

中易开工盛典双端互动系统 · 部署手册

系统名称：中易物联网集团 2026 开工盛典互动系统（zeiot-checkin-mobile） 版本：
v3.0 | 更新时间：2026-02-25 架构说明：单一 Node.js 应用，手机端（/）与大屏端
（/bigscreen）通过路由区分，WebSocket 实现双端实时互动。技术栈：React 19 +
Tailwind CSS 4 + Express 4 + tRPC 11 + Drizzle ORM + MySQL 8 + WebSocket

目录

- [1. 系统架构概览](#)
- [2. 环境要求](#)
- [3. 服务器环境准备](#)
- [4. 获取代码](#)
- [5. 数据库配置](#)
- [6. 环境变量配置](#)
- [7. 安装依赖与构建](#)
- [8. 数据库初始化与迁移](#)
- [9. PM2 进程管理](#)
- [10. Nginx 反向代理](#)
- [11. SSL 证书配置](#)
- [12. 防火墙设置](#)
- [13. Docker 部署方案（可选）](#)
- [14. 管理员账号设置](#)
- [15. 上线前检查清单](#)
- [16. 常见问题排查](#)
- [17. 活动当天操作指南](#)

1. 系统架构概览

本系统为**单一代码库双端应用**，手机端与大屏端共用同一套后端服务，通过路由区分前端展示：

```
zeiot-checkin-mobile/
├── client/                      # 前端 (React 19 + Tailwind 4 + Vite 7)
│   └── src/pages/
│       ├── Home.tsx              # 手机端首页 (活动预热倒计时)
│       ├── Register.tsx          # 活动报名
│       ├── Checkin.tsx           # AI签到
│       ├── Schedule.tsx          # 活动日程
│       ├── Awards.tsx            # 荣誉殿堂
│       ├── Quiz.tsx              # AI知识问答
│       ├── WishCard.tsx          # 心愿卡
│       ├── Profile.tsx           # 个人中心
│       ├── BigScreen.tsx          # 大屏端展示 (投屏专用)
│       └── Admin.tsx              # 管理员后台
├── server/                      # 后端 (Express 4 + tRPC 11)
│   ├── routers.ts                # tRPC API 路由
│   ├── db.ts                      # 数据库查询函数
│   └── _core/
│       ├── index.ts              # 服务器入口 (含 WebSocket)
│       ├── llm.ts                 # AI 大模型调用
│       └── imageGeneration.ts    # AI 图像生成
└── drizzle/                     # 数据库 Schema 和迁移文件
    ├── schema.ts                # 表结构定义
    ├── 0000_*.sql                # 初始迁移
    ├── 0001_*.sql                # 注册表迁移
    └── 0002_*.sql                # 补充迁移
└── dist/                         # 构建产物 (生产环境)
    ├── public/                   # 前端静态文件
    └── index.js                  # 后端入口
```

技术栈一览：

层级	技术	说明
前端框架	React 19 + TypeScript	组件化 UI
样式系统	Tailwind CSS 4 + CSS 变量	喜庆红金主题
构建工具	Vite 7	快速热更新
后端框架	Express 4 + tRPC 11	类型安全 API
数据库 ORM	Drizzle ORM	轻量级 MySQL 操作
数据库	MySQL 8.0 / TiDB	关系型数据存储
实时通信	WebSocket (ws)	双端实时互动
AI 集成	Manus Forge API	LLM + 图像生成
认证	Manus OAuth 2.0	统一身份认证

2. 环境要求

组件	最低版本	推荐版本	备注
Node.js	18.x	22.x LTS	必须支持 ES Modules
pnpm	9.x	10.4.x	项目指定包管理器
MySQL	5.7	8.0+	或 TiDB 兼容版本
Nginx	1.18	1.24+	反向代理和 SSL 终止
操作系统	Ubuntu 20.04	Ubuntu 22.04 LTS	推荐
内存	1 GB	2 GB+	AI 功能需要更多内存
磁盘	5 GB	10 GB+	含日志和构建产物

3. 服务器环境准备

3.1 安装 Node.js (使用 nvm)

```
# 安装 nvm
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
source ~/.bashrc

# 安装并使用 Node.js 22 LTS
nvm install 22
nvm use 22
nvm alias default 22

# 验证
node -v    # 应输出 v22.x.x
```

