Chi Cheat Sheet

Lightweight · Middleware · Router · RESTful

SETUP & ROUTER

```
go get github.com/go-chi/chi/v5
r := chi.NewRouter()
r.Use(middleware.Logger)
r.Use(middleware.Recoverer)
http.ListenAndServe(":3000", r)
```

ROUTING

```
r.Get("/users", listUsers)
r.Post("/users", createUser)
r.Put("/users/{id}", updateUser)
r.Delete("/users/{id}", deleteUser)
r.Patch  r.Head  r.Options
r.HandleFunc("/path", handler)  //
net/http compat
```

MIDDLEWARE

```
r.Use(middleware.RequestID)
r.Use(middleware.RealIP)
r.Use(middleware.Logger)
r.Use(middleware.Recoverer)
r.Use(middleware.Compress(5))
r.Use(middleware.Timeout(60 *
time.Second))
r.Use(middleware.Throttle(100))
```

ROUTE GROUPS & MOUNTING

```
r.Route("/api", func(r chi.Router) {
r.Use(JWTMiddleware)
r.Get("/users", listUsers)
r.Post("/users", createUser)
})
// Mount sub-router:
r.Mount("/admin", adminRouter())
```

URL PARAMS

```
func getUser(w http.ResponseWriter, r
*http.Request) {
id := chi.URLParam(r, "id")
// or from context:
id :=
r.Context().Value(chi.RouteCtxKey).(*chi.Context).URLParam(
w.Header().Set("Content-Type",
"application/json")
json.NewEncoder(w).Encode(user)
}
```

SUB-ROUTERS

```
func usersRouter() http.Handler {
r := chi.NewRouter()
r.Use(AuthRequired)
r.Get("/", listUsers)
r.Post("/", createUser)
r.Route("/{id}", func(r chi.Router) {
r.Use(UserCtx)
r.Get("/", getUser)
r.Put("/", updateUser)
})
return r
}
```

RESTFUL PATTERN

```
r.Route("/articles/{id}", func(r
chi.Router) {
r.Use(ArticleCtx)  // load article from
DB
r.Get("/", getArticle)
r.Put("/", updateArticle)
r.Delete("/", deleteArticle)
})
// ArticleCtx middleware sets article in
context
ctx := context.WithValue(r.Context(),
articleKey, article)
```

BUILT-IN MIDDLEWARE

```
middleware.RequestID   // adds
X-Request-Id header
middleware.RealIP      // sets
RemoteAddr to X-Real-IP
middleware.Logger      // structured
request logging
middleware.Recoverer   // recover from
panics
middleware.Compress(5) // gzip (level 5)
middleware.Timeout(t)  // request
timeout
middleware.CleanPath   // normalize URL
paths
```

COMMON GOTCHAS

Chi uses {param} not :param style
Wildcard: r.Get("/files/*",
serveFiles)
Context values require typed keys to
avoid collisions
r.Mount strips path prefix from
sub-router
middleware.Logger prints to stdout
use custom for structured logs