outsystems

Debugging and Monitoring

Topics

- Debugging in Service Studio
  - During Debugging
  - Inspecting Variables
  - Debugging a Producer Module
  - Testing / Debugging Mobile Apps
- Monitoring in OutSystems
  - Service Center Logs

Troubleshooting applications

Code is subject to many errors during execution

Troubleshooting focuses on finding those errors so they can be fixed

Troubleshooting is generally a combination of:

- Debugging
  - Suspends code execution at given points
  - Analyze the value of variables
  - Developer can execute the code step by step
- Monitoring
  - Analyzing and correlating system logs
  - OutSystems records events in logs
    - Errors, slow queries, web service calls, …

Debugging

Debugging in Service Studio

Place breakpoints on elements that generate code

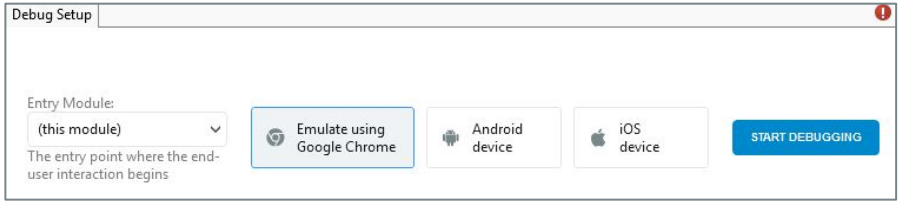
- Server and Client Actions

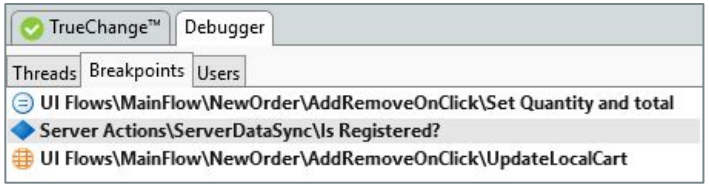
Start the debugger before opening the app in the browser

- Service Studio registers the breakpoints
- Debugging can be done using the browser emulator or the device

Breakpoints are listed in the Debug panel

- Can be temporarily disabled
- Can be removed





outsystems

During Debugging

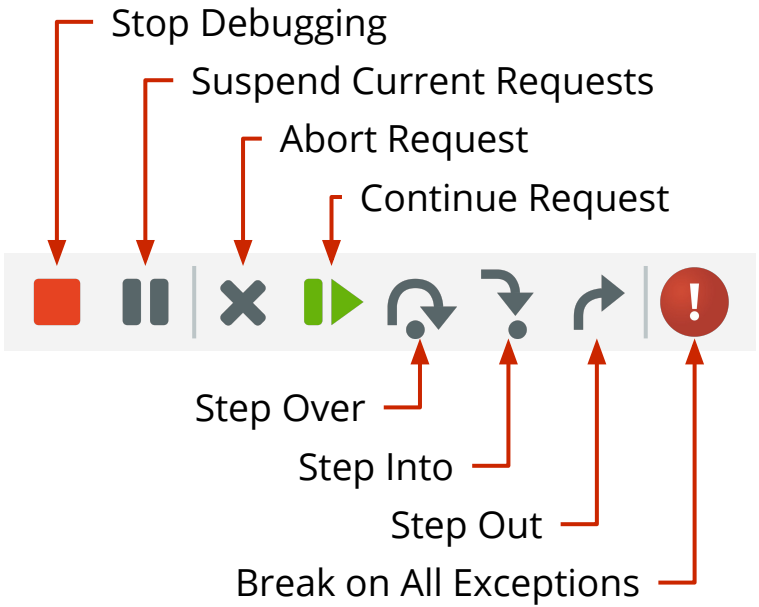
The following commands are available once code execution stops at a breakpoint

Regular debugging operations

- Stop, Continue, Suspend, Abort
- Step Over, Step Into, Step Out

Break on All Exceptions forces the debugger to suspend the execution of threads when exceptions occur

