🚗 Skoda Chatbot API Documentation

Base URL

```
http://localhost:5000
```

📡 Endpoints

1. GET /greeting

Get the welcome message when user opens chat

Request:

```
GET /greeting
```

Response:

```
{
  "message": "Hello! 👋 Welcome to Skoda Customer Care!\n\nI'm your AI
assistant..."
}
```

---

2. POST /chat

Send user message and get bot response

Request:

```
POST /chat
Content-Type: application/json

{
  "message": "Tell me about Kushaq",
  "session_id": "user_123"  // Optional: Use for multi-user support
}
```

Response:

```
{
  "reply": "The Skoda Kushaq is our India-specific compact SUV available with
1.0L TSI (115 PS) and 1.5L TSI (150 PS) engines. Prices start from ₹10.89
Lakhs. Would you like to know about specific features?",
  "needs_escalation": false,
  "support_info": {
```

```
    "phone": "+91-1800-103-5000",
    "email": "customercare@skoda.co.in"
  }
}
```

When escalation is needed:

```
{
  "reply": "I'd be happy to help you book a test drive! For scheduling,
please contact our team directly.",
  "needs_escalation": true,
  "support_info": {
    "phone": "+91-1800-103-5000",
    "email": "customercare@skoda.co.in"
  }
}
```

---

3. GET /farewell

Get goodbye message when user ends chat

Request:

```
GET /farewell
```

Response:

```
{
  "message": "Thank you for choosing Skoda! 🚗👋\n\nIt was a pleasure assisting
you..."
}
```

---

4. POST /reset-session

Clear conversation history for a user

Request:

```
POST /reset-session
Content-Type: application/json

{
  "session_id": "user_123"
}
```

Response:

```
{
```

```
  "message": "Session reset successfully"
}
```

---

5. GET /health

Check if API is running

Request:

```
GET /health
```

Response:

```
{
  "status": "healthy",
  "service": "Skoda Chatbot API"
}
```

🎯 Integration Flow

1. When User Opens Chat

```
// GET /greeting
const response = await fetch('http://localhost:5000/greeting');
const data = await response.json();
displayMessage(data.message, 'bot');
```

2. When User Sends Message

```
// POST /chat
const response = await fetch('http://localhost:5000/chat', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    message: userInput,
    session_id: userId  // Use unique ID per user
  })
});

const data = await response.json();
displayMessage(data.reply, 'bot');

// If escalation needed, show contact info
if (data.needs_escalation) {
  showContactInfo(data.support_info);
}
```

3. When User Closes Chat

```
// GET /farewell
const response = await fetch('http://localhost:5000/farewell');
const data = await response.json();
displayMessage(data.message, 'bot');
```

📦 Example JavaScript Integration

```
class SkodaChatbot {
  constructor() {
    this.apiUrl = 'http://localhost:5000';
    this.sessionId = this.generateSessionId();
  }

  generateSessionId() {
    return 'user_' + Math.random().toString(36).substr(2, 9);
  }

  async getGreeting() {
    const response = await fetch(`${this.apiUrl}/greeting`);
    const data = await response.json();
    return data.message;
  }

  async sendMessage(message) {
    const response = await fetch(`${this.apiUrl}/chat`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        message: message,
        session_id: this.sessionId
      })
    });
    return await response.json();
  }

  async getFarewell() {
    const response = await fetch(`${this.apiUrl}/farewell`);
    const data = await response.json();
    return data.message;
  }

  async resetSession() {
    await fetch(`${this.apiUrl}/reset-session`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        session_id: this.sessionId
```

```
      })
    });
  }
}

// Usage
const chatbot = new SkodaChatbot();

// On page load
chatbot.getGreeting().then(msg => console.log(msg));

// On user message
chatbot.sendMessage("Tell me about Kushaq").then(response => {
  console.log(response.reply);
  if (response.needs_escalation) {
    console.log("Contact:", response.support_info);
  }
});

// On chat close
chatbot.getFarewell().then(msg => console.log(msg));
```

💼 Example React Integration

```
import { useState, useEffect } from 'react';

function SkodaChat() {
  const [messages, setMessages] = useState([]);
  const [input, setInput] = useState('');
  const API_URL = 'http://localhost:5000';
  const SESSION_ID = 'user_' + Math.random().toString(36).substr(2, 9);

  useEffect(() => {
    // Load greeting on mount
    fetch(`${API_URL}/greeting`)
      .then(res => res.json())
      .then(data => {
        setMessages([{ text: data.message, sender: 'bot' }]);
      });
  }, []);

  const sendMessage = async () => {
    if (!input.trim()) return;

    // Add user message
    setMessages(prev => [...prev, { text: input, sender: 'user' }]);

    // Send to API
    const response = await fetch(`${API_URL}/chat`, {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({
        message: input,
        session_id: SESSION_ID
```

```
      })
    });

    const data = await response.json();

    // Add bot response
    setMessages(prev => [...prev, {
      text: data.reply,
      sender: 'bot',
      escalation: data.needs_escalation,
      support: data.support_info
    }]);

    setInput('');
  };

  return (
    <div className="chat-container">
      <div className="messages">
        {messages.map((msg, i) => (
          <div key={i} className={`message ${msg.sender}`}>
            <p>{msg.text}</p>
            {msg.escalation && (
              <div className="contact-info">
                <p>📞 {msg.support.phone}</p>
                <p>📧 {msg.support.email}</p>
              </div>
            )}
          </div>
        ))}
      </div>
      <div className="input-area">
        <input
          value={input}
          onChange={(e) => setInput(e.target.value)}
          onKeyPress={(e) => e.key === 'Enter' && sendMessage()}
        />
        <button onClick={sendMessage}>Send</button>
      </div>
    </div>
  );
}
```

⚡Quick Setup for Frontend Dev

Step 1: Start the API

```
python skoda_api.py
```

Step 2: Test with cURL

```
# Test greeting
```

```
curl http://localhost:5000/greeting

# Test chat
curl -X POST http://localhost:5000/chat \
  -H "Content-Type: application/json" \
  -d '{"message": "Tell me about Kushaq", "session_id": "test123"}'

# Test farewell
curl http://localhost:5000/farewell

# Check health
curl http://localhost:5000/health
```

Step 3: Integrate in Frontend

Use the JavaScript/React examples above

🔧 Important Notes

1. **Session Management**:
   - Use unique `session_id` for each user
   - Same `session_id` maintains conversation context
   - Call `/reset-session` to clear history
2. **Escalation Handling**:
   - When `needs_escalation: true`, show contact info prominently
   - Suggest calling or emailing for bookings, emergencies, complex issues
3. **Error Handling**:
   - Always check response status
   - API returns error messages in `error` field
   - Show fallback message on errors
4. **CORS**:
   - Already enabled for all origins
   - Safe for local development
   - Update CORS settings for production

🚀 Production Deployment

When deploying:

1. Change `debug=False` (already set)
2. Use production WSGI server (gunicorn):
3. `pip install gunicorngunicorn -w 4 -b 0.0.0.0:5000 skoda_api:app`
4. Set environment variables:

5. `export GROQ_API_KEY=your_key_here`
6. Update CORS for your domain:
7. `CORS(app, origins=["https://yourdomain.com"])`

📞 Support

For API issues, contact the backend team.

That's it! Clean, simple, and ready to integrate! 🎉