

Diario di lavoro

Luogo	Canobbio
Data	17.10.2019

Lavori svolti

Durante le prime due ore di lezione ho aggiunto un nuovo *handler* nella classe API che verrà utilizzato per le operazioni sui dati relativi alle prenotazioni.

```
/*
 * Action url: api/booking/add
 * Permission needed: visione_prenotazioni
 * Extra data: none
 */
elseif($action == "add" && $_SERVER["REQUEST_METHOD"] == "POST"){
    if(PermissionManager::getPermissions()->canInserirePrenotazioni()){
        // Sanitize POST data and promote user
        $result = BookingModel::add(filter_input_array( type: INPUT_POST, definition: FILTER_SANITIZE_STRING));

        // If it detects errors
        if(is_array($result)){
            $GLOBALS["NOTIFIER"]->add_all($result);
        }
    }
}
else{
    $GLOBALS["NOTIFIER"]->add("Non hai i permessi necessari per inserire le prenotazioni");
}

RedirectManager::redirect( path: "home/prenotazioni");
```

Come si può vedere nello screenshot il metodo controlla anche se l'utente ha effettivamente i permessi necessari per l'utilizzo di questa funzione (l'utente necessita i permessi per l'inserimento di prenotazioni).

Dopo il controllo dei permessi utilizzo il nuovo metodo *add* creato all'interno della classe *BookingModel*. Questo nuovo metodo non fa altro che prendere i dati ricevuti dalle API, validarli e, se tutto va a buon fine, inserire i dati all'interno del database. Questo è il codice:

```
/**
 * This method is used to add a new booking to the database.
 *
 * @param $data array Booking data array
 * @return array|bool Return array if the booking was added to the database, otherwise it returns an error as an
 * array
 */
public static function add(array $data){
    // Check booking data
    if(self::validateBookingData($data)){
        // Insert data into database
        $result = DB::insert('riservazione', array(
            'data' => $data['dataInizio']->format(BOOKING_DATE_FORMAT),
            'ora_inizio' => $data['dataInizio']->format(BOOKING_TIME_FORMAT),
            'ora_fine' => $data['dataFine']->format(BOOKING_TIME_FORMAT),
            'osservazioni' => (empty($data['osservazioni']) ? null : $data['osservazioni']),
            'utente' => $_SESSION['username']
        ));

        // Return true (if the insert query was successful) otherwise it returns an error as array
        return (! $result ? array("C'è stato un errore durante l'inserimento dei dati
            all'interno del database. Contattare un amministratore.") : true);
    }
    // Booking data not valid
    else{
        $GLOBALS["NOTIFIER"]->add_all(self::$errors);
        RedirectManager::redirect('path: "admin/utenti"');
    }
}
```

Successivamente ho aiutato per una quindicina di minuti Mattia Toscanelli e Mattia Lazzaroni per delle parti di codice.

Prima della fine della lezione mi sono ricordato di un requisito: *ci deve sempre essere almeno un amministratore*. Dato che nella gestione utenti non avevo implementato questo controllo ho creato un metodo nella classe *UserModel* che conta tutti gli amministratori presenti all'interno del database.

Problemi riscontrati e soluzioni adottate

Non ho riscontrato nessun problema

Punto della situazione rispetto alla pianificazione

Sono in anticipo rispetto alla pianificazione iniziale.

Programma di massima per la prossima giornata di lavoro

Creare il form per aggiungere gli utenti e testare il sistema.