# OClock

Let create a simple time service

Create new folder for oclock project.

```
mkdir -p $GOPATH/src/oclock
```

Open editor

```
atom $GOPATH/src/oclock
```

## Test

Lets start with a test.

create main_test.go

```
package main

import (
    "net/http/httptest"
    "testing"
)

func TestOClock(t *testing.T) {
    req := httptest.NewRequest("GET", "http://example.com", nil)
    w := httptest.NewRecorder()
    OClock(w, req)

    if w.Code != 200 {
        t.Errorf("Expected 200 Go %d", w.Code)
    }

}
```

Run tests

```
go test -v oclock
```

```
# oclock
../go/src/oclock/main_test.go:11: undefined: OClock
FAIL    oclock [build failed]
```

# Implement

create oclock/main.go

```
package main

import "net/http"

//OClock Gives you the time of day
func OClock(w http.ResponseWriter, r *http.Request) {
}
```

```
go test -v oclock
```

```
=== RUN   TestOClock
--- PASS: TestOClock (0.00s)
PASS
ok      oclock  0.011s
```

# Test

```
package main

import (
    "net/http/httptest"
    "testing"
)

func TestOClock_Status(t *testing.T) {
    req := httptest.NewRequest("GET", "http://example.com", nil)
    w := httptest.NewRecorder()
    OClock(w, req)

    if w.Code != 200 {
        t.Errorf("Expected 200 Result was %d", w.Code)
    }

}

func TestOClock_Body(t *testing.T) {
    req := httptest.NewRequest("GET", "http://example.com", nil)
    w := httptest.NewRecorder()
    OClock(w, req)
    expected := time.Now().Format("2006-01-02T15:04")
    result := w.Body.String()
    if result != expected {
        t.Errorf("Expected %q Result %q", expected, result)
    }

}
```

```
=== RUN   TestOClock
--- PASS: TestOClock (0.00s)
=== RUN   TestOClock_Body
--- FAIL: TestOClock_Body (0.00s)
    main_test.go:27: Expected "2017-02-15T20:27" Result ""
FAIL
exit status 1
FAIL    oclock  0.011s
```

# Implement

modify oclock/main.go

```
package main

import "net/http"

//OClock Gives you the time of day
func OClock(w http.ResponseWriter, r *http.Request) {
    w.Write([]byte(time.Now().Format("2006-01-02T15:04")))
}
```

# Test

```
go test -v oclock
```

```
=== RUN   TestOClock
--- PASS: TestOClock (0.00s)
=== RUN   TestOClock_Body
--- PASS: TestOClock_Body (0.00s)
PASS
ok      oclock  0.010s
```

# Implements

No that we have a handler lets add it to a server

```
package main

import (
    "log"
    "net/http"
    "time"
)

//OClock Gives you the time of day
func OClock(w http.ResponseWriter, r *http.Request) {
    w.Write([]byte(time.Now().Format("2006-01-02T15:04")))
}

func main() {
    http.HandleFunc("/oclock", OClock)
    log.Fatal(http.ListenAndServe(":8080", nil))
}
```

# Install

```
go install oclock
```

# Run

```
oclock
```

http://localhost:8080/oclock

Ship It!