
Smile Documentation

Inomial Pty Ltd & contributors.

May 29, 2018

1	Overview	1
1.1	What is GraphQL?	1
1.2	Why is Inomial using GraphQL?	1
1.3	Endpoint	1
1.4	About this reference	2
2	Resources	3
3	Query	5
3.1	Required by	7
4	BigDecimal	9
4.1	Required by	9
5	Boolean	11
5.1	Required by	11
6	Int	13
6.1	Required by	13
7	Long	17
7.1	Required by	17
8	Milliseconds	19
8.1	Required by	19
9	String	21
9.1	Required by	21
10	UUID	23
10.1	Required by	23
11	__DirectiveLocation	25
11.1	Required by	26
12	__TypeKind	27
12.1	Required by	27

13 Account	29
13.1 Required by	30
14 AccountAddress	31
14.1 Required by	32
15 AccountContact	33
15.1 Required by	34
16 AccountCostCentre	35
16.1 Required by	35
17 AllAccountCursor	37
17.1 Required by	37
18 AllDraftInvoiceCursor	39
18.1 Required by	39
19 AllInvoiceItemCursor	41
19.1 Required by	41
20 AllPlanCursor	43
20.1 Required by	43
21 AllSubscriptionCursor	45
21.1 Required by	45
22 AllTicketCursor	47
22.1 Required by	47
23 Binding	49
23.1 Required by	50
24 BindingCursor	51
24.1 Required by	51
25 Cdr	53
25.1 Required by	54
26 CdrCursor	55
26.1 Required by	55
27 CdrError	57
27.1 Required by	57
28 DraftInvoice	59
28.1 Required by	60
29 ItemSpecification	61
29.1 Required by	62
30 PaymentMethod	63
30.1 Required by	63
31 Plan	65
31.1 Required by	65

32 SearchResult	67
32.1 Required by	67
33 Service	69
33.1 Required by	69
34 Subscription	71
34.1 Required by	72
35 SubscriptionBinding	75
35.1 Required by	75
36 SubscriptionBindingCursor	77
36.1 Required by	77
37 SubscriptionCursor	79
37.1 Required by	79
38 TariffType	81
38.1 Required by	81
39 Ticket	83
39.1 Required by	84
40 TicketCursor	85
40.1 Required by	85
41 Tx	87
41.1 Required by	88
42 TxDetail	89
42.1 Required by	90
43 TxItem	91
43.1 Required by	92
44 TxItemCursor	93
44.1 Required by	93
45 Ucdr	95
45.1 Required by	95
46 UnbilledChargeSummary	97
46.1 Required by	97
47 __Directive	99
47.1 Required by	99
48 __EnumValue	101
48.1 Required by	101
49 __Field	103
49.1 Require by	103
50 __InputValue	105
50.1 Require by	105

51	__Schema	107
51.1	Required by	107
52	__Type	109
52.1	Required by	109
53	AccountAddressFilter	111
53.1	Required by	112
54	AccountContactFilter	113
54.1	Required by	114
55	AccountCostCentreFilter	115
55.1	Required by	115
56	AccountFilter	117
56.1	Required by	118
57	BindingFilter	119
57.1	Required by	120
58	CdrErrorFilter	121
58.1	Required by	121
59	CdrFilter	123
59.1	Required by	124
60	DraftInvoiceFilter	125
60.1	Required by	126
61	ItemSpecificationFilter	127
61.1	Required by	128
62	PaymentMethodFilter	129
62.1	Required by	129
63	PlanFilter	131
63.1	Required by	131
64	ServiceFilter	133
64.1	Required by	133
65	SubscriptionBindingFilter	135
65.1	Required by	135
66	SubscriptionFilter	137
66.1	Required by	138
67	TariffTypeFilter	139
67.1	Required by	139
68	TicketFilter	141
68.1	Required by	142
69	TxDetailFilter	143
69.1	Required by	144

70 TxFilter	145
70.1 Required by	146
71 TxItemFilter	147
71.1 Required by	148
72 include	149
72.1 Required by	149
73 skip	151
73.1 Required by	151
74 create	153
74.1 URL	153
74.2 Method	153
74.3 URL Params	153
74.4 Document Format	153
74.5 Success Response	159
74.6 Error Response	160
74.7 Sample Call	161
75 account	165
75.1 URL	165
75.2 Method	165
75.3 URL Params	165
75.4 Document Format	165
75.5 Success Response	169
75.6 Error Response	169
75.7 Sample Call	170
76 subscription	171
76.1 URL	171
76.2 Method	171
76.3 URL Params	171
76.4 Document Format	171
76.5 Success Response	172
76.6 Error Response	172
76.7 Sample Call	173

1.1 What is GraphQL?

GraphQL is a specification for an expressive data query language for APIs. It allows you to describe your data, request what you want from your data and receive a response back in the same shape as your request - only the fields you requested with the same field names. GraphQL is a strongly typed server-side runtime allowing you to traverse from the entry point to related data, following relationships defined in the schema, in a single request.

Many options for implementation on the server and client side are available.

GraphQL was internally developed by Facebook. It was publicly released in 2015.

For more information about the history of GraphQL see [GraphQL.org](https://graphql.org).

1.2 Why is Inomial using GraphQL?

Inomial chose GraphQL because..

Our GraphQL API is accessed by a GraphiQL console. GraphiQL is an interactive in-browser Integrated Development Environment (IDE) for interacting with GraphQL API calls, enabling the means to query data and perform mutations.

1.3 Endpoint

Smile GraphQL is available at the following endpoint:

```
https://showcase.inomial.net/live/api/ql
```

1.4 About this reference

This is a reference to every type available in the Smile GraphQL API. You can use the sidebar to view the available types. Start from [Query](#) to see the API's entry-point fields.

You can also access reference material using the Smile GraphQL App Docs sidebar.

CHAPTER 2

Resources

- [GraphQL.org](https://graphql.org)
- [GraphQL Basics](#)
- [How to GraphQL - GraphQL Fundamentals](#)

CHAPTER 3

Query

Smile's GraphQL Query type

```
type Query {  
  
  # Returns the account for the given UID.  
  #  
  # Arguments  
  # uid: [Not documented]  
  # uuid: [Not documented]  
  # usn: [Not documented]  
  account(uid: Int, uuid: UUID, usn: String): Account  
  
  # Returns the subscription for the given UID.  
  #  
  # Arguments  
  # uid: [Not documented]  
  # uuid: [Not documented]  
  # usn: [Not documented]  
  subscription(uid: Int, uuid: UUID, usn: String): Subscription  
  
  # Returns (possibly draft) transaction details.  
  #  
  # Arguments  
  # uuid: [Not documented]  
  # number: [Not documented]  
  txDetail(uuid: UUID, number: String): TxDetail  
  
  # Returns transaction.  
  #  
  # Arguments  
  # uuid: [Not documented]  
  # number: [Not documented]  
  # tx: [Not documented]  
  tx(uuid: UUID, number: String, tx: String): Tx  
}
```

(continues on next page)

(continued from previous page)

```
# Performs the equivalent of a UI search
#
# Arguments
# q: The query string
# limit: The number of results to return, up to 200
search(q: String, limit: Int): [SearchResult]

# Returns UCDRs.
#
# Arguments
# uuid: [Not documented]
ucdr(uuid: UUID): Ucdr

# Returns the ticket for the given ID.
#
# Arguments
# id: [Not documented]
# uuid: [Not documented]
# number: [Not documented]
ticket(id: Int, uuid: UUID, number: String): Ticket

# Returns the draft invoice for the given ID.
#
# Arguments
# uuid: [Not documented]
# number: [Not documented]
draftInvoice(uuid: UUID, number: String): DraftInvoice

# Returns all unbilled charge summary.
unbilledChargeSummary: [UnbilledChargeSummary]

# Returns all draft invoices.
#
# Arguments
# filter: [Not documented]
allDraftInvoices(filter: [DraftInvoiceFilter]): AllDraftInvoiceCursor

# Returns all tickets.
#
# Arguments
# filter: [Not documented]
allTickets(filter: [TicketFilter]): AllTicketCursor

# Returns all accounts.
#
# Arguments
# filter: [Not documented]
allAccounts(filter: [AccountFilter]): AllAccountCursor

# Returns all subscriptions.
#
# Arguments
# filter: [Not documented]
allSubscriptions(filter: [SubscriptionFilter]): AllSubscriptionCursor

# Returns all plans.
```

(continues on next page)

(continued from previous page)

```
#
# Arguments
# filter: [Not documented]
allPlans(filter: [PlanFilter]): AllPlanCursor

# Returns all invoice items.
#
# Arguments
# filter: [Not documented]
allInvoiceItems(filter: [ItemSpecificationFilter]): AllInvoiceItemCursor
}
```

