

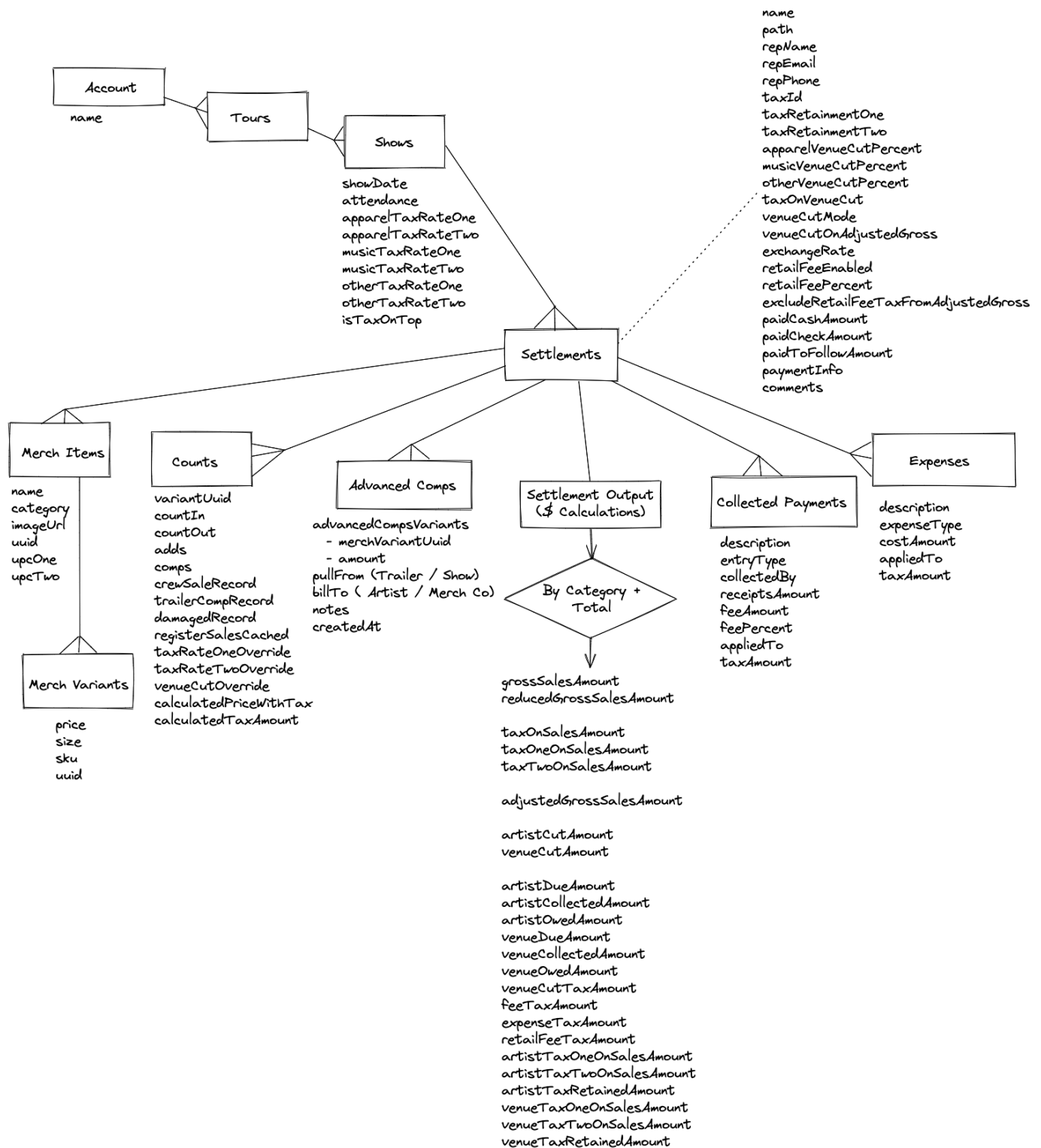
Settlements and Counts Read Specific Examples

- [Overview](#)
 - [Query](#)
 - [getFinalPayments for a Settlement](#)
 - [Paginating \(Paging\)](#)
 - [Filtering by Show date range](#)
 - [Filtering by Show status](#)
 - [Counts](#)
 - [Tax Summary](#)
 - [European Tax Summary](#)
 - [Canadian Tax Summary](#)
 - [Advanced Comps](#)
- [Appendix 1 - Schema](#)

Overview

The Settlements and Counts API is a public subset of atVenu's internal GraphQL API. In order to access your data, there is a small amount of domain knowledge that will be helpful. Figure 1 below outlines the structure and hierarchy of the data and types in the graph.

