

**C** SecurityConfig

**f** userDetailsService UserDetailsServiceImpl

**m** configure(AuthenticationManagerBuilder) void

**m** configure(HttpSecurity) void

**m** bCryptPasswordEncoder() BCryptPasswordEncoder

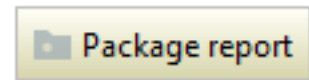
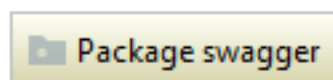
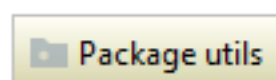
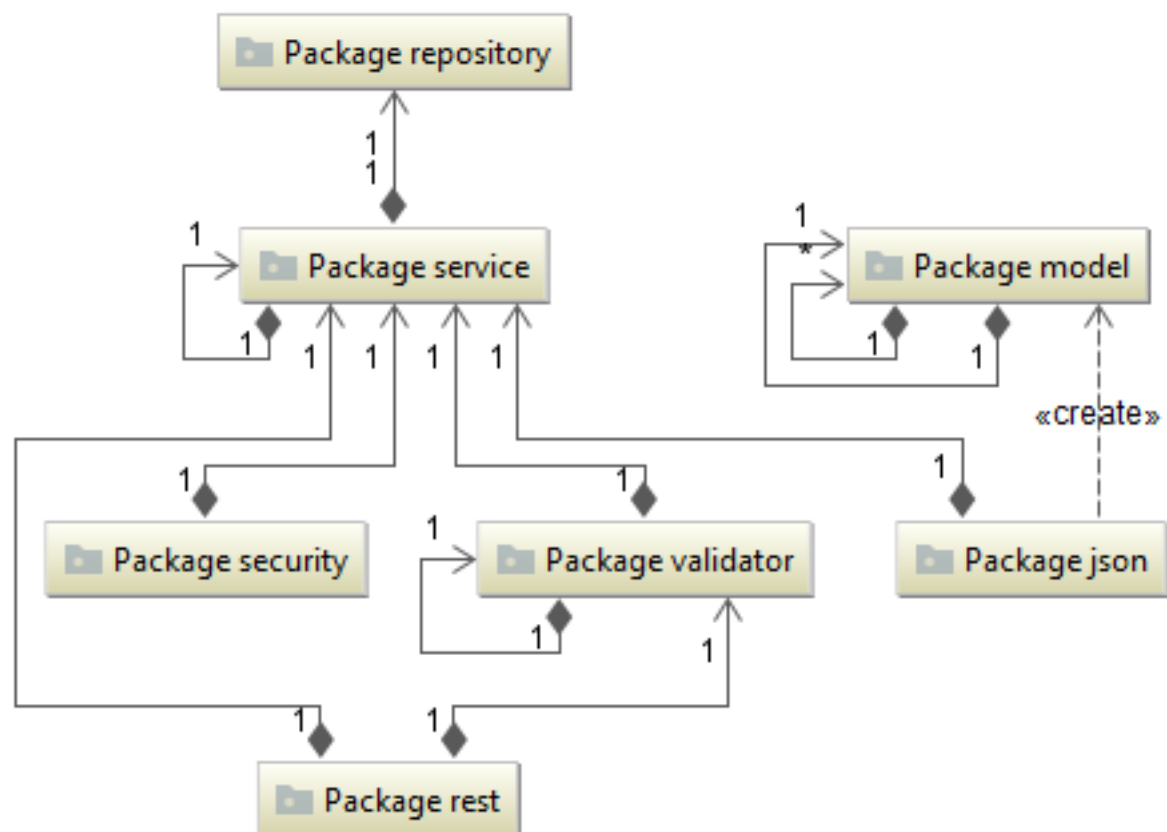
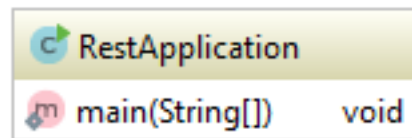
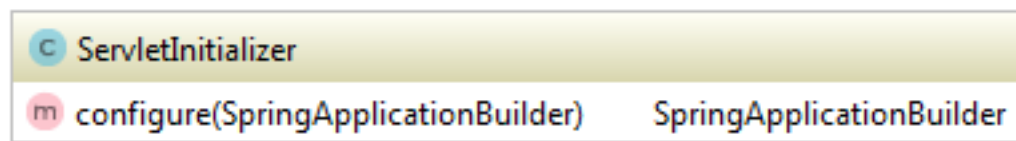
**m** authenticationProvider() DaoAuthenticationProvider