3.1 Required by

This element is not required

Built-in java.math.BigDecimal

```
scalar BigDecimal
```

4.1 Required by

Cdr *Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.*

CdrFilter *null*

DraftInvoice *Draft invoice*

DraftInvoiceFilter *DraftInvoiceFilter*

ItemSpecification *Item Description*

ItemSpecificationFilter *null*

Tx *Information about a credit or debit that has been applied to an Account*

TxDetail *Detailed information about a (possibly draft) transaction*

TxDetailFilter *null*

TxFilter *null*

TxItem *Transaction line item*

TxItemFilter *null*

UnbilledChargeSummary *Charge view summary*

Built-in Boolean

```
scalar Boolean
```

5.1 Required by

- Account* A debtor record containing transactions
- AccountFilter* null
- Binding* Account’s binding
- BindingFilter* null
- ItemSpecification* Item description
- ItemSpecificationFilter* null
- PaymentMethod* Saved payment details for an account
- PaymentMethodFilter* null
- Plan* Plan
- PlanFilter* Plan filter
- Subscription* A subscription to a service
- SubscriptionBinding* Subscription’s binding
- SubscriptionBindingFilter* null
- SubscriptionFilter* null
- Ticket* Tickets (work orders)
- TicketFilter* null

TxItem Transaction line item

TxItemFilter null

__Directive null

__EnumValue null

__Field null

__Type null

Int

Built-in Int

```
scalar Int
```

6.1 Required by

- Account* A debtor record containing transactions
- AccountAddress* A debtor record containing transactions
- AccountAddressFilter* null
- AccountContact* Contact details for an account
- AccountContactFilter* null
- AccountCostCentre* Account cost centre
- AccountCostCentreFilter* null
- AccountFilter* null
- AllAccountCursor* AllAccountCursor
- AllDraftInvoiceCursor* AllDraftInvoiceCursor
- AllInvoiceItemCursor* AllInvoiceItemCursor
- AllPlanCursor* AllPlanCursor
- AllSubscriptionCursor* AllSubscriptionCursor
- AllTicketCursor* AllTicketCursor
- Binding* Account's binding
- BindingCursor* null

BindingFilter null

Cdr Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.

CdrCursor null

CdrError CDR processing errors. These are deleted once a CDR has been successfully processed.

CdrErrorFilter null

CdrFilter null

DraftInvoice Draft invoice

DraftInvoiceFilter DraftInvoiceFilter

ItemSpecification Item Description

ItemSpecificationFilter null

PaymentMethod Saved payment details for an account

PaymentMethodFilter null

Plan Plan

PlanFilter Plan filter

Query Smile's GraphQL Query type

SearchResult A single result from a search query

Service Service

ServiceFilter null

Subscription A subscription to a service

SubscriptionBinding Subscription's binding

SubscriptionBindingCursor null

SubscriptionBindingFilter null

SubscriptionCursor null

SubscriptionFilter null

TariffType Tariff type

TariffTypeFilter null

Ticket Tickets (work orders)

TicketCursor null

TicketFilter null

Tx Information about a credit or debit that has been applied to an Account

TxDetail Detailed information about a (possibly draft) transaction

TxDetailFilter null

TxFilter null

TxItem Transaction line item

TxItemCursor null

TxItemFilter null

Ucdr Unmediated CDR

Long

Long type

scalar Long

7.1 Required by

Account *A debtor record containing transactions*

AccountContact *Contact details for an account*

AccountContactFilter *null*

AccountFilter *null*

Cdr *Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.*

CdrError *CDR processing errors. These are deleted once a CDR has been successfully processed.*

CdrErrorFilter *null*

CdrFilter *null*

Subscription *A subscription to a service*

SubscriptionFilter *null*

TariffType *Tariff type*

TariffTypeFilter *null*

Tx *Information about a credit or debit that has been applied to an Account*

TxDetail *Detailed information about a (possibly draft) transaction*

TxDetailFilter *null*

TxFilter *null*

***TxItem** Transaction line item*

***TxItemFilter** null*

***UnbilledChargeSummary** Charge view summary*

Milliseconds

Milliseconds since 1970

scalar Milliseconds

8.1 Required by

- Account* A debtor record containing transactions
- AccountFilter* null
- Binding* Account’s binding
- BindingFilter* null
- Cdr* Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.
- CdrError* CDR processing errors. These are deleted once a CDR has been successfully processed.
- CdrErrorFilter* null
- CdrFilter* null
- DraftInvoice* Draft invoice
- DraftInvoiceFilter* DraftInvoiceFilter
- Subscription* A subscription to a service
- SubscriptionBinding* Subscription’s binding
- SubscriptionBindingFilter* null
- SubscriptionFilter* null
- Ticket* Tickets (work orders)
- TicketFilter* null

Tx Information about a credit or debit that has been applied to an Account

TxDetail Detailed information about a (possibly draft) transaction

TxDetailFilter null

TxFilter null

Ucdr Unmediated CDR

String

Built-in String

```
scalar String
```

9.1 Required by

Account *A debtor record containing transactions*

AccountAddress *A debtor record containing transactions*

AccountAddressFilter *null*

AccountContact *Contact details for an account*

AccountContactFilter *null*

AccountCostCentre *Account cost centre*

AccountCostCentreFilter *null*

AccountFilter *null*

Binding *Account's binding*

BindingFilter *null*

Cdr *Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.*

CdrError *CDR processing errors. These are deleted once a CDR has been successfully processed.*

CdrErrorFilter *null*

CdrFilter *null*

DraftInvoice *Draft invoice*

DraftInvoiceFilter *DraftInvoiceFilter*

ItemSpecification *Item Description*

ItemSpecificationFilter *null*

PaymentMethod *Saved payment details for an account*

PaymentMethodFilter *null*

Plan *Plan*

PlanFilter *Plan filter*

Query *Smile's GraphQL Query type*

Service *Service*

ServiceFilter *null*

Subscription *A subscription to a service*

SubscriptionBinding *Subscription's binding*

SubscriptionFilter *null*

TariffType *Tariff type*

TariffTypeFilter *null*

Ticket *Tickets (work orders)*

TicketFilter *null*

Tx *Information about a credit or debit that has been applied to an Account*

TxDetail *Detailed information about a (possibly draft) transaction*

TxDetailFilter *null*

TxFilter *null*

TxItem *Transaction line item*

TxItemFilter *null*

Ucdr *Unmediated CDR*

__Directive *null*

__EnumValue *null*

__Field *null*

__InputValue *null*

__Type *null*

UUID

UUID

scalar UUID

10.1 Required by

Account *A debtor record containing transactions*

Account *A debtor record containing transactions*

Binding *Account's binding*

BindingFilter *null*

Cdr *Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.*

CdrError *CDR processing errors. These are deleted once a CDR has been successfully processed.*

CdrErrorFilter *null*

CdrFilter *null*

DraftInvoice *Draft invoice*

DraftInvoiceFilter *DraftInvoiceFilter*

PaymentMethod *Saved payment details for an account*

PaymentMethodFilter *null*

Query *Smile's GraphQL Query type*

Subscription *A subscription to a service*

SubscriptionBinding *Subscription's binding*

SubscriptionBindingFilter *null*

SubscriptionFilter null

TariffType Tariff type

TariffTypeFilter null

Ticket Tickets (work orders)

TicketFilter null

Tx Information about a credit or debit that has been applied to an Account

TxDetail Detailed information about a (possibly draft) transaction

TxDetailFilter null

TxFilter null

TxItem Transaction line item

TxItemFilter null

Ucdr Unmediated CDR

__DirectiveLocation

An enum describing valid locations where a directive can be placed.

```
enum __DirectiveLocation {  
  
    # Indicates the directive is valid on queries.  
    QUERY  
  
    # Indicates the directive is valid on mutations.  
    MUTATION  
  
    # Indicates the directive is valid on fields.  
    FIELD  
  
    # Indicates the directive is valid on fragment definitions.  
    FRAGMENT_DEFINITION  
  
    # Indicates the directive is valid on fragment spreads.  
    FRAGMENT_SPREAD  
  
    # Indicates the directive is valid on inline fragments.  
    INLINE_FRAGMENT  
  
    # Indicates the directive is valid on a schema IDL definition.  
    SCHEMA  
  
    # Indicates the directive is valid on a scalar IDL definition.  
    SCALAR  
  
    # Indicates the directive is valid on an object IDL definition.  
    OBJECT  
  
    # Indicates the directive is valid on a field IDL definition.  
    FIELD_DEFINITION  
}
```

(continues on next page)

(continued from previous page)

```
# Indicates the directive is valid on a field argument IDL definition.
ARGUMENT_DEFINITION

# Indicates the directive is valid on an interface IDL definition.
INTERFACE

# Indicates the directive is valid on an union IDL definition.
UNION

# Indicates the directive is valid on an enum IDL definition.
ENUM

# Indicates the directive is valid on an input object IDL definition.
INPUT_OBJECT

# Indicates the directive is valid on an input object field IDL definition.
INPUT_FIELD_DEFINITION
}
```