In summary, Settlement data is accessed through the Settlement type, which belongs to a Show. In turn a Show belongs to a Tour, which belongs to your Account.



Query

The following examples are intended to be a starting point for working with the API. GraphQL was designed to be dynamic, any field present in the schema in [Appendix 1](#) can be added and queried as required.

getFinalPayments for a Settlement

Query


An example query to access Final Payments for the Shows on a Tour or an Account:

```
query getFinalPayments(
  $firstForAccounts: Int!, $afterForAccounts: String,
  $firstForTours: Int!, $firstForShows: Int!) {
  organization {
    name
    accounts(first: $firstForAccounts, after: $afterForAccounts) {
      pageInfo {
        ...pageInfo
      }
      nodes {
        name
        tours(first: $firstForTours, open: true) {
          pageInfo {
            ...pageInfo
          }
          nodes {
            shows(first: $firstForShows) {
              pageInfo {
                ...pageInfo
              }
              nodes {
                uuid
                settlements {
                  name
                  uuid
                  paidCashAmount
                  paidCheckAmount
                  paidToFollowAmount
                }
              }
            }
          }
        }
      }
    }
  }
}

fragment pageInfo on PageInfo {
  endCursor
  hasNextPage
}
```

Variables

```
{
  "firstForAccounts": 20,
  "afterForAccounts": null,
  "firstForTours": 10,
  "firstForShows": 10,
  "firstForCounts": 10,
  "firstForItems": 10
}
```

 The max page size for Accounts is 20. Shows and Tours max page size is currently 200.

Paginating (Paging)

Paginating through collections *nested* within paginated collections is *not recommended*.

A paged collection nested in another paged collection that returns an object that has more than one page of data cannot be properly paged. The nested collection's parent should be queried directly so that the collection that requires paging can be properly paged at the top level

Example: The above sample query selects 4 collections, three of them nested:

- organization
 - accounts collection Can be paged because it is the first collection
 - tours collection nested, but will there will only be 1 tour for Venues
 - shows collection may need paging depending on arguments
 - settlements not a collection and does not require paging

To retrieve the settlements for a particular tour or account make a separate query for only that account and page through the shows

- account(uuid: \$accountUuid)
 - shows collection

Above example query uses the root of the graph to access the Settlements data. If you have the UUID for a specific Show or Settlement you could query those types directly using the node API (which allows direct access to individual nodes in the graph).

```
query specificShowFinalPayments {
  node(uuid: "<enter your uuid string here>") {
    ... on Show {
      settlements {
        uuid
        paidCashAmount
        paidCheckAmount
        paidToFollowAmount
      }
    }
  }
}
```

or

```

query specificSettlementFinalPayments {
  node(uuid: "<enter your uuid string here>") {
    ... on Settlement {
      uuid
      paidCashAmount
      paidCheckAmount
      paidToFollowAmount
    }
  }
}

```

It can also be useful to query specific Settlement values, such as Gross Sales, Adjusted Gross, amounts due to either party, and tax amounts. An example query:

```

query getSettlementOutput(
  $firstForAccounts: Int!, $afterForAccounts: String,
  $firstForTours: Int!, $firstForShows: Int!) {
  organization {
    name
    accounts(first: $firstForAccounts, after: $afterForAccounts) {
      pageInfo {
        ...pageInfo
      }
    }
    nodes {
      name
      tours(first: $firstForTours, open: true) {
        pageInfo {
          ...pageInfo
        }
      }
      nodes {
        shows(first: $firstForShows) {
          pageInfo {
            ...pageInfo
          }
        }
        nodes {
          uuid
          settlements {
            name
            uuid
            settlementOutput {
              apparel {
                summary {
                  grossSalesAmount
                  reducedGrossSalesAmount
                  adjustedGrossSalesAmount
                  taxOnSalesAmount
                  taxOneOnSalesAmount

