

# Diario di lavoro

Luogo	Canobbio
Data	04.10.2019

## Lavori svolti

Oggi ho creato una nuova libreria chiamata *User*. Essa permette di salvare al suo interno tutti i dati di un utente. Viene utilizzata dal programma per una gestione semplificata dei dati dell'utente.

Tramite il model *UserModel* posso adesso caricare tutti i dati degli utenti salvati all'interno della banca dati locale sotto forma di array di oggetti *User*. Questo viene utilizzato per scrivere i dati all'interno della tabella per la visualizzazione degli utenti nella pagina "*Gestione Utenti*" del pannello admin.

Tramite la libreria JavaScript *DataTables* ho potuto inserire delle funzioni molto utili per cercare/filtrare dati all'interno della tabella degli utenti:

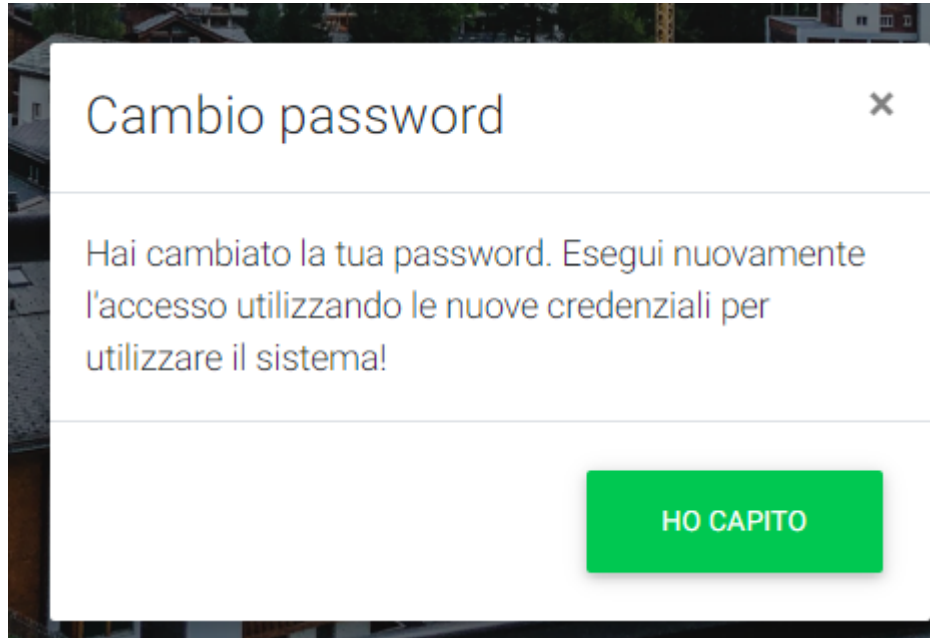
Username	Nome	cognome	Permessi	Password cambiata	Opzioni
luca.dibello	Luca	Di Bello	admin	No	<button>MODIFICA</button> <button>ELIMINA</button>
mattia.lazzaroni	Mattia	Lazzaroni	user	No	<button>MODIFICA</button> <button>ELIMINA</button>

Showing 1 to 2 of 2 entries

La libreria permette di fare una ricerca all'interno dei dati, di ordinare i dati per colonna (es: ordinare alfabeticamente per username) e di dividere i dati su più pagine (di default 10 record per pagina).

Dopo aver fatto questo ho messo apposto un altro problema relativo alla generazione dei link per il cambio password (spiegato in modo dettagliato nella sezione *Problemi riscontrati e soluzioni adottate*).

Dopo aver risolto il problema ho aggiunto un cookie che verrà utilizzato come flag nella pagina di login. Se un account ha cambiato password questo flag viene impostato a true e se entro 24 ore dal cambio password accede alla pagina di login viene mostrato un messaggio che conferma il cambio della password.



### Problemi riscontrati e soluzioni adottate

Ogni tanto quando cercavo di effettuare l'accesso con un nuovo utente l'applicazione mi dava l'errore di quando non trova il token per l'identificazione:

Hai fornito un id non valido.

Controllando per bene il token ho notato che criptandolo con il metodo *password\_hash* c'era la possibilità che qualche carattere all'interno del token diventavano degli '/' (slash).

```
localhost:8123/changepassword/id/$2y$10$rTOsP.XwQLRIltNDmZp0OcHe/jrtQZU/2U8jeEYLhQBhT.VRIQYq
```

Proprio per questo ho deciso di virare su un MD5. Dato che questo metodo non genera automaticamente un *salt* ho al posto di criptare un numero random tra 1 e 10, critto il timestamp del sistema per essere sicuri di avere sempre un token univoco. Ecco un esempio di link utilizzando il nuovo sistema:

```
localhost:8123/changepassword/id/bde80344dc4c32450c6ce5fd6c6bb63f
```

### Punto della situazione rispetto alla pianificazione

Sono in anticipo in anticipo rispetto alla pianificazione iniziale.

### Programma di massima per la prossima giornata di lavoro

Inserire la funzione di eliminazione ed aggiunta di utenti (tutto con validazione).