

# Configuration

## CorsConfiguration

```
package com.simplilearn.capstone2.configuration;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.stereotype.Component;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration
@Component
public class CorsConfiguration {

    private static final String GET = "GET";
    private static final String POST = "POST";
    private static final String DELETE = "DELETE";
    private static final String PUT = "PUT";

    @Bean
    public WebMvcConfigurer corsConfigurer() {
        return new WebMvcConfigurer() {
            @Override
            public void addCorsMappings(CorsRegistry registry) {
                registry.addMapping("/**")
                    .allowedMethods(GET, PUT, POST, DELETE)
                    .allowedHeaders("**")
                    .allowedOriginPatterns("**")
            }
        };
    }
}
```

```

        .allowCredentials(true);
    }

};

}

}

```

### **JwtAuthenticationEntryPoint**

```

package com.simplilearn.capstone2.configuration;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.security.core.AuthenticationException;
import org.springframework.security.web.AuthenticationEntryPoint;
import org.springframework.stereotype.Component;

@Component
public class JwtAuthenticationEntryPoint implements AuthenticationEntryPoint {

    @Override
    public void commence(HttpServletRequest request, HttpServletResponse response,
        AuthenticationException authException) throws IOException, ServletException {
        //sending msg that access is unauthorized
        response.sendError(HttpServletResponse.SC_UNAUTHORIZED, "Unauthorized");
    }
}

```

```
}
```

```
}
```

### **JwtRequestFilter**

```
package com.simplilearn.capstone2.configuration;
```

```
import java.io.IOException;
```

```
import javax.servlet.FilterChain;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.context.annotation.Lazy;
```

```
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
```

```
import org.springframework.security.core.context.SecurityContextHolder;
```

```
import org.springframework.security.core.userdetails.UserDetails;
```

```
import org.springframework.security.web.authentication.WebAuthenticationDetailsSource;
```

```
import org.springframework.stereotype.Component;
```

```
import org.springframework.web.filter.OncePerRequestFilter;
```

```
import com.simplilearn.capstone2.service.JwtService;
```

```
import com.simplilearn.capstone2.util.JwtUtil;
```

```
import io.jsonwebtoken.ExpiredJwtException;
```

@Component

```
public class JwtRequestFilter extends OncePerRequestFilter {
```

```
    public static String CURRENT_USER = "";
```

```
    @Autowired
```

```
    private JwtUtil jwtutil;
```

```
    @Lazy
```

```
    @Autowired
```

```
    private JwtService jwtservice;
```

```
    @Override
```

```
    protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response,
```

```
        FilterChain filterChain)
```

```
        throws ServletException, IOException {
```

```
        final String header = request.getHeader("Authorization");
```

```
        String jwtToken = null;
```

```
        String userName = null;
```

```
        if (header != null && header.startsWith("Bearer ")) {
```

```
            jwtToken = header.substring(7);
```

```
            try {
```

```
                userName = jwtutil.getUserNameFromToken(jwtToken);
```

```
                CURRENT_USER = userName;
```

```
            } catch (IllegalArgumentException e) {
```

```

        System.out.println("Unable to get JWT token");
    } catch (ExpiredJwtException e) {
        System.out.println("Jwt token is expired");
    }

    } else {
        System.out.println("Jwt token does not start with bearer");
    }

    if (userName != null && SecurityContextHolder.getContext().getAuthentication() == null)
{

        UserDetails userDetails = jwtService.loadUserByUsername(userName);

        if (jwtutil.ValidateToken(jwtToken, userDetails)) {
            UsernamePasswordAuthenticationToken
usernamePasswordAuthenticationToken =
                                new
UsernamePasswordAuthenticationToken(userDetails,

                                null, userDetails.getAuthorities());

            usernamePasswordAuthenticationToken.setDetails
            (new WebAuthenticationDetailsSource().buildDetails(request));

            SecurityContextHolder.getContext().setAuthentication(usernamePasswordAuthenticationToken)
;

        }

    }
}