```

```

        taxTwoOnSalesAmount
    }
}
other {
    summary {
        grossSalesAmount
        reducedGrossSalesAmount
        adjustedGrossSalesAmount
        taxOnSalesAmount
        taxOneOnSalesAmount
        taxTwoOnSalesAmount
    }
}
merch {
    grossSalesAmount
    reducedGrossSalesAmount
    adjustedGrossSalesAmount
    taxOnSalesAmount
    taxOneOnSalesAmount
    taxTwoOnSalesAmount
}
music {
    summary {
        grossSalesAmount
        reducedGrossSalesAmount
        adjustedGrossSalesAmount
        taxOnSalesAmount
        taxOneOnSalesAmount
        taxTwoOnSalesAmount
    }
}
total {
    grossSalesAmount
    reducedGrossSalesAmount
    adjustedGrossSalesAmount
    taxOnSalesAmount
    taxOneOnSalesAmount
    taxTwoOnSalesAmount

    artistTaxRetainedAmount
    venueTaxRetainedAmount
    artistDueAmount
    artistOwedAmount
    venueDueAmount
    venueOwedAmount
}
}
}
}
}
}

```

```

    }
  }
}

fragment pageInfo on PageInfo {
  endCursor
  hasNextPage
}

```



The apparel, music and other categories are distinct categories, based on the “category” of the items on your counts page. The merch category, is an aggregate of the apparel and other categories, due to the history of our settlements process.

Filtering by Show date range

The Shows connection can be provided with a `showsOverlap` argument, as a hash with a start and end date, to get Shows for a date range or specific day.

Query

```

...
  shows(first: $firstForShows, showsOverlap: $range) {
...

```

Variables

The variables used in this example will list Shows on Sept. 26, 2022.

```

{
  "firstForShows": 20,
  "range": {"start": "2022-09-26", "end": "2022-09-26"},
  ...
}

```

Include `$range: DateRange` at the start of your query definition, as GraphQL requires it for type check.

Filtering by Show status

To only view completed Shows use `filterViewState` argument.

Query

```
...
  shows(first: $firstForShows, filterViewState: $filter) {
...

```

Variables

```
{
  "firstForShows": 20,
  "filterViewState": "COMPLETED"
  ...
}
```

Include `$filterViewState: ShowListFilterEnum` at the start of your query definition, as GraphQL requires it for type check.

Counts

When accessing the Counts data, include the Counts connection from a Settlement selection. The query below includes `CountIn`, `CountOut`, `MerchAdds`, and `comps`.

Query

```
query getCounts(
  $firstForAccounts: Int!, $afterForAccounts: String,
  $firstForTours: Int!, $firstForShows: Int!,
  $firstForCounts: Int!) {
  organization {
    name
    accounts(first: $firstForAccounts, after: $afterForAccounts) {
      pageInfo {
        ...pageInfo
      }
      nodes {
        name
        tours(first: $firstForTours, open: true) {
          pageInfo {
            ...pageInfo
          }
        }
        nodes {
          shows(first: $firstForShows) {
            pageInfo {
              ...pageInfo
            }
          }
          nodes {
            uuid
            settlements {
              name

```



```

        uuid
        mainCounts(first: $firstForCounts) {
          nodes {
            merchVariantUuid
            merchItemUuid
            countIn
            countOut
            comps
            priceOverride
            merchAdds {
              quantity
            }
          }
        }
      }
    }
  }
}

fragment pageInfo on PageInfo {
  endCursor
  hasNextPage
}

```



Important: Counts data is normalized from the variant data and requires joining the merchVariantUuid field from the count with variants from the Show.

```

query getCountsWithVariantUuids (
  $firstForAccounts: Int!, $afterForAccounts: String,
  $firstForTours: Int!, $firstForShows: Int!,
  $firstForItems: Int!, $firstForCounts: Int!) {
  organization {
    name
    accounts(first: $firstForAccounts, after: $afterForAccounts) {
      pageInfo {
        ...pageInfo
      }
    }
    nodes {
      name
      tours(first: $firstForTours, open: true) {
        pageInfo {
          ...pageInfo
        }
      }
    }
  }
}

```

```

nodes {
  shows(first: $firstForShows) {
    pageInfo {
      ...pageInfo
    }
    nodes {
      uuid
      settlements {
        name
        uuid
        merchItems(first: $firstForItems) {
          nodes {
            name
            merchVariants {
              uuid
              size
              sku
            }
          }
        }
      }
      mainCounts(first: $firstForCounts) {
        nodes {
          merchVariantUuid
          countIn
          countOut
          comps
          merchAdds {
            quantity
          }
        }
      }
    }
  }
}

fragment pageInfo on PageInfo {
  endCursor
  hasNextPage
}

```

Tax Summary

To obtain the Tax Summary totals from the Settlements page, for a Canadian or EU / UK Shows try the following example.

