

Fiber Cheat Sheet

Express-style · Fast · Middleware · WebSockets

worksheetdownload.com



SETUP & APP

```
go get github.com/gofiber/fiber/v2
app := fiber.New(fiber.Config{
  Prefork: true,           //
  multi-process
  BodyLimit: 4 * 1024 * 1024, // 4MB
})
app.Listen(":3000")
app.ListenTLS(":443", "cert.pem",
  "key.pem")
```

ROUTING

```
app.Get("/users", listUsers)
app.Post("/users", createUser)
app.Put("/users/:id", updateUser)
app.Delete("/users/:id", deleteUser)
app.All("/debug", handler) // all
methods
app.Use(handler) // matches all paths
```

CTX METHODS

```
c.Params("id")           // route param
c.Query("page", "1")     // query string
with default
c.Body()                  // raw request
body
c.BodyParser(&req)        // unmarshal
body to struct
return c.JSON(data)       // send JSON
return c.Status(201).JSON(data)
return c.SendString("text")
return c.SendFile("./file.pdf")
```

MIDDLEWARE

```
import
'github.com/gofiber/fiber/v2/middleware/logger'
app.Use(logger.New())
app.Use(recover.New())
app.Use(cors.New(cors.Config{AllowOrigins:
  ""}))
// Custom:
app.Use(func(c *fiber.Ctx) error {
  c.Locals("requestID", uuid.New())
  return c.Next()
})
```

GROUPING

```
api := app.Group("/api")
v1 := api.Group("/v1", JWTMiddleware())
v1.Get("/users", listUsers)
admin := v1.Group("/admin", AdminOnly())
admin.Get("/stats", getStats)
Group middleware applies to all
routes in group
```

STATIC FILES

```
app.Static("/", "./public")
app.Static("/public", "./public",
  fiber.Static{
    Compress: true,
    CacheDuration: 10 * time.Second,
    MaxAge: 3600,
  })
// SPA fallback:
app.Static("/*", "./public/index.html")
```

BODY PARSING

```
type User struct {
  Name string `json:"name"
  validate:"required"`
}
var u User
if err := c.BodyParser(&u); err != nil {
  return
  c.Status(400).JSON(fiber.Map{"error":
    err.Error()})
}
// File upload:
file, _ := c.FormFile("document")
c.SaveFile(file,
  "uploads/"+file.Filename)
```

COMMON CTX METHODS

```
c.Locals("key", val) // request-scoped
storage
c.Locals("key")       // retrieve
c.IP()                // client IP
c.Get("Authorization") // request header
c.Set("X-Custom", val) // response
header
c.Method()            // GET POST etc
c.Path()              // URL path
```

COMMON GOTCHAS

Fiber uses fasthttp net/http
handlers incompatible
c.Body() returns []byte, not
io.Reader save before parsing
Prefork mode: goroutines/channels
don't work across processes
c.Locals() cleared after request
don't use as global cache
JSON tags required on structs
unexported fields ignored