11.1 Required by

__Directive null

CHAPTER 12

__TypeKind

```
enum __TypeKind {
  # Indicates this type is a scalar.
  SCALAR

  # Indicates this type is an object. `fields` and `interfaces` are valid fields.
  OBJECT

  # Indicates this type is an interface. `fields` and `possibleTypes` are valid
  # fields.
  INTERFACE

  # Indicates this type is a union. `possibleTypes` is a valid field.
  UNION

  # Indicates this type is an enum. `enumValues` is a valid field.
  ENUM

  # Indicates this type is an input object. `inputFields` is a valid field.
  INPUT_OBJECT

  # Indicates this type is a list. `ofType` is a valid field.
  LIST

  # Indicates this type is a non-null. `ofType` is a valid field.
  NON_NULL
}
```

12.1 Required by

__Type null

CHAPTER 13

Account

A debtor record containing transactions.

```
type Account {  
    account: Long  
    accountUuid: UUID  
    usn: String  
    alternateAccountNumber: String  
    accountType: Int  
    accountTerms: Int  
    accountDisposition: Int  
    discount: Int  
    creationTime_iso: String  
    creationTime: Milliseconds  
    accountCostCentre: Int  
    description: String  
    taxable: Boolean  
    taxSchedule: Int  
    currency: String
```

(continues on next page)

```

timezone: Int

uoobject: Int

custom: String

# Arguments
# filter: [Not documented]
Subscriptions(filter: [SubscriptionFilter]): SubscriptionCursor

# Arguments
# filter: [Not documented]
Tx(filter: [TxFilter]): [Tx]

# Arguments
# filter: [Not documented]
Contact(filter: [AccountContactFilter]): [AccountContact]

# Arguments
# filter: [Not documented]
Address(filter: [AccountAddressFilter]): [AccountAddress]

# Arguments
# filter: [Not documented]
CostCentre(filter: [AccountCostCentreFilter]): AccountCostCentre

# Arguments
# filter: [Not documented]
Tickets(filter: [TicketFilter]): TicketCursor

# Arguments
# filter: [Not documented]
Bindings(filter: [BindingFilter]): BindingCursor

# Arguments
# filter: [Not documented]
PaymentMethods(filter: [PaymentMethodFilter]): [PaymentMethod]
}

```

13.1 Required by

AllAccountCursor *AllAccountCursor*

DraftInvoice *Draft invoice*

Query *Smile's GraphQL Query type*

SearchResult *A single result from a search query*

Subscription *A subscription to a service*

Ticket *Tickets (work orders)*

Tx *Information about a credit or debit that has been applied to an Account*

TxDetail *Detailed information about a (possibly draft) transaction*

CHAPTER 14

AccountAddress

A debtor record containing transactions.

```
type AccountAddress {  
    account: Int  
    addressType: Int  
    addressName: String  
    companyName: String  
    addressDetail: String  
    buildingName: String  
    subUnit: String  
    floor: String  
    lot: String  
    postalDeliveryType: String  
    streetNumber: String  
    streetName: String  
    streetType: String  
    suburb: String  
    state: String  
}
```

(continues on next page)

(continued from previous page)

```
postcode: String
country: String
}
```

14.1 Required by

Account A debtor record containing transactions

CHAPTER 15

AccountContact

Contact details for an account

```
type AccountContact {  
    account: Long  
    contactType: Int  
    contactName: String  
    companyName: String  
    tradingName: String  
    abn: String  
    contactTitle: String  
    familyName: String  
    givenName: String  
    dateOfBirth: String  
    gender: String  
    position: String  
    phoneHome: String  
    phoneWork: String  
    phoneMobile: String  
}
```

(continues on next page)

(continued from previous page)

```
    fax: String  
}
```

15.1 Required by

Account A debtor record containing transactions

CHAPTER 16

AccountCostCentre

Account cost centre

```
type AccountCostCentre {  
    accountCostCentre: Int  
    account: Int  
    name: String  
}
```

16.1 Required by

Account A debtor record containing transactions

Subscription A subscription to a service

TxItem Transaction line item

CHAPTER 17

AllAccountCursor

AllAccountCursor

```
type AllAccountCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [Account]  
}
```

17.1 Required by

Query Smile's GraphQL Query type

AllDraftInvoiceCursor

AllDraftInvoiceCursor

```
type AllDraftInvoiceCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [DraftInvoice]  
}
```

18.1 Required by

Query Smile's GraphQL Query type

AllInvoiceItemCursor

AllInvoiceItemCursor

```
type AllInvoiceItemCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [ItemSpecification]  
}
```

19.1 Required by

Query Smile's GraphQL Query type

AllPlanCursor

AllPlanCursor

```
type AllPlanCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [Plan]  
}
```

20.1 Required by

Query Smile's GraphQL Query type

AllSubscriptionCursor

AllSubscriptionCursor

```
type AllSubscriptionCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [Subscription]  
}
```

21.1 Required by

Query Smile's GraphQL Query type

AllTicketCursor

AllTicketCursor

```
type AllTicketCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [Ticket]  
}
```

22.1 Required by

Query Smile's GraphQL Query type

Account's binding

```
type Binding {  
    tid: Int  
    bindingUuid: UUID  
    accountUuid: UUID  
    planUuid: UUID  
    planName: String  
    planCode: String  
    startTime_iso: String  
    startTime: Milliseconds  
    endTime_iso: String  
    endTime: Milliseconds  
    cancelled: Boolean  
    issueCycle: Int  
    issueFrequency: Int  
    chargeCycle: Int  
    chargeFrequency: Int
```

(continues on next page)

(continued from previous page)

```
timezone: Int

active: Boolean

# Arguments
# filter: [Not documented]
SubscriptionBindings(filter: [SubscriptionBindingFilter]): SubscriptionBindingCursor
}
```

23.1 Required by

BindingCursor null

SubscriptionBinding Subscription's binding

TxItem Transaction line item

CHAPTER 24

BindingCursor

```
type BindingCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [Binding]  
}
```

24.1 Required by

***Account** A debtor record containing transactions*

Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.

```
type Cdr {  
    tid: Int  
    ucdrUuid: UUID  
    cdrUuid: UUID  
    version: Int  
    modified_iso: String  
    modified: Milliseconds  
    subscriptionUuid: UUID  
    stid: Long  
    bindingUuid: UUID  
    aggregateUuid: UUID  
    effective_iso: String  
    effective: Milliseconds  
    duration: Long  
    cdrType: String  
    src: Long
```

(continues on next page)

(continued from previous page)

```
dst: Long

tariffCode: String

quantity: Long

cost: BigDecimal

tariffType: Long

chargeAmount: BigDecimal

# Arguments
# filter: [Not documented]
Subscription(filter: [SubscriptionFilter]): Subscription

# Arguments
# filter: [Not documented]
Errors(filter: [CdrErrorFilter]): [CdrError]

# Arguments
# filter: [Not documented]
TariffType(filter: [TariffTypeFilter]): TariffType
}
```

25.1 Required by

CdrCursor *null*

Ucdr *Unmediated CDR*

CHAPTER 26

CdrCursor

```
type CdrCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [Cdr]  
}
```

26.1 Required by

Subscription A subscription to a service

CdrError

CDR processing errors. These are deleted once a CDR has been successfully processed.

```
type CdrError {  
    error: Long  
    logTime_iso: String  
    logTime: Milliseconds  
    tid: Int  
    ucdrUuid: UUID  
    cdrUuid: UUID  
    errorSource: String  
    errorType: String  
    errorLevel: String  
    args: [String]  
}
```

27.1 Required by

Cdr Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.

CHAPTER 28

DraftInvoice

Draft invoice

```
type DraftInvoice {  
    txdetailUuid: UUID  
    txDetail: Int  
    number: String  
    account: Int  
    ledgerTime_iso: String  
    ledgerTime: Milliseconds  
    creationTime_iso: String  
    creationTime: Milliseconds  
    closedDate_iso: String  
    closedDate: Milliseconds  
    dueDate_iso: String  
    dueDate: Milliseconds  
    amount: BigDecimal  
    taxAmount: BigDecimal  
    reportingAccount: Int  
}
```

(continues on next page)

(continued from previous page)

```
comments: String

version: Int

discount: BigDecimal

lateFeeTxItem: Int

# Arguments
# filter: [Not documented]
TxItems(filter: [TxItemFilter]): TxItemCursor

# Arguments
# filter: [Not documented]
Account(filter: [AccountFilter]): Account
}
```

28.1 Required by

AllDraftInvoiceCursor *AllDraftInvoiceCursor*

Query *Smile's GraphQL Query type*

ItemSpecification

Item Description

```
type ItemSpecification {  
    itemSpecification: Int  
    itemCode: String  
    name: String  
    reportingAccount: Int  
    quantityDivisor: Int  
    quantityFormat: String  
    displayRate: String  
    displayOrder: Int  
    taxable: Boolean  
    taxSummaryItem: Boolean  
    quantityType: Int  
    defaultRate: BigDecimal  
    currency: String  
    taxTreatment: Int  
    custom: String  
}
```