Query

```
{
  node(uuid: "<enter your uuid string here>") {
    ... on Show {
      showDate
      uuid
      settlements {
        settlementOutput {
          total {
            taxOnSalesAmount
            taxOneOnSalesAmount
            taxTwoOnSalesAmount
            artistTaxOneOnSalesAmount
            artistTaxTwoOnSalesAmount
            venueTaxOneOnSalesAmount
            venueTaxTwoOnSalesAmount
            venueCutTaxAmount

            feeTaxAmount
            feeTaxOneAmount
            artistFeeTaxOneAmount
            venueFeeTaxOneAmount
            feeTaxTwoAmount
            artistFeeTaxTwoAmount
            venueFeeTaxTwoAmount

            expenseTaxAmount
            artistExpenseTaxAmount
            venueExpenseTaxAmount

            retailFeeTaxAmount
            artistRetailFeeTaxAmount
            venueRetailFeeTaxAmount

            artistTaxOneNetRetainedAmount
            artistTaxTwoNetRetainedAmount
            artistTaxRetainedAmount
            venueTaxOneNetRetainedAmount
            venueTaxTwoNetRetainedAmount
            venueTaxRetainedAmount
          }
        }
      }
    }
  }
}
```

Figure 2 and Figure 3 show the mapping of the fields to the UI.

European Tax Summary

Tax Summary ▾							Artist Tax: £84.36	Venue Tax: £95.70
Summary of Taxes								
Tax Type	Tax Rate	Retained By	Apparel	Other	Music	Total		
Sales	7%	Artist	£163.55	£16.51	£0.00	£180.06	1	
Tax on Cuts	7%	Venue	£26.45	£3.00	(£2.50)	£26.95	2	
Tax on Receipts	7%	Venue	£3.50	£0.00	£0.00	£3.50	3	
Tax on Adjustments	10%	Venue	£7.00	£0.00	£10.00	£17.00	4	
Tax on Retail Fee	7%	Venue	£43.75	£4.50	£0.00	£48.25	5	
Taxes Assessed								
Tax				Artist		Venue		
Tax on Sales				£180.06		£0.00		
Total Tax				£180.06		£0.00		
Tax on Cuts				(£26.95)		£26.95		
Tax on Receipts				(£3.50)		£3.50		
Tax on Adjustments				(£17.00)		£17.00		
Tax on Retail Fee				(£48.25)	6	£48.25	7	
Net Taxes Retained				£84.36	8	£95.70	9	

```
"settlements": [
{
  "settlementOutput": {
    "total": {
      1 "taxOnSalesAmount": "1.8006e+2",
        "taxOneOnSalesAmount": "1.8006e+2",
        "taxTwoOnSalesAmount": "0e+0",
        "artistTaxOneOnSalesAmount": "1.8006e+2",
        "artistTaxTwoOnSalesAmount": "0e+0",
        "venueTaxOneOnSalesAmount": "0e+0",
        "venueTaxTwoOnSalesAmount": "0e+0",
      2 "venueCutTaxAmount": "2.695e+1",
      3 "feeTaxAmount": "3.5e+0",
        "feeTaxOneAmount": "3.5e+0",
        "artistFeeTaxOneAmount": "-3.5e+0",
        "venueFeeTaxOneAmount": "3.5e+0",
        "feeTaxTwoAmount": "0e+0",
        "artistFeeTaxTwoAmount": "0e+0",
        "venueFeeTaxTwoAmount": "0e+0",
      4 "expenseTaxAmount": "1.7e+1",
        "artistExpenseTaxAmount": "-1.7e+1",
        "venueExpenseTaxAmount": "1.7e+1",
      5 "retailFeeTaxAmount": "4.825e+1",
      6 "artistRetailFeeTaxAmount": "-4.825e+1",
      7 "venueRetailFeeTaxAmount": "4.825e+1",
        "artistTaxOneNetRetainedAmount": "8.436e+1",
        "artistTaxTwoNetRetainedAmount": "0e+0",
      8 "artistTaxRetainedAmount": "8.436e+1",
        "venueTaxOneNetRetainedAmount": "9.57e+1",
        "venueTaxTwoNetRetainedAmount": "0e+0",
      9 "venueTaxRetainedAmount": "9.57e+1"
    }
  }
}
```

