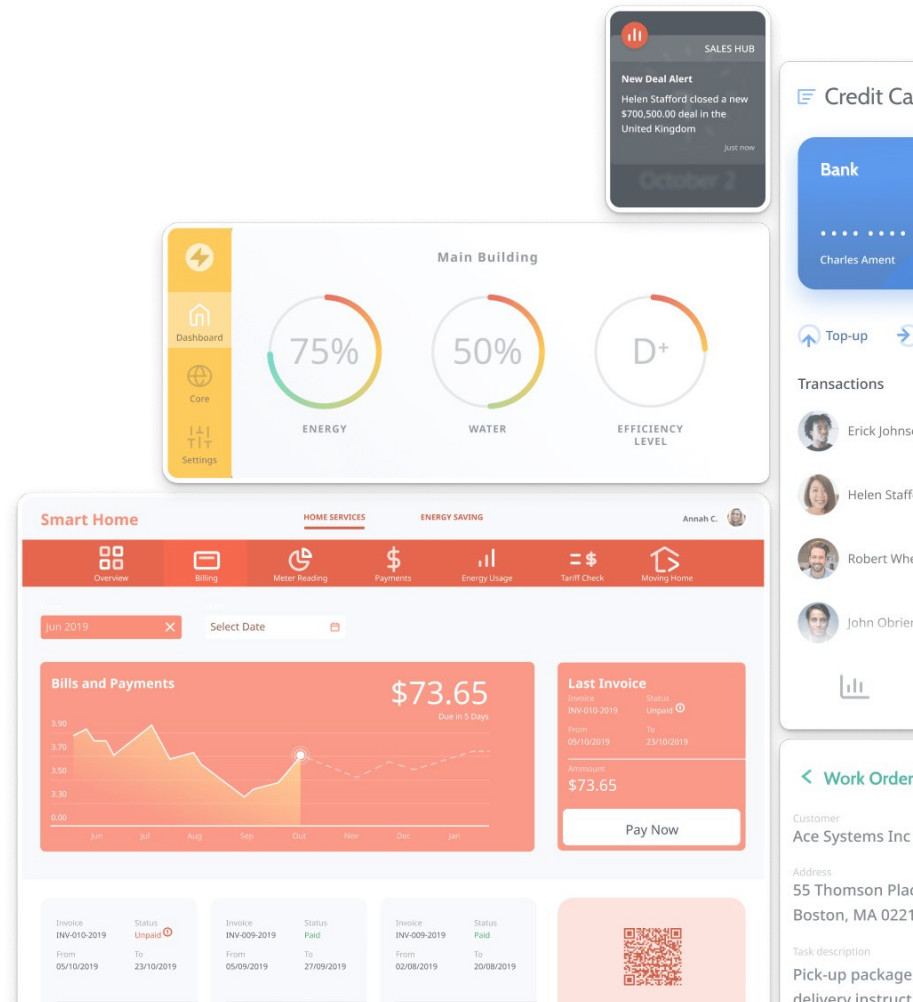




Debugging Apps

Reactive Developer Boot Camp

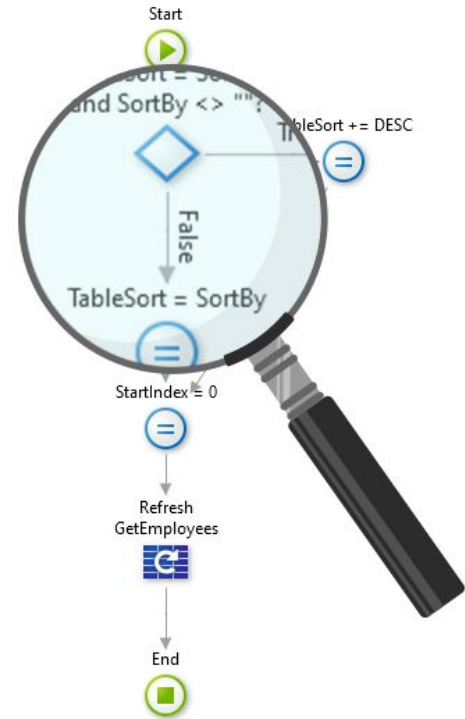


What you will learn here

- Troubleshooting Applications
- Debugging in Service Studio
- Inspecting Variables
- Debugging a Producer Module

