

Overview of Server

News

1. Economists such as Dale Jorgenson of Harvard University, who specialise in measuring economic productivity, report that in recent years the only increase in total-factor productivity in the US economy has been in the information technology-producing industries.
2. Recruiting: Big four auditing firm PricewaterhouseCoopers (PwC) is the top recruiter for blockchain-related jobs on recruitment platform Indeed, search results showed at press time on March 30. PwC is responsible for 40 blockchain-related job offers on the platform. Other big four auditing firms are apparently recruiting professionals in this niche on the platform: more precisely, Ernst & Young posted 17 such announcements, while Deloitte posted 10.
3. Argentina's Central Bank states "as blockchain based technology develops the need for USD as a central currency will diminish."

Imposter Syndrome

The syndrome.

How to Launder Money.

From: https://www.washingtonpost.com/outlook/trumps-businesses-are-full-of-dirty-russian-money-the-scandal-is-thats-legal/2019/03/29/11b812da-5171-11e9-88a1-ed346f0ec94f_story.html

"According to Jonathan Winer, who served as deputy assistant secretary of state for international law enforcement in the Clinton administration, ... "If you are doing a transaction with no mortgage, there is no financial institution that needs to know where the money came from, particularly if it's a wire transfer from overseas," ... "The customer obligations that are imposed on all kinds of financial institutions are not imposed on people selling real estate. ..."

"And without such regulations, prosecutors' hands are tied.

"All of which made it easier for the Russian Mafia to expand throughout the United States."

Server

```

1 package main
2
3 import (
4     "context"
5     "crypto/tls"
6     "database/sql"
7     "flag"
8     "fmt"
9     "log"
10    "net/http"
11    "os"
12    "os/signal"
13    "strings"
14    "sync"
15    "time"
16
17    "github.com/Univ-Wyo-Education/S19-4010/l/23/ssvr/ReadConfig"
18    "github.com/pschlump/MiscLib"
19    "github.com/pschlump/filelib"
20    "github.com/pschlump/godebug"
21 )
22
23 // -----
24 // Notes:
25 //   Graceful Shutdown: https://stackoverflow.com/questions/39320025/how-to-stop-ht
26 // -----
27
28 var Cfg = flag.String("cfg", "cfg.json", "config file for this call")
29 var HostPort = flag.String("hostport", ":9004", "Host/Port to listen on")
30 var DbFlag = flag.String("db_flag", "", "Additional Debug Flags")
31 var TLS_crt = flag.String("tls_crt", "", "TLS Signed Publick Key")
32 var TLS_key = flag.String("tls_key", "", "TLS Signed Private Key")
33
34 type GlobalConfigData struct {
35     TickerSeconds int `json:"ticker_seconds" default:"30" // Time Ticker Seconds
36
37     DBSqlite      string `json:"sqlite_name" default:"mydb.db"`
38     LogFileName   string `json:"log_file_name"`
39
40     // debug flags:
41     //      GetVal          Show fetching of values from GET/POST/.
42     //      HandleCRUD      General CRUD
43     //      HandleCRUD.GenWhere  CRUD generating "where" clause
44     //      HandleCRUDSP     CRUD calling stored procedures
45     //      IsAuthKeyValid   Check of 'auth_key'
46     //      dump-db-flag     Dump flags turned on.
47     DebugFlag string `json:"db_flag"`
48
49     AuthKey string `json:"auth_key" default:"" // Auth key by default is turned of
50

```

```

51 // Default file for TLS setup (Shoud include path), both must be specified.
52 // These can be over ridden on the command line.
53 TLS_crt string `json:"tls_crt" default:""``
54 TLS_key string `json:"tls_key" default:""``
55
56 // Path where files are temporary uploaded to
57 UploadPath string `json:"upload_path" default:"./files"``
58
59 TemplateDir string `default:"./tpl"``
60
61 StaticPath string `json:"static_path" default:"www"``
62 }
63
64 var gCfg GlobalConfigData
65 var logFilePtr *os.File
66 var logFileName = ""
67 var DB *sql.DB
68 var db_flag map[string]bool
69 var wg sync.WaitGroup
70 var httpServer *http.Server
71 var logger *log.Logger
72 var shutdownWaitTime = time.Duration(1)
73 var isTLS bool
74
75 func init() {
76     isTLS = false
77     db_flag = make(map[string]bool)
78     logger = log.New(os.Stdout, "", 0)
79 }
80
81 func main() {
82
83     flag.Parse() // Parse CLI arguments to this, --cfg <name>.json
84
85     fns := flag.Args()
86     if len(fns) != 0 {
87         fmt.Printf("Extra arguments are not supported [%s]\n", fns)
88         os.Exit(1)
89     }
90
91     if Cfg == nil {
92         fmt.Printf("--cfg is a required parameter\n")
93         os.Exit(1)
94     }
95
96     // -----
97     // Read in Configuraiton
98     // -----
99     err := ReadConfig.ReadFile(*Cfg, &gCfg)
100     if err != nil {
101         fmt.Fprintf(os.Stderr, "Unable to read configuration: %s error %s\n", *Cfg,