Canadian Tax Summary

Tax Summary ▾							Artist Tax: \$483.75	Venue Tax: \$1,076.25
Summary of Taxes								
Tax Type	Tax Rate	Retained By	Apparel	Other	Music	Total		
GST/HST on Sales	5%	Split	\$500.00	\$100.00	\$50.00	\$650.00	1	
PST on Sales	7%	Venue	\$700.00	\$140.00	\$70.00	\$910.00	2	
GST/HST on Credit Card Fees	5%	Venue	\$5.00	\$0.00	\$0.00	\$5.00	3	
PST on Credit Card Fees	7%	Venue	\$7.00	\$0.00	\$0.00	\$7.00	4	
Taxes Assessed								
Tax				Artist		Venue		
GST/HST on Sales				\$487.50	5	\$162.50	6	
GST/HST on Credit Card Fees				(\$3.75)	7	\$3.75	8	
GST/HST Subtotal				\$483.75	9	\$166.25	10	
PST on Sales				\$0.00		\$910.00		
PST on Credit Card Fees				\$0.00		\$0.00		
PST Subtotal				\$0.00		\$910.00		
Net Taxes Retained				\$483.75	11	\$1,076.25	12	

```
"settlements": [
{
  "settlementOutput": {
    "total": {
      "taxOnSalesAmount": "1.56e+3",
      1 "taxOneOnSalesAmount": "6.5e+2",
      2 "taxTwoOnSalesAmount": "9.1e+2",
      5 "artistTaxOneOnSalesAmount": "4.875e+2",
        "artistTaxTwoOnSalesAmount": "0e+0",
      6 "venueTaxOneOnSalesAmount": "1.625e+2",
        "venueTaxTwoOnSalesAmount": "9.1e+2",
        "venueCutTaxAmount": "0e+0",
        "feeTaxAmount": "1.2e+1",
      3 "feeTaxOneAmount": "5e+0",
      7 "artistFeeTaxOneAmount": "-3.75e+0",
      8 "venueFeeTaxOneAmount": "3.75e+0",
      4 "feeTaxTwoAmount": "7e+0",
        "artistFeeTaxTwoAmount": "0e+0",
        "venueFeeTaxTwoAmount": "0e+0",
        "expenseTaxAmount": "0e+0",
        "artistExpenseTaxAmount": "0e+0",
        "venueExpenseTaxAmount": "0e+0",
        "retailFeeTaxAmount": "0e+0",
        "artistRetailFeeTaxAmount": "0e+0",
        "venueRetailFeeTaxAmount": "0e+0",
      9 "artistTaxOneNetRetainedAmount": "4.8375e+2",
        "artistTaxTwoNetRetainedAmount": "0e+0",
      11 "artistTaxRetainedAmount": "4.8375e+2",
      10 "venueTaxOneNetRetainedAmount": "1.6625e+2",
        "venueTaxTwoNetRetainedAmount": "9.1e+2",
      12 "venueTaxRetainedAmount": "1.07625e+3"
    }
  }
}
```

Advanced Comps

For accounts with Advanced Comps enabled they will have the ability to view which comps were billed and to whom they were billed. Try the following example.


Query

```

query getAdvancedComps($firstForItems: Int) {
  node(uuid: "<enter your uuid string here>") {
    ... on Show {
      uuid
      settlements {
        name
        uuid
        merchItems(first: $firstForItems) {
          pageInfo {
            ...pageInfo
          }
          nodes {
            name
            merchVariants {
              uuid
              size
              sku
            }
          }
        }
      }
      advancedComps {
        advancedCompVariants {
          merchVariantUuid
          quantity
        }
        pullFrom
        billTo
        notes
        createdAt
      }
    }
  }
}

fragment pageInfo on PageInfo {
  endCursor
  hasNextPage
}

```

 Remember that all queries can use the node API when the Uuid is known for that a type.

