# Client-side Authentication with Joi Validation

This document explains how to perform basic client-side authentication in a TypeScript project using Joi for validation.

## 1. Prerequisites

- TypeScript must be configured.  
- You should have installed Joi earlier in the project.  
- SignalR client library should be available.  
- A working server endpoint for JWT authentication is required.

## 2. Joi Validation Schema

Create a file (e.g., `authValidator.ts`) with the following schema:

import Joi from "joi";  
  
export const authSchema = Joi.object({  
 username: Joi.string().min(3).required().messages({  
 "string.empty": "Username is required",  
 "string.min": "Username must be at least 3 characters long"  
 }),  
 password: Joi.string().min(4).required().messages({  
 "string.empty": "Password is required",  
 "string.min": "Password must be at least 4 characters long"  
 }),  
});

## 3. Using the Validator

In your `auth-client.ts`, validate credentials before making the authentication request:

import { authSchema } from "./authValidator";  
  
const credentials = { username: "user", password: "pass" };  
const { error } = authSchema.validate(credentials);  
if (error) {  
 console.error("Validation error:", error.details.map(d => d.message).join("; "));  
 process.exit(1);  
}

## 4. Why Client-side Validation?

Validating data on the client ensures only well-formed requests are sent to the server, which reduces server-side processing and improves user feedback.