

T_EXtacular

Trevor Maglione

11 March 2014

Overview

Idea

- Make \LaTeX accessible to everyone
- Simplify creation of beautiful documents

Audience

- Users who prefer point-and-click and simple forms over manual creation of \TeX source
- Those less technically inclined

Implementation

Initial view

The initial view of the ToXtacular application window shows a form for creating a document. The form has tabs for 'Letter', 'Article', and 'Invoice'. The 'Letter' tab is selected. The form includes fields for 'Address', 'To', 'Date', 'Opening', 'Content', 'Closing', and 'Signature'. A large preview panel on the right is empty. At the bottom, there are buttons for 'Compile', 'View', 'Clear', and 'Options'.

View after generating an invoice

The view of the ToXtacular application window after generating an invoice shows the 'Invoice' tab selected. The form is populated with data for an invoice. The 'Company' field is 'Foo Bar', the 'Address' is '123 Baz St % Sacramento, CA', the 'To' field is 'John Smith', and the 'Date' is '12 December 2013'. The 'Rate' is '65'. The 'Entries' table is populated with two rows: 'Did something' with a rate of 2.0 and 'Did something else' with a rate of 3.0. The 'Note' field is 'Browsed Reddit' and the 'Hours' field is '10.0'. The preview panel on the right shows the generated invoice for 'Woo Bar'.

Description of Services	Quantity	Unit Price	Amount
Did something	2.00 hours	\$65.00	\$130.00
Did something else	3.00 hours	\$65.00	\$195.00
Subtotal	5.00 hours		\$325.00
Balance Due			\$325.00

Implementation Details

- Main frame contains a concrete `TemplatePanel` with inputs, a few buttons at the bottom, and a `PreviewPanel` on the right
- A `TeXHandler` class handles compilation of $\text{T}_{\text{E}}\text{X}$ source
 - Uses `latexmk` and a script (`res/bin/compile`) to compile
- `PreviewPanel` updates automatically with the contents of the compiled PDF
- Third party libraries
 - `MiGLayout`
 - `Sun PDF-Renderer`
 - `PgsLookAndFeel`

Difficulties & Concerns

- How best to handle template types
- Reasonable method of laying out components without a completely different strategy for each template type
- Auto-updating PDF panel and zooming (conversion of PDF to a `BufferedImage` – far from perfect
- Handling application and `latexmk` compilation errors

Future

- Improve scrolling inside preview panel
 - Currently click and drag but can be flaky
- More general, approachable way for building templates
- Improve scaling of PDF preview
- Add “Save”, “Load”, “Print” options
- Add Windows support
- Add options dialogue for font/package selection and paper size