

# PDF-Formulare ausfüllen mit iText & Co.

## Den Papierkrieg gewinnen...

Marcus Bitzl  
marcus@bitzl.com

DANTE-Herbsttagung in Köln  
2. November 2013

# Situation: Tutorbetrieb Informatik (TUM)

mehr als

80

Tutoren jedes Semester

# Situation: Tutorbetrieb Informatik (TUM)

mehr als

80

Tutoren jedes Semester

# Situation: Tutorbetrieb Informatik (TUM)

ca.

25

Angaben je Einstellungsvorschlag

## Situation: Tutorbetrieb Informatik (TUM)

mehr als

2000

Angaben in den Einstellungsvorschlägen eines Semesters

## Situation: Tutorbetrieb Informatik (TUM)

mehr als

140

Hospitationen

# 2-3

Mitarbeiter (Voll- und Teilzeit)

Formulare :-)



# Unsere Lösung

Eine Webanwendung, die

- ▶ alle relevanten Informationen zum Einstellungsprozess verwaltet bzw. berechnet
- ▶ ... und damit den Einstellungsvorschlag ausfüllt
- ▶ alle relevanten Informationen zu den Hospitationen verwaltet
- ▶ ... und damit die Orga-Daten des Hospitationsbogens ausfüllt

# Unsere Lösung – Werkzeuge

- ▶ Google Webtoolkit (GWT) für die Oberfläche
- ▶ Servlets für die Formulare
- ▶ MySQL
- ▶ Tomcat

# Verworfen: TeX generieren und PDF erzeugen

Der Hospitationsbogen stammt von uns, den Einstellungsvorschlag könnten wir nachbauen. Also warum nicht

- ▶ TeX-Datei mit der Webanwendung erzeugen
- ▶ PDF für den Benutzer mit TeX generieren

Problem:

- ▶ asynchrone Ausführung: Webanwendung startet  $\text{\LaTeX}$  und muss prüfen, wann der Lauf beendet ist
- ▶ der Client darf nicht »hängen«
- ▶ ein  $\text{\LaTeX}$ -Lauf dauert...
- ▶ Entwickelt auf Windows, läuft unter Linux: Fehlerquellen beim Testen

⇒ zu komplex

# Unsere Lösung: PDF-Formulare ausfüllen

- ▶ Der Hospitationsbogen wird als PDF-Formular mit  $\text{\LaTeX}$  erzeugt
- ▶ Die Webanwendung füllt die Felder mit Werten aus der Datenbank
- ▶ Der Benutzer erhält das Dokument innerhalb von Sekunden
- ▶ Die Ausführung ist synchron

# Formulare mit L<sup>A</sup>T<sub>E</sub>X: Minihosptiationsbogen

Name

---

Uncool    ○ ○ ○ ○ ○    Cool

# Formulare mit $\text{\LaTeX}$ : Minihostptionsbogen

```
\begin{framed}
  \begin{Form}
    \TextField[name=name]{Name} % <--- Formularfeld
  \end{Form}

  \hrulefill

  \begin{center}
    \Huge
    Uncool\hspace{0.5em}
    $\circ\circ\circ\circ\circ$
    \hspace{0.5em} Cool
  \end{center}
  \smallskip
\end{framed}
```

# Formulare mit hyperref

## Form

Alle Felder eines Formulars werden eingeschlossen in

```
\begin{Form}[action=http://example.com]  
...  
\end{Form}
```

- ▶ Nur ein Form je Datei
- ▶ Der Optionale Parameter `action` erlaubt die Formulardaten an eine URL zu senden.

# Fomulare mit hyperref

## Einfache Felder

### Beispiel

- ▶ TextField
- ▶ CheckBox

### Code

```
\TextField[value={42}]{TextField}
```

```
\CheckBox[checked]{CheckBox}
```

### Anmerkung

Mit hyperref erzeugte Checkboxes funktionieren nicht mit allen PDF-Werkzeugen reibungslos: Adobe Acrobat zeigt das »checked« von hyperref nicht an. iText kann nur Haken entfernen, aber nicht setzen.



# Fomulare mit hyperref

## Choices

### Beispiel

- ▶ ChoiceMenu
- ▶ Combo
- ▶ Radio Eins      Zwei      Drei

### Code

```
\ChoiceMenu{ChoiceMenu1}{Eins,Zwei,Drei}  
\ChoiceMenu[combo]{Combo}{Eins,Zwei,Drei}  
\ChoiceMenu[radio]{Radio}{Eins,Zwei,Drei}
```

# Fomulare mit hyperref

## Buttons

### Beispiel

- ▶ PushButton
- ▶ Submit
- ▶ Reset

### Code

```
\PushButton[onclick={app.alert("Hallo Dante!",  
    2);}]{PushButton}  
\Submit{Submit}  
\Reset{Reset}
```

# Fomulare mit hyperref

Wie funktioniert das Layout (Beispiel)

Die Kombination aus Label und Textfeld wird so gesetzt:

```
\LayoutTextField{#1}{#2}
```

Per default #1 #2, ergibt also Label Feld.

# Ausfüllen mit iText

iText ist eine freie Programmbibliothek zum Erstellen und Bearbeiten von PDF-Dateien:

`http://itextpdf.com`

- ▶ Java, .Net und Android
- ▶ GNU Affero General Public License (AGPL)
- ▶ Kommerzielle Lizenz verfügbar.

# Ausfüllen mit iText

Zurück zu unserem Minibogen:

```
PdfReader reader = new PdfReader("minibogen.pdf");  
OutputStream out = new FileOutputStream("out.pdf");
```

```
PdfStamper stamper = new PdfStamper(reader, out);  
AcroFields form = stamper.getAcroFields();  
form.setField("Name", "Hans Wurst");
```

```
stamper.close();  
reader.close();
```

# Ausfüllen mit iText

Senden von einer Webanwendung (Servlet):

```
protected void doGet(HttpServletRequest req,
                      HttpServletResponse resp) {
    PdfReader reader = new PdfReader("minibogen.pdf");
    OutputStream out = response.getOutputStream();

    response.setContentType("application/pdf");
    PdfStamper stamper = new PdfStamper(reader, out);
    AcroFields form = stamper.getAcroFields();
    form.setField("Name", "Hans Wurst");

    stamper.close();
    reader.close();
}
```

# Minihospitationsbogen II

Name \_\_\_\_\_

Uncool

Cool

# Minihospitationsbogen II

Wir erweitern den Bogen:

```
\begin{center}
  \Huge Uncool\hspace{0.5em}
  \large
  \ChoiceMenu[radio,name=coolness]{}{
    {}={uncool},
    {}={fast uncool},
    {}={wer ist das?},
    {}={eher cool},
    {}={saucool}
  }
  \Huge \hspace{0.5em} Cool
\end{center}
```



# Minihospitationsbogen II

Und können den Tutor automatisch bewerten ;-)

```
String field = "coolness";  
String[] states = form.getAppearanceStates(field);  
int evaluation = (int) (Math.random() * states.length);  
form.setField(field, states[evaluation]);
```

## Alternative: PDFtk

```
# Daten via FDF-File
```

```
pdftk form.pdf fill_form data.fdf output form.filled.pdf
```

```
# Daten via STDIN
```

```
pdftk form.pdf fill_form - output form.filled.pdf
```

Mehr unter

```
http://www.pdflabs.com/docs/pdftk-man-page/  
#dest-op-fill-form
```

# Material im Internet

`https://github.com/bitzl/dante-herbsttagung-formulare`

Vielen Dank für die  
Aufmerksamkeit!

Fragen?