- Handled or unhandled



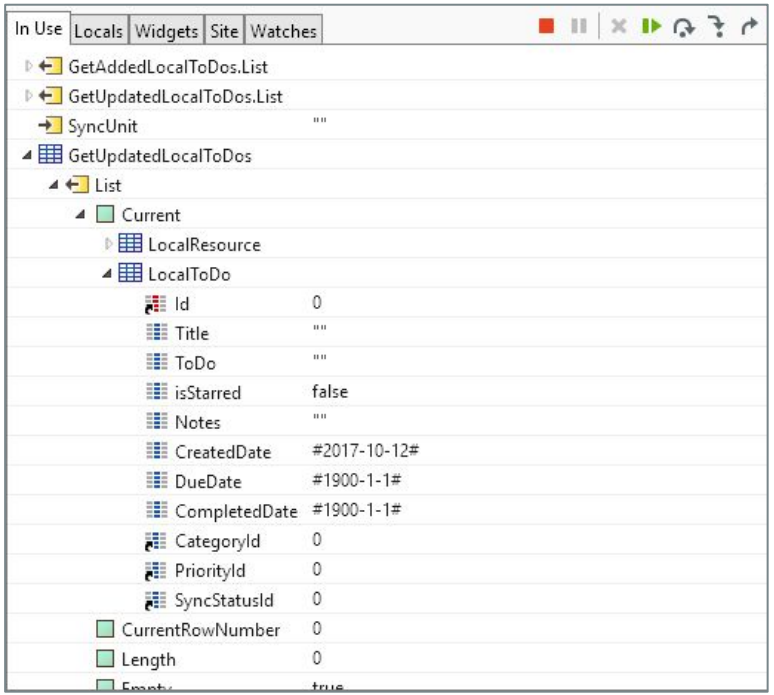
outsystems

Inspecting Variables during Debugging

Inspect the values of variables when code execution is stopped

- **In Use** by the current code statement
- **Locals** to the current execution scope
- Stored in this Screen's **Widgets**
- Global to the **Site** (i.e. application)
- Selected by the developer for quick access to always be under **Watch**

Watches are always displayed, even if out of scope



outsystems

Debugging a Producer Module

By default, breakpoints stop execution of requests made to that module *directly*

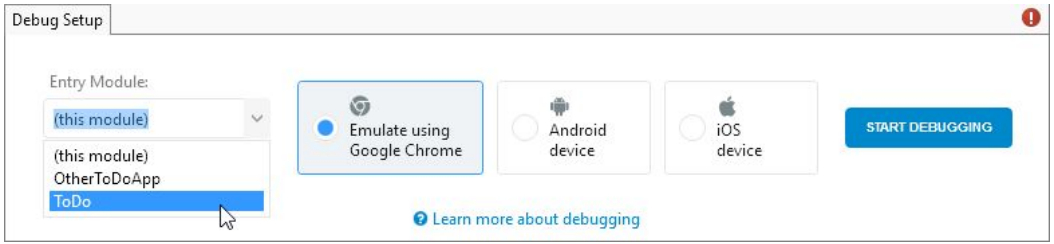
- Requests to *https://myserver.com/MyApp/* will stop on breakpoints inside MyApp

It is possible to stop the execution in another module's breakpoints

- Request to a consumer can stop in a producer
- e.g. Requests to *https://myserver.com/ToDo/* to stop on ToDo_Core breakpoints

To stop the execution in ToDo_Core, ToDo must be selected as entry module

- Requests start in the consumer module, but can stop on Core's breakpoints
- This can be done in any producer module of the application



outsystems

Debugging Options

On Mobile apps there are three debug options:

Emulate using Google Chrome

Android device

iOS device

Outsystems Now can also be used on Android to debug mobile apps running on the device

Testing Mobile Apps

- Google Chrome emulation allows to quickly test mobile apps
  - Inspect the source code
  - Does not test offline and plugin behaviors
  - OutSystems Now on Android is an alternative to debug the app on the device

- Android and iOS Debugging
  - Requires installing the native generated app
  - Connect the device with an USB Cable
  - Allows to better troubleshoot the performance and user experience
    - Offline scenarios
    - Plugin interaction
  - https://success.outsystems.com/Documentation/11/Developing_an_Application/Troubleshooting_Applications/Debugging_Applications

Monitoring

Monitoring in OutSystems

OutSystems tracks the occurrence of many events in distinct **Logs**

Logged events include situations for which diagnostics may be required:

- Errors
- Database query durations
- Web Service calls
- Mobile requests
- Native app generation