```

```
102         os.Exit(1)
103     }
104
105     // -----
106     // Logging File
107     // -----
108     if gCfg.LogFileName != "" {
109         LogFile(gCfg.LogFileName)
110     }
111
112     // -----
113     // TLS parameter check / setup
114     // -----
115     if *TLS_crt == "" && gCfg.TLS_crt != "" {
116         TLS_crt = &gCfg.TLS_crt
117     }
118     if *TLS_key == "" && gCfg.TLS_key != "" {
119         TLS_key = &gCfg.TLS_key
120     }
121
122     if *TLS_crt != "" && *TLS_key == "" {
123         log.Fatalf("Must supply both .crt and .key for TLS to be turned on - fatal")
124     } else if *TLS_crt == "" && *TLS_key != "" {
125         log.Fatalf("Must supply both .crt and .key for TLS to be turned on - fatal")
126     } else if *TLS_crt != "" && *TLS_key != "" {
127         if !filelib.Exists(*TLS_crt) {
128             log.Fatalf("Missing file ->%s<-\\n", *TLS_crt)
129         }
130         if !filelib.Exists(*TLS_key) {
131             log.Fatalf("Missing file ->%s<-\\n", *TLS_key)
132         }
133         isTLS = true
134     }
135
136     // -----
137     // Debug Flag Processing
138     // -----
139     if gCfg.DebugFlag != "" {
140         ss := strings.Split(gCfg.DebugFlag, ",")
141         // fmt.Printf("gCfg.DebugFlag ->%s<-\\n", gCfg.DebugFlag)
142         for _, sx := range ss {
143             // fmt.Printf("Setting ->%s<-\\n", sx)
144             db_flag[sx] = true
145         }
146     }
147     if *DbFlag != "" {
148         ss := strings.Split(*DbFlag, ",")
149         // fmt.Printf("gCfg.DebugFlag ->%s<-\\n", gCfg.DebugFlag)
150         for _, sx := range ss {
151             // fmt.Printf("Setting ->%s<-\\n", sx)
152             db_flag[sx] = true
```

```

153     }
154 }
155 if db_flag["dump-db-flag"] {
156     fmt.Fprintf(os.Stderr, "%sDB Flags Enabled Are:%s\n", MiscLib.ColorGreen, M
157     for x := range db_flag {
158         fmt.Fprintf(os.Stderr, "%s\t%s%s\n", MiscLib.ColorGreen, x, MiscLib.Col
159     }
160 }
161
162 fmt.Fprintf(os.Stderr, "%sAT %s, %s\n", MiscLib.ColorGreen, godebug.LF(), MiscL
163
164 // -----
165 // Connect to Database
166 // -----
167 ConnectToSqlite()
168 CreateTables()
169
170 fmt.Fprintf(os.Stderr, "%sConnected to Sqlite%s\n", MiscLib.ColorGreen, MiscLib
171
172 // -----
173 // Setup to use the ledger and send stuff to Eth.
174 // -----
175 SetupGeth()
176
177 // -----
178 // Setup HTTP End Points
179 // -----
180 mux := http.NewServeMux()
181 mux.Handle("/api/v1/status", http.HandlerFunc(HandleStatus)) //
182 mux.Handle("/status", http.HandlerFunc(HandleStatus)) //
183 mux.Handle("/api/v1/exit-server", http.HandlerFunc(HandleExitServer)) //
184 mux.Handle("/login", http.HandlerFunc(HandleLogin)) // //
185 mux.HandleFunc("/upload", UploadFileClosure(gCfg.UploadPath)) // URL to
186
187 // For the list of end-points (URI Paths) see ./handle.go
188 HandleTables(mux)
189
190 mux.Handle("/", http.FileServer(http.Dir(gCfg.StaticPath)))
191
192 // -----
193 // Setup signal capture
194 // -----
195 stop := make(chan os.Signal, 1)
196 signal.Notify(stop, os.Interrupt)
197
198 // -----
199 // Setup / Run the HTTP Server.
200 // -----
201 if isTLS {
202     cfg := &tls.Config{
203         MinVersion:          tls.VersionTLS12,

