pdfium-render v0.8.35

A high-level idiomatic Rust wrapper around Pdfium, the C++ PDF library used by the Google Chromium project.

#pdf #pdfium

Idiomatic Rust bindings for Pdfium

`pdfium-render` provides an idiomatic high-level Rust interface to Pdfium, the C++ PDF library used by the Google Chromium project. Pdfium can render pages in PDF files to bitmaps, load, edit, and extract text and images from existing PDF files, and create new PDF files from scratch.

```
use pdfium_render::prelude::*;

/// Renders each page in the PDF file at the given path to a separate JPEG file.
fn export_pdf_to_jpegs(path: &impl AsRef<Path>, password: Option<&str>) -> Result<(), PdfiumError> {
    // Bind to a Pdfium library in the same directory as our Rust executable.
    // See the "Dynamic linking" section below.

    let pdfium = Pdfium::default();

    // Load the document from the given path...

    let document = pdfium.load_pdf_from_file(path, password)?;

    // ... set rendering options that will be applied to all pages...

    let render_config = PdfRenderConfig::new()
        .set_target_width(2000)
        .set_maximum_height(2000)
        .rotate_if_landscape(PdfPageRenderRotation::Degrees90, true);

    // ... then render each page to a bitmap image, saving each image to a JPEG file.

    for (index, page) in document.pages().iter().enumerate() {
        page.render_with_config(&render_config)?
            .as_image() // Renders this page to an image::DynamicImage...
            .into_rgb8() // ... removes the alpha channel for compatibility with the Jpeg format...
            .save_with_format(
                format!("page-{}.jpg", index),
                image::ImageFormat::Jpeg
            ) // ... and saves the image to a file.
            .map_err(|_| PdfiumError::ImageError)?;
    }

    Ok(())
}
```

`pdfium-render` binds to a Pdfium library at run-time, allowing for flexible selection of system-provided or bundled Pdfium libraries and providing idiomatic Rust error handling in situations where a Pdfium library is not available. A key advantage of binding to Pdfium at run-time rather than compile-time is that a Rust application using `pdfium-render` can be compiled to WASM for running in a browser alongside a WASM-packaged build of Pdfium.