Monitoring is the act of analyzing and correlating these logs

Service Center Logs

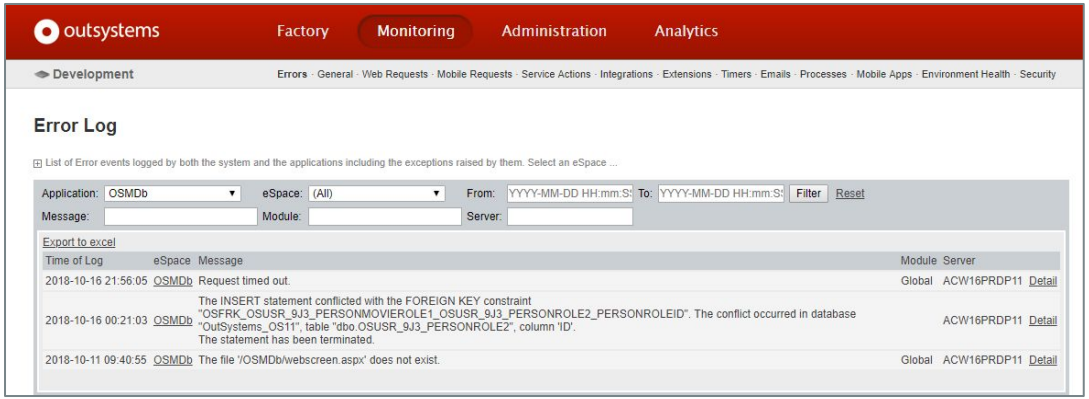
Service Center provides log reports

- **Errors**
- **General** audits
- **Mobile** requests
- **Integration** calls
- **Extension** calls
- Overall **Environment Health**
- … and more

Reports can be filtered by:

- application
- time window
- message content

and can be exported to Excel



outsystems

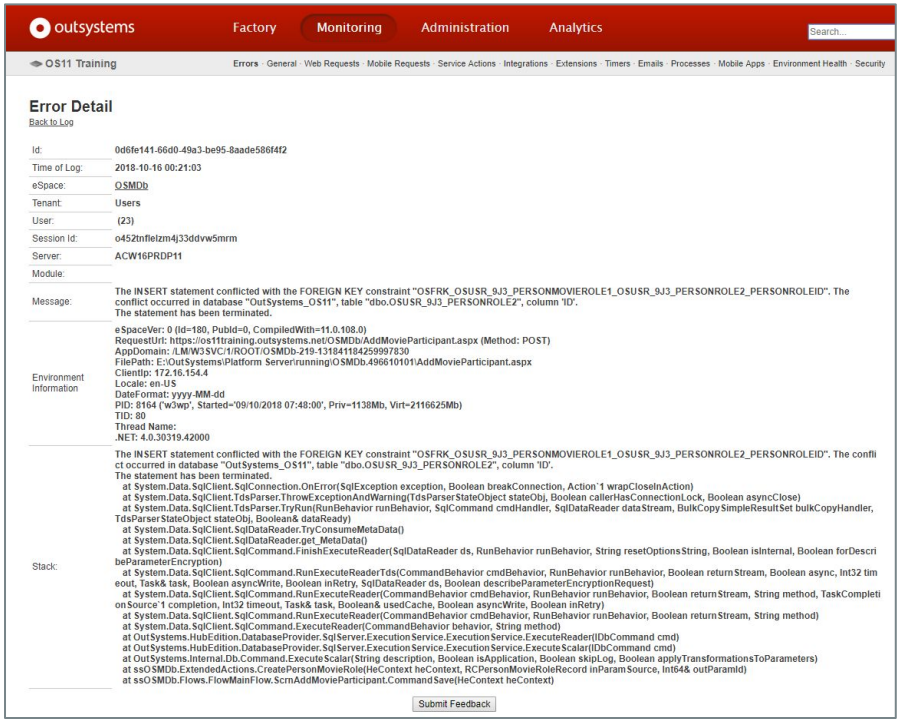
Log Detail

Clicking on the ‘Detail’ of an Error log shows extensive information on the logged event

Shows details like

- Message text
- Call Stack

Important for debugging an application during development



outsystems

Summary

- Debugging in Service Studio
  - During Debugging
  - Inspecting Variables
  - Debugging a Producer Module
  - Testing / Debugging Mobile Apps
- Monitoring in OutSystems
  - Service Center Logs


Debugging and
Monitoring
Thank You!