3.2 安装 pnpm

```
npm install -g pnpm@10.4.1
pnpm -v    # 应输出 10.4.1
```

3.3 安装 PM2

```
npm install -g pm2
pm2 -v
```

3.4 安装 Nginx

```
sudo apt update
sudo apt install -y nginx
sudo systemctl enable nginx
sudo systemctl start nginx
```

3.5 安裝 MySQL 8.0

```
sudo apt install -y mysql-server  
sudo systemctl enable mysql  
sudo systemctl start mysql  
  
# 安全初始化  
sudo mysql_secure_installation
```

4. 获取代码

方式一：从 GitHub 克隆

```
git clone https://github.com/your-org/zeiot-checkin-mobile.git  
/var/www/zeiot-checkin-mobile  
cd /var/www/zeiot-checkin-mobile
```

方式二：从 Manus 平台导出

在 Manus 管理界面 → **Code** 面板 → 点击「Download all files」下载压缩包，上传至服务器后解压：

```
# 上传后在服务器执行  
sudo mkdir -p /var/www/zeiot-checkin-mobile  
sudo chown -R $USER:$USER /var/www/zeiot-checkin-mobile  
unzip zeiot-checkin-mobile.zip -d /var/www/zeiot-checkin-mobile  
cd /var/www/zeiot-checkin-mobile
```

5. 数据库配置

5.1 创建数据库和用户

```
-- 以 root 身份登录 MySQL
sudo mysql -u root

-- 创建数据库
CREATE DATABASE zeiot_checkin
    CHARACTER SET utf8mb4
    COLLATE utf8mb4_unicode_ci;

-- 创建专用用户
CREATE USER 'zeiot'@'localhost' IDENTIFIED BY 'your_strong_password_here';

-- 授权
GRANT ALL PRIVILEGES ON zeiot_checkin.* TO 'zeiot'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

5.2 验证连接

```
mysql -u zeiot -p -e "SELECT 'Connected!' AS status;"
# 输入密码后应看到 Connected!
```

5.3 数据库连接字符串格式

```
mysql://zeiot:your_strong_password_here@localhost:3306/zeiot_checkin
```

云数据库说明：如使用阿里云 RDS、腾讯云 CDB 或 TiDB Cloud，将 `localhost:3306` 替换为对应的主机和端口，并确保服务器公网 IP 已加入数据库白名单。

6. 环境变量配置

在项目根目录创建 .env 文件（**切勿提交到代码仓库**）：

```
cd /var/www/zeiot-checkin-mobile  
nano .env
```

填入以下内容：

```
# ===== 运行环境 =====  
NODE_ENV=production  
PORT=3000  
  
# ===== 数据库 =====  
DATABASE_URL=mysql://zeiot:your_strong_password_here@localhost:3306/zeiot_checkin  
  
# ===== 认证密钥 =====  
# 建议使用 openssl rand -base64 32 生成  
JWT_SECRET=your_random_jwt_secret_at_least_32_chars  
  
# ===== Manus OAuth =====  
VITE_APP_ID=your_manus_app_id  
OAUTH_SERVER_URL=https://api.manus.im  
VITE_OAUTH_PORTAL_URL=https://manus.im  
OWNER_OPEN_ID=your_owner_open_id  
OWNER_NAME=管理员姓名  
  
# ===== Manus AI API (AI 功能必填) =====  
BUILT_IN_FORGE_API_URL=https://api.manus.im/forge  
BUILT_IN_FORGE_API_KEY=your_server_side_forge_api_key  
VITE_FRONTEND_FORGE_API_KEY=your_frontend_forge_api_key  
VITE_FRONTEND_FORGE_API_URL=https://api.manus.im/forge
```

环境变量说明：

变量名	必填	说明
DATABASE_URL	✓	MySQL 连接字符串
JWT_SECRET	✓	Session 签名密钥，32 位以上随机字符串
VITE_APP_ID	✓	Manus OAuth 应用 ID
OAUTH_SERVER_URL	✓	Manus OAuth 后端地址
VITE_OAUTH_PORTAL_URL	✓	Manus 登录门户地址
OWNER_OPEN_ID	✓	管理员的 Manus OpenID
OWNER_NAME	✓	管理员姓名
BUILT_IN_FORGE_API_URL	✓	Manus AI API 地址（服务端）
BUILT_IN_FORGE_API_KEY	✓	Manus AI API 密钥（服务端，严格保密）
VITE_FRONTEND_FORGE_API_KEY	✓	Manus AI API 密钥（前端）
VITE_FRONTEND_FORGE_API_URL	✓	Manus AI API 地址（前端）
PORT	✗	服务端口，默认 3000

生成安全 JWT_SECRET：