```

```
        filterChain.doFilter(request, response);
    }

}
```

### **WebSecurityConfiguration**

```
package com.simplilearn.capstone2.configuration;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Lazy;
import org.springframework.http.HttpHeaders;
import org.springframework.security.authentication.AuthenticationManager;
import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import
org.springframework.security.config.annotation.method.configuration.EnableGlobalMethodSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.config.http.SessionCreationPolicy;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;
import org.springframework.stereotype.Component;
```

```
import com.simplilearn.capstone2.service.JwtService;
```

```
@Configuration
```

```
@EnableWebSecurity
```

```
@EnableGlobalMethodSecurity(prePostEnabled = true)
```

```
@Component
```

```
public class WebSecurityConfiguration extends WebSecurityConfigurerAdapter {
```

```
    @Autowired
```

```
    private JwtAuthenticationEntryPoint jwtauthenticationentrypoint;
```

```
    //@Lazy
```

```
    @Autowired
```

```
    private JwtRequestFilter jwtrequestfilter;
```

```
    //@Lazy
```

```
    @Autowired
```

```
    private UserDetailsService jwtservice;
```

```
    @Bean
```

```
    @Override
```

```
    public AuthenticationManager authenticationManagerBean() throws Exception {
```

```
        return super.authenticationManagerBean();
```

```
    }
```

```
    @Override
```

```
    protected void configure(HttpSecurity httpsecurity) throws Exception {
```

```
        httpsecurity.cors();
```

```

        httpsecurity.csrf().disable().authorizeRequests()

            .antMatchers("/authenticate","/registerNewUser","/movieticket/showall","getTicketById/{ticket
            Id}","/getTicketDetails/{isSingleTicketCheckout},{ticketId}")

                .permitAll().antMatchers(HttpHeaders.ALLOW)

                    .permitAll().anyRequest().authenticated()

                    .and()

                    .exceptionHandling()

            .authenticationEntryPoint(jwtauthenticationentrypoint).and().sessionManagement()

                .sessionCreationPolicy(SessionCreationPolicy.STATELESS);

        httpsecurity.addFilterBefore(jwtrequestfilter,
        UsernamePasswordAuthenticationFilter.class);
    }

    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }

    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder authenticatnManagerBuilder)
    throws Exception {

        authenticatnManagerBuilder.userDetailsService(jwtService).passwordEncoder(passwordEncoder
        ());
    }

}

```



# Controller

## CartController

```
package com.simplilearn.capstone2.controller;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.security.access.prepost.PreAuthorize;
```

```
import org.springframework.web.bind.annotation.DeleteMapping;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
import com.simplilearn.capstone2.entity.Cart;
```

```
import com.simplilearn.capstone2.service.CartService;
```

```
@RestController
```

```
public class CartController {
```

```
    @Autowired
```

```
    private CartService cartService;
```

```
    @PreAuthorize("hasRole('User')")
```

```
    @GetMapping("/{}/addToCart/{ticketId}")
```

```
    public Cart addToCart(@PathVariable(name="ticketId")Integer ticketId) {
```

```
        return cartService.addToCart(ticketId);
```

```
    }
```

```
    @PreAuthorize("hasRole('User')")
```

```
    @DeleteMapping("/{}/deleteCartItem/{cartId}")
```

```

        public void deleteCartItem(@PathVariable (name="cartId")Integer cartId) {
            cartService.deleteCartItem(cartId);
        }

        @PreAuthorize("hasRole('User')")
        @GetMapping("/{getCartDetails}")
        public List<Cart> getCartDetails() {
            return cartService.getCartDetails();
        }
    }
}

```

### **JwtController**

```

package com.simplilearn.capstone2.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import com.simplilearn.capstone2.entity.JwtRequest;
import com.simplilearn.capstone2.entity.JwtResponse;
import com.simplilearn.capstone2.service.JwtService;

@RestController
@CrossOrigin
public class JwtController {

    @Autowired

```

```

        private JwtService jwtService;

        @PostMapping("/{authenticate}")
        public JwtResponse CreateJwtToken(@RequestBody JwtRequest jwtrequest)throws Exception {
            return jwtService.createJwtToken(jwtrequest);
        }
    }

```

### **MovieTicketController**

```

package com.simplilearn.capstone2.controller;

import java.io.IOException;

import java.util.HashSet;
import java.util.List;
import java.util.Set;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestPart;

