Chatbot Backend

3 Joke Generator — Send API Request → **4** Translation API — Get API Answer → **5** Translation Handler — Send API Answer → **4** Websocket Endpoint

3 Rage Handler — Generate random Answer → **4** Websocket Endpoint — Send Answer →

1 Client — Send Data → **2** Websocket Endpoint — Validate Object Type → **3** Translation Handler — Send API Request → **4** Translation API — Get API Answer → **5** Translation Handler — Send API Answer → **4** Websocket Endpoint — Send Answer → **5** Client
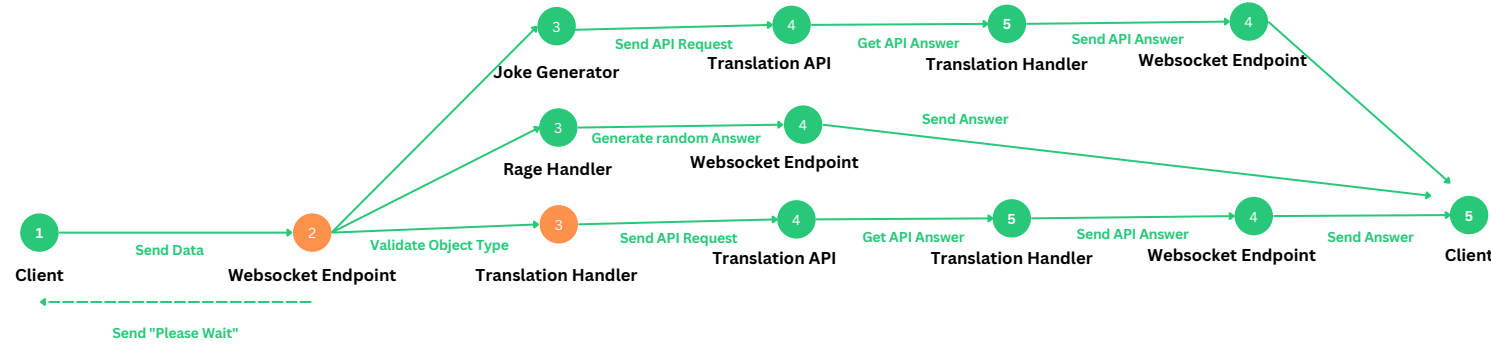
Send "Please Wait"

Websocket (Input)

- Detect requested Handler
- Pass Data to right Handler
- Respond to Client (Waiting/Error)

Translation Handler

- Filter wanted Language
- Pattern: LanguageInfo: TextData
- Examples:
  - Translate the following into german: I'm hungry
  - German: I'm hungry
  - Please in french: I'm hungry

Data from Frontend

```
{
"chatBot": "translationBot",
"message": "Translate the following into german: I'm hungry"
}
```

3rd Party API

```
.../translate?language=german&text=I%27m%20hungry
```

Translation Handler

```
Pattern:
before ":" -> scan for language
after ":" -> text to be translated
```

Data from Frontend

```
{
"chatBot": "weatherBot",
"message": "What's the weather for Stuttgart, tomorrow"
}
```

3rd Party API

```
.../weather?city=Stuttgart&time=tomorrow
```

Weather Handler

```
Pattern:
scan for known cities
scan for date terms -> if not: today
```