29.1 Required by

AllInvoiceItemCursor *AllInvoiceItemCursor*

TxItem *Transaction line item*

CHAPTER 30

PaymentMethod

Saved payment details for an account

```
type PaymentMethod {  
    tid: Int  
    accountId: UUID  
    paymentMethod: UUID  
    hint: String  
    expiryDate: String  
    isDefault: Boolean  
}
```

30.1 Required by

***Account** A debtor record containing transactions*

Plan

Plan

```
type Plan {  
  plan: String  
  service: Int  
  name: String  
  userDescription: String  
  operatorDescription: String  
  isSuspensionPlan: Boolean  
  firstAvailableDate: String  
  lastAvailableDate: String  
  custom: String  
}
```

31.1 Required by

AllPlanCursor AllPlanCursor

SearchResult

A single result from a search query.

```
type SearchResult {  
  
  # Highest ranking match for this account.  
  accountRank: Int!  
  
  # Rank of this result within the account.  
  resultRank: Int!  
  
  # The type of this result.  
  resultType: Int!  
  
  # Debtor item ID for this result, if any  
  debtorItem: Int  
  
  # Account associated with this result  
  account: Account!  
  
  # Subscription associated with this result, if any  
  sub: Subscription  
}
```

32.1 Required by

Query Smile's GraphQL Query type

Service

Service

```
type Service {  
    service: Int  
    name: String  
    realm: String  
    description: String  
    usernameLabel: String  
}
```

33.1 Required by

Subscription A subscription to a service

CHAPTER 34

Subscription

A subscription to a service

```
type Subscription {  
    subscription: Long  
    subscriptionUuid: UUID  
    usn: String  
    username: String  
    account: Int  
    masterSubscription: Int  
    service: Int  
    ratingCycle: String  
    ratingCycleDay: Int  
    discount: Int  
    created_iso: String  
    created: Milliseconds  
    accountCostCentre: Int  
    description: String  
    taxable: Boolean
```

(continues on next page)

(continued from previous page)

```

currency: String

timezone: Int

provisioned_iso: String

provisioned: Milliseconds

activated_iso: String

activated: Milliseconds

ordered_iso: String

ordered: Milliseconds

enabled_iso: String

enabled: Milliseconds

uoobject: Int

subscriptionStatus: String

purchaseOrder: UUID

custom: String

# Arguments
# filter: [Not documented]
Account(filter: [AccountFilter]): Account

# Arguments
# filter: [Not documented]
CostCentre(filter: [AccountCostCentreFilter]): AccountCostCentre

# Arguments
# filter: [Not documented]
Service(filter: [ServiceFilter]): Service

# Arguments
# filter: [Not documented]
Cdrs(filter: [CdrFilter]): CdrCursor

# Arguments
# filter: [Not documented]
SubscriptionBindings(filter: [SubscriptionBindingFilter]): SubscriptionBindingCursor
}

```

34.1 Required by

AllSubscriptionCursor AllSubscriptionCursor

Cdr Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.

Query Smile's GraphQL Query type

SearchResult A single result from a search query

SubscriptionCursor null

Ticket Tickets (work orders)

TxItem Transaction line item

SubscriptionBinding

Subscription's binding

```
type SubscriptionBinding {  
  
    tid: Int  
  
    subscriptionUuid: UUID  
  
    bindingUuid: UUID  
  
    startTime_iso: String  
    startTime: Milliseconds  
  
    endTime_iso: String  
    endTime: Milliseconds  
  
    timezone: Int  
  
    active: Boolean  
  
    # Arguments  
    # filter: [Not documented]  
    Binding(filter: [BindingFilter]): Binding  
}
```

35.1 Required by

SubscriptionBindingCursor null

SubscriptionBindingCursor

```
type SubscriptionBindingCursor {  
  
    count: Int  
  
    # Arguments  
    # offset: [Not documented]  
    # limit: [Not documented]  
    objects(offset: Int, limit: Int): [SubscriptionBinding]  
}
```

36.1 Required by

Binding Account's binding

Subscription A subscription to a service

SubscriptionCursor

```
type SubscriptionCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [Subscription]  
}
```

37.1 Required by

***Account** A debtor record containing transactions*

TariffType

Tariff type

```
type TariffType {  
    tid: Int  
    tariffType: Long  
    rateType: String  
    path: [Long]  
    name: String  
    tariffTypeUuid: UUID  
}
```

38.1 Required by

***Cdr** Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.*

CHAPTER 39

Ticket

Tickets (work orders)

```
type Ticket {  
    ticket: Int  
    number: String  
    ticketType: Int  
    ticketTypeName: String  
    ticketPriority: Int  
    ticketPriorityName: String  
    ticketStatus: Int  
    ticketStatusName: String  
    ticketGroup: Int  
    account: Int  
    targetSubscription: Int  
    targetTx: Int  
    submittingSubscription: Int  
    assignedOperator: Int  
    creationTime_iso: String  
}
```

(continues on next page)

(continued from previous page)

```
creationTime: Milliseconds

closedTime_iso: String

closedTime: Milliseconds

heldUntil_iso: String

heldUntil: Milliseconds

seconds: Int

summary: String

statusText: String

unread: Boolean

description: String

ticketUuid: UUID

# Arguments
# filter: [Not documented]
Account(filter: [AccountFilter]): Account

# Arguments
# filter: [Not documented]
AssignedOperator(filter: [SubscriptionFilter]): Subscription
}
```

39.1 Required by

AllTicketCursor *AllTicketCursor*

Query *Smile's GraphQL Query type*

TicketCursor *null*

CHAPTER 40

TicketCursor

```
type TicketCursor {  
  
  count: Int  
  
  # Arguments  
  # offset: [Not documented]  
  # limit: [Not documented]  
  objects(offset: Int, limit: Int): [Ticket]  
}
```

40.1 Required by

***Account** A debtor record containing transactions*

CHAPTER 41

Tx

Information about a credit or debit that has been applied to an Account

```
type Tx {  
    tx: Long  
    txUuid: UUID  
    txDetail: Long  
    account: Long  
    txType: Int  
    txNumber: String  
    ledgerDate: String  
    ledgerTime_iso: String  
    ledgerTime: Milliseconds  
    currency: String  
    amount: BigDecimal  
    taxAmount: BigDecimal  
    unallocatedAmount: BigDecimal  
    creatingOperator: Int  
    creationTime_iso: String
```

(continues on next page)

(continued from previous page)

```
creationTime: Milliseconds

comments: String

reportingAccount: Int

paymentPlan: Int

txDisposition: Int

referencedTx: Int

documentType: Int

stationery: Int

# Arguments
# filter: [Not documented]
TxDetail(filter: [TxDetailFilter]): TxDetail

# Arguments
# filter: [Not documented]
Account(filter: [AccountFilter]): Account

# Arguments
# filter: [Not documented]
TxItems(filter: [TxItemFilter]): TxItemCursor
}
```

41.1 Required by

***Account** A debtor record containing transactions*

***Query** Smile's GraphQL Query type*

TxDetail

Detailed information about a (possibly draft) transaction

```
type TxDetail {  
    txdetailUuid: UUID  
    txDetail: Long  
    number: String  
    account: Long  
    ledgerTime_iso: String  
    ledgerTime: Milliseconds  
    creationTime_iso: String  
    creationTime: Milliseconds  
    closedDate_iso: String  
    closedDate: Milliseconds  
    dueDate_iso: String  
    dueDate: Milliseconds  
    amount: BigDecimal  
    taxAmount: BigDecimal  
    reportingAccount: Int  
}
```

(continues on next page)

(continued from previous page)

```
comments: String

version: Int

discount: BigDecimal

lateFeeTxItem: Int

# Arguments
# filter: [Not documented]
TxItems(filter: [TxItemFilter]): TxItemCursor

# Arguments
# filter: [Not documented]
Account(filter: [AccountFilter]): Account
}
```

42.1 Required by

Query Smile's GraphQL Query type

Tx Information about a credit or debit that has been applied to an Account

CHAPTER 43

TxItem

Transaction line item

```
type TxItem {  
    txItem: Long  
    txDetail: Long  
    itemCode: String  
    lineNumber: Int  
    ratingPeriod: Int  
    itemSpecification: Int  
    taxDate: String  
    taxDurationDays: Int  
    description: String  
    quantity: BigDecimal  
    eventCount: Int  
    displayRate: String  
    generated: Boolean  
    amount: BigDecimal  
    subtotalAmount: BigDecimal
```

(continues on next page)

(continued from previous page)

```
subscriptionUuid: UUID
taxSummaryItem: Boolean
taxSchedule: Int
tax: BigDecimal
taxable: Boolean
discount: Int
discountAmount: BigDecimal
purchaseOrder: UUID
rollupTxItem: Int
bindingUuid: UUID
costCentre: Int

# Arguments
# filter: [Not documented]
Subscription(filter: [SubscriptionFilter]): Subscription

# Arguments
# filter: [Not documented]
Binding(filter: [BindingFilter]): Binding

# Arguments
# filter: [Not documented]
CostCentre(filter: [AccountCostCentreFilter]): AccountCostCentre

# Arguments
# filter: [Not documented]
ItemSpecification(filter: [ItemSpecificationFilter]): ItemSpecification
}
```