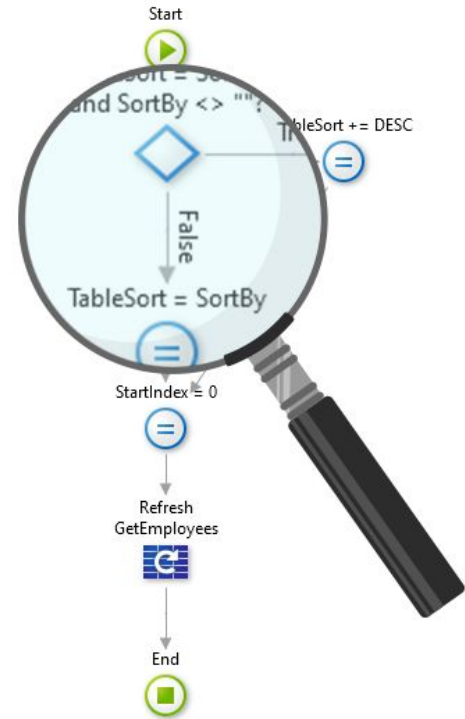
Troubleshooting Applications

- Code is subject to errors
 - Static Analysis (TrueChange) can help
 - Some are only detected during execution



Troubleshooting Applications

- Code is subject to errors
 - Static Analysis (TrueChange) can help
 - Some are only detected during execution
- Troubleshooting an application focuses on finding and fixing errors



What is Debugging?

Debugging is a **troubleshooting technique** that helps **find errors** in code.

Debugging Steps

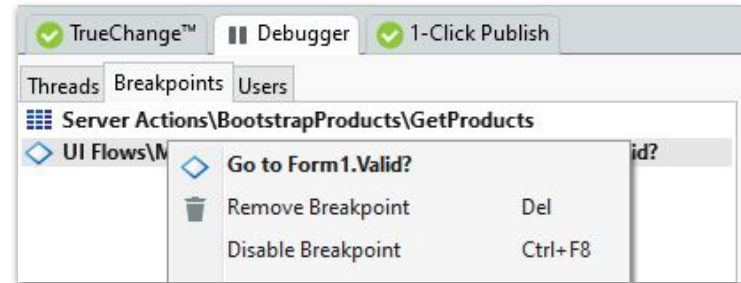
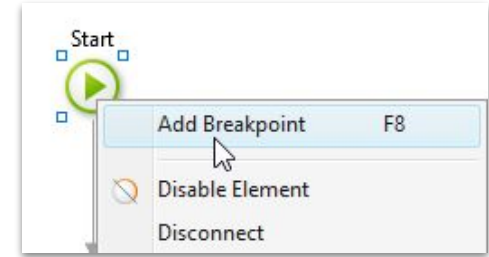


Debugging Applications



Add breakpoints

- Server and Client Actions
- Service Studio registers the breakpoints
- Breakpoints are listed in the Debug panel
 - Can be temporarily disabled
 - Can be removed



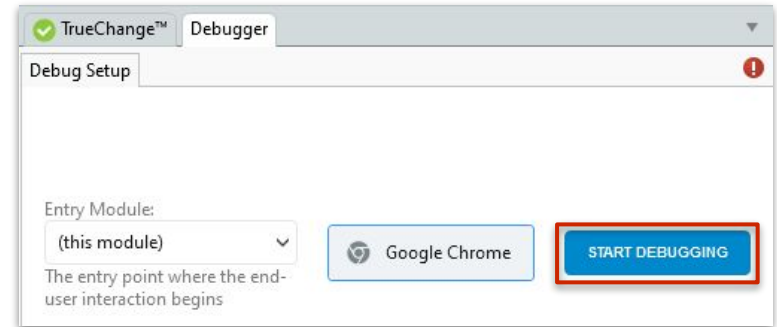
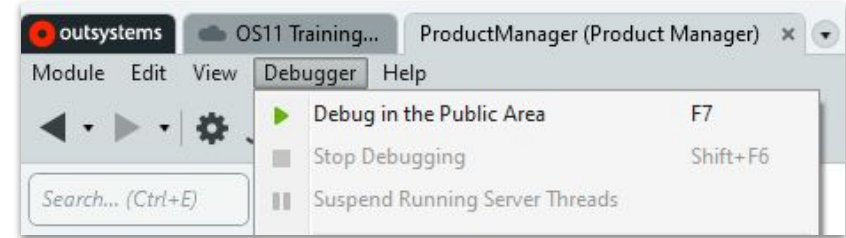
Debugging Applications



