

# 露营系统数据库设计表及 API 接口文档

这是根据你的要求修改后的完整版《露营系统数据库设计表及 API 接口文档》。

## 主要优化点：

- 逻辑简化：**移除了超卖并发锁、超时自动释放等复杂逻辑，完全专注于“业务闭环”。
- 价格计算：**修正了冗余，`/booking/create` 接口不再接收价格参数，而是由后端根据数据库当前价格重新计算，确保数据安全与一致性。
- 高级特性体现：**显式增加了视图 (View) 用于财务报表，触发器 (Trigger) 用于审计日志，索引 (Index) 用于时间段查询优化，并在逻辑说明中强调了事务 (Transaction) 控制。
- 新增接口：**增加了 `/booking/pay` 模拟支付接口，闭环订单状态流转。

# 露营系统数据库设计表及 API 接口文档

## 一、数据库设计 (Database Design)

### 1. UserTable (用户表)

字段名	类型	属性 / 说明
userID	BigInt	Unsigned, not null, primary_key, auto_increment
username	varchar(255)	not null, unique
password	varchar(255)	not null
phone	varchar(20)	not null, unique
role	varchar(20)	default 'USER' (USER: 普通用户 , ADMIN: 管理员)
createTime	TIMESTAMP	default CURRENT_TIMESTAMP

### 2. SiteTypeTable (营位 / 房型类型表)

字段名	类型	属性 / 说明
typeID	BigInt	Unsigned, not null, primary_key, auto_increment
typeName	varchar(255)	not null (如: "星空泡泡屋", "草地营位")
billingMode	varchar(20)	not null (核心字段: 'DAY'为日营, 'NIGHT'为过夜)
basePrice	DECIMAL(10,2)	not null (平日基础价格)
capacity	INT	default 2 (容纳人数)
coverImg	varchar(512)	default null (图片 URL)

### 3. SiteTable (具体营位 / 房间表)

字段名	类型	属性 / 说明
siteID	BigInt	Unsigned, not null, primary_key, auto_increment
siteNo	varchar(50)	not null (编号, 如 "A01", "Room-202")
status	TINYINT(1)	default 1 (1: 正常, 0: 维护中)
typeID	BigInt	not null, foreign_key, reference SiteTypeTable(typeID)

### 4. DailyPriceTable (价格日历表 - 浮动定价)

字段名	类型	属性 / 说明
priceID	BigInt	Unsigned, not null, primary_key, auto_increment
specificDate	DATE	not null (指定日期, 如 2025-10-01)
price	DECIMAL(10,2)	not null (当日特殊价格)
typeID	BigInt	not null, foreign_key, reference SiteTypeTable(typeID)

## 5. EquipmentTable (装备表)

字段名	类型	属性 / 说明
equipID	BigInt	Unsigned, not null, primary_key, auto_increment
equipName	varchar(255)	not null (如 "烧烤架", "天幕")
price	DECIMAL(10,2)	not null (单价 / 次)
totalStock	INT	not null (总物理库存)

## 6. BookingTable (订单表)

字段名	类型	属性 / 说明
bookingID	BigInt	Unsigned, not null, primary_key, auto_increment
checkInTime	DATETIME	not null (入住时间)
checkOutTime	DATETIME	not null (离店时间)
totalPrice	DECIMAL(10,2)	not null (订单总金额: 后端计算写入)
status	TINYINT	default 0 (0: 待支付, 1: 已支付, 2: 已取消)
createTime	TIMESTAMP	default CURRENT_TIMESTAMP
userID	BigInt	not null, foreign_key, reference UserTable(userID)
siteID	BigInt	not null, foreign_key, reference SiteTable(siteID)

## 7. BookingEquipTable (订单装备关联表)

字段名	类型	属性 / 说明
id	BigInt	Unsigned, not null, primary_key, auto_increment
quantity	INT	not null (该订单租赁的数量)
bookingID	BigInt	not null, foreign_key, reference BookingTable(bookingID)
equipID	BigInt	not null, foreign_key, reference EquipmentTable(equipID)