```
# 方式一：openssl
openssl rand -base64 32

# 方式二：Node.js
node -e "console.log(require('crypto').randomBytes(32).toString('hex'))"
```

7. 安装依赖与构建

```
cd /var/www/zeiot-checkin-mobile

# 安装生产依赖 (锁定版本)
pnpm install --frozen-lockfile

# 构建生产版本 (同时构建前端和后端)
pnpm build

# 验证构建产物
ls dist/
# 应看到：
#   public/   ← 前端静态文件 (含 index.html 和 assets/)
#   index.js  ← 后端入口文件
```

构建说明： `pnpm build` 执行 `vite build` (前端) 和 `esbuild` (后端) 两个步骤，前端产物输出到 `dist/public/`，后端产物为 `dist/index.js`。

8. 数据库初始化与迁移

8.1 执行 Schema 迁移 (创建表结构)

```
cd /var/www/zeiot-checkin-mobile

# 方式一：使用 drizzle-kit 自动迁移 (推荐)
pnpm drizzle-kit migrate

# 方式二：手动按顺序执行 SQL 文件
mysql -u zeiot -p zeiot_checkin < drizzle/0000_noisy_power_pack.sql
mysql -u zeiot -p zeiot_checkin < drizzle/0001_steady_gargoyle.sql
mysql -u zeiot -p zeiot_checkin < drizzle/0002_empty_slipstream.sql
```

8.2 验证表结构

```
mysql -u zeiot -p zeiot_checkin -e "SHOW TABLES;"
```

应看到以下表：

表名	用途
users	用户账户 (OAuth 登录)
registrations	活动报名信息
checkins	签到记录
wish_cards	心愿卡内容
awards	奖项定义
award_winners	获奖记录
quiz_questions	AI 知识问答题库 (20 题)
quiz_answers	答题记录
lottery_events	抽奖活动
lottery_results	抽奖结果
event_config	活动配置

8.3 插入初始数据

```
-- 连接数据库
mysql -u zeiot -p zeiot_checkin

-- 插入活动基本配置
INSERT INTO event_config (configKey, configValue) VALUES
('event_name', '2026开工盛典'),
('event_date', '2026-03-01'),
('event_location', '中易物联集团总部'),
('event_theme', 'AI智启·同心聚力·焕新出发'),
('checkin_open_time', '2026-03-01T08:00:00+08:00')
ON DUPLICATE KEY UPDATE configValue = VALUES(configValue);

-- 插入奖项
INSERT INTO awards (name, description, icon, `order`) VALUES
('AI效率革命奖', '年度AI工具应用最佳实践，引领团队效率提升的先锋', '🤖', 1),
('年度优秀员工奖', '综合表现突出，为公司发展做出卓越贡献的员工', '⭐', 2),
('最佳团队协作奖', '跨部门协作典范，推动项目高效落地的团队', '🤝', 3),
('创新突破奖', '勇于创新，推动业务模式或技术方案重大突破', '💡', 4),
('客户之星奖', '客户服务和满意度方面表现卓越的员工', '🌟', 5)
ON DUPLICATE KEY UPDATE name = VALUES(name);

-- 插入抽奖活动
INSERT INTO lottery_events (name, description, prizePool, maxWinners) VALUES
('现金盲盒大作战', '随机抽取幸运员工，赢取神秘现金盲盒', 500, 5),
('AI幸运大抽奖', '年度压轴大奖，最高奖金', 1000, 1)
ON DUPLICATE KEY UPDATE name = VALUES(name);
```

8.4 数据备份与恢复

```
# 备份 (活动前必做)
mysqldump -u zeiot -p zeiot_checkin > \
backup_zeiot_$(date +%Y%m%d_%H%M%S).sql

