

Den Checkout-Prozess verstehen und optimieren

(für Entwickler)

Tobias Vogt

0 52 51/ 6 81 12 – 16

tobias.vogt@code-x.de

www.code-x.de

- Tobias Vogt aus Paderborn
- 25 Jahre
- Magento seit 2008
- (bald) Wirtschaftsinformatiker

Projekte

- www.webguys.de
- www.magento-podcast.de
- Artikel im PHP-Magazin zu Theming + Widgets
- Empfehlungssysteme mit Magento



Wer bin ich?

- ~~Wer bin ich?~~
- Die Frage nach dem „Warum“
- Grundlagen
- Theming
- Steps hinzufügen oder entfernen

- Diskussion: Warum eigentlich JS?

Agenda

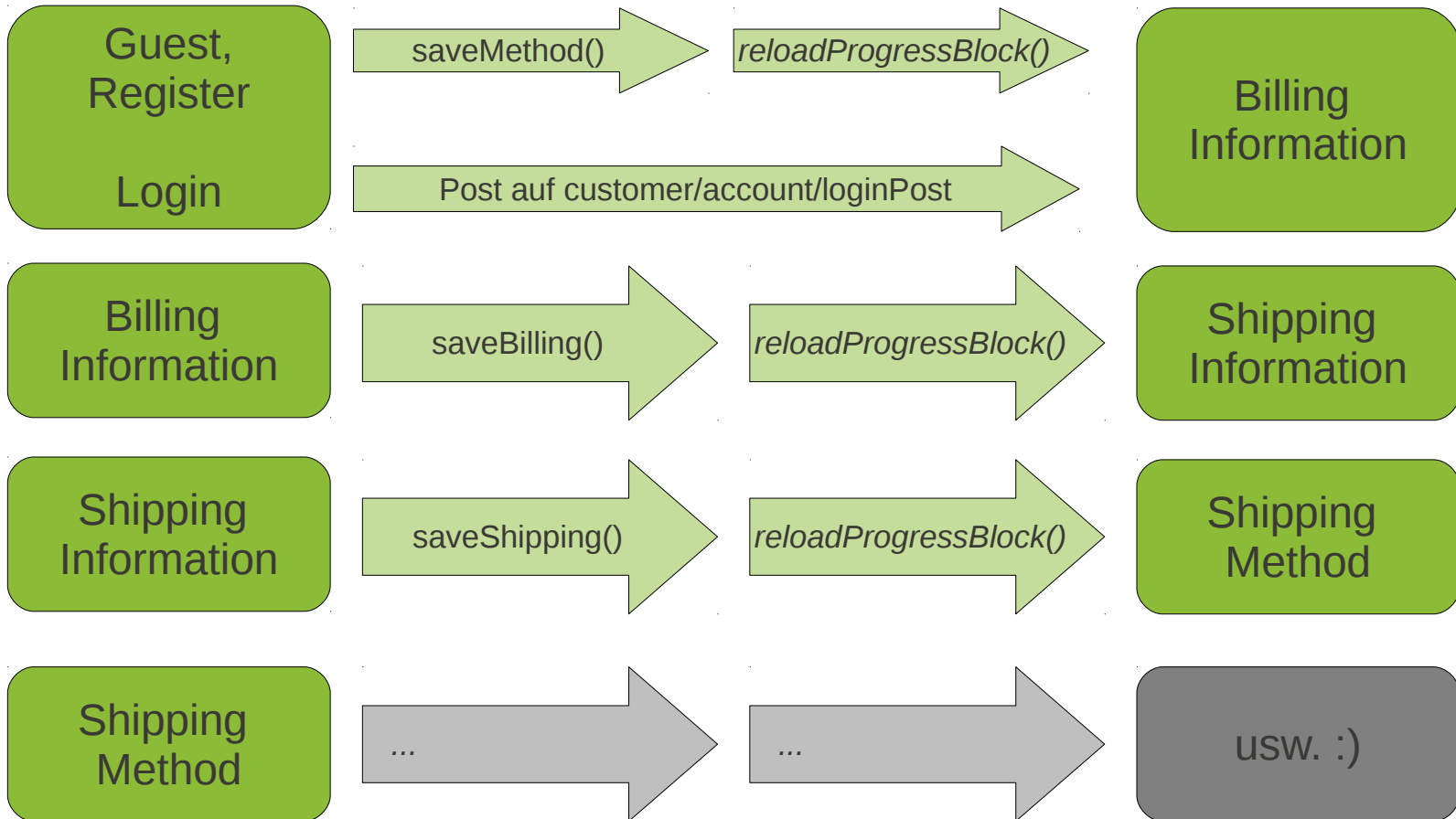
Steigerung der Conversion-Rate

Warum?