```

```

import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.multipart.MultipartFile;

import com.simplilearn.capstone2.dao.MovieTicketDao;
import com.simplilearn.capstone2.entity.MovieTickets;
import com.simplilearn.capstone2.entity.Posters;
import com.simplilearn.capstone2.service.MovieTicketService;

@RestController
@CrossOrigin
public class MovieTicketController {

    @Autowired
    private MovieTicketService mtService;

    // @Autowired
    // private MovieTicketDao mtDao;

    @PreAuthorize("hasRole('Admin')")
    @PostMapping(value = { "/movieticket/add" }, consumes = {
MediaType.MULTIPART_FORM_DATA_VALUE })
    public MovieTickets addNewTicket(@RequestPart("movieticket") MovieTickets mt,
                                     @RequestPart("posters") MultipartFile[] file) {
        // return mtService.addNewTicket(mt);
        try {
            Set<Posters> posters = uploadPoster(file);
            mt.setMoviePoster(posters);
            return mtService.addNewTicket(mt);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}

```

```

        return null;
    }

}

// for processing these uploaded images
public Set<Posters> uploadPoster(MultipartFile[] multipartFiles) throws IOException {
    Set<Posters> posters = new HashSet<>();

    for (MultipartFile file : multipartFiles) {
        Posters p = new Posters(file.getOriginalFilename(), file.getContentType(),
file.getBytes());

        posters.add(p);
    }

    return posters;
}

// @PreAuthorize("hasRole('Admin')")
@GetMapping({ "/movieticket/showall" })
public List<MovieTickets> getAllTickets() {
    return mtService.getAllTickets();
}

@PreAuthorize("hasRole('Admin')")
@DeleteMapping({ "/deleteticket/{ticketId}" })
public void deleteTicket(@PathVariable("ticketId") Integer ticketId) {
    mtService.deleteTicket(ticketId);
}

```

```

    @GetMapping("getTicketById/{ticketId}")
    public MovieTickets getTicketById(@PathVariable Integer ticketId) {
        return mtService.getTicketsById(ticketId);
    }

    @PreAuthorize("hasRole('User')")
    @GetMapping({ "/getTicketDetails/{isSingleTicketCheckout}/{ticketId}" })
    public List<MovieTickets> getTicketDetails(@PathVariable(name = "isSingleTicketCheckout")
boolean isSingleTicketCheckout,
        @PathVariable(name = "ticketId")Integer ticketId) {
        return mtService.getTicketDetails(isSingleTicketCheckout, ticketId);
    }
}

```

### **OrderDetailController**

```

package com.simplilearn.capstone2.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

```

```

import com.simplilearn.capstone2.entity.OrderDetail;
import com.simplilearn.capstone2.entity.OrderInput;
import com.simplilearn.capstone2.entity.TransactionDetails;
import com.simplilearn.capstone2.service.OrderDetailService;

@RestController

public class OrderDetailController {

    @Autowired
    private OrderDetailService orderDetailService;

    @PreAuthorize("hasRole('User')")
    @PostMapping("/{place/order/{isSingleTicketCheckout}}")
    public void placeOrder(@PathVariable (name="isSingleTicketCheckout") boolean
isSingleTicketCheckout,

        @RequestBody OrderInput orderInput) {
        orderDetailService.placeOrder(orderInput,isSingleTicketCheckout);

    }

    @PreAuthorize("hasRole('User')")
    @GetMapping("/{getOrderDetails}")
    public List<OrderDetail> getOrderDetails() {
        return orderDetailService.getOrderDetails();
    }

    @PreAuthorize("hasRole('Admin')")
    @GetMapping("/{getAllOrderDetails/{status}}")
    public List<OrderDetail> getAllOrderDetails(@PathVariable(name = "status")String status){

```

```

        return orderDetailService.getAllOrderDetails(status);
    }

    @PreAuthorize("hasRole('Admin')")
    @GetMapping("/{orderId}/markOrderWatched/{orderId}")
    public void markOrderWatched(@PathVariable(name = "orderId")Integer orderId ) {
        //this can be used for marked order as delivered in other cases.
        orderDetailService.markOrderAsWatched(orderId);
    }

    @PreAuthorize("hasRole('User')")
    @GetMapping("/{amount}/createTransaction/{amount}")
    public TransactionDetails CreateTransaction(@PathVariable(name= "amount")Double amount) {
        return orderDetailService.createTransaction(amount);
    }
}

