

## Package user

```
import "goInAction2Assignment/user"
```

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

### Overview ▾

Package user contains all the registered user for the application and login logout feature.

### Index ▾

Variables  
`func AlreadyLoggedIn(req *http.Request) bool`  
`func GenerateUser()`  
`func Login(res http.ResponseWriter, req *http.Request)`  
`func Logout(res http.ResponseWriter, req *http.Request)`  
`func Signup(res http.ResponseWriter, req *http.Request)`  
`func init()`  
`func multiLogin(username string) bool`  
type User  
`func GetUser(res http.ResponseWriter, req *http.Request) User`

### Package files

[loginlogout.go](#)

### Variables

```
var (
    MapUsers      = map[string]User{} //mapUsers uses username as a key and map with all its related information in user
    struct.
    MapSessions = map[string]string{} //mapSessions map registered user to a particular cookie value.
    tpl          *template.Template

    //Unique policy creation for the life of the program.
    Policy = bluemonday.UGCPolicy()
    // use a single instance of Validate
    validate *validator.Validate
)
```

### func AlreadyLoggedIn

```
func AlreadyLoggedIn(req *http.Request) bool
```

AlreadyLoggedIn function enabled quick check of if a particular session is checked in.

### func GenerateUser

```
func GenerateUser()
```

GenerateUser initiate preliminary user list and brcrypt their password in a secure manner.

### func Login

```
func Login(res http.ResponseWriter, req *http.Request)
```

Login function allow registered user to login to the system. This function will also check for the existence of multiple login. Concurrent/multiple login is not allowed for this apps.

### func Logout

```
func Logout(res http.ResponseWriter, req *http.Request)
```

Logout function is able to access by user in all the pages. Once registered user is log out, the current session cookies will be cleared.

## func Signup

```
func Signup(res http.ResponseWriter, req *http.Request)
```

Signup function enabled all the unregistered user to signup before they are able to access registered user feature.

## func init

```
func init()
```

## func multiLogin

```
func multiLogin(username string) bool
```

multiLogin is to check through all the sessions to see if particular user login twice

## type User

User struct is a type that stored all registered user information.

```
type User struct {  
    FirstName string  
    LastName  string  
    UserName  string  
    Password  []byte  
}
```

## func GetUser

```
func GetUser(res http.ResponseWriter, req *http.Request) User
```

GetUser function can helped to check if a particular session is map to any registered user. If an cookie value is able to map to registered user, it will return it. Else, the return would be empty.

Build version go1.16.

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](#)