@ EnableSwagger2

@ Configuration  
value() String

SwaggerConfig  
api() Docket

Powered by yFiles

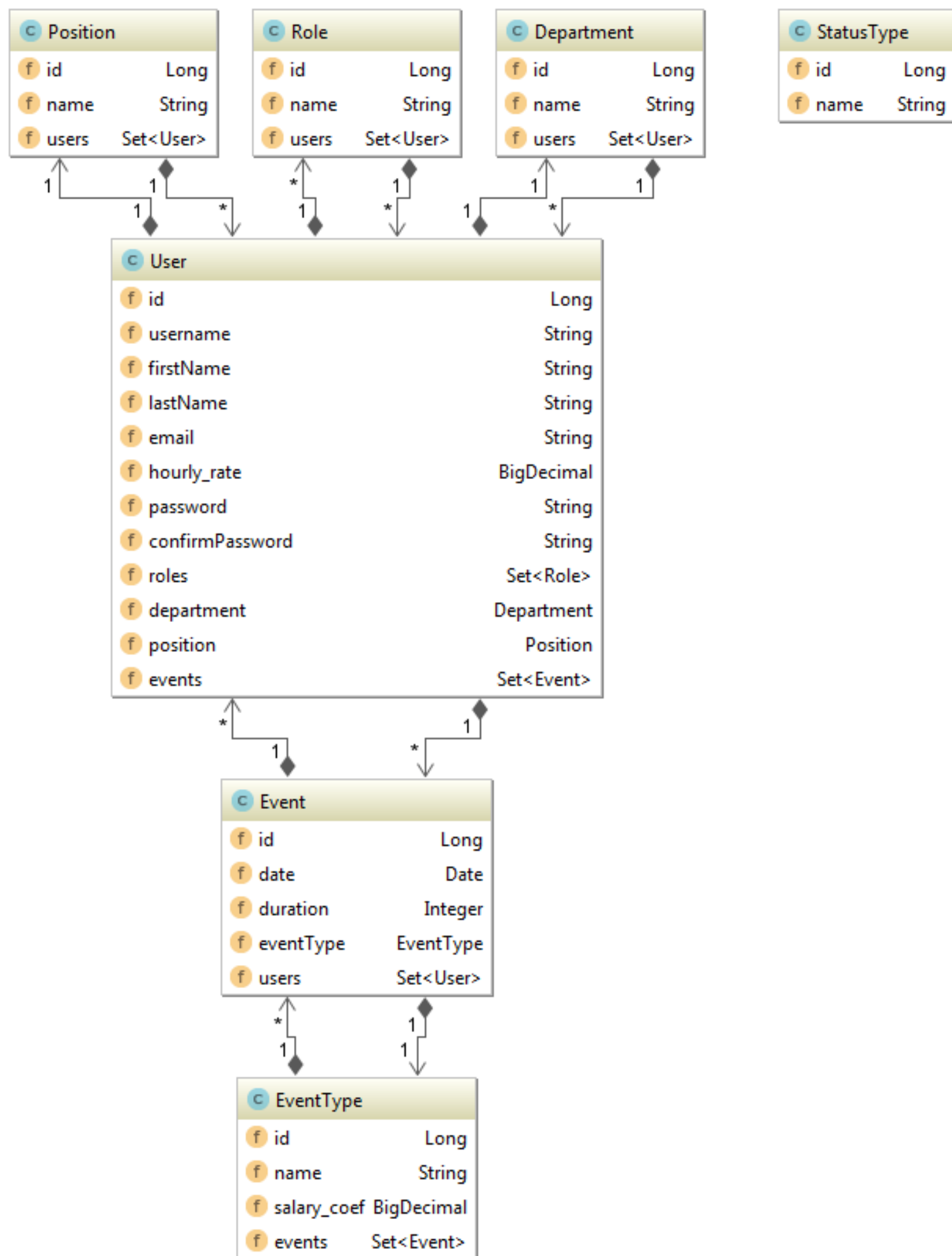










C CustomEventDeserializer		
f	eventTypeService	EventTypeService
f	userService	UserService
f	dateFormatter	SimpleDateFormat
m	deserialize(JsonParser, DeserializationContext) Event	

C CustomUserSalarySerializer		
f	dateFormatter	SimpleDateFormat
m	serialize(UserSalaryAggregation, JsonGener	






C CustomEventSerializer		
m	serialize(Event, JsonGenerator, SerializerProvider)	void













### UserMonthlySalaryAggregation


 userId	Long
 year	Integer
 month	Integer
 username	String
 salary	BigDecimal
 email	String