43.1 Required by

TxItemCursor *null*

```
type TxItemCursor {  
  
    count: Int  
  
    # Arguments  
    # offset: [Not documented]  
    # limit: [Not documented]  
    objects(offset: Int, limit: Int): [TxItem]  
}
```

44.1 Required by

DraftInvoice Draft invoice

Tx Information about a credit or debit that has been applied to an Account

TxDetail Detailed information about a (possibly draft) transaction

Unmediated CDR

```
type Ucdr {  
  tid: Int  
  ucdrUuid: UUID  
  created_iso: String  
  created: Milliseconds  
  cdr: String  
  source: Int  
  
  # Arguments  
  # filter: [Not documented]  
  Cdrs(filter: [CdrFilter]): [Cdr]  
}
```

45.1 Required by

Query Smile's GraphQL Query type

UnbilledChargeSummary

Charge view summary

```
type UnbilledChargeSummary {  
  count: Long  
  sum: BigDecimal  
  issuePeriodEnd: Long  
}
```

46.1 Required by

Query Smile's GraphQL Query type


```
type __Directive {  
  name: String  
  description: String  
  locations: [__DirectiveLocation!]  
  args: [__InputValue!]!  
  onOperation: Boolean @deprecated( reason: "Use `locations`." )  
  onFragment: Boolean @deprecated( reason: "Use `locations`." )  
  onField: Boolean @deprecated( reason: "Use `locations`." )  
}
```

47.1 Required by

__Schema A GraphQL Introspection defines the capabilities of a GraphQL server. It exposes all available types and directives on the server, the entry points for query, mutation, and subscription operations.

`__EnumValue`

```
type __EnumValue {  
  name: String!  
  description: String  
  isDeprecated: Boolean!  
  deprecationReason: String  
}
```

48.1 Required by

`__Type` *null*

`__Field`

```
type __Field {  
  name: String!  
  description: String  
  args: [__InputValue!]!  
  type: __Type!  
  isDeprecated: Boolean!  
  deprecationReason: String  
}
```

49.1 Require by

`__Type` *null*

__InputValue

```
type __InputValue {  
  name: String!  
  description: String  
  type: __Type!  
  defaultValue: String  
}
```

50.1 Require by

__Directive null

__Field null

__Type null

A GraphQL Introspection defines the capabilities of a GraphQL server. It exposes all available types and directives on the server, the entry points for query, mutation, and subscription operations.

```
type __Schema {  
  
  # A list of all types supported by this server.  
  types: [__Type!]!  
  
  # The type that query operations will be rooted at.  
  queryType: __Type!  
  
  # If this server supports mutation, the type that mutation operations will be  
  # rooted at.  
  mutationType: __Type  
  
  # 'A list of all directives supported by this server.  
  directives: [__Directive!]!  
  
  # 'If this server support subscription, the type that subscription operations will  
  # be rooted at.  
  subscriptionType: __Type  
}
```

51.1 Required by

This element is not required

__Type

```
type __Type {  
  kind: __TypeKind!  
  name: String  
  description: String  
  
  # Arguments  
  # includeDeprecated: [Not documented]  
  fields(includeDeprecated: Boolean): [__Field!]  
  
  interfaces: [__Type!]  
  
  possibleTypes: [__Type!]  
  
  # Arguments  
  # includeDeprecated: [Not documented]  
  enumValues(includeDeprecated: Boolean): [__EnumValue!]  
  
  inputFields: [__InputValue!]  
  
  ofType: __Type  
}
```

52.1 Required by

__Field null

__InputValue null

__Schema A GraphQL Introspection defines the capabilities of a GraphQL server. It exposes all available types and directives on the server; the entry points for query, mutation, and subscription operations.

__Type null

AccountAddressFilter

```
input AccountAddressFilter {  
  
    account: Int  
  
    addressType: Int  
  
    addressName: String  
  
    companyName: String  
  
    addressDetail: String  
  
    buildingName: String  
  
    subUnit: String  
  
    floor: String  
  
    lot: String  
  
    postalDeliveryType: String  
  
    streetNumber: String  
  
    streetName: String  
  
    streetType: String  
  
    suburb: String  
  
    state: String  
  
    postcode: String  
}
```

(continues on next page)

(continued from previous page)

```
country: String
}
```

53.1 Required by

Account A debtor record containing transactions

AccountContactFilter

```
input AccountContactFilter {  
  
    account: Long  
  
    contactType: Int  
  
    contactName: String  
  
    companyName: String  
  
    tradingName: String  
  
    abn: String  
  
    contactTitle: String  
  
    familyName: String  
  
    givenName: String  
  
    dateOfBirth: String  
  
    gender: String  
  
    position: String  
  
    phoneHome: String  
  
    phoneWork: String  
  
    phoneMobile: String  
  
    fax: String  
}
```

54.1 Required by

Account A debtor record containing transactions

AccountCostCentreFilter

```
input AccountCostCentreFilter {  
    accountCostCentre: Int  
    account: Int  
    name: String  
}
```

55.1 Required by

Account *A debtor record containing transactions*

Subscription *A subscription to a service*

TxItem *Transaction line item*

CHAPTER 56

AccountFilter

```
input AccountFilter {  
  
    account: Long  
  
    accountUuid: UUID  
  
    usn: String  
  
    alternateAccountNumber: String  
  
    accountType: Int  
  
    accountTerms: Int  
  
    accountDisposition: Int  
  
    discount: Int  
  
    creationTime: Milliseconds  
  
    accountCostCentre: Int  
  
    description: String  
  
    taxable: Boolean  
  
    taxSchedule: Int  
  
    currency: String  
  
    timezone: Int  
  
    uobject: Int  
}
```

(continues on next page)

(continued from previous page)

```
    custom: String
  }
```

56.1 Required by

DraftInvoice Draft invoice

Query Smile's GraphQL Query type

Subscription A subscription to a service

Ticket Tickets (work orders)

Tx Information about a credit or debit that has been applied to an Account

TxDetail Detailed information about a (possibly draft) transaction

BindingFilter

```
input BindingFilter {  
  
    tid: Int  
  
    bindingUuid: UUID  
  
    accountUuid: UUID  
  
    planUuid: UUID  
  
    planName: String  
  
    planCode: String  
  
    startTime: Milliseconds  
  
    endTime: Milliseconds  
  
    cancelled: Boolean  
  
    issueCycle: Int  
  
    issueFrequency: Int  
  
    chargeCycle: Int  
  
    chargeFrequency: Int  
  
    timezone: Int  
  
    active: Boolean  
}
```

57.1 Required by

Account A debtor record containing transactions

SubscriptionBinding Subscription's binding

TxItem Transaction line item

```
input CdrErrorFilter {  
    error: Long  
    logTime: Milliseconds  
    tid: Int  
    ucdrUuid: UUID  
    cdrUuid: UUID  
    errorSource: String  
    errorType: String  
    errorLevel: String  
    args: [String]  
}
```

58.1 Required by

***Cdr** Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.*

CHAPTER 59

CdrFilter

```
input CdrFilter {  
    tid: Int  
    ucdrUuid: UUID  
    cdrUuid: UUID  
    version: Int  
    modified: Milliseconds  
    subscriptionUuid: UUID  
    stid: Long  
    bindingUuid: UUID  
    aggregateUuid: UUID  
    effective: Milliseconds  
    duration: Long  
    cdrType: String  
    src: Long  
    dst: Long  
    tariffCode: String  
    quantity: Long
```

(continues on next page)

(continued from previous page)

```
cost: BigDecimal

tariffType: Long

chargeAmount: BigDecimal
}
```

59.1 Required by

Subscription A subscription to a service

Ucdr Unmediated CDR

CHAPTER 60

DraftInvoiceFilter

DraftInvoiceFilter

```
input DraftInvoiceFilter {  
  
    txdetailUuid: UUID  
  
    txDetail: Int  
  
    number: String  
  
    account: Int  
  
    ledgerTime: Milliseconds  
  
    creationTime: Milliseconds  
  
    closedDate: Milliseconds  
  
    dueDate: Milliseconds  
  
    amount: BigDecimal  
  
    taxAmount: BigDecimal  
  
    reportingAccount: Int  
  
    comments: String  
  
    version: Int  
  
    discount: BigDecimal  
  
    lateFeeTxItem: Int  
}
```