```

### **RoleController**

```

package com.simplilearn.capstone2.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import com.simplilearn.capstone2.entity.Role;
import com.simplilearn.capstone2.service.RoleService;

@RestController

```



```

public class RoleController {

    @Autowired

    private RoleService roleservice;


    @PostMapping("/{createNewRole"})

    public Role createNewRole(@RequestBody Role role) {

        return roleservice.createNewRole(role);

    }

}

```

### **UserController**

```

package com.simplilearn.capstone2.controller;


import javax.annotation.PostConstruct;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;


import com.simplilearn.capstone2.entity.User;
import com.simplilearn.capstone2.service.UserService;


@RestController

public class UserController {

    @Autowired

    private UserService userservice;

```

@PostConstruct//will run each the program starts

```
public void initRolesAndUsers() {  
    userservice.initRolesAndUser();  
}
```

@PostMapping("/{registerNewUser}")

```
public User RegisterNewUser(@RequestBody User user) {  
    return userservice.RegisterNewUser(user);  
}
```

@GetMapping("/{forAdmin}")

@PreAuthorize("hasRole('Admin')") //this will give access only to users having role of Admin

```
public String forAdmin() {  
    return "This URL is only accessible to admin";  
}
```

@GetMapping("/{forUser}")

@PreAuthorize("hasRole('User')")

```
public String forUser() {  
    return "This URL is only accessible to User";  
}
```

```
}
```

## **Dao**

### **CartDao**

```
package com.simplilearn.capstone2.dao;
```

```
import java.util.List;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.simplilearn.capstone2.entity.Cart;
```

```
import com.simplilearn.capstone2.entity.User;
```

```
@Repository
```

```
public interface CartDao extends CrudRepository<Cart, Integer>{
```

```
    public List<Cart> findByUser(User user);
```

```
}
```

### **MovieTicketDao**

```
package com.simplilearn.capstone2.dao;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.simplilearn.capstone2.entity.MovieTickets;
```

```
@Repository
```

```
public interface MovieTicketDao extends CrudRepository<MovieTickets, Integer> {
```

```
}
```

### **OrderDetailDao**

```
package com.simplilearn.capstone2.dao;
```

```
import java.util.List;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.simplilearn.capstone2.entity.OrderDetail;
```

```
import com.simplilearn.capstone2.entity.User;
```

```
@Repository
```

```
public interface OrderDetailDao extends CrudRepository<OrderDetail, Integer> {
```

```
    public List<OrderDetail> findByUser(User user);
```

```
    //this will take object of user and will return the list of orderdetail
```

```
    //in this method findBy is mandatory followed by variable name in OrderDetail class i.e is user in  
    this case.
```

```
    public List<OrderDetail> findByOrderStatus(String status);
```

```
}
```

### **RoleDao**

```
package com.simplilearn.capstone2.dao;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.simplilearn.capstone2.entity.Role;
```

```
@Repository
```

```
public interface RoleDao extends CrudRepository<Role, String>{
```

```
}
```

### **UserDao**

```
package com.simplilearn.capstone2.dao;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import com.simplilearn.capstone2.entity.User;
```

```
@Repository
```

```
public interface UserDao extends CrudRepository<User, String> {
```

```
}
```

## **Entity**

### **Cart**

```
package com.simplilearn.capstone2.entity;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.OneToOne;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table
```

```
public class Cart {
```

```
    @Id
```

```
@GeneratedValue(strategy = GenerationType.AUTO)

private Integer cartId;

@OneToOne
private MovieTickets movieTicket;

@OneToOne
private User user;


public Cart() {

}


public Cart(MovieTickets movieTicket, User user) {
    super();
    this.movieTicket = movieTicket;
    this.user = user;
}

public Integer getCartId() {
    return cartId;
}

public void setCartId(Integer cartId) {
    this.cartId = cartId;
}

public MovieTickets getMovieTicket() {
    return movieTicket;
}

public void setMovieTicket(MovieTickets movieTicket) {
    this.movieTicket = movieTicket;
}
```

```
}  
  
public User getUser() {  
    return user;  
}  
  
public void setUser(User user) {  
    this.user = user;  
}  
  
}
```

