Parsing PDF files in Go

Parsing PDF files in Go is useful when you need to extract text or metadata from PDF documents for analysis or transformation. In this article, we’ll see how to read a PDF file using the open-source library `rsc.io/pdf`.

Installing the Library

The `rsc.io/pdf` library is one of the simplest and most straightforward solutions for parsing PDFs in Go. To install it, simply run:

```
go get rsc.io/pdf
```

Reading Text from a PDF

Once the library is installed, we can write a simple program to open a PDF file and extract text from each page. Here’s a complete example:

```
package main

import (
    "fmt"
    "log"
    "rsc.io/pdf"
)

func main() {
    reader, err := pdf.Open("document.pdf")
    if err != nil {
        log.Fatalf("Error opening file: %v", err)
    }
    for pageIndex := 1; pageIndex <=
reader.NumPage(); pageIndex++ {
        page := reader.Page(pageIndex)
        if page.V.IsNull() {
            continue
        }

        content := page.Content()
        for _, text := range content.Text {
            fmt.Print(text.S)
        }
    }
}
```

Considerations

The `rsc.io/pdf` library is sufficient for basic text parsing operations, but it has limitations. For example, it doesn’t fully support all complex or encrypted PDF formats. In such cases, you might consider alternatives like `unidoc/unipdf`, which also offers PDF editing and generation features but requires a commercial license for business use.

Conclusion

With just a few steps, you can read and analyze the content of a PDF file in Go using `rsc.io/pdf`. This solution is ideal for simple tasks like extracting text from PDF documents.