GoDoc

Package hashcsv

```
import "goInAction2/assignment/packages/hashcsv"
```


Overview

Implements funtions to initialize and read to / write from csv files using the csv package, with additional checks and error handling against an SHA256 checksum to detect file tampering. HashCSVs are safe for concurrent use via the inclusion of a Mutex with each struct. Also includes a file to track the last saved time and date of each CSV file. Functions to update the CSV save files are called whenever a user logs out of their session.

Index

Package files

hashcsv.go

type **HashCSV**

```
type HashCSV struct {
    Name          string
    FilePath      string
    ChecksumPath  string
    LastsavedPath string
    Reader        *csv.Reader
    Writer        *csv.Writer
    // contains filtered or unexported fields
}
```

func Init

```
func Init(name string) *HashCSV
```

Init creates a new csv file as well as its associated hash, and returns a pointer to a HashCSV for that file.

func (*HashCSV) LoadProducts

```
func (hcsv *HashCSV) LoadProducts() *[]string
```

LoadProducts loads a products slice from an existing csv file, and returns that newly-loaded products slice's address.

func (*HashCSV) LoadSubmissions

```
func (hcsv *HashCSV) LoadSubmissions() *[]dsa.Ticket
```

LoadSubmissions loads a submissions heap (implemented in the dsa package) from an existing csv file, and returns that newly-loaded heap's address.

func (*HashCSV) LoadTickets

```
func (hcsv *HashCSV) LoadTickets() *dsa.TicketNode
```

LoadTickets loads a tickets AVL tree (implemented in the dsa package) from an existing csv file, and returns that newly-loaded AVL tree's address.

func (*HashCSV) LoadUsers

```
func (hcsv *HashCSV) LoadUsers() *[]*dsa.UserNode
```

LoadUsers loads a users hash table (implemented in the dsa package) from an existing csv file, and returns that newly-loaded users hash table's address.

func (*HashCSV) SaveProducts

```
func (hcsv *HashCSV) SaveProducts(products *[]string)
```

SaveProducts saves a products slice to an existing csv file, overwriting any existing data in the file, and updates the associated hash.

func (*HashCSV) SaveSubmissions

```
func (hcsv *HashCSV) SaveSubmissions(submissions *[]dsa.Ticket)
```

SaveSubmissions saves an existing submissions heap (implemented in the dsa package) to an existing csv file, overwriting any existing data in the file, and updates the associated hash.

func (*HashCSV) SaveTickets

```
func (hcsv *HashCSV) SaveTickets(Tickets *dsa.AVLtree)
```

SaveTickets saves an existing ticket AVL tree (implemented in the dsa package) to an existing csv file, overwriting any existing data in the file, and updates the associated hash.

func (*HashCSV) SaveUsers

```
func (hcsv *HashCSV) SaveUsers(users *[]*dsa.UserNode)
```

SaveUsers saves a users hash table to an existing csv file, overwriting any existing data in the file, and updates the associated hash.