### **JwtRequest**

```
package com.simplilearn.capstone2.entity;  
  
public class JwtRequest {  
    private String userName;  
    private String userPassword;  
    public String getUserName() {  
        return userName;  
    }  
    public void setUserName(String userName) {  
        this.userName = userName;  
    }  
    public String getUserPassword() {  
        return userPassword;  
    }  
    public void setUserPassword(String userPassword) {  
        this.userPassword = userPassword;  
    }  
}
```

```
}
```

### JwtResponse

```
package com.simplilearn.capstone2.entity;
```

```
public class JwtResponse {
```

```
    private User user;
```

```
    private String jwtToken;
```

```
    public JwtResponse(User user, String jwtToken) {
```

```
        super();
```

```
        this.user = user;
```

```
        this.jwtToken = jwtToken;
```

```
    }
```

```
    public User getUser() {
```

```
        return user;
```

```
    }
```

```
    public void setUser(User user) {
```

```
        this.user = user;
```

```
    }
```

```
    public String getJwtToken() {
```

```
        return jwtToken;
```

```
    }
```

```
    public void setJwtToken(String jwtToken) {
```

```
        this.jwtToken = jwtToken;
```

```
    }
```



```
}
```

### **MovieTickets**

```
package com.simplilearn.capstone2.entity;
```

```
import java.util.Set;
```

```
import javax.persistence.CascadeType;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.FetchType;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.JoinColumn;
```

```
import javax.persistence.JoinTable;
```

```
import javax.persistence.ManyToMany;
```

```
@Entity
```

```
public class MovieTickets {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    private int ticketId;
```

```
    private String movieName;
```

```
    private String genre;
```

```
    private String actors;
```

```
    private String director;
```

```
    private String language;
```

```
    private String description;
```

```
    private double actualPrice;
```

```
private double discountedPrice;

@ManyToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)

@JoinTable(name = "ticket_posters", joinColumns = { @JoinColumn(name = "ticket_id") },
inverseJoinColumns = {

    @JoinColumn(name = "poster_id") })

private Set<Posters> moviePoster;


public Set<Posters> getMoviePoster() {

    return moviePoster;

}


public void setMoviePoster(Set<Posters> moviePoster) {

    this.moviePoster = moviePoster;

}


public String getLanguage() {

    return language;

}


public void setLanguage(String language) {

    this.language = language;

}


public String getDirector() {

    return director;

}
```

```
public void setDirector(String director) {  
    this.director = director;  
}
```

```
public int getTicketId() {  
    return ticketId;  
}
```

```
public void setTicketId(int ticketId) {  
    this.ticketId = ticketId;  
}
```

```
public String getMovieName() {  
    return movieName;  
}
```

```
public void setMovieName(String movieName) {  
    this.movieName = movieName;  
}
```

```
public String getGenre() {  
    return genre;  
}
```

```
public void setGenre(String genre) {  
    this.genre = genre;  
}
```

```
public String getActors() {
```

```
        return actors;
    }
}
```

```
public void setActors(String actors) {
    this.actors = actors;
}
```

```
public String getDescription() {
    return description;
}
```

```
public void setDescription(String description) {
    this.description = description;
}
```

```
public double getActualPrice() {
    return actualPrice;
}
```

```
public void setActualPrice(double actualPrice) {
    this.actualPrice = actualPrice;
}
```

```
public double getDiscountedPrice() {
    return discountedPrice;
}
```

```
public void setDiscountedPrice(double discountedPrice) {
    this.discountedPrice = discountedPrice;
}
```

```
}
```

```
public MovieTickets(int ticketId, String movieName, String genre, String actors, String director,  
                    String description, String language, double actualPrice, double discountedPrice)  
{  
    super();  
    this.ticketId = ticketId;  
    this.movieName = movieName;  
    this.genre = genre;  
    this.actors = actors;  
    this.director = director;  
    this.language = language;  
    this.description = description;  
    this.actualPrice = actualPrice;  
    this.discountedPrice = discountedPrice;  
}
```

```
public MovieTickets() {  
    super();  
    // TODO Auto-generated constructor stub  
}
```

```
}
```

