package com.bookwhale.chatroom.docs;

import static org.springframework.restdocs.headers.HeaderDocumentation.headerWithName; import static org.springframework.restdocs.headers.HeaderDocumentation.requestHeaders; import static org.springframework.restdocs.mockmvc.MockMvcRestDocumentation.document; import static org.springframework.restdocs.operation.preprocess.Preprocessors.preprocessRequest; import static org.springframework.restdocs.operation.preprocess.Preprocessors.preprocessResponse; import static org.springframework.restdocs.operation.preprocess.Preprocessors.prettyPrint; import static org.springframework.restdocs.payload.PayloadDocumentation.fieldWithPath; import static org.springframework.restdocs.payload.PayloadDocumentation.requestFields; import static org.springframework.restdocs.payload.PayloadDocumentation.responseFields; import static org.springframework.restdocs.request.RequestDocumentation.parameterWithName; import static org.springframework.restdocs.request.RequestDocumentation.pathParameters;

import org.springframework.http.HttpHeaders; import org.springframework.restdocs.mockmvc.RestDocumentationResultHandler; import org.springframework.restdocs.payload.FieldDescriptor; import org.springframework.restdocs.payload.JsonFieldType;

public class ChatRoomDocumentations {

```
public static RestDocumentationResultHandler createChatRoom() {
    return document("chatRoom/createChatRoom",
        preprocessRequest(prettyPrint()),
        requestHeaders(
            headerWithName(HttpHeaders.AUTHORIZATION).description("□□ □□ □□□ □□ JWT")
        ),
        requestFields(
            fieldWithPath("articleId").type(JsonFieldType.NUMBER).description("□□□
ID"),
            fieldWithPath("sellerId").type(JsonFieldType.NUMBER).description("□□□ ID")
        )
    );
}
```

```
public static RestDocumentationResultHandler findChatRooms() {
    FieldDescriptor[] response = new FieldDescriptor[]{
        fieldWithPath("roomId").type(JsonFieldType.NUMBER).description("□□□ ID"),
        fieldWithPath("articleId").type(JsonFieldType.NUMBER).description("□□□ ID"),
        fieldWithPath("articleTitle").type(JsonFieldType.STRING).description("□□□ □□"),
        fieldWithPath("articleImage").type(JsonFieldType.STRING).description("□□□ □□
□□□"),
        fieldWithPath("opponentIdentity").type(JsonFieldType.STRING).description("□□□
□□□"),
        fieldWithPath("opponentProfile").type(JsonFieldType.STRING).description("□□□ □□□
□□"),
        fieldWithPath("opponentDelete").type(JsonFieldType.BOOLEAN).description(
            "□□□□ □□ □□□ □□"),

fieldWithPath("lastContent").type(JsonFieldType.STRING).optional().description(
            "□□□□ □□□ □□□ (* □□ □□□□ null)")
    };
```

```
    return document("chatRoom/findChatRooms",
        preprocessResponse(prettyPrint()),
        requestHeaders(
            headerWithName(HttpHeaders.AUTHORIZATION).description("□□ □□ □□□ □□ JWT")
        ),
        responseFields(fieldWithPath("[]").description("An arrays of chatRoom"))
            .andWithPrefix("[].", response)
    );
}
```

```
    public static RestDocumentationResultHandler deleteChatRoom() {
        return document("chatRoom/deleteChatRoom",
            preprocessRequest(prettyPrint()),
            requestHeaders(
                headerWithName(HttpHeaders.AUTHORIZATION).description("□□ □□ □□□ □□ JWT")
            ),
            pathParameters(
                parameterWithName("roomId").description("□□□ ID")
            )
        );
    }
}
```