

Go Web Examples Courses Easy to follow Video Courses
about Go. Now in early access!

Learn more → ×

Forms

Hey, Philipp here!

I'd like to tell you, that my platform **Go Web Examples Courses** just launched. Enjoy easy to follow video courses about web development in Go. Make sure to check out the special offer I have for early supporters.

We'll see us over there! :)

Learn more



Forms

This example will show how to simulate a contact form and parse the message into a struct.

```
// forms.go
package main

import (
    "html/template"
    "net/http"
)

type ContactDetails struct {
    Email    string
    Subject  string
```

```

    Message string
}

func main() {
    tpl := template.Must(template.ParseFiles("forms.html"))

    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        if r.Method != http.MethodPost {
            tpl.Execute(w, nil)
            return
        }

        details := ContactDetails{
            Email:    r.FormValue("email"),
            Subject: r.FormValue("subject"),
            Message: r.FormValue("message"),
        }

        // do something with details
        _ = details

        tpl.Execute(w, struct{ Success bool }{true})
    })

    http.ListenAndServe(":8080", nil)
}

```

```

<!-- forms.html -->
{{if .Success}}
    <h1>Thanks for your message!</h1>
{{else}}
    <h1>Contact</h1>
    <form method="POST">
        <label>Email:</label><br />
        <input type="text" name="email"><br />
        <label>Subject:</label><br />
        <input type="text" name="subject"><br />
        <label>Message:</label><br />
        <textarea name="message"></textarea><br />
        <input type="submit">
    </form>
{{end}}

```

```
$ go run forms.go
```

Contact

Email:

Subject:

Message:

Submit

[Legal Disclosure](#) [Privacy Statement](#)