Appendix 1 - Schema

""

```

A Show at a location and span in time
"""
type Show implements NodeInterface {
  apparelTaxRateOne: Decimal
  apparelTaxRateTwo: Decimal
  attendance: Int
  capacity: Int # How many people can attend this show, may be
different then location capacity
  currencyFormat: CurrencyFormat! # Local currency format for the show
  externalId: String # External Id associated with the Show; Max Length
of 255 Characters

  """
  True if the show is on a tour and the report currency of the tour
  does not match the currency of the show
  """
  isNotInTourCurrency: Boolean!

  """
  True if tax is calculated on top of the price, false if tax is
  included in the price
  """
  isTaxOnTop: Boolean!

  """
  List of itemized transactions at a show in reverse chronological
order. Note:
  pageInfo.hasPreviousPage is not supported for this connection.
  """
  itemizedTransactions: ItemizedTransactionConnection!
  location: Location # Where the show is taking place
  mobileOrdersDirectShowLink: String # Direct Link to a Mobile Orders
Site for this Show, if available
  musicTaxRateOne: Decimal
  musicTaxRateTwo: Decimal
  nights: Int! #Number of nights
  otherTaxRateOne: Decimal
  otherTaxRateTwo: Decimal
  settlements: [Settlement!]!
  showDate: Date! # Start date for the show
  showEndDate: Date! # End date for the show
  state: ShowStateEnum! # Current settlement state of the Show
  uuid: UUID! # Unique atVenu ID for the show, will start with 'show'
to indicate it is a show UUID
}

type Settlement {
  advancedComps: [AdvancedComp!]!
  apparelVendFee: Decimal # Apparel vend fee percentage.
  apparelVenueCutPercent: Decimal!

```

```

    cashFromCrewSalesAmount: Decimal!
    collectedPayments: [CollectedPayment!]! # The summary of payments
collected for this show
    comments: String
    exchangeRate: Decimal
    excludeRetailFeeTaxFromAdjustedGross: Boolean! # Applies to EU and UK
Settlements
    expenses: [Expense!]! # The list of expenses for this show
    legacySettlementOutput: Snapshot # Snapshotted values for legacy
atVenu settlements. Returns nil for current settlements.

    """
    The existing main counts for this settlement. May be less than the
number of merch variants.
    """
    mainCounts: MainCountConnection!
    merchItems: MerchItemConnection! # The merch items available for this
settlement
    musicVendFee: Decimal # Music vend fee percentage.
    musicVenueCutPercent: Decimal!
    name: String!
    otherVendFee: Decimal # Other vend fee percentage.
    otherVenueCutPercent: Decimal!
    paidCashAmount: Decimal
    paidCheckAmount: Decimal
    paidToFollowAmount: Decimal
    path: String! # A path to the settlement, for easy access in a web
browser.
    paymentInfo: String
    postedApparelTaxRateOne: Decimal # Apparel tax one at time of
settlement. Null for unsettled shows.
    postedApparelTaxRateTwo: Decimal # Apparel tax two at time of
settlement. Null for unsettled shows.
    postedMusicTaxRateOne: Decimal # Music tax one at time of settlement.
Null for unsettled shows.
    postedMusicTaxRateTwo: Decimal # Music tax two at time of settlement.
Null for unsettled shows.
    postedOtherTaxRateOne: Decimal # Other tax one at time of settlement.
Null for unsettled shows.
    postedOtherTaxRateTwo: Decimal # Other tax two at time of settlement.
Null for unsettled shows.
    repEmail: String
    repName: String
    repPhone: String
    retailFeeEnabled: Boolean! # Applies to EU and UK Settlements
    retailFeePercent: Decimal # Applies to EU and UK Settlements
    settledByEmail: String
    settledByName: String

    """

```

```

    The calculated settlement values based on your counts and settlements
page
    inputs. Returns a value snapshotted at the time of settlement for
settled
    shows. Returns nil for shows settled in the legacy atVenu settlement
process.
    """
    settlementOutput: SettlementOutput
    show: Show! # The show associated with the settlement
    status: SettlementStatusEnum!
    taxId: String
    taxOnVenueCut: Boolean # Applies to EU and UK Settlements
    taxRetainmentOne: SettlementTaxRetainmentOneEnum!
    taxRetainmentTwo: SettlementTaxRetainmentTwoEnum!
    uuid: UUID! # Unique atVenu ID for the settlement, will start with
'settlement' to indicate it is a settlement UUID
    venueCutMode: SettlementVenueCutModeEnum! # Applies to EU and UK
Settlements, otherwise always VENUE_CUT_ON_ADJUSTED_GROSS_WITH_TAX
}

type MainCount {
    """
    Price with tax calculated for the variant. Note: This field will be
null for unsettled shows.
    """
    calculatedPriceWithTax: Decimal

    """
    Total tax amount calculated for the variant. Note: This field will be
null for unsettled shows.
    """
    calculatedTaxAmount: Decimal
    comps: Int
    countIn: Int
    countOut: Int
    crewSalesRecord: PostSettlementAdjustment
    damagedRecord: PostSettlementAdjustment
    merchAdds: [MerchAdd!]! # List of adds for main counts
    merchItemUuid: UUID!
    merchVariantUuid: UUID!

    """
    Price associated with the count's variant . Populated with the
variants
    default price, or a user entered price of an override.
    """
    priceOverride: Decimal
    registerSalesCached: Int # Quantity of the variant sold on Register.
Cached amount as it is not a real-time value during a show.
    taxRateOneOverride: Decimal # This is count specific, but should be

