Solution Review: Writing to a WIKI Page

This lesson discusses the solution to the challenge given in the previous lesson.

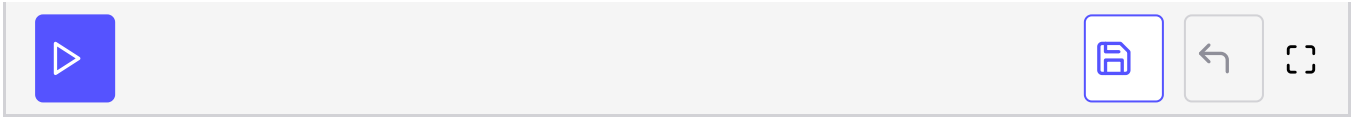
```
package main
import (
        "fmt"
        "io/ioutil"
)

type Page struct {
        Title string
        Body  []byte
}

func (p *Page) save() error {
        filename := p.Title + ".txt"
        return ioutil.WriteFile(filename, p.Body, 0600)
}

func load(title string) (*Page, error) {
        filename := title + ".txt"
        body, err := ioutil.ReadFile(filename)
        if err != nil {
                return nil, err
        }
        return &Page{Title: title, Body: body}, nil
}

func main() {
        p1 := &Page{Title: "TestPage", Body: []byte("This is a sample Page.")}
        p1.save()
        p2, _ := load("TestPage")
        fmt.Println(string(p2.Body))
}
```



Writing to a File

In the code above, we define a struct of type `Page` struct at **line** 7 with two fields: `Title` and `Body`.

Now, look at the header of the `load` method at **line 17**. It takes `title`, the *filename* as a parameter (apart from the extension .txt), and returns a pointer

to a `Page` struct and an error. At **line 19**, the file is read into a variable `body` with the `ioutil.ReadFile` method. The usual error-handling is done from **line 20** to **line 22**, returning a *non-nil* `err` variable in case of an error. At **line 23**, a `Page` struct is made with `title` and `body`, and a reference to it is returned from the function together with `nil` for the `err` variable.

Look at the header of the `save` method at **line 12**. It works on a pointer to a `Page` struct as a receiver. It constructs the filename at **line 13** with the `Title` field and writes the `Body` field out to the file at **line 14**.

In the `main()` function, we first initialize a `Page` struct at **line 27** and save it to a file at **line 28**. Then, to test our code, we load it at **line 29** and print its contents out at **line 30**.

That's it for the solution. In the next lesson, you'll see how Go provides support for copying mechanisms.