### **MovieTickets**

```
package com.simplilearn.capstone2.entity;
```

```
import java.util.Set;
```

```
import javax.persistence.CascadeType;
```

```
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
```

```
@Entity
```

```
public class MovieTickets {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    private int ticketId;
```

```
    private String movieName;
```

```
    private String genre;
```

```
    private String actors;
```

```
    private String director;
```

```
    private String language;
```

```
    private String description;
```

```
    private double actualPrice;
```

```
    private double discountedPrice;
```

```
    @ManyToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
```

```
    @JoinTable(name = "ticket_posters", joinColumns = { @JoinColumn(name = "ticket_id") },
inverseJoinColumns = {
```

```
        @JoinColumn(name = "poster_id") })
```

```
    private Set<Posters> moviePoster;
```

```
public Set<Posters> getMoviePoster() {  
    return moviePoster;  
}
```

```
public void setMoviePoster(Set<Posters> moviePoster) {  
    this.moviePoster = moviePoster;  
}
```

```
public String getLanguage() {  
    return language;  
}
```

```
public void setLanguage(String language) {  
    this.language = language;  
}
```

```
public String getDirector() {  
    return director;  
}
```

```
public void setDirector(String director) {  
    this.director = director;  
}
```

```
public int getTicketId() {  
    return ticketId;  
}
```

```
public void setTicketId(int ticketId) {  
    this.ticketId = ticketId;  
}
```

```
public String getMovieName() {  
    return movieName;  
}
```

```
public void setMovieName(String movieName) {  
    this.movieName = movieName;  
}
```

```
public String getGenre() {  
    return genre;  
}
```

```
public void setGenre(String genre) {  
    this.genre = genre;  
}
```

```
public String getActors() {  
    return actors;  
}
```

```
public void setActors(String actors) {  
    this.actors = actors;  
}
```



```
public String getDescription() {  
    return description;  
}
```

```
public void setDescription(String description) {  
    this.description = description;  
}
```

```
public double getActualPrice() {  
    return actualPrice;  
}
```

```
public void setActualPrice(double actualPrice) {  
    this.actualPrice = actualPrice;  
}
```

```
public double getDiscountedPrice() {  
    return discountedPrice;  
}
```

```
public void setDiscountedPrice(double discountedPrice) {  
    this.discountedPrice = discountedPrice;  
}
```

```
public MovieTickets(int ticketId, String movieName, String genre, String actors, String director,  
    String description, String language, double actualPrice, double discountedPrice)  
{  
    super();  
    this.ticketId = ticketId;
```

```

        this.movieName = movieName;

        this.genre = genre;

        this.actors = actors;

        this.director = director;

        this.language = language;

        this.description = description;

        this.actualPrice = actualPrice;

        this.discountedPrice = discountedPrice;
    }

    public MovieTickets() {
        super();
        // TODO Auto-generated constructor stub
    }
}

```

### **OrderDetail**

```

package com.simplilearn.capstone2.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;

@Entity
public class OrderDetail {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)

```

```
private Integer orderId;  
private String orderFullname;  
private String orderFullAddress;  
private String orderContactNumber;  
private String orderAlternateNumber;  
private String orderStatus;  
private double orderAmount;  
@OneToOne  
private MovieTickets movieTicket;  
@OneToOne  
private User user;  
private String transactionId;
```

```
public String getTransactionId() {  
    return transactionId;  
}  
public void setTransactionId(String transactionId) {  
    this.transactionId = transactionId;  
}  
public OrderDetail() {  
  
}  
public MovieTickets getMovieTicket() {  
    return movieTicket;  
}  
public void setMovieTicket(MovieTickets movieTicket) {
```

```
        this.movieTicket = movieTicket;
    }
    public User getUser() {
        return user;
    }
    public void setUser(User user) {
        this.user = user;
    }
    public Integer getOrderId() {
        return orderId;
    }
    public void setOrderId(Integer orderId) {
        this.orderId = orderId;
    }
    public String getOrderFullname() {
        return orderFullname;
    }
    public void setOrderFullname(String orderFullname) {
        this.orderFullname = orderFullname;
    }
    public String getOrderFullAddress() {
        return orderFullAddress;
    }
    public void setOrderFullAddress(String orderFullAddress) {
        this.orderFullAddress = orderFullAddress;
    }
    public String getOrderContactNumber() {
        return orderContactNumber;
    }
}
```

```

    public void setOrderContactNumber(String orderContactNumber) {
        this.orderContactNumber = orderContactNumber;
    }
    public String getOrderAlternateNumber() {
        return orderAlternateNumber;
    }
    public void setOrderAlternateNumber(String orderAlternateNumber) {
        this.orderAlternateNumber = orderAlternateNumber;
    }
    public String getOrderStatus() {
        return orderStatus;
    }
    public void setOrderStatus(String orderStatus) {
        this.orderStatus = orderStatus;
    }
    public double getOrderAmount() {
        return orderAmount;
    }
    public void setOrderAmount(double orderAmount) {
        this.orderAmount = orderAmount;
    }

    public OrderDetail( String orderFullname, String orderFullAddress, String orderContactNumber,
        String orderAlternateNumber, String orderStatus, double orderAmount,
MovieTickets movieTicket, User user,
        String transactionId) {
        super();

        this.orderFullname = orderFullname;
        this.orderFullAddress = orderFullAddress;