Die Säulen des Checkouts

- `optcheckout.js`
Steuert Ablauf mittels Java-Script und JSON-Rückgaben
- `Templates+Blöcke in Checkout/onepage/..`
generieren HTML, rufen JS-Calls auf
- `Mage_Checkout_OnepageController`
Reicht Daten an Model `checkout/type_onepage` weiter, gibt JSON zurück

Der Ablauf



Grundlagen des Checkouts

optcheckout.js (1) (skin/frontend/base/default/js)

nextStep → Auswertung des Requests

- If (response.redirect)
 location.href = response.redirect;
- If (response.success)
 window.location=this.successUrl;
- If (!response.success)
 Fehlerausgabe mittels alert(msg);

optcheckout.js (2)

- ```
if (response.update_section) {
 $('checkout-'+response.update_section.name+'-load')
 .update(response.update_section.html);
}
```
- ```
if (response.goto_section) {  
    checkout.gotoSection(response.goto_section);  
    checkout.reloadProgressBlock();  
}
```


optcheckout.js (3)

reloadProgressBlock → Fortschrittsbalken aktualisieren

- Jeweils ein separater Ajax-Request
- Container mit der Id 'checkout-progress-wrapper' wird aktualisiert

```
new Ajax.Updater('checkout-progress-wrapper',  
    this.progressUrl, [..] );
```

Mage_Checkout_OnepageController (1)

save[Method/Billing/Shipping/ShippingMethod/Payment/Order]Action();

- Nimmt POST-Daten entgegen
- Gibt Daten an Model OnePage weiter
 - Bei Fehlern: Fehler als JSON aufbereiten oder heilen
 - Bei Erfolg: NextStep-JSON aufbereiten
→ Was kommt als nächstes?
- Geben (fast) immer JSON zurück

Grundlagen des Checkouts

Mage_Checkout_OnepageController (2)

saveShippingMethodAction()

- ```
$result['goto_section'] = 'payment';
$result['update_section'] = array(
 'name' => 'payment-method',
 'html' => $this->_getPaymentMethodsHtml()
);
```
- ```
$this->getResponse()->setBody(  
    Mage::helper('core')->jsonEncode($result)  
);
```

Mage_Checkout_Model_Type_OnePage

save[Billing/Shipping/ShippingMethod/Payment/Order]();

- Daten vom Controller entgegennehmen

Array('firstname' => 'Tobias', 'lastname' => 'Vogt')

- Validieren

*return array('error' => 1, '
message' => \$this->_helper->__('Customer
Address is not valid.'));*

- Im Quote speichern

Grundlagen des Checkouts

Mage_Sales_Model_Quote

- Unfertige Bestellung
 - addItem, addProduct, getItemByProduct
 - getPayment, setPayment, getTotals
- Mittels Mage_Sales_Model_Convert_Quote::toOrder zu Bestellung konvertieren

Blöcke

Mage_Checkout_Block_Onepage

Mage_Checkout_Block_Onepage_Billing

Mage_Checkout_Block_Onepage_Payment

Mage_Checkout_Block_Onepage_Shipping

USW.

- Bereiten Daten auf
- geben Daten an das Template zur Darstellung

Grundlagen des Checkouts

Probleme beim Theming

- Aufwendig da kein Page-Reload möglich
 - Für CSS: ReCSS verwenden
(<http://david.dojotoolkit.org/recss.html>)
 - Für JS + HTML: Tjaa...
 - Idee: In *Mage_Checkout_Block_Onepage* die Reihenfolge der 'stepCodes' ändern

```
$stepCodes = array('billing', 'shipping', 'shipping_method', 'payment', 'review');
```

Steps entfernen oder hinzufügen

ToDo-List

- Neuen Block erstellen
- Im eigenen Layout-XML verankern (vgl. checkout.xml)
- Block unter Mage_Checkout_Block_Onepage hinzufügen
- Next-Steps der benachbarten Controller-Actions ändern

Das heißt..

- viel Arbeit mit umfangreichen Tests
- kritischer Prozess, muss fehlerfrei funktionieren

→ An Wirtschaftlichkeit denken!

Warum Java-Script?

Probleme

- JS ist ungewohnt für Benutzer (Vor, Zurück, Timeouts..)
- JS ist fehleranfälliger + komplizierter zu testen
- Daten werden erst am Ende gespeichert
(Beim Reload ist manchmal alles weg..)

Lösungen

- JS entfernen
- Pro Schritt eine Action im Controller
(d.H. echte Formulare, neue Controller, erweiterte Blöcke..)
→ <http://code.google.com/p/simplecheckout/>

Platz für Diskussion

.. Danke!

Folgt mir!

www.twitter.com/tobi_pb

www.facebook.com/webguys

www.webguys.de

www.xing.com/profile/Tobias_Vogt16

Tobias Vogt

0 52 51/ 6 81 12 – 16

tobias.vogt@code-x.de

www.code-x.de