OClock - Part Three

Let extend the clock to return time from NTP

Add Dependancy

NTP library

go get github.com/beevik/ntp

Code Change

```

//OClock Gives you the time of day
func OClock(w http.ResponseWriter, r *http.Request) {
    ntpTime, err := ntp.Time("pool.ntp.org")

    if err != nil {
        fmt.Printf("Can't get NTP time - %v", err.Error())
        w.WriteHeader(http.StatusInternalServerError)
    }

    c := clock{Time: jsonTime(ntpTime)}

    cJSON, err := json.Marshal(c)
    if err != nil {
        fmt.Printf("Can't marshal time - %v", err.Error())
        w.WriteHeader(http.StatusInternalServerError)
    }

    w.Write([]byte(cJSON))
}
```

Multiple NTP Servers

```
var chann = make(chan clock)

func ntpWorker(host string) {
    t, err := ntp.Time(host)
    if err != nil {
        fmt.Printf("Can't get NTP time %v - %v", host, err.Error())
        return
    }
    // write to a channel
    chann <- clock{Server: host, Time: jsonTime(t)}
}

//OClock Gives you the time of day
func OClock(w http.ResponseWriter, r *http.Request) {
    go ntpWorker("europe.pool.ntp.org")
    go ntpWorker("africa.pool.ntp.org")
    go ntpWorker("north-america.pool.ntp.org")

    // read from a channel
    c := <-chann

    cJSON, err := json.Marshal(c)
    if err != nil {
        fmt.Printf("Can't marshal time - %v", err.Error())
        w.WriteHeader(http.StatusInternalServerError)
    }

    w.Write([]byte(cJSON))
}
```