Debugging the Frigate Mobile App

Viewing Logs on Android

Option 1: Use Logcat (Recommended)

```Bash Terminal

# View all logs from the app

adb logcat -s ReactNativeJS:V

# Or filter for our specific logs

adb logcat | grep “[FrigateAPI]|[AuthScreen]“
```

```
### Option 2: Enable Remote JS Debugging

1. Shake your device or press the menu button
2. Select "Debug" from the developer menu
3. Open Chrome DevTools at `chrome://inspect`
4. View console logs in the browser

### Option 3: Use React Native Debugger

Install the standalone app:
```Bash Terminal
brew install --cask react-native-debugger
```

Cloud Error Tracking with Sentry (Recommended)

Sentry automatically captures errors and sends them to the cloud for analysis.

Setup:

1. **Create free Sentry account**: https://sentry.io/signup/
2. **Create a new project**:
   - Select “React Native” as platform
   - Copy the DSN (looks like: `https://xxx@sentry.io/yyy`)
3. **Add DSN to your app**:
   ```Bash Terminal
   # Create .env file
   cp .env.example .env

# Edit .env and add your DSN
echo ‘EXPO_PUBLIC_SENTRY_DSN=https://your-dsn@sentry.io/project-id’ >> .env
```

1. **Rebuild the app**:
   ```
   Bash Terminal
      eas build --platform android --profile preview
   ```
2. **View errors**: Go to https://sentry.io/issues/ to see all errors with full stack traces

What Sentry Captures:

- All JavaScript errors and crashes
- Network request failures
- Login errors with full response data
- Custom error messages
- User device info (OS version, device model, etc.)
- Breadcrumbs (user actions leading to error)

Free tier: 5,000 errors/month

Quick Sentry Setup:

Your DSN has been configured. Just rebuild the APK to enable error tracking:
```Bash Terminal
eas build –platform android –profile preview

```
Then check https://sentry.io/issues/ to see all errors from your app!

## Common Issues & Solutions

### "Network Error" on Login

**Symptoms**: Login fails with "Network error" message

**Causes & Solutions**:

1. **Wrong URL format**
   - ✅ Correct: `http://192.168.1.100:5000` or `https://frigate.yourdomain.com`
   - ❌ Wrong: `192.168.1.100` (missing http://), `http://frigate.local` (might not r
esolve)

2. **Not on same WiFi network** (for local URLs)
   - Phone must be on same WiFi as Frigate server
   - Try using the remote URL instead

3. **Firewall blocking connections**
   - Check Frigate server firewall
   - Ensure port 5000 is open

4. **Frigate not running**
   - Verify Frigate is accessible in browser on phone
   - Open `http://192.168.1.100:5000` in Chrome/Safari

5. **Android network security policy**
   - HTTP (not HTTPS) requests might be blocked
   - Try using HTTPS URL instead

### "Login endpoint not found" (404) or "Method not allowed" (405)

**Cause**: Your reverse proxy is forwarding to the wrong port

**Solution**:
- Frigate has two ports:
  - **Port 5000**: Unauthenticated (no `/api/login` endpoint)
  - **Port 8971**: Authenticated (has `/api/login` endpoint) ← Required!
- Configure your reverse proxy to forward to port 8971
- The app uses `/api/login` with body format: `{user, password}`

### "Invalid username or password" (401)

**Cause**: Credentials don't match Frigate user accounts

**Solution**:
- Verify credentials work in Frigate web UI
- Create a new user in Frigate settings
- Username and password are case-sensitive

### Cameras not streaming

**Symptoms**: Cameras appear but video doesn't load

**Solutions**:

1. **WebRTC not working**
   - go2rtc must be enabled and accessible on port 1984
   - Try switching to "High" or "Low" quality mode

2. **Port 1984 not accessible**
```

```
   - Check firewall settings
   - Verify go2rtc is running: `http://your-ip:1984`

3. **Camera disabled in Frigate**
   - Check Frigate config: all cameras must be enabled

## Debugging Network Issues

### Test connectivity from your phone:

1. **Open browser on phone** (Chrome/Safari)
2. **Try accessing Frigate directly**:
   - Local: `http://192.168.1.100:5000`
   - Remote: `https://frigate.yourdomain.com`
3. **Try accessing go2rtc**:
   - `http://192.168.1.100:1984` (should show go2rtc interface)

### Check from terminal:

```Bash Terminal
# From your Mac/PC, test if Frigate is reachable
curl http://192.168.1.100:5000/api/config

# Check if authentication works
curl -X POST http://192.168.1.100:5000/login \
  -H "Content-Type: application/json" \
  -d '{"username":"your_username","password":"your_password"}'
```

Getting Help

When reporting issues, include:

1. **Error message** (exact text)
2. **Frigate version** (from web UI)
3. **Frigate URL** you're using (local or remote)
4. **Android/iOS version**
5. **Logs** (from Logcat or Sentry)
6. **Browser test result** (does Frigate work in phone's browser?)

Useful Commands

```Bash Terminal

# View real-time logs while testing

adb logcat -s ReactNativeJS:V

# Clear app data and start fresh

adb shell pm clear com.frigate.nvr.mobile
```
```

Reinstall app

adb install -r app.apk

Check if device is connected

adb devices

Forward port for debugging

adb reverse tcp:5000 tcp:5000

```
## Advanced Debugging

### Enable network traffic logging:

Add to `frigateApi.ts`:

```typescript
import axios from 'axios';

// Add request interceptor
axios.interceptors.request.use(request => {
  console.log('[Request]', request.method?.toUpperCase(), request.url);
  console.log('[Headers]', request.headers);
  console.log('[Data]', request.data);
  return request;
});

// Add response interceptor
axios.interceptors.response.use(
  response => {
    console.log('[Response]', response.status, response.config.url);
    return response;
  },
  error => {
    console.error('[Response Error]', error.message);
    console.error('[Error Details]', error.toJSON());
    return Promise.reject(error);
  }
);
```

Test without authentication:

If your Frigate doesn't have authentication enabled, temporarily modify `frigateApi.ts`:

```
// Skip login and connect directly
this.baseUrl = frigateUrl;
this.client = axios.create({
  baseURL: this.baseUrl,
  timeout: 30000,
});
```
```