### UserSalaryAggregation


 userId	Long
 startDate	Date
 endDate	Date
 username	String
 salary	BigDecimal


I UserSalaryRepository		
	query	String
	getUserSalaryForPeriod(Long, Date, Date)	ArrayList<UserSalary>
	getUserSalaryForYearMonth(Long, Integer, Integer)	ArrayList<UserSalary>
	storeUserMonthlySalary(Long, Integer, Integer, Double)	Boolean
	findByUserIdAndYearAndMonth(Long, Integer, Integer)	ArrayList<UserSalary>
	findByYearAndMonth(Integer, Integer)	ArrayList<UserSalary>
	deleteByUserIdAndYearAndMonth(Long, Integer, Integer)	Boolean
	deleteByYearAndMonth(Integer, Integer)	Boolean


I UserRepository		
	findByUsername(String)	User
	findByUserIdAndEventDate(Long, Date)	ArrayList<User>

I EventRepository		
	findByDateAndEventType(Long, Date)	Long

I DepartmentRepository		
	findByName(String)	Department

I EventTypeRepository		
	findByName(String)	EventType

I PositionRepository		
	findByName(String)	Position

I RoleRepository		
	findByName(String)	Role

UserSalaryController		
f	userSalaryService	UserSalaryService
f	userService	UserService
f	monthlySalaryService	MonthlySalaryService
f	emailService	EmailService
f	dateFormatter	SimpleDateFormat
f	salaryValidator	SalaryValidator
f	roleUserValidator	RoleUserValidator
m	getUserSalary(long, String, String, Authentication)	ResponseEntity<?>
m	setUserMonthlySalary(long, String, String)	ResponseEntity<?>
m	deleteUserMonthlySalary(long, String, String)	ResponseEntity<?>
m	deleteMonthlySalary(String, String)	ResponseEntity<?>
m	setMonthlySalary(String, String)	ResponseEntity<?>
m	sendUserMonthlySalary(long, String, String)	ResponseEntity<?>

DepartmentController		
f	departmentService	DepartmentService
f	userService	UserService
m	getAllDepartments()	ResponseEntity<List<Department>>
m	getDepartment(long)	ResponseEntity<Department>
m	createDepartment(Department, UriComponentsBuilder)	ResponseEntity<?>
m	updateDepartment(long, Department, UriComponentsBuilder)	ResponseEntity<?>
m	deleteDepartment(long)	ResponseEntity<Department>
m	getDepartmentUsers(long)	ResponseEntity<List<User>>
m	addUserToDepartment(long, long, UriComponentsBuilder)	ResponseEntity<Void>
m	checkDepartment(Department)	Map<String, String>

PositionController		
f	positionService	PositionService
f	userService	UserService
m	getAllPositions()	ResponseEntity<List<Position>>
m	getPosition(long)	ResponseEntity<Position>
m	createPosition(Position, UriComponentsBuilder)	ResponseEntity<?>
m	updatePosition(long, Position)	ResponseEntity<?>
m	deletePosition(long)	ResponseEntity<Position>
m	getPositionUsers(long)	ResponseEntity<List<User>>
m	addUserOnPosition(long, long, UriComponentsBuilder)	ResponseEntity<Void>
m	checkPosition(Position)	Map<String, String>

EventController		
f	eventService	EventService
f	dateFormatter	SimpleDateFormat
f	userService	UserService
m	getAllEvents()	ResponseEntity<List<Event>>
m	getEvent(long)	ResponseEntity<Event>
m	createEvent(Event, UriComponentsBuilder)	ResponseEntity<?>
m	deleteEvent(long)	ResponseEntity<Event>
m	checkEvent(Event)	Map<String, String>

UserController		
f	userService	UserService
f	roleUserValidator	RoleUserValidator
m	getAllUsers()	ResponseEntity<List<User>>
m	getUser(long, Authentication)	ResponseEntity<?>
m	createUser(User, UriComponentsBuilder)	ResponseEntity<?>
m	updateUser(long, User)	ResponseEntity<?>
m	deleteUser(long)	ResponseEntity<User>
m	checkUser(User)	Map<String, String>

RoleController		
f	roleService	RoleService
f	userService	UserService
m	getAllRoles()	ResponseEntity<List<Role>>
m	getRoleUsers(long)	ResponseEntity<List<User>>
m	addRoleToUser(long, long, UriComponentsBuilder)	ResponseEntity<?>
m	removeRoleFromUser(long, long, UriComponentsBuilder)	ResponseEntity<?>

EventTypeController		
f	eventTypeService	EventTypeService
m	getAllEventTypes()	ResponseEntity<List<EventType>>

