```
<leer>
```

- Response: leer

**3. Quote-Request:**

- Request: POST /v3/quotes

**Body: (identisch mit Schritt 1)**

```
{
  "depotId" : "1234_depot_UUID_1234",
  "orderType" : "QUOTE",
  "side" : "BUY",
  "instrumentId" : "WKN123",
  "quantity" : {"value":"10", "unit": "XXX"},
  "venueId" : "1234_venue_UUID_LIVETRADING_1234"
}
```

- Response: uuld als eindeutige Referenz für eine Quoteld mit Quote

**Body:**

```
{
  "depotId": "1234_depot_UUID_1234",
  "side": "BUY",
  "instrumentId": "WKN123",
  "venueId": "1234_venue_UUID_LIVETRADING_1234",
  "quantity": {
    "value": "10",
    "unit": "XXX"
  },
  "quoteId": "1234_quoteId_1234",
  "validity": 5000,
  "creationDateTimeStamp": "2019-11-01T09:02:35,116000+01",
  "limit": {
    "value": "53.7700",
    "unit": "EUR"
  },
  "expectedValue": {
    "value": "537.70000",
    "unit": "EUR"
  }
}
```

#### 4. Trade-Request/Validierung Quote-Order:

- Request: POST /v3/orders/validation mit Order-Objekt einschl. quoteTicketId als Referenz auf die TAN-Einstellung und Angabe der Quote-Id als Referenz auf den angefragten Quote

##### Body:

```
{
  "depotId" : "1234_depot_UUID_1234",
  "orderType" : "QUOTE",
  "side" : "BUY",
  "instrumentId" : "WKN123",
  "quantity" : {"value":"10", "unit": "XXX"},
  "venueId" : "1234_venue_UUID_LIVETRADING_1234"
  "quoteId" : "1234_quoteId_1234"
  "quoteTicketId" : "1233_quoteTicketId_1234"
  "limit": {
    "value": "53.7700",
    "unit": "EUR"
  },
  "creationTimestamp" : "2019-11-01T09:02:35,116000+01"
}
```

- Response: Order-Objekt und uuld als eindeutige Referenz auf eine OrderId; die Quoteld wird formal geprüft, ebenfalls die quoteTicketId

##### Header (Auszug):

```
x-Once-Authentication-Info : {"id":"3434343","typ":"TAN_FREI"}
```

##### Body (identisch zu Request)

```
{
  "depotId" : "1234_depot_UUID_1234",
  "orderType" : "QUOTE",
  "side" : "BUY",
  "instrumentId" : "WKN123",
  "quantity" : {"value":"10", "unit": "XXX"},
  "venueId" : "1234_venue_UUID_LIVETRADING_1234"
  "quoteId" : "1234_quoteId_1234"
  "quoteTicketId" : "1233_quoteTicketId_1234"
  "limit": {
    "value": "53.7700",
    "unit": "EUR"
  },
  "creationTimestamp" : "2019-11-01T09:02:35,116000+01"
}
```

#### 5. Trade-Request/Einstellung Quote-Order:



- Request: POST /v3/orders mit Order-Objekt einschließlich quoteTicketId als Referenz auf die TAN-Einstellung und Angabe der Quote-Id als Referenz auf den angefragten Quote.

**Header Auszug:**

x-Once-Authentication-Info: {"id":"3434343"}

**Body (identisch mit Schritt 4):**

```
{
  "depotId" : "1234_depot_UUID_1234",
  "orderType" : "QUOTE",
  "side" : "BUY",
  "instrumentId" : "WKN123",
  "quantity" : {"value":"10", "unit": "XXX"},
  "venueId" : "1234_venue_UUID_LIVETRADING_1234"
  "quoteId" : "1234_quoteId_1234"
  "quoteTicketId" : "1233_quoteTicketId_1234"
  "limit": {
    "value": "53.7700",
    "unit": "EUR"
  },
  "creationTimestamp" : "2019-11-01T09:02:35,116000+01"
}
```

- Response: Order-Objekt und uuld als eindeutige Referenz auf eine OrderId; die gespeicherten Quote Request-Daten werden mit der Quoteld validiert.

**Body:**

```
{
  "depotId" : "1234_depot_UUID_1234",
  "settlementAccountId": "1234_account_UUID_1234",
  "orderId": "1234_order_UUID_1234",
  "creationTimestamp": "2019-11-01T09:02:35,116000+01",
  "legNumber": 1,
  "bestEx": false,
  "orderType": "QUOTE",
  "orderStatus": "EXECUTED",
  "side": "BUY",
  "instrumentId": "ISIN_1234",
  "venueId": "1234_venue_UUID_LIVETRADING_1234",
  "quantity": {
    "value": "10",
    "unit": "XXX"
  },
  "executedQuantity": {
    "value": "10",
    "unit": "XXX"
  },
}
```

```

    "validityType": "GTD",
    "validity": "2019-11-30",
    "expectedValue": {
        "value": "537.70000",
        "unit": "EUR"
    },
    "executions": [
        {
            "executionId": "1234_execution_UUID_1234",
            "executionNumber": 1,
            "executedQuantity": {
                "value": "10",
                "unit": "XXX"
            },
            "executionPrice": {
                "value": "53.7700",
                "unit": "EUR"
            },
            "executionTimestamp": null
        }
    ]
}

```

## 11.2 Beispiele für die Orderanlage

Nachfolgende Beispiele dienen Ihnen zur Orientierung, wie ein Order Request Body zu befüllen ist.

