Production Testing Guide

A step-by-step guide for testing Oracle SQL Server connections, PeopleSoft SOAP authentication, and running real SmartForm SQL queries.

Table of Contents

Quick Start Checklist

Before testing, ensure you have:

- [ ] Access to an Oracle database server (hostname, port, service name)
- [ ] Valid Oracle database credentials (username/password)
- [ ] Access to a PeopleSoft web server (hostname, port, site name)
- [ ] Valid PeopleSoft credentials with Component Interface access
- [ ] Network connectivity to both servers (VPN if required)
- [ ] Your production SmartForm SQL query ready to copy

Environment Variables Reference

All configuration is done in the `.env` file at the project root. This file is git-ignored, so your credentials remain private.

Application Mode

Variable	Values	Description
`VITE_APP_MODE`	`development` / `production`	Controls whether mock data or real connections are used

Oracle SQL Variables

Variable	Required	Default	Description
`VITE_ORACLE_HOSTNAME`	Yes	-	Oracle server hostname or IP address
`VITE_ORACLE_PORT`	No	`1521`	Oracle listener port
`VITE_ORACLE_SERVICE_NAME`	Yes	-	Oracle service name (ask your DBA)

PeopleSoft SOAP Variables

Variable	Required	Default	Description
`VITE_PS_PROTOCOL`	No	`https`	Protocol (`http` or `https`)
`VITE_PS_SERVER`	Yes	-	PeopleSoft web server hostname
`VITE_PS_PORT`	No	`443`	Web server port (typically 443 or 8443)
`VITE_PS_SITE_NAME`	Yes	-	PeopleSoft site name (e.g., HRPRD, HRTST)
`VITE_PS_PORTAL`	Yes	-	Portal name (e.g., EMPLOYEE)
`VITE_PS_NODE`	Yes	`PT_LOCAL`	Default local node name

Optional Settings

Variable	Default	Description
`VITE_PS_LANGUAGE_CODE`	`ENG`	Language code for SOAP requests
`VITE_SOAP_BLOCKING_FACTOR`	`40`	Records per batch for bulk operations
`VITE_ENABLE_DEBUG_LOGGING`	`false`	Verbose console logging
`VITE_ENABLE_AUDIT_TRAIL`	`false`	Audit logging for data modifications
`VITE_ALLOWED_ORIGINS`	-	CORS origins for production (comma-separated)

Configuring Oracle SQL Connection

Step 1: Open your `.env` file

```
# From project root
code .env   # or notepad .env
```

Step 2: Update Oracle settings

Find the Oracle section and replace the placeholder values:

```
# ========================================
# ORACLE SQL CONNECTION
# ========================================
# Uses node-oracledb Thin Mode - NO Oracle Client installation needed!
# Requires Oracle Database 12.1 or later

# [REQUIRED] Oracle database server hostname or IP
VITE_ORACLE_HOSTNAME=your-actual-oracle-server.company.com

# [OPTIONAL] Oracle listener port (default: 1521)
VITE_ORACLE_PORT=1521

# [REQUIRED] Oracle service name (ask your DBA if unsure)
VITE_ORACLE_SERVICE_NAME=HRPRD
```

Example for a typical HR database:

```
VITE_ORACLE_HOSTNAME=hr-db-prod.mycompany.com
VITE_ORACLE_PORT=1521
VITE_ORACLE_SERVICE_NAME=HRPRD
```

Configuring PeopleSoft SOAP Connection

Step 3: Update PeopleSoft settings

Find the PeopleSoft SOAP section and update:

```
# ==========================================
# PEOPLESOFT SOAP CONNECTION
# ==========================================

# [OPTIONAL] Protocol - use 'https' for production (default: https)
VITE_PS_PROTOCOL=https

# [REQUIRED] PeopleSoft web server hostname
VITE_PS_SERVER=your-actual-ps-server.company.com

# [OPTIONAL] PeopleSoft web server port (default: 443 for https)
VITE_PS_PORT=8443

# [REQUIRED] PeopleSoft site name
VITE_PS_SITE_NAME=HRPRD

# [REQUIRED] Portal name
VITE_PS_PORTAL=EMPLOYEE

# [REQUIRED] Default local node name
VITE_PS_NODE=PT_LOCAL
```

Example for a typical PeopleSoft environment:

```
VITE_PS_PROTOCOL=https
VITE_PS_SERVER=ps-web-prod.mycompany.com
VITE_PS_PORT=8443
VITE_PS_SITE_NAME=HRPRD
VITE_PS_PORTAL=EMPLOYEE
VITE_PS_NODE=PT_LOCAL
```

Switching to Production Mode

Step 4: Enable production mode

Change `VITE_APP_MODE` from `development` to `production`:

```
# ==========================================
# APPLICATION MODE
# ==========================================
# Comment out development mode:
#VITE_APP_MODE=development

# Enable production mode:
VITE_APP_MODE=production
```

> **WARNING**: In production mode, all actions affect live data. Credential verification will use actual servers!