```



```

synchronized across all variants for the item
    taxRateTwoOverride: Decimal # This is count specific, but should be
synchronized across all variants for the item
    trailerCompsRecord: PostSettlementAdjustment
    venueCutOverride: Decimal # This is count specific, but should be
synchronized across all variants for the item
}

type SettlementOutput {
    apparel: CategorySettlement!
    merch: SettlementOutputFields!
    music: CategorySettlement!
    other: CategorySettlement!
    outputCurrencyFormat: CurrencyFormat!
    total: SettlementOutputTotals!
}

type CategorySettlement {
    summary: SettlementOutputFields
}

type SettlementOutputFields {
    adjustedGrossSalesAmount: Decimal! # Value of goods sold minus
discounts, taxes, fees
    artistCutAmount: Decimal! # Artist cut for category
    artistRetailFeeAmount: Decimal
    artistRetailFeeTaxAmount: Decimal
    artistTaxOneNetRetainedAmount: Decimal # Tax one retained by artist
after accounting for tax on cuts, fees, expenses and retail fee.
    artistTaxOneOnSalesAmount: Decimal # Tax one on sales collected by
the artist
    artistTaxRetainedAmount: Decimal! # Total tax retained by artist
after accounting for tax on cuts, fees, expenses and retail fee.
    artistTaxTwoNetRetainedAmount: Decimal # Tax two retained by artist
after accounting for tax on cuts, fees, expenses and retail fee.
    artistTaxTwoOnSalesAmount: Decimal # Tax two on sales collected by
the artist
    expenseTaxAmount: Decimal! # Total tax calculated on expenses.
Available for EU and UK settlements.
    feeTaxAmount: Decimal! # Total tax calculated on payment fees.
Available for Canadian, EU and UK settlements.
    feeTaxOneAmount: Decimal # Tax one on fees calculated as part of the
settlement.
    feeTaxTwoAmount: Decimal # Tax two on fees calculated as part of the
settlement.
    grossSalesAmount: Decimal! # Value of goods sold
    reducedGrossSalesAmount: Decimal! # Value of goods sold minus
discounts
    retailFeeAmount: Decimal
    retailFeeTaxAmount: Decimal # Total tax calculated on payment fees,

```

when using the retail fee option. Available for EU and UK settlements.

taxOnSalesAmount: Decimal! # Tax charged on sales

taxOneOnSalesAmount: Decimal! # Tax one charged on sales

taxTwoOnSalesAmount: Decimal! # Tax two charged on sales

venueCutAmount: Decimal! # Venue cut for category

venueCutTaxAmount: Decimal! # Total tax on cuts taken by a venue.

Available for settlements in the EU and UK.

venueRetailFeeAmount: Decimal

venueRetailFeeTaxAmount: Decimal

venueTaxOneNetRetainedAmount: Decimal # Tax one retained by venue after accounting for tax on cuts, fees, expenses and retail fee.

venueTaxOneOnSalesAmount: Decimal # Tax one on sales collected by the venue

venueTaxRetainedAmount: Decimal! # Total tax retained by venue after accounting for tax on cuts, fees, expenses and retail fee.

venueTaxTwoNetRetainedAmount: Decimal # Tax two retained by venue after accounting for tax on cuts, fees, expenses and retail fee.

venueTaxTwoOnSalesAmount: Decimal # Tax two on sales collected by the venue

}

type SettlementOutputTotals implements SettlementOutputFieldsInterface {

adjustedGrossSalesAmount: Decimal! # Value of goods sold minus discounts, taxes, fees

artistCashFromShowAmount: Decimal! # For artist accounts only.

artistCashOnHandAmount: Decimal! # For artist accounts only.

artistCollectedAmount: Decimal!

artistCutAmount: Decimal! # Artist cut for category

artistDueAmount: Decimal!

artistExpenseAmount: Decimal! # Total of expenses paid to the artist.

artistExpenseTaxAmount: Decimal! # Tax on expenses being retained by artist. Will be negative if venue is being credited for the expense tax.

artistFeeAmount: Decimal! # Total fees on credit cards / other payments collected by the artist.

