

```

Header Header    // 请求头字段
Body io.ReadCloser // 请求体
GetBody func() (io.ReadCloser, error)
ContentLength int64
TransferEncoding []string
Close bool
Host string
// 请求报文中的一些参数, 包括表单字段等
Form url.Values
PostForm url.Values
MultipartForm *multipart.Form
Trailer Header
RemoteAddr string
RequestURI string
TLS *tls.ConnectionState
Cancel <-chan struct{}
Response *Response
ctx context.Context
}

```

Request 结构体主要用于返回 HTTP 请求的响应, 是 HTTP 处理请求中非常重要的一部分。只有正确地解析请求数据, 才能向客户端返回响应。接下来通过简单示例来测试一下。

下方是 Go 服务器端的代码, 用于解析 Request 结构体中各成员 (或者说是属性)。

代码 chapter3/request.go      Request 结构体解析返回示例的代码

```

package main

import (
    "fmt"
    "log"
    "net/http"
    "strings"
)

func request(w http.ResponseWriter, r *http.Request) {
    // 这些信息是输出到服务器端的打印信息
    fmt.Println("Request 解析")
    // HTTP 方法
    fmt.Println("method", r.Method)
    // RequestURI 是被客户端发送到服务器端的请求行中未修改的请求 URI
    fmt.Println("RequestURI:", r.RequestURI)
    // URL 类型, 下方分别列出 URL 的各成员
    fmt.Println("URL_path", r.URL.Path)
    fmt.Println("URL_RawQuery", r.URL.RawQuery)
    fmt.Println("URL_Fragment", r.URL.Fragment)
}

```