

## Package booking

```
import "goInAction2Assignment/booking"
```

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

### Overview ▾

Package booking stores all the booking variables and functions. It includes function to initialise preloaded bookings, performing venue booking search, and lastly generate booking record

### Index ▾

Variables  
`func BookVenue(res http.ResponseWriter, req *http.Request)`  
`func GenerateBookings() error`  
`func init()`  
type booking  
type bookingList  
    `func (b *bookingList) BrowseBooking(res http.ResponseWriter, req *http.Request)`  
    `func (b *bookingList) RemoveBooking(res http.ResponseWriter, req *http.Request)`  
    `func (b *bookingList) addBooking(option int, userName string, bookings []int, duration int, size int, bookingError chan error) error`

### Package files

[booking.go](#) [browseDeleteBooking.go](#)

### Variables

```
var (
    MainBookingList = &bookingList{nil, 0} //MainBookingList store all bookings in linked list.
    bookingCount    int64                  //bookingCount will be treated as booking ID. It is locked inside mutex so
    that there will be access once at a time.
    mu              sync.Mutex              //mu is used to lock the booking process where only one booking can be done at
    a time.
    tpl             *template.Template
    wg              sync.WaitGroup //wg is used to wait the go routine that being launched to preloaded some bookings.
    err             error
    mapBookingID    = map[string][]int{} //mapBookingID store the searchCriteria result from the booking query for every
    user.
)
```

### func BookVenue

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

BookVenue function will first perform a search based on the booking input by user. After that, all available venue that matches the criteria will be listed out. Registered user can select the venue and hit the submit button to book it.

### func GenerateBookings

```
func GenerateBookings() error
```

GenerateBookings initiate the application with some preloaded booking list

### func init

```
func init()
```

### type booking

booking struct is a type of every booking information.

```
type booking struct {
    BookingID    int64
```

```
Venue      int
BookingSlot []int
Duration   int
participantSize int
Host       string
next       *booking
}
```

## type bookingList

bookingList struct is setup for linked list of booking.

```
type bookingList struct {
    head *booking
    size int
}
```

## func (\*bookingList) BrowseBooking

```
func (b *bookingList) BrowseBooking(res http.ResponseWriter, req *http.Request)
```

BrowseBooking function will be triggered in two scenario. 1.Registered user can choose to browse their own booking record. 2.Admin can browse the whole booking list. He/she can also filter the booking list by user name.

## func (\*bookingList) RemoveBooking

```
func (b *bookingList) RemoveBooking(res http.ResponseWriter, req *http.Request)
```

RemoveBooking function will perform remove booking feature based on booking ID. Only admin can perform remove booking.He/she can select the booking ID based on the full booking list displayed below.

## func (\*bookingList) addBooking

```
func (b *bookingList) addBooking(option int, userName string, bookings []int, duration int, size int, bookingError chan error) error
```

addBooking function traverse through booking linked list to add new booking record.

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](#)