Error Solving Guide

Common Error Types and Solutions

1. GIF Capture Errors

Error: frameDelay is not defined

Location: `server_side/services/dashboardCreationService.js` **Error Message**:

```
[DEBUG] Error capturing Grafana GIF: Error [ReferenceError]: frameDelay is not
defined
```

Cause: Environment variables for GIF capture settings are missing or not properly loaded. **Solution**:

1. Check `.env` file has these variables:

```
GIF_CAPTURE_DIR=captureGrafanaGIF
CAPTURE_WIDTH=1920
CAPTURE_HEIGHT=1080
CAPTURE_FRAMES=10
CAPTURE_FRAME_DELAY=500
CAPTURE_TIMEOUT=120000
```

2. Verify `process.env` values are being read in `captureGrafanaGIF` function
3. Add error handling for missing environment variables

Error: GIF File Not Found

Location: `server_side/services/slackService.js` **Error Message**: `ENOENT: no such file or directory` **Solution**:

1. Check multiple path formats:

```
const possiblePaths = [
    dashboard.local_snapshot_url,
    path.join(__dirname, '..', 'captureGrafanaGIF',
path.basename(dashboard.local_snapshot_url)),
    path.join(process.cwd(), 'captureGrafanaGIF',
path.basename(dashboard.local_snapshot_url)),
    path.join('captureGrafanaGIF', path.basename(dashboard.local_snapshot_url))
];
```

2. Ensure GIF directory exists and has proper permissions

3. Check file path normalization for Windows systems

2. Database Connection Errors

Error: Connection Refused

Location: `server_side/config/dbConfig.js` **Error Message**: `ECONNREFUSED` **Solution**:

1. Check database host and port in `.env`:

```
MASTER_DB_HOST=172.16.16.26
MASTER_DB_PORT=5432
MASTER_DB_USER=postgres
MASTER_DB_PASS=root
```

2. Verify PostgreSQL is running
3. Check firewall settings
4. Verify connection pool settings:

```
DB_POOL_MAX=20
DB_POOL_TIMEOUT=5000
DB_POOL_IDLE_TIMEOUT=30000
```

Error: Database Pool Exhaustion

Location: `server_side/config/dbConfig.js` **Error Message**: `timeout exceeded when trying to connect` **Solution**:

1. Implement pool refresh mechanism:

```
const refreshInterval = parseInt(process.env.DB_POOL_REFRESH_INTERVAL ||
DEFAULT_CONFIG.refresh_interval);
setInterval(refreshPool, refreshInterval);
```

2. Add proper error handling for pool events
3. Implement connection validation

3. Grafana API Errors

Error: Authentication Failed

Location: `server_side/services/grafanaService.js` **Solution**:

1. Verify Grafana credentials in `.env`:

```
GRAFANA_BASE_URL=http://172.16.16.26:3000
GRAFANA_USER=slack
GRAFANA_PASSWORD=Welcom@123
```

2. Check if Grafana service is accessible
3. Verify API key permissions
4. Implement retry mechanism for transient failures

Error: Dashboard Creation Failed

Location: `server_side/services/dashboardCreationService.js` **Solution**:

1. Check dashboard template validity
2. Verify datasource UIDs are correct
3. Ensure proper permissions in Grafana
4. Add error logging for debugging

4. Slack Integration Errors

Error: Invalid Token

Location: `server_side/services/slackService.js` **Solution**:

1. Check user's Slack credentials in database:

```sql
SELECT * FROM credentials
WHERE key_name IN ('SLACK_BOT_TOKEN', 'SLACK_CHANNEL_ID');
```

2. Verify bot token permissions
3. Ensure channel ID is correct
4. Implement token refresh if needed

Error: File Upload Failed

Location: `server_side/services/slackService.js` **Solution**:

1. Check file size limits
2. Verify file permissions
3. Implement multi-step upload process:

```javascript
// Step 1: Get upload URL
const step1 = await
axios.post("https://slack.com/api/files.getUploadURLExternal"...);

// Step 2: Upload file
const form = new FormData();
form.append("file", fs.createReadStream(normalizedPath));
```

```
// Step 3: Complete upload
await axios.post("https://slack.com/api/files.completeUploadExternal"...);
```

5. Frontend Errors

Error: API Connection Failed

Location: `frontend/src/api/axiosClient.js` **Solution**:

1. Check API URL in frontend `.env`:

```
REACT_APP_API_URL=http://localhost:8050
REACT_APP_API_TIMEOUT=30000
```

2. Verify CORS settings in backend
3. Implement retry logic:

```
const axiosClient = axios.create({
  baseURL: process.env.REACT_APP_API_URL,
  timeout: parseInt(process.env.REACT_APP_API_TIMEOUT)
});
```

Error: State Management Issues

Location: `frontend/src/pages/CredentialsPage.jsx` **Solution**:

1. Implement proper state updates
2. Add error boundaries
3. Use proper React hooks
4. Add loading states

6. Environment Variable Issues

Error: Missing Environment Variables

Location: Various files **Solution**:

1. Create comprehensive `.env.example`:

```
# Server Configuration
PORT=8050
HOST=0.0.0.0

# Database Configuration
MASTER_DB_HOST=localhost
```

```
MASTER_DB_PORT=5432

# Grafana Configuration
GRAFANA_BASE_URL=http://localhost:3000

# GIF Configuration
CAPTURE_FRAME_DELAY=500
CAPTURE_FRAMES=10
```

2. Implement environment validation
3. Add default values where appropriate

Debugging Tips

Backend Debugging

1. Enable debug logging:

```
LOG_LEVEL=debug
```

2. Check application logs:

```
tail -f logs/app.log
```

3. Monitor database connections:

```
SELECT * FROM pg_stat_activity;
```

Frontend Debugging

1. Enable React Developer Tools
2. Use browser console logging
3. Implement error boundaries:

```
class ErrorBoundary extends React.Component {
  componentDidCatch(error, errorInfo) {
    console.error('Error:', error);
    // Log to service
  }
}
```

Network Debugging

1. Use browser Network tab
2. Monitor API responses
3. Check CORS headers

Quick Solutions for Common Issues

1. Server Won't Start

```
# Check logs
pm2 logs
# Restart server
pm2 restart all
```

2. Database Issues

```
# Check PostgreSQL status
systemctl status postgresql
# Restart PostgreSQL
systemctl restart postgresql
```

3. GIF Capture Issues

```
# Clear GIF directory
rm -rf captureGrafanaGIF/*
# Check directory permissions
chmod 755 captureGrafanaGIF
```

4. Frontend Build Issues

```
# Clear node modules
rm -rf node_modules
npm install
# Clear cache
npm cache clean --force
```

File Locations for Common Issues

Backend Issues

- Database Configuration: `server_side/config/dbConfig.js`
- Grafana Integration: `server_side/services/grafanaService.js`
- Slack Integration: `server_side/services/slackService.js`
- GIF Capture: `server_side/services/dashboardCreationService.js`

Frontend Issues

- API Client: `frontend/src/api/axiosClient.js`
- Credential Management: `frontend/src/pages/CredentialsPage.jsx`
- User Management: `frontend/src/pages/ManageUserPage.jsx`
- Layout Issues: `frontend/src/components/Layout.jsx`

Monitoring and Maintenance

Regular Checks

1. Database connection pool health
2. GIF directory size and cleanup
3. API response times
4. Frontend performance metrics

Preventive Measures

1. Implement proper logging
2. Set up monitoring alerts
3. Regular database maintenance
4. Automated testing