# 恢复
mysql -u zeiot -p zeiot_checkin < backup_zeiot_20260301_080000.sql
```

9. PM2 进程管理

9.1 创建 PM2 配置文件

```
cat > /var/www/zeiot-checkin-mobile/ecosystem.config.js << 'EOF'
module.exports = {
  apps: [
    {
      name: 'zeiot-checkin',
      script: './dist/index.js',
      cwd: '/var/www/zeiot-checkin-mobile',
      instances: 1,
      exec_mode: 'fork',
      // 从 .env 文件加载环境变量
      env_file: '.env',
      env: {
        NODE_ENV: 'production',
        PORT: 3000,
      },
      // 日志配置
      out_file: '/var/log/zeiot-checkin/out.log',
      error_file: '/var/log/zeiot-checkin/error.log',
      log_date_format: 'YYYY-MM-DD HH:mm:ss',
      // 自动重启策略
      watch: false,
      max_memory_restart: '512M',
      restart_delay: 3000,
      max_restarts: 10,
    },
  ],
};

EOF
```

9.2 创建日志目录

```
sudo mkdir -p /var/log/zeiot-checkin
sudo chown -R $USER:$USER /var/log/zeiot-checkin
```

9.3 启动应用

```
cd /var/www/zeiot-checkin-mobile
pm2 start ecosystem.config.cjs

# 保存进程列表（重启服务器后自动恢复）
pm2 save

# 设置开机自启（按提示执行输出的 sudo 命令）
pm2 startup
```

9.4 常用 PM2 命令

pm2 status	# 查看所有进程状态
pm2 logs zeiot-checkin	# 查看实时日志
pm2 logs zeiot-checkin --lines 100	# 查看最近 100 行日志
pm2 restart zeiot-checkin	# 重启应用
pm2 stop zeiot-checkin	# 停止应用
pm2 reload zeiot-checkin	# 零停机重载（更新代码后使用）
pm2 delete zeiot-checkin	# 删除进程
pm2 monit	# 实时监控面板

9.5 更新部署流程

```
cd /var/www/zeiot-checkin-mobile

# 拉取最新代码
git pull origin main

# 重新安装依赖（如有变化）
pnpm install --frozen-lockfile

# 重新构建
pnpm build

# 零停机重载
pm2 reload zeiot-checkin
```

10. Nginx 反向代理

10.1 创建站点配置

```
sudo nano /etc/nginx/sites-available/zeiot-checkin
```

填入以下配置（将 `your-domain.com` 替换为实际域名）：

```
# HTTP → HTTPS 重定向
server {
    listen 80;
    listen [::]:80;
    server_name your-domain.com www.your-domain.com;
    return 301 https://$host$request_uri;
}

# HTTPS 主配置
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name your-domain.com www.your-domain.com;

    # SSL 证书 (见第 11 节)
    ssl_certificate      /etc/letsencrypt/live/your-domain.com/fullchain.pem;
    ssl_certificate_key  /etc/letsencrypt/live/your-domain.com/privkey.pem;
    ssl_protocols        TLSv1.2 TLSv1.3;
    ssl_ciphers          HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

    # 安全响应头
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";
    add_header X-XSS-Protection "1; mode=block";
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains";

    # 上传大小限制 (AI 头像生成需要)
    client_max_body_size 50m;

    # ===== WebSocket 代理 (双端实时互动核心) =====
    # 必须在其他 location 之前配置
    location /ws {
        proxy_pass http://127.0.0.1:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        # WebSocket 长连接，设置较长超时 (24小时)
        proxy_read_timeout 86400s;
        proxy_send_timeout 86400s;
        proxy_connect_timeout 60s;
    }
}
```

```

# ===== API 代理 =====
location /api/ {
    proxy_pass http://127.0.0.1:3000;
    proxy_http_version 1.1;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_connect_timeout 60s;
    proxy_send_timeout 120s;
    proxy_read_timeout 120s;
}

# ===== 前端静态资源 (长期缓存) =====
location /assets/ {
    proxy_pass http://127.0.0.1:3000;
    add_header Cache-Control "public, max-age=31536000, immutable";
}