60.1 Required by

Query Smile's GraphQL Query type

CHAPTER 61

ItemSpecificationFilter

```
input ItemSpecificationFilter {  
  
    itemSpecification: Int  
  
    itemCode: String  
  
    name: String  
  
    reportingAccount: Int  
  
    quantityDivisor: Int  
  
    quantityFormat: String  
  
    displayRate: String  
  
    displayOrder: Int  
  
    taxable: Boolean  
  
    taxSummaryItem: Boolean  
  
    quantityType: Int  
  
    defaultRate: BigDecimal  
  
    currency: String  
  
    taxTreatment: Int  
  
    custom: String  
}
```

61.1 Required by

Query Smile's GraphQL Query type

TxItem Transaction line item

PaymentMethodFilter

```
input PaymentMethodFilter {  
    tid: Int  
    accountId: UUID  
    paymentMethod: UUID  
    hint: String  
    expiryDate: String  
    isDefault: Boolean  
}
```

62.1 Required by

Account A debtor record containing transactions

PlanFilter

PlanFilter

```
input PlanFilter {  
  plan: String  
  service: Int  
  name: String  
  userDescription: String  
  operatorDescription: String  
  isSuspensionPlan: Boolean  
  firstAvailableDate: String  
  lastAvailableDate: String  
  custom: String  
}
```

63.1 Required by

Query Smile's GraphQL Query type

CHAPTER 64

ServiceFilter

```
input ServiceFilter {  
    service: Int  
    name: String  
    realm: String  
    description: String  
    usernameLabel: String  
}
```

64.1 Required by

Subscription A subscription to a service

SubscriptionBindingFilter

```
input SubscriptionBindingFilter {  
    tid: Int  
    subscriptionUuid: UUID  
    bindingUuid: UUID  
    startTime: Milliseconds  
    endTime: Milliseconds  
    timezone: Int  
    active: Boolean  
}
```

65.1 Required by

Binding Account's binding

Subscription A subscription to a service

CHAPTER 66

SubscriptionFilter

```
input SubscriptionFilter {  
  
    subscription: Long  
  
    subscriptionUuid: UUID  
  
    usn: String  
  
    username: String  
  
    account: Int  
  
    masterSubscription: Int  
  
    service: Int  
  
    ratingCycle: String  
  
    ratingCycleDay: Int  
  
    discount: Int  
  
    created: Milliseconds  
  
    accountCostCentre: Int  
  
    description: String  
  
    taxable: Boolean  
  
    currency: String  
  
    timezone: Int  
}
```

(continues on next page)

(continued from previous page)

```
provisioned: Milliseconds
activated: Milliseconds
ordered: Milliseconds
enabled: Milliseconds
uobject: Int
subscriptionStatus: String
purchaseOrder: UUID
custom: String
}
```

66.1 Required by

Account A debtor record containing transactions

Cdr Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.

Query Smile's GraphQL Query type

Ticket Tickets (work orders)

TxItem Transaction line item

TariffTypeFilter

```
input TariffTypeFilter {  
    tid: Int  
    tariffType: Long  
    rateType: String  
    path: [Long]  
    name: String  
    tariffTypeUuid: UUID  
}
```

67.1 Required by

***Cdr** Charge Data Record (CDR). In Smile, a CDR is a record of a single element of a billable event.*

CHAPTER 68

TicketFilter

```
input TicketFilter {  
  
    ticket: Int  
  
    number: String  
  
    ticketType: Int  
  
    ticketTypeName: String  
  
    ticketPriority: Int  
  
    ticketPriorityName: String  
  
    ticketStatus: Int  
  
    ticketStatusName: String  
  
    ticketGroup: Int  
  
    account: Int  
  
    targetSubscription: Int  
  
    targetTx: Int  
  
    submittingSubscription: Int  
  
    assignedOperator: Int  
  
    creationTime: Milliseconds  
  
    closedTime: Milliseconds  
}
```

(continues on next page)

(continued from previous page)

```
heldUntil: Milliseconds
seconds: Int
summary: String
statusText: String
unread: Boolean
description: String
ticketUuid: UUID
}
```

68.1 Required by

***Account** A debtor record containing transactions*

***Query** Smile's GraphQL Query type*

CHAPTER 69

TxDetailFilter

```
input TxDetailFilter {  
  
    txdetailUuid: UUID  
  
    txDetail: Long  
  
    number: String  
  
    account: Long  
  
    ledgerTime: Milliseconds  
  
    creationTime: Milliseconds  
  
    closedDate: Milliseconds  
  
    dueDate: Milliseconds  
  
    amount: BigDecimal  
  
    taxAmount: BigDecimal  
  
    reportingAccount: Int  
  
    comments: String  
  
    version: Int  
  
    discount: BigDecimal  
  
    lateFeeTxItem: Int  
}
```

69.1 Required by

Tx Information about a credit or debit that has been applied to an Account

CHAPTER 70

TxFilter

```
input TxFilter {  
  
    tx: Long  
  
    txUuid: UUID  
  
    txDetail: Long  
  
    account: Long  
  
    txType: Int  
  
    txNumber: String  
  
    ledgerDate: String  
  
    ledgerTime: Milliseconds  
  
    currency: String  
  
    amount: BigDecimal  
  
    taxAmount: BigDecimal  
  
    unallocatedAmount: BigDecimal  
  
    creatingOperator: Int  
  
    creationTime: Milliseconds  
  
    comments: String  
  
    reportingAccount: Int  
  
}
```

(continues on next page)

(continued from previous page)

```
paymentPlan: Int
txDisposition: Int
referencedTx: Int
documentType: Int
stationery: Int
}
```

70.1 Required by

Account A debtor record containing transactions

CHAPTER 71

TxItemFilter

```
input TxItemFilter {  
  
    txItem: Long  
  
    txDetail: Long  
  
    itemCode: String  
  
    lineNumber: Int  
  
    ratingPeriod: Int  
  
    itemSpecification: Int  
  
    taxDate: String  
  
    taxDurationDays: Int  
  
    description: String  
  
    quantity: BigDecimal  
  
    eventCount: Int  
  
    displayRate: String  
  
    generated: Boolean  
  
    amount: BigDecimal  
  
    subtotalAmount: BigDecimal  
  
    subscriptionUuid: UUID  
}
```

(continues on next page)

(continued from previous page)

```
taxSummaryItem: Boolean
taxSchedule: Int
tax: BigDecimal
taxable: Boolean
discount: Int
discountAmount: BigDecimal
purchaseOrder: UUID
rollupTxItem: Int
bindingUuid: UUID
costCentre: Int
}
```

71.1 Required by

DraftInvoice Draft invoice

Tx Information about a credit or debit that has been applied to an Account

TxDetail Detailed information about a (possibly draft) transaction

include

Directs the executor to include this field or fragment only when the `if` argument is true.

```
directive @include(if: Boolean!) on FIELD | FRAGMENT_SPREAD | INLINE_FRAGMENT
```

72.1 Required by

This element is not required

skip

Directs the executor to skip this field or fragment when the `if` argument is true.

```
directive @skip(if: Boolean!) on FIELD | FRAGMENT_SPREAD | INLINE_FRAGMENT
```

73.1 Required by

This element is not required

create

The create REST API posts a structured JSON document to create accounts and services.

74.1 URL

`https://smile.example.com/stage/api/orders`

74.2 Method

POST

74.3 URL Params

No URL params exist for this request.

74.4 Document Format

Example:

```
{
  "accounts" : [
    {
      "taxable" : true,
      "packageId" : 27,
      "comments" : "Comments",
      "fax" : "(03) 9663 3555",
      "ratingCycleDay" : 31,
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    "invoicingCycleDay" : 31,
    "phoneContact" : {
      "work" : "(03) 9663 3554",
      "home" : "(03) 9663 3554",
      "mobile" : "0411 888 000"
    },
    "releaseDelay" : 0,
    "contactTitle" : "MR",
    "givenName" : "John",
    "familyName" : "Smith",
    "abn" : "94 088 172 301",
    "serviceAddress" : {
      "country" : "Australia",
      "postcode" : "3000",
      "suburb" : "Melbourne",
      "streetNumber" : "120",
      "streetType" : "Road",
      "streetName" : "Main",
      "addressDetail" : "Main Building",
      "state" : "Victoria"
    },
    "companyName" : "Example Telecom",
    "billAddress" : {
      "streetNumber" : "620",
      "streetType" : "Street",
      "streetName" : "Bourke",
      "country" : "Australia",
      "suburb" : "Melbourne",
      "state" : "Victoria",
      "postcode" : "3000"
    },
    "timezone" : "Australia/Victoria",
    "alternateAccountNumber" : "abc111115",
    "currency" : "AUD",
    "tradingName" : "Example Telco",
    "emailAddress" : "john.smith@example.com",
    "dob" : "1995-12-08"
  },
  "custom" : {
    "sms_subscribe" : false,
    "referrer" : "Acme Telecoms"
  }
},
"subscriptions" : [
  {
    "releaseDelay" : 0,
    "startTime" : 1514725200000,
    "description" : "The A2 Start D Plan",
    "plan" : "a2startd",
    "username" : "john.adsl@example.com",
    "currency" : "AUD",
    "ratingCycleDay" : 31,
    "invoicingCycleDay" : 31,
    "timezone" : "Australia/Victoria",
    "serviceId" : 382,
    "custom" : {
      "newsletter_subscribe" : true,