```

```
        this.orderContactNumber = orderContactNumber;

        this.orderAlternateNumber = orderAlternateNumber;

        this.orderStatus = orderStatus;

        this.orderAmount = orderAmount;

        this.movieTicket = movieTicket;

        this.user = user;

        this.transactionId = transactionId;

    }
```

```
}
```

### **OrderInput**

```
package com.simplilearn.capstone2.entity;
```

```
import java.util.List;
```

```
public class OrderInput {
```

```
    private String fullName;

    private String fullAddress;

    private String contactNumber;

    private String alternateContactNumber;

    private List<OrderTicketQuantity> orderTicketQuantity;

    private String transactionId;
```

```
public String getTransactionId() {  
    return transactionId;  
}  
  
public void setTransactionId(String transactionId) {  
    this.transactionId = transactionId;  
}  
  
public String getFullName() {  
    return fullName;  
}  
  
public void setFullName(String fullName) {  
    this.fullName = fullName;  
}  
  
public String getFullAddress() {  
    return fullAddress;  
}  
  
public void setFullAddress(String fullAddress) {  
    this.fullAddress = fullAddress;  
}  
  
public String getContactNumber() {  
    return contactNumber;  
}  
  
public void setContactNumber(String contactNumber) {  
    this.contactNumber = contactNumber;  
}  
  
public String getAlternateContactNumber() {  
    return alternateContactNumber;  
}  
  
public void setAlternateContactNumber(String alternateContactNumber) {  
    this.alternateContactNumber = alternateContactNumber;  
}
```

```

    }

    public List<OrderTicketQuantity> getOrderTicketQuantity() {
        return orderTicketQuantity;
    }

    public void setOrderTicketQuantity(List<OrderTicketQuantity> orderTicketQuantity) {
        this.orderTicketQuantity = orderTicketQuantity;
    }
}

```

### **OrderTicketQuantity**

```
package com.simplilearn.capstone2.entity;
```

```

public class OrderTicketQuantity {

    private Integer ticketId;
    private Integer quantity;

    public Integer getTicketId() {
        return ticketId;
    }

    public void setTicketId(Integer ticketId) {
        this.ticketId = ticketId;
    }

    public Integer getQuantity() {
        return quantity;
    }

    public void setQuantity(Integer quantity) {
        this.quantity = quantity;
    }
}

```



```
}
```

### Posters

```
package com.simplilearn.capstone2.entity;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name="movie_poster")
```

```
public class Posters {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    private long posterId;
```

```
    private String name;
```

```
    private String type;
```

```
    @Column(length = 5000000)
```

```
    private byte[] pic;
```

```
    public Posters(String name, String type, byte[] pic) {
```

```
        super();
```

```
        this.name = name;
```

```
        this.type = type;
```

```
        this.pic = pic;
```

```
    }
```

```
public Posters() {  
    super();  
    // TODO Auto-generated constructor stub  
}
```

```
public long getPosterId() {  
    return posterId;  
}
```

```
public void setPosterId(long posterId) {  
    this.posterId = posterId;  
}
```

```
public String getName() {  
    return name;  
}
```

```
public void setName(String name) {  
    this.name = name;  
}
```

```
public String getType() {  
    return type;  
}
```

```
public void setType(String type) {  
    this.type = type;  
}
```

```
public byte[] getPic() {  
    return pic;  
}
```

```
public void setPic(byte[] pic) {  
    this.pic = pic;  
}
```

```
}
```

```
}
```