Step 5: (Optional) Configure CORS for local testing

If running `npm run dev` with production mode enabled, add localhost to allowed origins:

```
# For local production testing
VITE_ALLOWED_ORIGINS=http://localhost:5173,http://127.0.0.1:5173
```

Updating the SmartForm SQL Query

The SmartForm SQL query file determines what data appears in the DataTable. Here's how to replace it with your production query.

Location

```
src/server/sql/server/smartform-pending-transactions.sql
```

Required Columns

Your SQL query **must** return these columns (used by the type system):

Column	Type	Description
`TRANSACTION_NBR`	VARCHAR	Unique transaction identifier (displayed as hyperlink)
`MGR_CUR`	NUMBER	Queue filter: `1` = Manager queue, `0` = Other queue
`EMPLID`	VARCHAR	Employee ID
`EMPLOYEE_NAME`	VARCHAR	Full employee name

Column	Type	Description
`WEB_LINK`	VARCHAR	Full URL for transaction hyperlink (hidden from display)

Optional Known Columns (with special formatting)

Column	Type	Formatting
`NEW_EFFDT`	DATE	Formatted as MM/DD/YYYY; highlighted if matches CUR_EFFDT
`CUR_EFFDT`	DATE	Formatted as MM/DD/YYYY
`CUR_POS`	VARCHAR	Monospace font
`EMPL_RCD`	NUMBER	Standard text
`POSITION_CREATE_CI`	VARCHAR	Standard text (nullable)
`POSITION_UPDATE_CI`	VARCHAR	Standard text (nullable)
`JOB_UPDATE_CI`	VARCHAR	Standard text (nullable)
`DEPT_CO_UPDATE_CI`	VARCHAR	Standard text (nullable)
`FIELD_DIFFERENCES`	VARCHAR	Standard text (nullable)

Adding Custom Columns

Any additional columns in your SQL query will automatically appear in the DataTable!

The DataTable dynamically builds columns from the first row's keys. For example, if your query returns:

- `DEPARTMENT_NAME`
- `SUPERVISOR_ID`
- `REQUESTED_BY`

These will all appear as additional text columns in the table.

Step 6: Replace the SQL file

1. Open the placeholder SQL file:

   ```
   src/server/sql/server/smartform-pending-transactions.sql
   ```

2. Replace the entire contents with your production query. Keep the comment header for documentation:

```sql
-- SmartForm Pending Transactions Query
--
-- Retrieves all pending CI transactions awaiting approval.
-- Results are used to populate the SmartForm data table.
--
-- Required columns:
--   TRANSACTION_NBR    - Unique transaction identifier
--   MGR_CUR            - Manager current flag (1 = Manager, 0 = Other)
--   EMPLID             - Employee ID
--   EMPLOYEE_NAME      - Full employee name
--   WEB_LINK           - Full URL for transaction hyperlink
--
-- Additional columns are dynamically displayed in the DataTable.

SELECT
    t.TRANSACTION_NBR,
    t.MGR_CUR,
    t.EMPLID,
    t.EMPL_RCD,
    e.NAME AS EMPLOYEE_NAME,
    t.NEW_EFFDT,
    t.CUR_EFFDT,
    t.CUR_POS,
    t.POSITION_CREATE_CI,
    t.POSITION_UPDATE_CI,
    t.JOB_UPDATE_CI,
    t.DEPT_CO_UPDATE_CI,
    t.FIELD_DIFFERENCES,
    'https://your-ps-server.com/psp/portal/txn/' || t.TRANSACTION_NBR AS WEB_LINK
    -- Add your custom columns here
FROM YOUR_PENDING_TXN_TABLE t
JOIN PS_PERSONAL_DATA e ON e.EMPLID = t.EMPLID
WHERE t.STATUS = 'PENDING'
ORDER BY t.TRANSACTION_NBR
```

3. Save the file.

Important Notes

- **No restart required** - SQL files are read at query execution time
- **Test in development first** - Use mock data to verify column rendering before production
- **Keep the WHERE clause** - Ensure your query only returns pending transactions

Testing the Full Flow

Step 7: Start the application

```
npm run dev
```

Watch the console for connection configuration messages:

```
[Oracle] Service ready
[SOAP] Service initialized
[SOAP] Server: https://ps-web-prod.mycompany.com:8443
[SOAP] Site: HRPRD/EMPLOYEE/PT_LOCAL
```

Step 8: Open the application

Navigate to: `http://localhost:5173`

Step 9: Test Oracle SQL connection

1. In the **Connection Panel** (left side), find **Oracle SQL**
2. Enter your Oracle **username**
3. Enter your Oracle **password**
4. Click **Connect**

Success indicators:

- Green "Connected" indicator appears
- Console shows: `[Oracle] Connected successfully`

Step 10: Test PeopleSoft SOAP connection

1. In the **Connection Panel**, find **PeopleSoft**
2. Enter your PeopleSoft **username**
3. Enter your PeopleSoft **password**
4. Click **Connect**

Success indicators:

- Green "Connected" indicator appears
- Console shows: `[SOAP] Connection test successful`

Step 11: Run the SmartForm query

1. Navigate to the **SmartForm** tab
2. Click **Run Query**
3. Observe the DataTable populating with your production data

What to verify:

- Row count matches expected transactions
- Manager/Other counts are correct (filtered by MGR_CUR)
- All columns are visible and properly formatted
- TRANSACTION_NBR displays as clickable hyperlink
- Dates display as MM/DD/YYYY

Understanding DataTable Column Rendering

The DataTable uses a **dynamic column generation** system. Here's how it works:

Column Build Process

```
Oracle Query Results
       ↓
First Row Keys Extracted
       ↓
Filter: Remove HIDDEN_SMARTFORM_FIELDS
       ↓
Map: Apply special formatting
  - MONOSPACE_SMARTFORM_FIELDS → monospace font
  - DATE_SMARTFORM_FIELDS → MM/DD/YYYY format
  - TRANSACTION_NBR → hyperlink using WEB_LINK
       ↓
Render: Display in DataTable
```

Modifying Field Behavior

To change how fields are rendered, edit `src/types/smartform.ts`:

```
// Fields hidden from table display
export const HIDDEN_SMARTFORM_FIELDS = ['MGR_CUR', 'WEB_LINK', 'status', 'errorMessage'] as const;

// Fields with monospace font
export const MONOSPACE_SMARTFORM_FIELDS = ['EMPLID', 'TRANSACTION_NBR', 'CUR_POS'] as const;

// Fields formatted as dates
export const DATE_SMARTFORM_FIELDS = ['NEW_EFFDT', 'CUR_EFFDT'] as const;
```

Example: Adding a new monospace field

If your query returns a `BADGE_ID` column and you want it monospace:

```
export const MONOSPACE_SMARTFORM_FIELDS = ['EMPLID', 'TRANSACTION_NBR', 'CUR_POS', 'BADGE_ID'] as const;
```

Troubleshooting

Oracle Connection Issues

Error	Cause	Solution
`ORA-01017`	Invalid username/password	Verify credentials
`ORA-12154`	Bad hostname or service name	Check `VITE_ORACLE_HOSTNAME` and `VITE_ORACLE_SERVICE_NAME`
`ORA-12541`	No listener	Verify Oracle is running; check port
`ORA-12170`	Connection timeout	Check network/VPN/firewall
`ORA-12514`	Wrong service name	Contact DBA for correct service name
`ORA-28000`	Account locked	Contact DBA to unlock
`ORA-28001`	Password expired	Reset password in Oracle

PeopleSoft SOAP Issues

Error	Cause	Solution
`AUTHENTICATION_FAILED`	Wrong credentials	Verify PeopleSoft username/password
`SOAP Fault: Invalid user`	User locked or doesn't exist	Check user status in PeopleSoft

Error	Cause	Solution
`Connection refused`	Server down or wrong port	Verify server is running; check port
`Connection timed out`	Network issue	Check VPN/firewall
`SSL certificate error`	Self-signed cert	May need proper SSL configuration
`SOAP Fault: Not authorized`	Missing CI permissions	User needs Component Interface access

SmartForm Query Issues

Symptom	Cause	Solution
Empty table	Query returns no rows	Check WHERE clause; verify data exists
Missing columns	Column not in SELECT	Add column to SQL query
Wrong column order	Oracle returns columns in query order	Reorder SELECT columns
Date format wrong	Date not in DATE_SMARTFORM_FIELDS	Add column name to array
Link not working	WEB_LINK column missing or malformed	Verify WEB_LINK returns full URL

Network Issues

1. **VPN Required**: Corporate servers often require VPN connection
2. **Firewall**: Ensure ports 1521 (Oracle) and 443/8443 (PeopleSoft) are open
3. **Proxy**: Corporate proxy may block direct database connections

---

Full `.env` Example

```
# ==========================================
# APPLICATION MODE
# ==========================================
VITE_APP_MODE=production

# ==========================================
# ORACLE SQL CONNECTION
# ==========================================
VITE_ORACLE_HOSTNAME=hr-db-prod.mycompany.com
VITE_ORACLE_PORT=1521
VITE_ORACLE_SERVICE_NAME=HRPRD

# ==========================================
# PEOPLESOFT SOAP CONNECTION
# ==========================================
VITE_PS_PROTOCOL=https
VITE_PS_SERVER=ps-web-prod.mycompany.com
VITE_PS_PORT=8443
VITE_PS_SITE_NAME=HRPRD
VITE_PS_PORTAL=EMPLOYEE
VITE_PS_NODE=PT_LOCAL

# ==========================================
# OPTIONAL SETTINGS
# ==========================================
VITE_ENABLE_DEBUG_LOGGING=true
VITE_ENABLE_AUDIT_TRAIL=false

# For local production testing (npm run dev with VITE_APP_MODE=production)
VITE_ALLOWED_ORIGINS=http://localhost:5173,http://127.0.0.1:5173
```

Security Reminders

1. **Never commit `.env` to git** - It contains sensitive connection info
2. **Use HTTPS** - Always use `VITE_PS_PROTOCOL=https` in production
3. **Least privilege** - Use accounts with minimal required permissions
4. **Audit logging** - Enable `VITE_ENABLE_AUDIT_TRAIL=true` for production tracking
5. **Memory-only passwords** - Credentials are never persisted to disk