### Bitte beachten:

Damit ein Order Request erfolgreich platziert werden kann, ist es zunächst erforderlich, die Schnittstelle **GET /brokerage/v3/orders/dimensions** abzufragen, um die UUIDs der verfügbaren Handelsplätze zu erhalten.

In den Order Dimensions sehen Sie zudem, welcher Ordertyp bei dem jeweiligen Handelsplatz erlaubt ist.

### 11.2.1 Market Order

Request-Body:

```

{
    "depotId": "1234_depot_UUID_1234",
    "side": "BUY",
    "instrumentId": "WKN123",
    "orderType": "MARKET",
    "quantity": {"value": "1", "unit": "XXX"},
    "venueId": "1234_venue_UUID_1234",
}

```

```

    "validityType": "GFD"
  }

```

### 11.2.2 Tagesgültige Limit Order

Request-Body:

```

{
  "depotId":      "1234_depot_UUID_1234",
  "side":         "BUY",
  "instrumentId": "WKN123",
  "orderType":    "LIMIT",
  "quantity":     {"value": "1", "unit": "XXX"},
  "venueId":      "1234_venue_UUID_1234",
  "limit":        {"value": "1.50", "unit": "EUR"},
  "validityType": "GFD"
}

```

### 11.2.3 Tagesgültige Stop Limit Order

Request-Body:

```

{
  "depotId":      "1234_depot_UUID_1234",
  "side":         "SELL",
  "instrumentId": "WKN123",
  "orderType":    "STOP_LIMIT",
  "quantity":     {"value": "1", "unit": "XXX"},
  "venueId":      "1234_venue_UUID_1234",
  "triggerLimit": {"value": "9.50", "unit": "EUR"},
  "limit":        {"value": "9.00", "unit": "EUR"},
  "validityType": "GFD"
}

```

### 11.2.4 Trailing Stop Market Verkaufsoorder mit absolutem Abstand

Request-Body:

```

{
  "depotId":      "1234_depot_UUID_1234",
  "side":         "SELL",
  "instrumentId": "WKN123",
  "orderType":    "TRAILING_STOP_MARKET",
  "quantity":     {"value": "1", "unit": "XXX"},
  "venueId":      "1234_venue_UUID_1234",
  "triggerLimit": {"value": "10", "unit": "EUR"},
  "triggerLimitDistAbs": {"value": "1", "unit": "EUR"},
  "validityType": "GFD"
}

```

### 11.2.5 Trailing Stop Limit Verkaufsoorder mit relativem Abstand

Request-Body:

```
{
  "depotId":          "1234_depot_UUID_1234",
  "side":             "SELL",
  "instrumentId":     "WKN123",
  "orderType":        "TRAILING_STOP_LIMIT",
  "quantity":         {"value": "1", "unit": "XXX"},
  "venueId":          "1234_venue_UUID_1234",
  "limit":             {"value": "9", "unit": "EUR"},
  "triggerLimit":     {"value": "10", "unit": "EUR"},
  "triggerLimitDistRel": "preDecimalPlaces": "5", "decimalPlaces": "50"},
  "validityType":     "GFD"
}
```

### 11.2.6 Kombinationsorder des Typs One Cancels Other (OCO)

Das Rest-Objekt Order kann sich im Falle einer Kombinationsorder (OCO, NEO) aus zwei Order-Legs mit unterschiedlichen OrderIds zusammensetzen.

Request-Body:

```
{
  "depotId":          "1234_depot_UUID_1234",
  "orderType":        "ONE_CANCELS_OTHER",
  "subOrders":        [
    {
      "depotId":          "1234_depot_UUID_1234",
      "side":             "SELL",
      "instrumentId":     "WKN123",
      "orderType":        "STOP_MARKET",
      "quantity":         {"value": "1", "unit": "XXX"},
      "triggerLimit":     {"value": "15.50", "unit": "XXX"},
      "venueId":          "1234_venue_UUID_1234",
      "validityType":     "GTD",
      "validity":         "2019-12-01",
    },
    {
      "depotId":          "1234_depot_UUID_1234",
      "side":             "SELL",
      "instrumentId":     "WKN123",
      "orderType":        "LIMIT",
      "quantity":         {"value": "1", "unit": "XXX"},
      "limit":            {"value": "50", "unit": "XXX"},
      "venueId":          "1234_venue_UUID_1234",
      "validityType":     "GTD",
    }
  ]
}
```

```

        "validity":      "2019-12-01",
      }
    ]
  }

```

### 11.2.7 Kombinationsorder des Typs Next Order

Request-Body:

```

{
  "depotId":      "1234_depot_UUID_1234",
  "orderType":    "NEXT_ORDER",
  "subOrders":    [
    {
      "depotId":      "1234_depot_UUID_1234",
      "side":          "BUY",
      "instrumentId":  "WKN123",
      "orderType":     "LIMIT",
      "quantity":      {"value": "10", "unit": "XXX"},
      "limit":          {"value": "10.00", "unit": "XXX"},
      "venueId":       "1234_venue_UUID_1234",
      "validityType":  "GTD",
      "validity":       "2019-12-01",
    },
    {
      "depotId":      "1234_depot_UUID_1234",
      "side":          "SELL",
      "instrumentId":  "WKN123",
      "orderType":     "STOP_MARKET",
      "quantity":      {"value": "5", "unit": "XXX"},
      "triggerLimit":  {"value": "5.50", "unit": "XXX"},
      "venueId":       "1234_venue_UUID_1234",
      "validityType":  "GFD",
    }
  ]
}

```