# 🧑‍💻 Guide for the React Frontend Developer

## 🧠 Objective

You need to build a Telegram-like interface in React that connects to the Django backend.

## 🔌 Backend Base URL

http://localhost:8000/api/

## 📡 API Endpoints

POST send\_code/ - Send Telegram login code  
POST verify\_code/ - Verify login and get session  
GET user\_info/ - Get Telegram user info  
GET chats/ - Get chat list  
POST send\_message/ - Send a message to chat  
GET history/<chat\_id>/ - Get message history of a chat

## 📤 Login Flow

1. User enters phone number → Call send\_code/

2. Receive phone\_code\_hash from response

3. Ask user to enter the code they received on Telegram

4. Call verify\_code/ with phone, code, and phone\_code\_hash

## 🪪 Session Handling

Session strings are stored by backend. Treat it as stateful post login.

## 💡 Real-Time Messages

Messages are stored via receiver.py in the backend.

You can poll history endpoint or add WebSocket later.

## 🛠️ Developer Setup

git clone <repo>  
cd telegram-project  
python -m venv venv  
venv\Scripts\activate  
pip install -r requirements.txt  
python manage.py migrate  
python manage.py createsuperuser  
python manage.py runserver  
python telegramchat/receiver.py

## 🚀 Running the Backend Project

Follow these steps to set up and run the backend project:  
  
1. Clone or extract the project files into your development environment.  
2. Ensure Python 3.10 or newer is installed.  
3. Create and activate a virtual environment (optional but recommended).  
4. Install dependencies manually using:  
 pip install django  
 pip install telethon  
  
5. Apply migrations to set up the database:  
 python manage.py makemigrations  
 python manage.py migrate  
  
6. Create a superuser for admin access (optional but recommended):  
 python manage.py createsuperuser  
  
7. Run the Django development server:  
 python manage.py runserver  
  
8. To start listening for incoming messages from logged-in Telegram users, run:  
 python telegramchat/receiver.py  
  
Ensure you replace the placeholders in settings.py with actual API credentials.