## 8. OperationLogTable (操作审计日志表 - 配合 Trigger 使用)

字段名	类型	属性 / 说明
logID	BigInt	Unsigned, not null, primary_key, auto_increment
operation	varchar(50)	not null (如: "INSERT_BOOKING", "CANCEL_BOOKING")
operatorID	BigInt	not null (操作的用户 ID)
description	varchar(255)	自动生成的描述信息
logTime	TIMESTAMP	default CURRENT_TIMESTAMP

## 二、数据库高级特性体现 (Advanced Features)

### 1. 索引设计 (Indexes)

为了提高时间段重叠查询的效率，必须建立以下索引：

- **idx\_booking\_time**: 在 `BookingTable` 上建立联合索引 `(checkInTime, checkOutTime)`，用于加速重叠时间段的检索。
- **idx\_booking\_status**: 在 `BookingTable` 上建立索引 `(status)`，用于快速过滤已取消的订单。
- **idx\_daily\_price**: 在 `DailyPriceTable` 上建立联合索引 `(typeID, specificDate)`，加速价格查询。

### 2. 视图设计 (Views)

- **View\_Daily\_Revenue (日营收报表视图)**
  - **定义**: 将 `BookingTable` 按照 `checkInTime` 的日期部分进行分组，聚合 `totalPrice`。
  - **作用**: 管理员无需编写复杂 SQL，直接 `SELECT * FROM View_Daily_Revenue` 即可查看每日收入情况，且数据实时更新。

### 3. 触发器设计 (Triggers)

- **Trigger\_Booking\_Audit**

- 触发时机: AFTER INSERT ON BookingTable
- 逻辑: 当有新订单生成时, 自动向 OperationLogTable 插入一条记录, 记录 operation 为 "NEW\_ORDER", description 为 "Order ID [ID] created for Site [ID]"。

## 4. 事务控制 (Transactions)

- 在后端代码 (Spring Boot) 的 /booking/create Service 层方法上添加 @Transactional 注解。
- 目的: 确保 "写入 BookingTable" 和 "写入 BookingEquipTable" 两个动作要么同时成功, 要么同时回滚, 防止出现 “有订单无装备” 或 “有装备无订单”的脏数据。

## 三、 API 接口文档 (API Documentation)

### 1. 全局数据结构

Java

```
1 public class Result<T> {  
2     private Integer code; // 1:成功, 0:失败  
3     private String msg;  
4     private T data;  
5 }
```

### 2. 用户模块 (User)

Method	URL	功能说明	请求数据格式	返回数据格式 (data 部分)	备注
POST	/user/register	用户注册	typescript <pre>{             username: String,             password: String,             phone: String           }</pre>	null	
POST	/user/login	用户登录	typescript <pre>{             username: String,             password: String           }</pre>	typescript <pre>{             userId: BigInteger,             username: String,             token: String,             role: String           }</pre>	返回 Token 用于后续鉴权

### 3. 资源与查询模块 (Resource)

Method	URL	功能说明	请求数据格式	返回数据格式 (data 部分)	备注
GET	/type/list	获取房型列表	无	typescript <pre>List&lt;SiteTypeItem&gt; [             {               typeId: BigInteger,               typeName: String,               coverImg: String,               billingMode: String,               basePrice: BigDecimal             }, ...           ]</pre>	
GET	/type/calendar	获取价格日历	typescript <pre>{             typeId: BigInteger,             startDate: String,             endDate: String           }</pre>	typescript <pre>List&lt;DailyPrice&gt; [             {               date: String,               price: BigDecimal             }, ...           ]</pre>	
GET	/equip/list	获取装备列表	无	typescript <pre>List&lt;EquipItem&gt; [             {               equipId: BigInteger,               equipName: String,               price: BigDecimal,               totalStock: Integer             }, ...           ]</pre>	