Add breakpoints

Start debugger

- Application will open in browser



Debugging Applications



Add breakpoints



Start debugger



Run the application

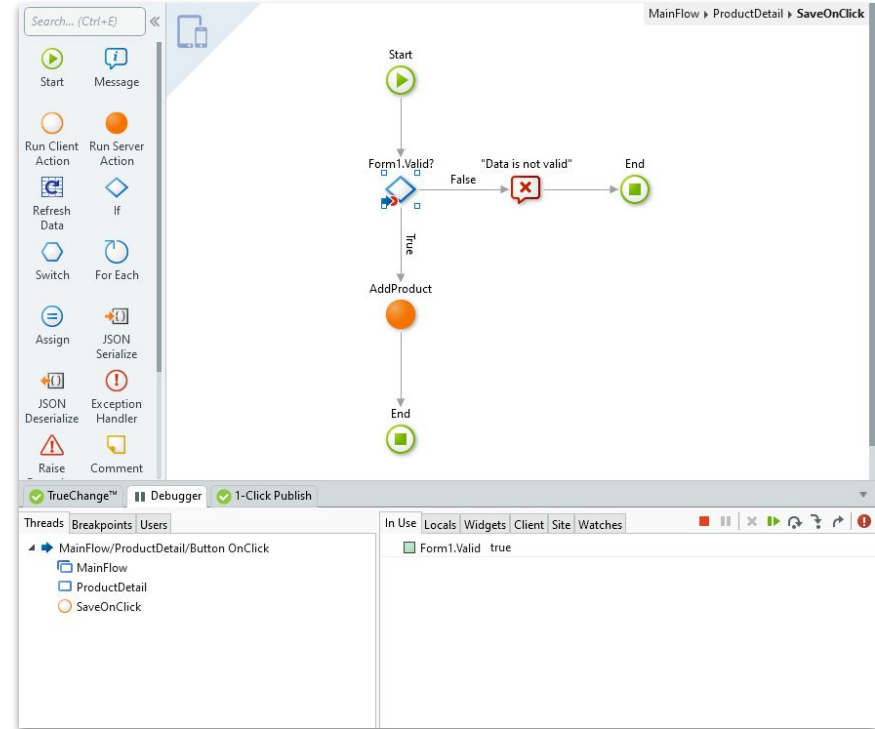
- Debugger suspends execution in breakpoints