```

(continues on next page)

(continued from previous page)

```
        "productDescription" : "3G Wireless Modem",
        "productCode" : "3gwifi",
        "colour" : "Purple"
    }
}
}
```


74.4.1 Account attributes

Field name	Data type	Description	Required?
USN	String		
alternateAccountNumber	String	AccountID from old system. Will be searchable in Smile and can be used for linking to the other import files.	
ratingCycleDay	Number	Default is the service's period day or the company's period day if there is no service's period day. Default is 31 when there is no service's period day or company's period day.	
invoicingCycleDay	Number	Default is the service's invoicing day or the company's invoicing day if there is no service's invoicing day. Default is 31 when there is no service's invoicing day or company's invoicing day.	
releaseDelay	Number	Default is company's release delay	
currency	String	Currency from Smile. The three letter ISO 4217 currency code of the payment amount. For example: AUD, USD, FJD. Default is package's currency.	
accountTerms	Number	Acterms ID from Smile. Default is package's account terms.	
taxable	Boolean	Default is True	
packageId	Number	Package ID from Smile	Yes
comments	String		
tradingName	String		
abn	String		
companyName	String		
contactTitle	String		
givenName	String		
familyName	String		
emailAddress	String		
serviceAddress	JsonObject	For more information, see <i>serviceAddress attributes</i> .	
billAddress	JsonObject	For more information, see <i>billAddress attributes</i> .	
phoneContact	JsonObject	For more information, see <i>phoneContact attributes</i> .	
fax	String		
timezone	String	Full time zone name. For example: Australia/Adelaide.	
dob	String	Format: yyyy-MM-dd For example: 2017-11-16	
custom	JsonObject	Data is in code/value pairs	

74.4.2 serviceAddress attributes

Field name	Data type	Description	Re-quired?
address-Detail	String		
street-Number	String		
street-Name	String		
street-Type	String	Allowed values: “select code,name from addressLookup where lookupType = 0;”. Can be code or name.	
suburb	String		
postcode	String		
state	String	Allowed values: “select code,name from addressLookup where lookupType = 1;”. Can be code or name.	
country	String		

74.4.3 billAddress attributes

Field name	Data type	Description	Re-quired?
address-Detail	String		
street-Number	String		
street-Name	String		
street-Type	String	Allowable values: “select code,name from addressLookup where lookupType = 0;”. Can be code or name.	
suburb	String		
postcode	String		
state	String	Allowed values: “select code,name from addressLookup where lookupType = 1;”. Can be code or name.	
country	String		

74.4.4 phoneContact attributes

Field name	Data type	Description	Required?
work	String		
home	String		
mobile	String		

74.4.5 Subscription attributes

Field name	Data type	Description	Required?
ac-coun-tId		Account UUID from Smile	Yes (if has no account in request)
servi-ceId		Service ID from Smile	Yes
user-name	String		
plan	String	Plan ID from Smile	
start-Time Number	Time in	milliseconds since the UNIX epoch (January 1, 1970 00:00:00 UTC) For example: 1514725200000	
time-zone	String	Full time zone name. For example: Australia/Adelaide. Default is inherited from account.	
de-scrip-tion	String		
rat-ing-Cycle-Day	Number	Default is inherited from account	
in-voic-ing-Cycle-Day	Number	Default is inherited from account	
re-leaseDe-lay	Number	Default is inherited from account	
currency	String	Currency from Smile. The three letter ISO 4217 currency code of the payment amount. For example: AUD, USD, FJD. Default is inherited from account.	
cus-tom	Json object	Data is in code/value pairs. For example: {"test_text":"value","is_bool":true,"test_date": "2018-12-08","test_number":123.45} Note: Date must be in yyyy-MM-dd format. If there is no matching custom field from provided code in Smile, a new one will be created.	

74.5 Success Response

This request returns a list of account UUID created and a list of subscription UUID created.

Note: Only 1 account create is supported.

74.5.1 Example:

Code: 200 (OK)

Content:

```
{
  "accounts" : {
    "0" : {
      "uuid" : "25f17058-35dd-4647-a00b-536c915c83cf"
    }
  },
  "subscriptions" : {
    "0" : {
      "uuid" : "dc120f6c-8c29-4e18-9944-c3b082ee1daa"
    },
    "1" : {
      "uuid" : "e0d540b7-5f3c-4c4f-a022-8447dfcba5df"
    }
  }
}
```

74.6 Error Response

74.6.1 Example:

Code: 500 (Internal Server Error)

Description: Returned if an import of more than one account or more than 100 subscriptions is attempted.

Content: Not applicable.

74.6.2 Example:

Code: 200

- CURRENCY_NOT_AVAILABLE
- CURRENCY_NOT_FOUND
- PACKAGE_NOT_PUBLISHED
- PACKAGE_NOT_FOUND
- PACKAGE_MISSING
- NO_PACKAGE_FOR_CURRENCY
- SERVICE_NOT_PUBLISHED
- SERVICE_NOT_FOUND
- SERVICE_NOT_VALID
- SERVICE_MISSING

- PLAN_NOT_FOUND
- PLAN_NOT_PUBLISHED
- PLAN_NOT_VALID
- NO_PLAN_FOR_SERVICE
- NO_PLAN_FOR_ACCOUNT
- NOT_ACCOUNT
- ACCOUNT_MISSING
- TIMEZONE_NOT_FOUND
- DUPLICATE_USERNAME
- DUPLICATE_LEGACY_ACCOUNT_NUMBER
- INTERNAL_ERROR

Content:

```
{
  "accounts" : {
    "0" : {
      "errors" : [
        "CURRENCY_NOT_FOUND",
        "PACKAGE_NOT_FOUND"
      ]
    }
  },
  "subscriptions" : {
    "0" : {
      "errors" : [
        "NOT_ACCOUNT",
        "CURRENCY_NOT_FOUND",
        "SERVICE_NOT_FOUND",
        "PLAN_NOT_FOUND"
      ]
    },
    "1" : {
      "errors" : [
        "NOT_ACCOUNT",
        "CURRENCY_NOT_FOUND",
        "SERVICE_NOT_FOUND",
        "PLAN_NOT_FOUND"
      ]
    }
  }
}
```

74.7 Sample Call

```
POST /api/orders HTTP/1.1
URL: https://smile.example.com/test/api/orders
Content-Type:application/json
Accept:application/json
```

(continues on next page)

(continued from previous page)

```
{
  "accounts" : [
    {
      "taxable" : true,
      "packageId" : 27,
      "comments" : "Comments",
      "fax" : "(03) 9663 3555",
      "ratingCycleDay" : 31,
      "invoicingCycleDay" : 31,
      "phoneContact" : {
        "work" : "(03) 9663 3554",
        "home" : "(03) 9663 3554",
        "mobile" : "0411 888 000"
      },
      "releaseDelay" : 0,
      "contactTitle" : "MR",
      "givenName" : "John",
      "familyName" : "Smith",
      "abn" : "94 088 172 301",
      "serviceAddress" : {
        "country" : "Australia",
        "postcode" : "3000",
        "suburb" : "Melbourne",
        "streetNumber" : "120",
        "streetType" : "Road",
        "streetName" : "Main",
        "addressDetail" : "Main Building",
        "state" : "Victoria"
      },
      "companyName" : "Example Telecom",
      "billAddress" : {
        "streetNumber" : "620",
        "streetType" : "Street",
        "streetName" : "Bourke",
        "country" : "Australia",
        "suburb" : "Melbourne",
        "state" : "Victoria",
        "postcode" : "3000"
      },
      "timezone" : "Australia/Victoria",
      "alternateAccountNumber" : "abc111115",
      "currency" : "AUD",
      "tradingName" : "Example Telco",
      "emailAddress" : "john.smith@example.com",
      "dob" : "1995-12-08"
      "custom" : {
        "sms_subscribe" : false,
        "referrer" : "Acme Telecoms"
      }
    }
  ],
  "subscriptions" : [
    {
      "releaseDelay" : 0,
      "startTime" : 1514725200000,
      "description" : "The A2 Start D Plan",
```

(continues on next page)

(continued from previous page)

```
"plan" : "a2startd",
"username" : "john.adsl@example.com",
"currency" : "AUD",
"ratingCycleDay" : 31,
"invoicingCycleDay" : 31,
"timezone" : "Australia/Victoria",
"serviceId" : 382,
"custom" : {
  "newsletter_subscribe" : true,
  "productDescription" : "3G Wireless Modem",
  "productCode" : "3gwifi",
  "colour" : "Purple"
}
]
```

account

The account REST API posts a structured JSON document to update an account.

75.1 URL

`https://smile.example.com/stage/api/account/{id}/update`

75.2 Method

POST

75.3 URL Params

Required:

`id=[string]`

id can be UUID or USN of account.