### **Posters**

```
package com.simplilearn.capstone2.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="movie_poster")
public class Posters {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long posterId;
    private String name;
    private String type;
    @Column(length = 5000000)
    private byte[] pic;

    public Posters(String name, String type, byte[] pic) {
        super();
        this.name = name;
        this.type = type;
    }
}
```

```
        this.pic = pic;
    }

    public Posters() {
        super();
        // TODO Auto-generated constructor stub
    }
```

```
    public long getPosterId() {
        return posterId;
    }

    public void setPosterId(long posterId) {
        this.posterId = posterId;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }

    public byte[] getPic() {
        return pic;
    }
```

```
        public void setPic(byte[] pic) {  
            this.pic = pic;  
        }  
  
    }  
}
```

### Role

```
package com.simplilearn.capstone2.entity;  
  
import javax.persistence.Entity;  
import javax.persistence.Id;  
  
@Entity  
public class Role {  
  
    @Id  
    private String roleName;  
    private String roleDescription;  
    public String getRoleName() {  
        return roleName;  
    }  
    public void setRoleName(String roleName) {  
        this.roleName = roleName;  
    }  
    public String getRoleDescription() {  
        return roleDescription;  
    }  
    public void setRoleDescription(String roleDescription) {
```

```
        this.roleDescription = roleDescription;
    }

}
```

### **TransactionDetails**

```
package com.simplilearn.capstone2.entity;
```

```
public class TransactionDetails {
```

```
    private String orderId;
    private String currency;
    private Integer amount;
    private String key;
```

```
    public TransactionDetails(String orderId, String currency, Integer amount, String key) {
        super();
        this.orderId = orderId;
        this.currency = currency;
        this.amount = amount;
        this.key = key;
    }
```

```
    public String getKey() {
        return key;
    }
```

```
public void setKey(String key) {  
    this.key = key;  
}
```

```
public String getOrderId() {  
    return orderId;  
}
```

```
public void setOrderId(String orderId) {  
    this.orderId = orderId;  
}
```

```
public String getCurrency() {  
    return currency;  
}
```

```
public void setCurrency(String currency) {  
    this.currency = currency;  
}
```

```
public Integer getAmount() {  
    return amount;  
}
```

```
public void setAmount(Integer amount) {  
    this.amount = amount;  
}
```

```
}
```

## User

```
package com.simplilearn.capstone2.entity;
```

```
import java.util.Set;
```

```
import javax.persistence.CascadeType;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.FetchType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.JoinColumn;
```

```
import javax.persistence.JoinTable;
```

```
import javax.persistence.ManyToMany;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name="users")
```

```
public class User {
```

```
    @Id
```

```
    private String userName;
```

```
    private String fullName;
```

```
    private String userpassword;
```

```
    // many users may have many different roles
```

```
    /*
```

```
    * By itself create a third table named USER_ROLE that will have user and its
```

```
    * associated role details this third table will have user_id and associated
```

```
    * role_id
```



```

    */

    @ManyToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)

    @JoinTable(name = "USER_ROLE", joinColumns = { @JoinColumn(name = "USER_ID") },
inverseJoinColumns = {

        @JoinColumn(name = "ROLE_ID") })

    private Set<Role> role;


    public String getUsername() {

        return userName;

    }


    public void setUsername(String userName) {

        this.userName = userName;

    }


    public String getFullName() {

        return fullName;

    }


    public void setFullName(String fullName) {

        this.fullName = fullName;

    }


    public String getUserpassword() {

        return userpassword;

    }


    public void setUserpassword(String userpassword) {

        this.userpassword = userpassword;

    }

```

```
}
```

```
public Set<Role> getRole() {
```

```
    return role;
```

```
}
```

```
public void setRole(Set<Role> role) {
```

```
    this.role = role;
```

```
}
```

```
}
```

