

# Package server

```
import "PGL/Client/server"
```

[Overview](#)[Index](#)

## Overview ▾

Package server provides the handles and mux to run the server

## Index ▾

### Variables

```
func LoggedIn(r *http.Request) bool
func Run()
func addinv(w http.ResponseWriter, r *http.Request)
func additem(w http.ResponseWriter, r *http.Request)
func checkHash(hashpwd, pw string) bool
func checkUser(username, password string) bool
func createSession(w http.ResponseWriter, r *http.Request, username string)
func edititem(w http.ResponseWriter, r *http.Request)
func editprofile(w http.ResponseWriter, r *http.Request)
func getUserID(r *http.Request) uint32
func getUsername(r *http.Request) string
func hash(password string) string
func home(w http.ResponseWriter, r *http.Request)
func index(w http.ResponseWriter, r *http.Request)
func inputCheck(args ...string) bool
func invs(w http.ResponseWriter, r *http.Request)
func isNum(args ...string) bool
func items(w http.ResponseWriter, r *http.Request)
func login(w http.ResponseWriter, r *http.Request)
func logout(w http.ResponseWriter, r *http.Request)
func profile(w http.ResponseWriter, r *http.Request)
func register(w http.ResponseWriter, r *http.Request)
func removeSession(w http.ResponseWriter, r *http.Request)
func routes(r *mux.Router)
func updateLogin(mode string, user models.User)
type SessionInfo
```

## Package files

[errors.go](#) [invhandles.go](#) [itemhandles.go](#) [routes.go](#) [session.go](#) [setup.go](#) [userhandles.go](#) [validation.go](#)

## Variables

variables for port and tpl pointer

```
var (  
    port = ":8081"  
    tpl  *template.Template  
)
```

error variables

```
var (  
    errChars = errors.New("unallowed characters")  
)
```

custom funcs for the templates

```
var functionMap = template.FuncMap{  
    "toDate": func(unix uint64) string {  
        if unix == 0 {  
            return "Nil"  
        }  
        return time.Unix(int64(unix), 0).Format("02/01/2006")  
    },  
    "toDay": func(unix uint64) string {  
        if unix == 0 {  
            return ""  
        }  
        t1 := time.Now()  
        t2 := time.Unix(int64(unix), 0)  
        y, m, d := t2.Date()  
        u2 := time.Date(y, m, d, 0, 0, 0, 0, time.UTC)  
        y, m, d = t1.In(t2.Location()).Date()  
        u1 := time.Date(y, m, d, 0, 0, 0, 0, time.UTC)  
        days := u2.Sub(u1) / (24 * time.Hour)  
        ret := strconv.Itoa(int(days))  
        if int(days) < 0 {  
            return "Expired"  
        } else {  
            return ret + " DaysToGo"  
        }  
    },  
    "toDate2": func(unix uint64) string {  
        if unix == 0 {  
            return "Nil"  
        }  
        return time.Unix(int64(unix), 0).Format("2006-01-02")  
    },  
}
```

session map

```
var mapSessions = map[string]SessionInfo{}
```

sync variables for the package

```
var (  
    //mutex sync.Mutex  
    wg sync.WaitGroup  
)
```

## func LoggedIn

```
func LoggedIn(r *http.Request) bool
```

function to check whether a user is logged in by requesting and checking the details from the cookies against the session map

## func Run

```
func Run()
```

runs the server

## func addinv

```
func addinv(w http.ResponseWriter, r *http.Request)
```

inv handle to add inv

## func additem

```
func additem(w http.ResponseWriter, r *http.Request)
```

handle to add the item

## func checkHash

```
func checkHash(hashedpw, pw string) bool
```

function to check the hashed password against the input

## func checkUser

```
func checkUser(username, password string) bool
```

function to check whether a user credentials are correct

## func createSession

```
func createSession(w http.ResponseWriter, r *http.Request, username string)
```

function to create a new session

## func edititem

```
func edititem(w http.ResponseWriter, r *http.Request)
```

handle to edit the item

## func editprofile

```
func editprofile(w http.ResponseWriter, r *http.Request)
```

handle for editing user profile

## func getUserID

```
func getUserID(r *http.Request) uint32
```

function to get userid

## func getUsername

```
func getUsername(r *http.Request) string
```

function to get the username from the mapped session

## func hash

```
func hash(password string) string
```

function to hash password

## func home

```
func home(w http.ResponseWriter, r *http.Request)
```

handle for userhome

## func index

```
func index(w http.ResponseWriter, r *http.Request)
```

index

## func inputCheck

```
func inputCheck(args ...string) bool
```

checks whether input is allowed based on regexp expression sanitization for inputs up till emails, prevents js scripts

## func invs

```
func invs(w http.ResponseWriter, r *http.Request)
```

inv handle to display inv

## func isNum

```
func isNum(args ...string) bool
```

checks whether input is alphanumeric based on regexp expression sanitization for inputs

## func items

```
func items(w http.ResponseWriter, r *http.Request)
```

handle to show the items

## func login

```
func login(w http.ResponseWriter, r *http.Request)
```

handle to log the user in and update the prevlogin and currlogin times

## func logout

```
func logout(w http.ResponseWriter, r *http.Request)
```

handle to log the user out

## func profile

```
func profile(w http.ResponseWriter, r *http.Request)
```

handle for user profile

## func register

```
func register(w http.ResponseWriter, r *http.Request)
```

handle to register the user

## func removeSession

```
func removeSession(w http.ResponseWriter, r *http.Request)
```

function to remove an existing session

## func routes

```
func routes(r *mux.Router)
```

routes

## func updateLogin

```
func updateLogin(mode string, user models.User)
```

handle to update the login timings

## type SessionInfo

session struct

```
type SessionInfo struct {  
    UserID    uint32  
    Username string  
}
```

Build version go1.15.7.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)