75.4 Document Format

Example:

```
{
  "taxable" : true,
  "releaseDelay" : 0,
  "comments" : "Comments",
```

(continues on next page)

(continued from previous page)

```
"contactTitle" : "MR",
"givenName" : "Alfred",
"familyName" : "Sample",
"abn" : "94 088 172 301",
"serviceAddress" : {
  "addressDetail" : "Main Building",
  "streetNumber" : "120",
  "streetType" : "Road",
  "streetName" : "Main",
  "suburb" : "Melbourne",
  "country" : "Australia",
  "postcode" : "3000"
},
"companyName" : "Example Telecom",
"tradingName" : "Example Telco",
"billAddress" : {
  "streetNumber" : "620",
  "streetType" : "Street",
  "streetName" : "Bourke",
  "suburb" : "Melbourne",
  "state" : "Victoria",
  "country" : "Australia",
  "postcode" : "3000"
},
"phoneContact" : {
  "work" : "(03) 9663 3554",
  "home" : "(03) 9663 3554",
  "mobile" : "0411 888 000"
},
"fax" : "(03) 9663 3555",
"emailAddress" : "alfred.sample@example.com",
"dob" : "1995-12-08",
"accountType" : 1,
"accountTerms" : 1,
"custom" : {
  "sms_subscribe" : false,
  "referrer" : "Acme Telecoms"
}
}
```

75.4.1 Account attributes

Field name	Data type	Description	Required?
ac-count-Terms	Number	Acterms ID from Smile.	
re-lease-Delay	Number		
tax-able	Boolean		
com-ments	String		
trad-ing-Name	String		
abn	String		
com-pa-ny-Name	String		
con-tact-Ti-tle	String		
given-Name	String		
fam-ily-Name	String		
emailAd-dress	String		
ser-viceAd-dress	JsonObject	For more information, see serviceAddress attributes .	
bil-lAd-dress	JsonObject	For more information, see billAddress attributes .	
phoneCon-tact	JsonObject	For more information, see phoneContact attributes .	
fax	String		
dob	String	Format: yyyy-MM-dd For example: 2017-11-16	
ac-count-Type	Number	Account type from Smile.	
cus-tom	JsonObject	Data is in code/value pairs. Value can be an object or array. For example: {"test_text": "value", "is_bool": true, "test_date": "2018-12-08", "test_number": 123.45, "note": ["Important note1", "Important note2"]} Note: Date must be in yyyy-MM-dd format. If there is no matching custom field from provided code in Smile, a new one will be created.	

75.4.2 serviceAddress attributes

Field name	Data type	Description	Re-quired?
address-Detail	String		
street-Number	String		
street-Name	String		
street-Type	String	Allowed values: “select code,name from addressLookup where lookupType = 0;”. Can be code or name.	
suburb	String		
postcode	String		
state	String	Allowed values: “select code,name from addressLookup where lookupType = 1;”. Can be code or name.	
country	String		

75.4.3 billAddress attributes

Field name	Data type	Description	Re-quired?
address-Detail	String		
street-Number	String		
street-Name	String		
street-Type	String	Allowable values: “select code,name from addressLookup where lookupType = 0;”. Can be code or name.	
suburb	String		
postcode	String		
state	String	Allowed values: “select code,name from addressLookup where lookupType = 1;”. Can be code or name.	
country	String		

75.4.4 phoneContact attributes

Field name	Data type	Description	Required?
work	String		
home	String		
mobile	String		

75.5 Success Response

This request returns an account ID updated and status.

75.5.1 Example:

Code: 200 (OK)

Content:

```
{
  "accountId": "e5f946bb-6b31-4bbd-ae6c-247c54a57b4b",
  "status": "UPDATED"
}
```

75.6 Error Response

75.6.1 Example:

Code: 500 (Internal Server Error)

Description: Returned if something fails during an update. You can find more details from the server's log.

Content:

```
{
  "accountId": "e5f946bb-6b31-4bbd-ae6c-247c54a57b4b",
  "status": "INTERNAL_ERROR"
}
```

75.6.2 Example:

Code: 200

- NOT_ACCOUNT
- ACCOUNT_MISSING
- TIMEZONE_NOT_FOUND
- ACCOUNT_TYPE_NOT_FOUND

Content:

```
{
  "accountId": "e5f946bb-6b31-4bbd-ae6c-247c54a57b4b",
  "status": "ERROR",
  "errors": [
    "NOT_ACCOUNT",
    "TIMEZONE_NOT_FOUND"
  ]
}
```

75.7 Sample Call

```
POST /api/account HTTP/1.1
URL: https://smile.example.com/test/api/account/e5f946bb-6b31-4bbd-ae6c-247c54a57b4b/
→update
Content-Type:application/json
Accept:application/json

{
  "taxable" : true,
  "releaseDelay" : 0,
  "comments" : "Comments",
  "contactTitle" : "MR",
  "givenName" : "Alfred",
  "familyName" : "Sample",
  "abn" : "94 088 172 301",
  "serviceAddress" : {
    "addressDetail" : "Main Building",
    "streetNumber" : "120",
    "streetType" : "Road",
    "streetName" : "Main",
    "suburb" : "Melbourne",
    "country" : "Australia",
    "postcode" : "3000"
  },
  "companyName" : "Example Telecom",
  "tradingName" : "Example Telco",
  "billAddress" : {
    "streetNumber" : "620",
    "streetType" : "Street",
    "streetName" : "Bourke",
    "suburb" : "Melbourne",
    "state" : "Victoria",
    "country" : "Australia",
    "postcode" : "3000"
  },
  "phoneContact" : {
    "work" : "(03) 9663 3554",
    "home" : "(03) 9663 3554",
    "mobile" : "0411 888 000"
  },
  "fax" : "(03) 9663 3555",
  "emailAddress" : "alfred.sample@example.com",
  "dob" : "1995-12-08",
  "accountType" : 1,
  "accountTerms" : 1,
  "custom" : {
    "sms_subscribe" : false,
    "referrer" : "Acme Telecoms"
  }
}
```

The subscription REST API posts a structured JSON document to update a subscription.

76.1 URL

`https://smile.example.com/stage/api/subscription/{id}/update`

76.2 Method

POST

76.3 URL Params

Required:

id=[string]

id can be UUID or USN of subscription.

76.4 Document Format

Example:

```
{
  "releaseDelay" : 0,
  "timezone" : "Australia/Victoria",
  "username" : "john.adsl@example.com",
```

(continues on next page)

(continued from previous page)

```

    "description" : "The A2 Start D Plan",
    "custom" : {
      "newsletter_subscribe" : true,
      "productDescription" : "3G Wireless Modem"
    }
  }
}

```

76.4.1 Subscription attributes

Field name	Data type	Description	Required?
user-name	String		
de-scription	String		
re-lease delay	Number	Default is inherited from account	
time-zone	String	Full time zone name. For example: Australia/Adelaide.	
cus-tom	Json object	Data is in code/value pairs. Value can be an object or array. For example: {“test_text”:”value”,”is_bool”:true,”test_date”: “2018-12-08”,”test_number”:123.45, “note”: [“Important note1”, “Important note2”]} Note: Date must be in yyyy-MM-dd format. If there is no matching custom field from provided code in Smile, a new one will be created.	

76.5 Success Response

This request returns a subscription ID updated and status

76.5.1 Example:

Code: 200 (OK)

Content:

```

{
  "subscriptionId": "e5f946bb-6b31-4bbd-ae6c-247c54a57b4b",
  "status": "UPDATED"
}

```

76.6 Error Response

76.6.1 Example:

Code: 500 (Internal Server Error)

Description: Returned if something fails during an update. You can find more details from the server's log.

Content:

```
{
  "subscriptionId":"e5f946bb-6b31-4bbd-ae6c-247c54a57b4b",
  "status":"INTERNAL_ERROR"
}
```

76.6.2 Example:

Code: 200

- NOT_SUBSCRIPTION
- SUBSCRIPTION_MISSING
- TIMEZONE_NOT_FOUND
- DUPLICATE_USERNAME

Content:

```
{
  "subscriptionId":"e5f946bb-6b31-4bbd-ae6c-247c54a57b4b",
  "status":"ERROR",
  "errors":[
    "NOT_SUBSCRIPTION",
    "TIMEZONE_NOT_FOUND"
  ]
}
```

76.7 Sample Call

```
POST /api/subscription HTTP/1.1
URL: https://smile.example.com/test/api/subscription/e5f946bb-6b31-4bbd-ae6c-
↪247c54a57b4b/update
Content-Type:application/json
Accept:application/json
```

```
{
  "releaseDelay" : 0,
  "timezone" : "Australia/Victoria",
  "username" : "john.adsl@example.com",
  "description" : "The A2 Start D Plan",
  "custom" : {
    "newsletter_subscribe" : true,
    "productDescription" : "3G Wireless Modem"
  }
}
```