# ===== 其他所有请求 (前端路由 SPA) =====
location / {
    proxy_pass http://127.0.0.1:3000;
    proxy_http_version 1.1;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
}

```

10.2 启用配置并重载

```

# 创建软链接
sudo ln -s /etc/nginx/sites-available/zeiot-checkin \
           /etc/nginx/sites-enabled/zeiot-checkin

# 测试配置语法 (必须无错误才能继续)
sudo nginx -t

# 重载 Nginx
sudo systemctl reload nginx

```

11. SSL 证书配置

方式一：Let's Encrypt 免费证书（推荐）

```
# 安装 Certbot
sudo apt install -y certbot python3-certbot-nginx

# 申请证书（自动修改 Nginx 配置）
sudo certbot --nginx -d your-domain.com -d www.your-domain.com

# 验证自动续期
sudo certbot renew --dry-run
```

方式二：自有证书

将证书文件上传至服务器，修改 Nginx 配置中的证书路径：

```
ssl_certificate      /path/to/your/fullchain.pem;
ssl_certificate_key  /path/to/your/privkey.pem;
```

12. 防火墙设置

```
# 开放必要端口
sudo ufw allow ssh          # 保留 SSH 访问（重要！）
sudo ufw allow 80/tcp        # HTTP
sudo ufw allow 443/tcp       # HTTPS
sudo ufw enable

# 验证规则
sudo ufw status
```

重要：应用端口 3000 无需对外开放，Nginx 作为反向代理处理所有外部请求，3000 端口仅在服务器内部使用。

13. Docker 部署方案（可选）

如需使用 Docker 容器化部署，可使用以下配置：

13.1 Dockerfile

```
FROM node:22-alpine AS builder
WORKDIR /app
COPY package.json pnpm-lock.yaml ./
RUN npm install -g pnpm@10.4.1 && pnpm install --frozen-lockfile
COPY . .
RUN pnpm build

FROM node:22-alpine AS runner
WORKDIR /app
COPY --from=builder /app/dist ./dist
COPY --from=builder /app/node_modules ./node_modules
COPY --from=builder /app/package.json ./
EXPOSE 3000
CMD ["node", "dist/index.js"]
```

13.2 docker-compose.yml

```
version: '3.8'
services:
  app:
    build: .
    ports:
      - "3000:3000"
    env_file: .env
    environment:
      NODE_ENV: production
    depends_on:
      db:
        condition: service_healthy
    restart: unless-stopped

  db:
    image: mysql:8.0
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: zeiot_checkin
      MYSQL_USER: zeiot
      MYSQL_PASSWORD: your_strong_password_here
    volumes:
      - mysql_data:/var/lib/mysql
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
      timeout: 20s
      retries: 10
    restart: unless-stopped

volumes:
  mysql_data:
```

```
# 启动
docker-compose up -d

# 查看日志
docker-compose logs -f app

# 停止
docker-compose down
```

14. 管理员账号设置

系统使用 Manus OAuth 登录，管理员需要先登录一次系统，然后通过以下方式设置权限：

方式一：通过 MySQL 命令行

```
-- 连接数据库
mysql -u zeiot -p zeiot_checkin

-- 查询所有用户 (找到目标用户)
SELECT id, openId, name, email, role FROM users;

-- 将目标用户设为管理员 (通过 openId)
UPDATE users SET role = 'admin' WHERE openId = 'target_user_open_id';

-- 或通过邮箱查找
UPDATE users SET role = 'admin' WHERE email = 'target@email.com';

