# Debugging Mobile

Servoy Mobile applications can be debugged while running on actual devices like phones or tablets or in a desktop browser.

The debugger is a special instrumented version of the Google Chrome Developer Tools. This means that the Debugger inside Servoy Developer is not used and the Google Chrome is required to do debugging.

This feature is available as of Servoy 7.3.

## Preparation

To be able to debug a mobile solution, the solution needs to be exported and deployed in debug mode:

1. In Servoy Developer, open the export wizard: **Export Solution** > **Mobile Export**
2. On the first page of wizard:
    a. set the **Application Server URL** to a url/ip address and port of Servoy Developer that can be reached by the device on which the solution is to be debugged.
    b. check the **Export in debug mode** checkbox
3. Finish the wizard and deploy the result

Note that solutions exported in debug mode require Servoy Developer to be running and accessible from the Mobile Client. When Servoy Developer is not available, the mobile solution will fail to launch.

## Debugging

Once the Mobile solution is deployed and running, the debugger can be opened in Google Chrome on the machine where Servoy Developer runs on localhost:8889. Depending if there are multiple mobile debug clients connected, the page will first offer a list of clients to debug or go straight to the debugger if there's only one debug client connected.

The UI of the debugger sports a range of tabs at the top of the screen. Of those only the following two are functional:

- **Console** tab - displays the console output the client generates and allows interaction with the client through it's command line
- **Sources** tab - provides access to the JavaScript source files of the solution
    - The source files can be access through the little arrow button in the top left corner
    - The source files part of the solution design are located under the node for the IP address and port of the Servoy Application Server the solution is connected to, e.g. localhost:8080 or 192.168.1.10:8080

The other tabs are not functional at this point in time.

Breakpoints can be set by clicking in the gutter of the sourcefile and when hit, the tools in the right pane provide options to inspect code and to step through the code.

**Note:** If you are using Chrome version 36 or higher, you may need to set "Allow insecure WebSocket from https origin" in chrome://flags, else the debugger won't start.

## Hot Code Replace

The source files can be edited and when saved the changes are simultaneously applied both in the running client as well as pushed into the workspace of Servoy Developer, so the changes are still applied after restarting either the debug client or Servoy Developer.