artistFeeTaxAmount: Decimal! # Tax on fees being retained by artist. Will be negative if venue is being credited for the fee tax.

artistFeeTaxOneAmount: Decimal # Tax one on fees being retained by artist. Will be negative if venue is being credited for the fee tax.

artistFeeTaxTwoAmount: Decimal # Tax two on fees being retained by artist. Will be negative if venue is being credited for the fee tax.

artistOwedAmount: Decimal!

artistReceiptAmount: Decimal! # Total of credit cards / other payments collected by the artist.

artistRetailFeeAmount: Decimal

artistRetailFeeTaxAmount: Decimal

artistTaxOneNetRetainedAmount: Decimal # Tax one retained by artist after accounting for tax on cuts, fees, expenses and retail fee.

artistTaxOneOnSalesAmount: Decimal # Tax one on sales collected by the artist

artistTaxRetainedAmount: Decimal! # Total tax retained by artist

after accounting for tax on cuts, fees, expenses and retail fee.

artistTaxTwoNetRetainedAmount: Decimal # Tax two retained by artist after accounting for tax on cuts, fees, expenses and retail fee.

artistTaxTwoOnSalesAmount: Decimal # Tax two on sales collected by the artist

cashFromCrewSalesAmount: Decimal!

expenseTaxAmount: Decimal! # Total tax calculated on expenses. Available for EU and UK settlements.

feeTaxAmount: Decimal! # Total tax calculated on payment fees. Available for Canadian, EU and UK settlements.

feeTaxOneAmount: Decimal # Tax one on fees calculated as part of the settlement.

feeTaxTwoAmount: Decimal # Tax two on fees calculated as part of the settlement.

grossSalesAmount: Decimal! # Value of goods sold

paymentFeesAmount: Decimal! # Total fees on credit cards / other payments collected.

paymentReceiptsAmount: Decimal! # Total of credit cards / other payments collected.

reducedGrossSalesAmount: Decimal! # Value of goods sold minus discounts

retailFeeAmount: Decimal

retailFeeTaxAmount: Decimal # Total tax calculated on payment fees, when using the retail fee option. Available for EU and UK settlements.

taxOnSalesAmount: Decimal! # Tax charged on sales

taxOneOnSalesAmount: Decimal! # Tax one charged on sales

taxTwoOnSalesAmount: Decimal! # Tax two charged on sales

totalOffBottomAmount: Decimal! # Total of 'Off the Bottom' expenses on the show.

totalWithheldAmount: Decimal! # Total of 'Tax Withholding' expenses on the show.

vendFeeAmount: Decimal! # Total of vend fee charged for the show. Requires vend fee enabled for the tour.

venueCollectedAmount: Decimal!

venueCutAmount: Decimal! # Venue cut for category

venueCutTaxAmount: Decimal! # Total tax on cuts taken by a venue. Available for settlements in the EU and UK.

venueDueAmount: Decimal!

venueExpenseAmount: Decimal! # Total of expenses paid to the venue.

venueExpenseTaxAmount: Decimal! # Tax on expenses being retained by venue. Will be negative if artist is being credited for the expense tax.

venueFeeAmount: Decimal! # Total fees on credit cards / other payments collected by the venue.

venueFeeTaxAmount: Decimal! # Tax on fees being retained by venue. Will be negative if artist is being credited for the fee tax.

venueFeeTaxOneAmount: Decimal # Tax one on fees being retained by venue. Will be negative if artist is being credited for the fee tax.

venueFeeTaxTwoAmount: Decimal # Tax two on fees being retained by venue. Will be negative if artist is being credited for the fee tax.

venueOwedAmount: Decimal!

```
venueReceiptAmount: Decimal! # Total of credit cards / other payments
collected by the venue.
venueRetailFeeAmount: Decimal
venueRetailFeeTaxAmount: Decimal
venueTaxOneNetRetainedAmount: Decimal # Tax one retained by venue
after accounting for tax on cuts, fees, expenses and retail fee.
venueTaxOneOnSalesAmount: Decimal # Tax one on sales collected by the
venue
venueTaxRetainedAmount: Decimal! # Total tax retained by venue after
accounting for tax on cuts, fees, expenses and retail fee.
venueTaxTwoNetRetainedAmount: Decimal # Tax two retained by venue
after accounting for tax on cuts, fees, expenses and retail fee.
venueTaxTwoOnSalesAmount: Decimal # Tax two on sales collected by the
venue
}
```