-- 验证
SELECT name, email, role FROM users WHERE role = 'admin';
```

方式二：通过 Manus 平台 Database 面板

在 Manus 管理界面 → **Database** 面板 → `users` 表 → 找到目标用户行 → 将 `role` 字段修改为 `admin` → 保存。

验证管理员权限

登录系统后访问 `/admin` 路径，如能正常进入管理后台（显示概览/签到/颁奖/抽奖/心愿五个 Tab），则权限设置成功。

15. 上线前检查清单

在活动开始前，请逐项确认以下内容：

系统功能检查

检查项	验证方法	状态
首页倒计时正常	访问 /，确认日期为 2026-03-01，倒计时实时更新	<input type="checkbox"/>
签到按钮时间锁	3月1日08:00前显示「签到将于3月1日08:00开启」	<input type="checkbox"/>
活动报名正常	访问 /register，完成报名并收到成功提示	<input type="checkbox"/>
AI签到功能	访问 /checkin，完成签到并生成AI头像	<input type="checkbox"/>
活动日程页	访问 /schedule，确认上午场+下午场内容正确	<input type="checkbox"/>
AI知识问答	访问 /quiz，完成一道题并查看详细解析	<input type="checkbox"/>
心愿卡提交	访问 /wishcard，提交一条心愿	<input type="checkbox"/>
大屏端显示	访问 /bigscreen，确认签到墙、统计、心愿墙正常	<input type="checkbox"/>
WebSocket 实时同步	手机端签到后，大屏端头像实时出现	<input type="checkbox"/>
管理后台	访问 /admin，确认五个 Tab 均可用	<input type="checkbox"/>
AI颁奖词生成	管理后台 → 颁奖 → 生成一条颁奖词	<input type="checkbox"/>
抽奖功能	管理后台 → 抽奖 → 执行一次抽奖	<input type="checkbox"/>
AI分组功能	管理后台 → 抽奖 → AI分组 → 确认领导分组正确	<input type="checkbox"/>

数据准备检查

检查项	操作	状态
清空测试数据	管理后台 → 概览 → 一键清空测试数据（二次确认）	<input type="checkbox"/>
奖项信息正确	管理后台 → 颁奖 → 核对奖项名称和描述	<input type="checkbox"/>
活动日期正确	首页倒计时目标日期为 2026-03-01	<input type="checkbox"/>
管理员账号	目标用户 role=admin，可正常访问 /admin	<input type="checkbox"/>

技术环境检查

```
# 检查应用进程状态 (应为 online)
pm2 status

# 检查端口监听
ss -tlnp | grep 3000

# 检查 Nginx 状态
sudo systemctl status nginx

# 检查 SSL 证书有效期
sudo certbot certificates

# 检查数据库连接
mysql -u zeiot -p -e "SELECT COUNT(*) as user_count FROM
zeiot_checkin.users;"

# 检查应用日志 (无 ERROR 级别日志)
pm2 logs zeiot-checkin --lines 20
```

16. 常见问题排查

问题一：应用启动失败

```
# 查看详细错误日志  
pm2 logs zeiot-checkin --lines 50  
  
# 常见原因及解决方案：  
# 1. DATABASE_URL 配置错误  
#   → 检查 .env 文件，确认用户名、密码、数据库名正确  
#   → 测试：mysql -u zeiot -p zeiot_checkin -e "SELECT 1;"  
  
# 2. 端口 3000 被占用  
#   → 查找占用进程：lsof -i :3000  
#   → 修改 .env 中的 PORT=3001  
  
# 3. 依赖未安装或版本不匹配  
#   → 重新安装：pnpm install --frozen-lockfile  
  
# 4. 未执行构建  
#   → 重新构建：pnpm build
```

问题二：WebSocket 连接失败（实时互动不工作）

```
# 确认 Nginx 配置中 /ws 路径有正确的 WebSocket 升级头：  
# proxy_set_header Upgrade $http_upgrade;  
# proxy_set_header Connection "upgrade";  
  
# 测试 WebSocket 连接（需要安装 wscat）  
npm install -g wscat  
wscat -c wss://your-domain.com/ws  
# 应看到：Connected (press CTRL+C to quit)  
# 并收到：{"type": "CONNECTED", "data": {"time": ...}}
```

问题三：AI 功能报错

```
# 检查 Forge API 密钥配置  
grep "BUILT_IN_FORGE_API_KEY" /var/www/zeiot-checkin-mobile/.env  
  