Debugging Applications

- Add breakpoints
- Start debugger
- Run the application

Debug code

- Developer can execute the code step by step
- Analyze the value of variables



During Debugging

- When the execution gets to a breakpoint, the focus returns to Service Studio

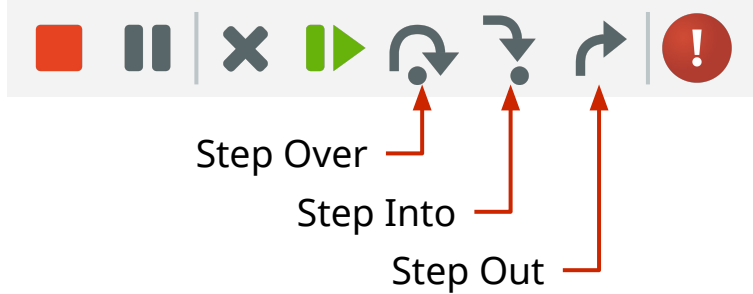


During Debugging



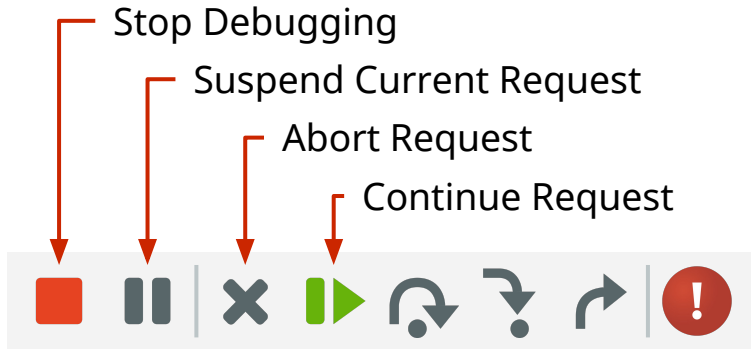
- When the execution gets to a breakpoint, the focus returns to Service Studio
- Regular debugging operations are available

During Debugging



- When the execution gets to a breakpoint, the focus returns to Service Studio
- Regular debugging operations are available
 - Step Over, Step Into, Step Out

During Debugging



- When the execution gets to a breakpoint, the focus returns to Service Studio
- Regular debugging operations are available
 - Step Over, Step Into, Step Out
 - Stop, Continue, Suspend, Abort

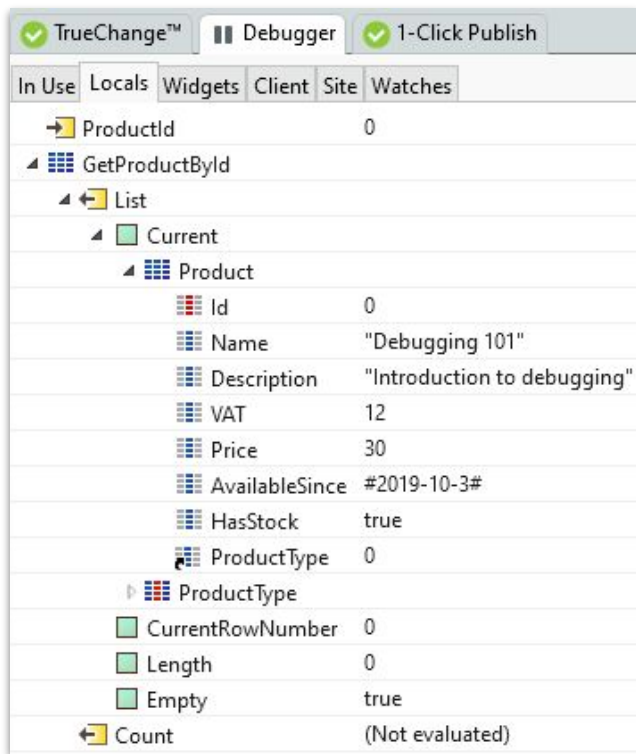
During Debugging



Break on All Exceptions

- When the execution gets to a breakpoint, the focus returns to Service Studio
- Regular debugging operations are available
 - Step Over, Step Into, Step Out
 - Stop, Continue, Suspend, Abort
 - Break on All Exceptions suspends the execution when exceptions occur

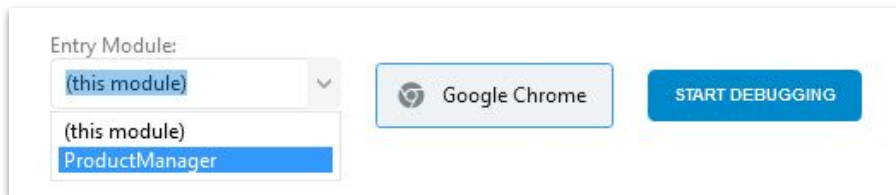
Inspecting Variables



- Values of the variables can be inspected when execution is stopped
 - **In use** by the current statement
 - **Locals** to the scope
 - Screen **widgets**
 - **Client** variables
 - Global to the app (or **Site**)
 - Selected by the developer to always be under **Watch**

Debugging a Producer Module

- Breakpoints stop execution of requests made *directly* to that module
 - Requests to `https://myserver.com/ProductManager/` will stop on breakpoints inside ProductManager
- It is possible to stop the execution in another module's breakpoints
 - Requests to `https://myserver.com/ProductManager/` to stop on Product_Core breakpoints
 - ProductManager must be selected as entry module in the Product_Core



Questions?



Thank You!

