**Deployment Service**

Prerequisites:

* Install MySQL 8.0
  1. <https://dev.mysql.com/downloads/installer/>
* Install .NET 8.0 Hosting Bundler
  1. https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-8.0.15-windows-hosting-bundle-installer
* Enable IIS
  1. <https://winsides.com/enable-internet-information-services-iis-in-windows-11/>

Running the REST Service

**Option #1 (Run the VS Studio code)**

* Clone REST/SourceCode/ folder
* Open the InstantMessagingPublisher.sln
* Run the Project
* This will start
  + <https://localhost:7189/swagger/index.html>
* If you are running MySQL on another PC other than local
  + Adjust the app.setting.json file with the location of MySQL server

**Option #2 (Deploy to IIS)**

* Clone REST/Dist folder
* Setup a new Website on IIS to run under port 7189 with HTTPS protocol
* Point file directory to the Dist folder
* Ensure that IIS\_USRS has read/write access to the Dist folder
* This will enable you to browse direct
* If you are running MySQL on another PC other than local
  + Adjust the app.setting.json file with the location of MySQL server

**Deployment Frontend**

# MyChat Frontend

This project is the frontend for the MyChat messaging system. It is built with React and Axios, designed to interact with a backend API. This guide will help you get the application running on your remote machine.

## Prerequisites

- \*\*Node.js\*\* (v14.x or later) and npm

[Download Node.js] (<https://nodejs.org>)

- \*\*Git\*\* (basic usage assumed)

## Setup and Running the Application

Follow these steps to get the MyChat frontend running on your remote machine:

1. Clone the Repository

Clone the repository to your remote machine using Git:

(should be done already)

2. Navigate to the Frontend Directory

Change into the project's frontend folder:

cd Frontend (from project root)

3. Install Dependencies

Install all required dependencies by running:

npm install (make sure node.js command set up)

This command reads your package.json file and installs React, Axios, react-router-dom, and other necessary packages.

Afterwards, enter this command to install NODE MODULES:

npm install react-scripts

DO NOT COMMIT THIS FOLDER< IT IS IN THE GITIGNORE FILE

4. Run the Application in Development Mode

Start the React development server:

npm start

The application will start on port 3000 by default and can be accessed at:

Locally: http://localhost:3000

Remotely: Use your server's IP address (e.g., http://<your-server-ip>:3000)

Note: Make sure port 3000 is open in your firewall settings if you are accessing the app remotely.

5. Create a Production Build (Optional)

If you need to create an optimized production build, run:

npm run build

This will generate a build/ folder containing the production-ready files.

By following these steps, you'll have the MyChat frontend up and running remotely. This setup makes it easy to test and develop the application before integrating it with the backend.

**Comparison Architecture Style**

We utilized same client for both the architecture, so that we can reuse the front-end components. The REST connectors provided interphase to back-end code, which would be replace by web sockets in the other implementations.