

# Checkout

# What is checkout?

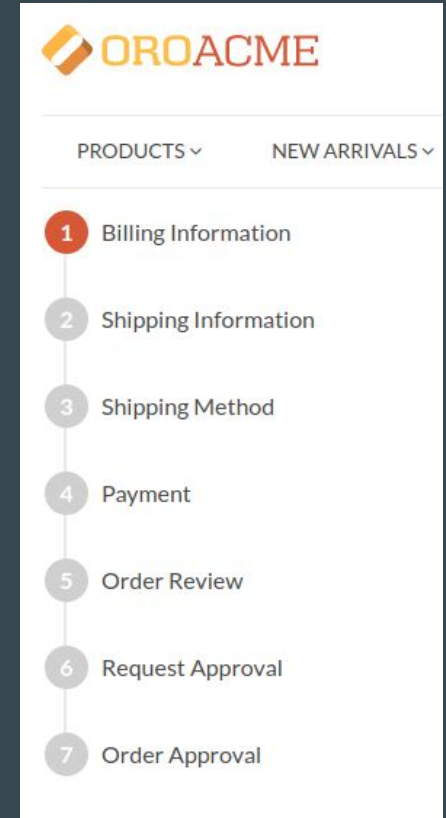
- Process for checking out the items in the cart
- Series of steps
  - Billing information
  - Shipping information
  - Shipping method
  - Payment
  - Order review
  - ...and more
- Checkout is not an order yet

# Checkout at OroCommerce

- Workflow based
  - Highly customizable
- Can be started from any entity
  - Shopping List
  - Quote
  - Order
- Can be multi page or single page
- Can be enabled for guests (anonymous users)
- OroCheckoutBundle

# Out of the box checkouts

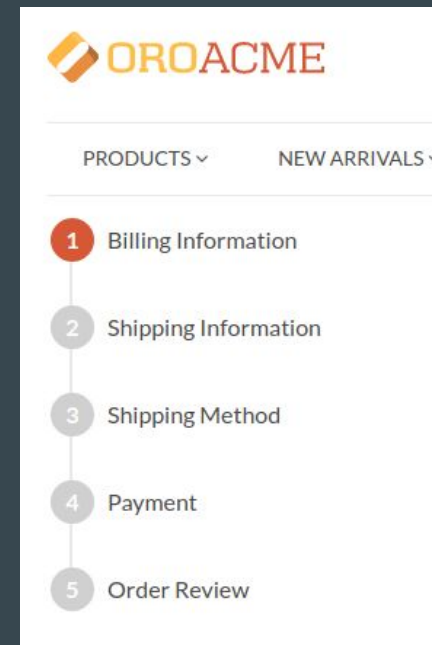
- Standard Checkout
- Alternative Checkout
  - Represents *an example* of the checkout workflow customization
  - Request and order approval steps
- Single Page Checkout



*Demonstration >>>*

# Standard checkout: Steps

- Billing information
- Shipping information
- Shipping method
- Payment
- Order review
- Order created
- [Configuration]



# Standard checkout: Start

- Start transitions
- Action group

# Standard checkout: Transitions

- Transition definitions
- Move to next step
- Move to previous steps
- Finish checkout

# Checkout Controller

- Transitions
- Checkout restart
  - checkout.checkout\_validate, Example
- Warning and error messages
  - Form errors
  - Custom flash messages defined in workflow
- Recommendation
  - good place to start debugging

*Demonstration >>>*



# Checkout UI

- Layout updates
  - Main tree with imports
  - Layout updates files
    - Billing step example
  - Templates
    - Billing step example
- Translations
  - Standard workflow translations

# Checkout Customization

- Copy checkout workflow
- Customize workflow
  - attributes
  - steps
  - transitions
  - transition definitions
- Customize layout updates
- Customize controller

# Best Practices

- Implement all standard (positive) scenarios
- Implement not standard (negative) scenarios
- Use workflow variables for the configuration
  - Not all have to be persistent
- Create custom conditions and actions for the complicated logic
- Add informational fields to the Order review step

# Best Practices

- Line items from shopping list / quote / order are converted to checkout line items using one of the implementations of `Oro\Bundle\CheckoutBundle\Model\CheckoutLineItemConverterInterface`
- Developer may add new entity source for checkout - need to add relation to `oro_checkout_source` table, implement `CheckoutLineItemConverterInterface` and start checkout workflow (e.g. using start transition)

# Quality Assurance

- Check all possible ways to start checkout
- Check scenarios with real data
- Check negative scenarios