



# Go Agent v0.1.7

배포일: 2022-01-04

## 베타 - 업데이트

### database/sql 라이브러리를 지원합니다.

database/sql 패키지의 `sql.Open` 함수 대신 `whatapsql.OpenContext` 함수를 사용합니다. `PrepareContext`, `QueryContext`, `ExecContext` 등 context를 전달하는 함수를 사용하기를 권장합니다.

전달하는 context는 `trace.Start()` 를 통해서 whatap TraceCtx 정보가 있어야합니다.

#### Install guide

```
import (
    _ "github.com/go-sql-driver/mysql"
    "github.com/whatap/go-api/instrumentation/database/sql/whatapsql"
)

func main() {
    config := make(map[string]string)
    trace.Init(config)
    defer trace.Shutdown()

    // whataphttp.Func 내부에서 whatap TraceCtx를 생성합니다.
    http.HandleFunc("/query", whataphttp.Func(func(w http.ResponseWriter, r *http.Request) {
        ctx := r.Context()

        // Use WhaTap Driver. whatap의 TraceCtx가 있는 context 를 전달합니다.
        db, err := whatapsql.OpenContext(ctx, "mysql", dataSource)
        if err != nil {
            fmt.Println("Error whatapsql.Open ", err)
            return
        }
        defer db.Close()

        ...
        query := "select id, subject from tbl_faq limit 10"

        // whatap TraceCtx가 있는 context 를 전달합니다.
        if rows, err := db.QueryContext(ctx, query); err == nil {
```

```
...
}
}
...
}
```

[참조 예제](#)

## 설정

```
go.sql_profile_enabled
```

```
#Default : true#
```

```
#Type : boolean#
```

database/sql 정보 수집여부를 설정합니다.

## github.com/go-gonic/gin 라이브러리 지원

Install guide

```
import (
    "github.com/go-gonic/gin"

    "github.com/whatap/go-api/trace"
    "github.com/whatap/go-api/instrumentation/github.com/go-gonic/gin/whatapgin"
)

func main() {
    config := make(map[string]string)
    trace.Init(config)
    defer trace.Shutdown()

    r := gin.Default()

    // Set the whatap
    r.Use(whatapgin.Middleware())

    r.GET("/", func(c *gin.Context) {
        c.JSON(http.StatusOK, gin.H{
            "message": "ok",
        })
    })
}
```

[참조 예제](#)

## github.com/gorilla/mux 라이브러리 지원

### Install guide

```
import (  
    "github.com/gorilla/mux"  
  
    "github.com/whatap/go-api/trace"  
    "github.com/whatap/go-api/instrumentation/github.com/gorilla/mux/whatapmux"  
)  
  
func main() {  
    config := make(map[string]string)  
    trace.Init(config)  
    defer trace.Shutdown()  
  
    r := mux.NewRouter()  
  
    // Set the whatap  
    r.Use(whatapmux.Middleware())  
  
    r.HandleFunc("/index", func(w http.ResponseWriter, r *http.Request) {  
        w.Header().Add("Content-Type", "text/html")  
        reply := "/index <br/>Test Body"  
        _, _ = w.Write([]byte(reply))  
        fmt.Println("Response -", r.Response)  
    })  
}
```

### [참조 예제](#)

## github.com/labstack/echo 라이브러리 지원

### Install guide

```
import (  
    "github.com/labstack/echo"  
  
    "github.com/whatap/go-api/trace"  
    "github.com/whatap/go-api/instrumentation/github.com/labstack/echo/whatapecho"  
)  
  
func main() {  
    config := make(map[string]string)  
    trace.Init(config)  
    defer trace.Shutdown()  
}
```

```
...  
  
e := echo.New()  
  
// Set the whatap  
e.Use(whatapecho.Middleware())  
  
e.GET("/", func(c echo.Context) error {  
    return c.String(http.StatusOK, "Hello, World!\n")  
})  
}
```

[참조 예제](#)