```

```

204     CurvePreferences:      []tls.CurveID{tls.CurveP521, tls.CurveP384, t
205     PreferServerCipherSuites: true,
206     CipherSuites: []uint16{
207         tls.TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
208         tls.TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
209         tls.TLS_RSA_WITH_AES_256_GCM_SHA384,
210         tls.TLS_RSA_WITH_AES_256_CBC_SHA,
211     },
212 }
213 // loggingHandler := NewApacheLoggingHandler(mux, os.Stderr)
214 loggingHandler := NewApacheLoggingHandler(mux, logFilePtr)
215 httpServer = &http.Server{
216     Addr:      *HostPort,
217     Handler: loggingHandler,
218     // Handler: mux,
219     TLSConfig:  cfg,
220     TLSNextProto: make(map[string]func(*http.Server, *tls.Conn, http.Handle
221 }
222 } else {
223     // loggingHandler := NewApacheLoggingHandler(mux, os.Stderr)
224     loggingHandler := NewApacheLoggingHandler(mux, logFilePtr)
225     httpServer = &http.Server{
226         Addr:      *HostPort,
227         Handler: loggingHandler,
228         // Handler: mux,
229     }
230 }
231
232 go func() {
233     wg.Add(1)
234     defer wg.Done()
235     if isTLS {
236         fmt.Fprintf(os.Stderr, "%sListening on https://%s%s\n", MiscLib.ColorGr
237         if err := httpServer.ListenAndServeTLS(*TLS_crt, *TLS_key); err != nil
238             logger.Fatal(err)
239     }
240 } else {
241     fmt.Fprintf(os.Stderr, "%sListening on http://%s%s\n", MiscLib.ColorGre
242     if err := httpServer.ListenAndServe(); err != nil {
243         logger.Fatal(err)
244     }
245 }
246 }()
247
248 // -----
249 // Catch signals from [Contro-C]
250 // -----
251 select {
252 case <-stop:
253     fmt.Fprintf(os.Stderr, "\nShutting down the server... Received OS Signal...
254     ctx, cancel := context.WithTimeout(context.Background(), shutdownWaitTime*t

```

```
255     defer cancel()
256     err := httpServer.Shutdown(ctx)
257     if err != nil {
258         fmt.Printf("Error on shutdown: [%s]\n", err)
259     }
260 }
261
262 // -----
263 // Wait for HTTP server to exit.
264 // -----
265 wg.Wait()
266 }
267
268 func SetupGeth() {
269     // TODO - xyzzy - some work to do.
270 }
```

Partials

Single Page App

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <!-- Required meta tags -->
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fi
7
8     <!-- Bootstrap CSS -->
9     <link rel="stylesheet" href="/bootstrap-4.1.3/dist/css/bootstrap.css">
10    <link rel="stylesheet" href="/css/bootstrap.curulean-theme.min.css">
11    <link rel="stylesheet" href="/css/bootstrap-datepicker.min.css">
12    <script src="https://cdn.jsdelivr.net/npm/signature_pad@2.3.2/dist/signature_pad
13
14    <title>PVP Documet Tracking</title>
15 <style>
16 .some-style {
17
18 }
19 </style>
20 </head>
21 <body>
22
23 <nav class="navbar navbar-expand-md navbar-dark bg-dark">
24     <div class="navbar-collapse collapse w-100 order-1 order-md-0 dual-collapse2">
25         <ul class="navbar-nav mr-auto">
```

