This project started life as a DOCX templating engine. It has now envolved to also support converting HTML to PDF using a headless version of webkit, phantomjs.

The DOCX templating is great for documents that end clients update and manage over time, particularly text heavy documents. For example I use it to auto generate some legal contracts, where simple replacements are made for attributes like First Name, Last Name, Company Name & Address. The client, an insurance company, can provide updated template word documents that might contain subtle changes to policies & other conditions.

The HTML to PDF engine is great for cases where greater control over the design of the document is required. It's also more natural for us programmers, using standard HTML & CSS, with a splash of Javscript.

How to Use, the basics

Both APIs are accessed through the main `Pdf` class.

To convert a word document into a PDF without any templating:

```
$pdf = Gears\Pdf::convert('/path/to/document.docx');
```

To save the generated PDF to a file:

```
Gears\Pdf::convert('/path/to/document.docx', '/path/to/document.pdf');
```

To convert a html document into a PDF:

```
$pdf = Gears\Pdf::convert('/path/to/document.html');
```

> NOTE: The save to file works just the same for a HTML document.

DOCX Templating

By default the DOCX backend defaults to using `libre-office-headless`, to use `unoconv`, override the converter like so:

```
$document = new Gears\Pdf('/path/to/document.docx');
$document->converter = function()
{
        return new Gears\Pdf\Docx\Converter\Unoconv();
};
$document->save('/path/to/document.pdf');
```

> NOTE: Currently the HTML backend only uses phantomjs.