



Demonstrate ability to use totals models

sales.xml

```
<section name="quote">
  <group name="totals">
    <item name="tax" instance="Magento\Tax\Model\Sales\Total\Quote\Tax"
sort_order="450">
      <renderer name="adminhtml"
instance="Magento\Sales\Block\Adminhtml\Order\Create\Totals\Tax"/>
      <renderer name="frontend" instance="Magento\Tax\Block\Checkout\Tax"/>
    </item>
  </group>
</section>
```

Classes:

- Quote\TotalsCollector - does the job, `collect` and `collectAddressTotals` .
(`collectQuoteTotals` not used)

Quote totals

- `quote.collectTotals`
- `total = totalsCollector.collect` - crunch numbers, return data to set on quote
 - create *quote total object* `quote\address\total`. all you set here will be copied to quote
 - event `sales_quote_collect_totals_before`
 - collect item qtys - `quote.itemsCount`, `quote.itemsQty`, `quote.virtualItemsQty`
 - zero total object amounts - subtotal, grand total etc.
 - collect each *address totals*:
 - *collect address totals* into new address total object
 - `address total = quote.collectAddressTotals` (see below)
 - add to quote total object: shipping amount, shipping description, subtotal, subtotal with discount, grant total (+ base versions when applicable)
 - validate max grand total amount 99999999
 - validate coupon code - check that at least one address has it, otherwise reset to "" - code is invalid
 - event `sales_quote_collect_totals_after`
- `quote.addData(total.getData)`

Address totals

`quote.collectAddressTotals` for each address (billing, shipping) - set address fields, return some fields for quote

- new shipping assignment obj:
 - `shipping = [method, address]`
 - items = address items. empty for normal checkout?
- create new *address total object* `quote\address\total`. all you set here will be copied to address
- event `sales_quote_address_collect_totals_before`
- get collectors:
 - `_initModels -> _initModelInstance -> model.processConfigArray` – tax collector can change its sort programmatically
 - `sales/totals_sort/*` - admin configurable *totals retrievers* ordering - display order
- *every collector[] collect*

- all data set on address total object will be copied to address. Main amounts will also copy to quote total object.
- has direct access to quote
- has access to address via shipping assignment.getShipping.getAddress
- event `sales_quote_address_collect_totals_after`
- `address.addData(address total.getData)`

Display totals

Totals rendered in cart in UI-component fashion with JS configuration.

Totals JS models extend `Magento_Checkout/js/view/summary/abstract-total` and implement:

- `getPureValue` - raw float
- `getValue` - formatted via `Magento_Catalog/js/price-utils` in format `window.checkoutConfig.priceFormat`.

Models get their values from model `quote.getTotals()`, e.g. `quote.getTotals()().subtotal`. Initial totals values are in `window.checkoutConfig.totalsData`, any extended attributes are merged into main totals.

`window.checkoutConfig` values are provided by `\Magento\Checkout\Model\CompositeConfigProvider`. Default config provider `\Magento\Checkout\Model\DefaultConfigProvider`. Register your classes via arguments.

Default config provider.`getTotalsData` and REST API reuse same repository.

`\Magento\Quote\Model\Cart\CartTotalRepository::get`:

- billing or shipping address.`getTotals`
 - `\Magento\Quote\Model\Quote\TotalsReader::fetch`
 - each collector[].`fetch`(quote, data)
 - can return `\Magento\Quote\Model\Quote\Address\Total` or array [code, title, value]
 - title can be object with method `render()` - by default this is just __ Phrase translation
 - can return multiple totals as once
 - can overwrite other totals by code
- convert address data to interface `Cart\Totals` - totals by code + quote visible items
- *totals segments* - totals by code as return from fetch:
 - `\Magento\Quote\Model\Cart\TotalsConverter::process` - convert address totals to total segment
 - title = `titleRenderer.render`
- add other info - coupon code, grand total, items, items qty, currency code

js model `quote.getTotals`:

```
[
  {address data fields = populate \Magento\Quote\Api\Data\TotalsInterface with
  address.getData},
  items = quote visible items,
  total_segments = as returned from totals collector.fetch
]
```

Invoice totals, credit memo totals

```

<section name="order_invoice">
    <group name="totals">
        <item name="subtotal"
instance="Magento\Sales\Model\Order\Invoice>Total\Subtotal" sort_order="50"/>
    </group>
</section>
<section name="order_creditmemo">
    <group name="totals">
        <item name="subtotal"
instance="Magento\Sales\Model\Order\Creditmemo>Total\Subtotal" sort_order="50"/>
    </group>
</section>

```

Invoice totals:

- \Magento\Sales\Model\Order\Invoice::collectTotals
- every total - \Magento\Sales\Model\Order\Invoice>Total\AbstractTotal::collect(invoice)
 - directly update invoice object

Credit memo totals:

- \Magento\Sales\Model\Order\Creditmemo::collectTotals
- every total - \Magento\Sales\Model\Order\Creditmemo>Total\AbstractTotal::collect
 - directly update credit memo object

Render totals in admin area:

Example - admin invoice totals:

handle `sales_order_invoice_view` :

```

<block class="Magento\Sales\Block\Adminhtml\Order\Invoice\Totals"
name="invoice_totals" template="Magento_Sales::order/totals.phtml">

```

- block totals. `__beforeToHtml`
- block order totals. `__initTotals` - add hardcoded totals - subtotal, shipping, discount, grand_total, base_grandtotal
- every child block. `initTotals`
- child blocks call `block order totals.addTotal(total, after={'first'|'last'|$code})` or `addTotalBefore(total, before)`
 - total = DataObject [code, value, base_value, label, optional: {area, block_name, is_formated, strong}]
- block order totals template
 - totals with area = 'footer'
 - totals with empty area
 - if 'block_name', `getChildHtml` - simply renders child block
 - otherwise print [label, value], value will be price formatted unless flag 'is_formated'

Describe how to modify the price calculation process in the shopping cart.

- when preparing product in product type model, can set `product.custom_price`
- register quote totals collector in sales.xml. Edit given `address_total`. Examples:
 -

```

$total->setGrandTotal($total->getGrandTotal() - $pointsCurrencyAmountUsed);

```

- call `$total->addTotalAmount(code, value)` . Grand total will then be sum of all total amounts - they can be negative.

How can you add a custom totals model or modify existing totals models?

- add quote totals collector - declare in sales.xml, implement `collect` and `fetch`
 - in collect can modify address totals object (preferred), or edit directly quote or quote address via shipping assignment
 - fetch can return null, or one or many declarations in array format or ready Address\Total objects
 - fetch can rewrite other totals with same 'code'
 - in cart block JS config, add view model and implement `getPureValue`, `getValue`. Can read values from model `quote.getTotals().totals_segments`
- Render totals in admin:
 - `sales_invoice_view`, referenceBlock 'invoice_totals', add block with method `initTotals` , call `parentBlock.addTotal`