**AirDrop 局域网文件共享系统设计说明文档**

**一、项目简介**

AirDrop 是一个基于 Spring Boot 和 WebSocket 的局域网文件共享系统，致力于为用户提供类似 Apple AirDrop 的无缝文件传输体验。系统支持自动发现局域网内设备、点对点大文件传输、实时进度反馈，并具备跨平台兼容性。

**二、系统架构**

1. 架构总览

前端：基于 HTML5、CSS3、JavaScript，采用 Thymeleaf 模板引擎渲染页面，通过 WebSocket 与后端实时通信。

后端：Spring Boot 3.2.0，核心功能包括设备注册、文件传输管理、WebSocket 消息处理。

通信协议：控制消息采用 JSON 格式，文件数据采用 WebSocket 二进制消息分块传输。

架构图建议

```

[用户A浏览器] ←→ [WebSocket] ←→ [Spring Boot后端] ←→ [WebSocket] ←→ [用户B浏览器]

```

所有设备通过WebSocket与后端建立连接，后端负责设备发现、消息转发、文件分块中转。

2. 主要模块

controller/DropController：页面路由控制，负责前端页面的访问入口。

handler/FileTransferHandler：WebSocket 处理器，负责消息分发、文件传输请求、数据分块收发。

service/DeviceRegistry：设备注册与管理，维护在线设备列表，支持设备上下线通知。

config/WebSocketConfig：WebSocket 端点配置，允许跨域访问。

config/WebSocketBinaryMessageSizeConfig：配置WebSocket消息缓冲区大小，支持大文件。

model/DeviceInfo：设备信息（ID、名称、会话）。

model/FileTransferMessage：消息体，支持多种类型（设备信息、传输请求、文件元数据等）。

**三、功能设计**

1. 设备发现与注册

新设备连接时自动分配唯一 ID 和昵称，注册到 DeviceRegistry。

设备上线/下线时，系统自动广播最新设备列表。

设备信息通过 `device\_info`、`device\_list` 消息同步。

2. 文件传输流程

1. 发送方选择目标设备，发起 `request\_transfer` 请求。

2. 接收方收到请求后可接受或拒绝（`transfer\_response`）。

3. 传输前发送文件元数据（`file\_metadata`），包括文件名、大小、类型、分块数、校验码等。

4. 文件通过 WebSocket 二进制分块（`file\_chunk`）传输，支持大文件断点续传。

5. 实时反馈传输进度、速度、剩余时间。

关键流程时序图建议

```

发送方 后端 接收方

| request\_transfer | |

|>| |

| | transfer\_request |

| |>|

| | |

| | <transfer\_response|

| <transfer\_response| |

| file\_metadata | |

|>| |

| file\_chunk(多次) | |

|>| |

```

3. 前端交互

自动发现并展示在线设备。

支持文件拖拽、选择发送。

展示传输状态、进度条、历史记录。

断点续传与失败重试提示。

**四、通信协议详细说明**

1. 控制消息（JSON）

`device\_info`：设备注册/上线

```json

{"type": "device\_info", "deviceId": "abc123", "deviceName": "设备A"}

```

`device\_list`：设备列表

```json

{"type": "device\_list", "deviceList": [{"deviceId": "abc123", "deviceName": "设备A"}, ...]}

```

`request\_transfer`：发起传输

```json

{"type": "request\_transfer", "sourceDeviceId": "abc123", "targetDeviceId": "def456", "fileName": "test.png", "fileSize": 123456}

```

`transfer\_response`：接收方响应

```json

{"type": "transfer\_response", "transferId": "uuid", "accepted": true}

```

`file\_metadata`：文件元数据

```json

{"type": "file\_metadata", "fileName": "test.png", "fileSize": 123456, "chunks": 10, "checksum": "md5..."}

```

`file\_chunk`：二进制分块（通过WebSocket二进制消息发送）

2. 异常与错误消息

`error`：

```json

{"type": "error", "message": "文件过大"}

```

**五、异常处理与健壮性**

连接断开自动注销设备，广播最新设备列表。

文件传输中断时支持重试与断点续传。

文件大小、类型校验，防止恶意上传。

传输过程中的异常通过 `error` 消息通知前端。

**六、安全性分析**

仅允许局域网访问，避免外网攻击。

WebSocket 端点可配置允许的来源（生产环境建议严格限制）。

文件传输采用分块与校验，防止数据篡改。

可扩展权限认证、传输加密（如WSS）、操作日志等。

**七、扩展性建议**

支持多文件并发传输。

支持历史记录与传输管理。

支持移动端适配与PWA。

支持用户自定义设备昵称、头像。

支持更丰富的传输状态与通知。

**八、目录结构简述**

`controller/`：页面路由控制

`handler/`：WebSocket 业务处理

`service/`：设备注册与管理

`model/`：数据模型

`config/`：WebSocket 及消息大小配置

`resources/`：静态资源与模板

**九、部署与运维**

主要配置项见 `application.yml`，如端口、文件大小限制、WebSocket 消息大小等。

构建命令：`mvn clean package`

运行命令：`java jar target/AirDrop1.0SNAPSHOT.jar`

局域网内任意设备通过浏览器访问 `http://[服务器IP]:8080` 即可使用。

日志级别可通过 `application.yml` 配置，便于问题追踪。

**十、参考与致谢**

参考 Apple AirDrop 用户体验

基于 Spring Boot 官方文档与社区最佳实践

感谢开源社区相关资料与灵感