```

26     <li class="nav-item show-logged-in">
27         <a class="nav-link" href="#" id="form00-render">New Document</a>
28     </li>
29     <li class="nav-item show-logged-in">
30         <a class="nav-link" href="#" id="form01-render">List Documetns</a>
31     </li>
32     <li class="nav-item show-anon">
33         <a class="nav-link" href="#" id="getStatus">Status</a>
34     </li>
35 </ul>
36 </div>
37 <div class="mx-auto order-0">
38     <a class="navbar-brand mx-auto" href="#">Document Signing</a>
39     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar-collapse">
40         <span class="navbar-toggler-icon"></span>
41     </button>
42 </div>
43 <div class="navbar-collapse collapse w-100 order-3 dual-collapse2">
44     <ul class="navbar-nav ml-auto">
45         <li class="nav-item">
46             <a class="nav-link" href="#" id="login">Login</a>
47         </li>
48     </ul>
49 </div>
50 </nav>
51
52 <div class="top-of-page">
53
54     <h1 id="speical-title">&nbsp;</h1>
55
56     <div class="content container" id="msg"></div>
57     <div class="content container" id="body"></div>
58
59 </div>
60
61 <script src="/js/jquery-3.3.1.js"></script>
62 <script src="/js/popper-1.14.7.js"></script>
63 <script src="/js/bootstrap.js"></script>
64 <script src="/js/bootstrap-datepicker.min.js"></script>
65
66 <script src="/js/cfg.js?_ran_=00004"></script>
67
68 <script src="/js/doc-index.js?_ran_=00004"></script>
69 <script src="/js/doc-form25.js?_ran_=0004"></script> <!-- login form -->
70 <script src="/js/doc-form00.js?_ran_=0004"></script> <!-- document upload sign -->
71 <script src="/js/doc-form01.js?_ran_=0004"></script> <!-- list documents -->
72 <script src="/js/doc-form09.js?_ran_=0004"></script> <!-- render welcome page -->
73
74 <script>
75 renderForm09(null);
76 $("#form00-render").click(renderForm00); // Attach to link to paint the partial

```



```
77 $("#form01-render").click(renderForm01);    // Attach to link to paint the partial
78 </script>
79
80 </body>
81 </html>
```

Render a Partial

```
1 // Login Form
2
3 function submitForm25 ( event ) {
4     event.preventDefault(); // Totally stop stuff happening
5
6     console.log ( "Click of submit button for form25 - login" );
7
8     var data = {
9         "username"      : $("#username").val()
10        , "password"     : $("#password").val()
11        , "__method__"   : "POST"
12        , "_ran_"       : ( Math.random() * 100000000 ) % 100000000
13    };
14    submitItData ( event, data, "/login", function(data){
15        console.log ( "data=", data );
16        if ( data && data.status && data.status == "success" ) {
17            user_id = data.user_id; // sample: -- see bottom of file: www/js/pdoc-fc
18            auth_token = data.auth_token;
19            LoggInDone ( auth_token );
20            $(".show-anon").hide();
21            $(".show-logged-in").show();
22            renderMessage ( "Successful Login", "You are now logged in<br>");
23        } else {
24            console.log ( "ERROR: ", data );
25            renderError ( "Failed to Login", data.msg );
26        }
27    }, function(data) {
28        console.log ( "ERROR: ", data );
29        renderError ( "Failed to Login - Network communication failed.", "Failed to
30    }
31    );
32 }
33
34 function renderForm25 ( event ) {
35     var form = [ ''
36         , '<div>'
37         , '<div class="row">'
38         , '<div class="col-sm-6">'
```

```
39         , '<div class="card bg-default">'
40         , '<div class="card-header"><h2>Login</h2></div>'
41         , '<div class="card-body">'
42         , '<form id="form01">'
43             , '<input name="app"                                t
44             , '<input name="auth_key"                          t
45             , '<div class="form-group">'
46                 , '<label for="username">Email</label>'
47                 , '<input type="text" class="form-control" id="us
48             , '</div>'
49             , '<div class="form-group">'
50                 , '<label for="password">Password</label>'
51                 , '<input type="password" class="form-control" id
52             , '</div>'
53             , '<button type="button" class="btn btn-primary" id="
54         , '</form>'
55         , '</div>'
56     , '</div>'
57 , '</div>'
58 , '</div>'
59 , '</div>'
60 ].join("\n");
61 $("#body").html(form);
62 // Add events
63 $("#form25-submit").click(submitForm25);
64 // xyzzy - additional click events forgot-pass, forgot-acct
65 }
66 $("#form25-render").click(renderForm25);    // Attach to link to paint the partial
```