## 4. 预订与订单模块 (Booking) - 核心逻辑

Method	URL	功能说明	请求数据格式	返回数据格式 (data 部分)	备注
POST	/booking /check	预订前试算与检查 检查营位和装备是否有库存，并计算价格展示给用户	<pre>typescript {   typeId: BigInteger,   checkIn: String,   checkOut: String,   equipments: [     {       equipId: 1,       count: 2     }   ] }</pre>	<pre>typescript {   isAvailable: Boolean,   msg: String,   totalPrice: BigDecimal,   priceDetail: String }</pre>	纯查询操作，不锁库存
POST	/booking /create	提交订单 开启事务：1. 再次校验库存 2. 重新计算总价 3. 自动分配 SiteID 4. 落库	<pre>typescript {   userId: BigInteger,   typeId: BigInteger,   checkIn: String,   checkOut: String,   equipments: [     {       equipId: 1,       count: 2     },     {       guestName: String,       guestPhone: String     }   ] }</pre>	<pre>typescript {   bookingId: BigInteger,   siteNo: String,   totalPrice: BigDecimal }</pre>	核心逻辑：若无可用营位或装备库存不足，抛出异常，事务回滚
POST	/booking /pay	支付订单 模拟支付成功	<pre>typescript {   bookingId: BigInteger }</pre>	<pre>typescript {   status: Integer // 1 }</pre>	修改 status 0 -> 1
GET	/booking /my	我的订单列表	<pre>typescript {   userId: BigInteger,   status: Integer }</pre>	<pre>typescript List&lt;BookingItem&gt; [   {     bookingId: BigInteger,     typeName: String,     siteNo: String,     checkIn: String,     checkOut: String,     status: Integer,     totalPrice: BigDecimal   } ]</pre>	
POST	/booking /cancel	取消订单	<pre>typescript {   bookingId: BigInteger }</pre>	null	修改 status -> 2 库存自动释放

## 5. 管理员模块 (Admin)

Method	URL	功能说明	请求数据格式	返回数据格式 (data 部分)	备注
POST	/admin/pricelist/set	设置浮动价格	<pre>typescript {     typeId: BigInteger,     dates: List&lt;String&gt;,     price: BigDecimal }</pre>	null	
GET	/admin/report/daily	获取营收报表 使用 View 查询	<pre>typescript {     start: String,     end: String }</pre>	<pre>typescript List&lt;ReportItem&gt; [     {         date: String,         revenue: BigDecimal     } ]</pre>	基于 View_Daily_Revenue

## 四、核心逻辑补充说明

### 1. 价格计算一致性 (Price Consistency)

- 后端不信任前端传递的价格。
- /booking/check 仅用于前端展示估算价格。
- /booking/create 必须在后端根据 DailyPriceTable 和 EquipmentTable 重新计算 totalPrice 并写入数据库。

### 2. 装备库存算法 (Equipment Availability)

- 定义：**装备是全局共享的，不绑定特定房间。
- 剩余可用量计算公式：**

## SQL

```
1 剩余量 = EquipmentTable.totalStock -  
2 (  
3   SELECT SUM(quantity)  
4   FROM BookingEquipTable be  
5   JOIN BookingTable b ON be.bookingID = b.bookingID  
6   WHERE be.equipID = ?  
7   AND b.status != 2 -- 排除已取消订单  
8   AND NOT (b.checkOutTime <= ? OR b.checkInTime >= ?) -- 时间段存在交集  
9 )
```

- 在创建订单事务中，若任意装备的 `剩余量 < 请求量`，则抛出异常，回滚整个订单。

### 3. 营位自动分配 (Auto-Assign Site)

- 系统根据用户选择的 `typeId` 和时间段，查找该时间段内未被占用的 `siteID`。
- 若找到多个，随机或按顺序选取一个分配给订单；若找不到，则提示“满房”。