# 查看 AI 相关错误日志  
pm2 logs zeiot-checkin | grep -i "forge\|llm\|image\|error"  
  
# 常见原因：  
# 1. API Key 未配置或已过期 → 在 Manus 控制台重新获取  
# 2. 网络无法访问 Manus API → 检查服务器出站网络
```

问题四：数据库连接失败

```
# 检查 MySQL 服务状态  
sudo systemctl status mysql  
  
# 测试数据库连接  
mysql -u zeiot -p -h localhost zeiot_checkin -e "SHOW TABLES;"  
  
# 检查数据库表是否已创建  
mysql -u zeiot -p zeiot_checkin -e "SHOW TABLES;"  
  
# 如果表不存在，重新执行迁移  
cd /var/www/zeiot-checkin-mobile  
pnpm drizzle-kit migrate
```

问题五：页面加载空白或报 404

```
# 确认构建产物存在  
ls /var/www/zeiot-checkin-mobile/dist/public/  
  
# 确认 NODE_ENV=production (生产模式下由 Express 提供静态文件)  
pm2 env zeiot-checkin | grep NODE_ENV  
  
# 查看 Nginx 错误日志  
sudo tail -50 /var/log/nginx/error.log
```

问题六：大屏端在投屏时显示异常

大屏端 /bigscreen 针对 1920×1080 分辨率优化。投屏时建议：

1. 使用 Chrome 浏览器，按 **F11** 全屏
2. 在 Chrome 地址栏输入 `chrome://settings/` → 外观 → 字体大小设为「中」
3. 如分辨率不是 1080p，可在 Chrome 开发者工具中设置缩放比例

17. 活动当天操作指南

活动前（3月1日 08:00 前）

```
# 1. 确认应用正常运行  
pm2 status    # 应显示 online  
  
# 2. 备份当前数据库（以防万一）  
mysqldump -u zeiot -p zeiot_checkin > backup_before_event.sql  
  
# 3. 清空所有测试数据（在管理后台操作）  
# 访问 https://your-domain.com/admin  
# → 概览 Tab → 点击「一键清空测试数据」→ 确认
```

活动中操作节点

时间	环节	操作	入口
09:00	签到开始	员工扫码访问，完成签到	手机访问首页
09:00	大屏投屏	打开大屏端，全屏展示	/bigscreen + F11
09:30	开场后	大屏自动轮播签到墙/心愿墙	自动
10:45	颁奖环节	生成颁奖词，大屏展示	/admin → 颁奖 Tab
14:00	游戏分组	执行 AI 随机分组	/admin → 抽奖 Tab → AI分组
16:00	抽奖环节	执行现金抽奖	/admin → 抽奖 Tab

颁奖操作步骤

1. 访问 /admin → 颁奖 Tab
2. 选择奖项，点击「AI生成颁奖词」
3. 等待 AI 生成（约 3-5 秒）
4. 点击「发送到大屏」，大屏端自动全屏展示颁奖词
5. 宣读完毕后，大屏端点击关闭或等待自动消失

抽奖操作步骤

1. 访问 /admin → 抽奖 Tab
2. 选择抽奖活动（现金盲盒 / 幸运大奖）
3. 点击「开始抽奖」，大屏端显示滚动动画
4. 点击「停止」，中奖结果实时显示在大屏

紧急重启

```
# 应用崩溃时立即重启
pm2 restart zeiot-checkin

# 如需完整重建（约 2-3 分钟）
cd /var/www/zeiot-checkin-mobile
pnpm build && pm2 restart zeiot-checkin
```

附录 A：手机端页面路由

路径	页面	功能
/	首页	活动预热倒计时、功能导航
/register	活动报名	填写姓名、部门、身份、饮食需求
/checkin	AI签到	签到并生成 AI 风格头像
/schedule	活动日程	上午场+下午场完整流程
/awards	荣誉殿堂	奖项展示与获奖名单
/quiz	AI知识问答	20 道 AI 知识题，含详细解析
/wish	心愿卡	填写并提交心愿，实时显示大屏
/profile	个人中心	签到状态、答题记录
/bigscreen	大屏展示	科技感 AI 大屏（投屏专用）
/admin	管理后台	签到/颁奖/抽奖/分组/心愿管理

附录 B：AI 功能说明

功能	实现方式	平均耗时
AI 头像生成	Manus Image Generation API	5-15 秒
AI 颁奖词生成	Manus LLM API (GPT-4 级别)	3-8 秒
AI 随机分组	Manus LLM API + 规则约束	2-5 秒
AI 知识问答	静态题库 (20 题，含解析)	即时

本手册适用于 zeiot-checkin-mobile v3.0，最后更新：2026-02-25。如有问题，请联系系统管理员。