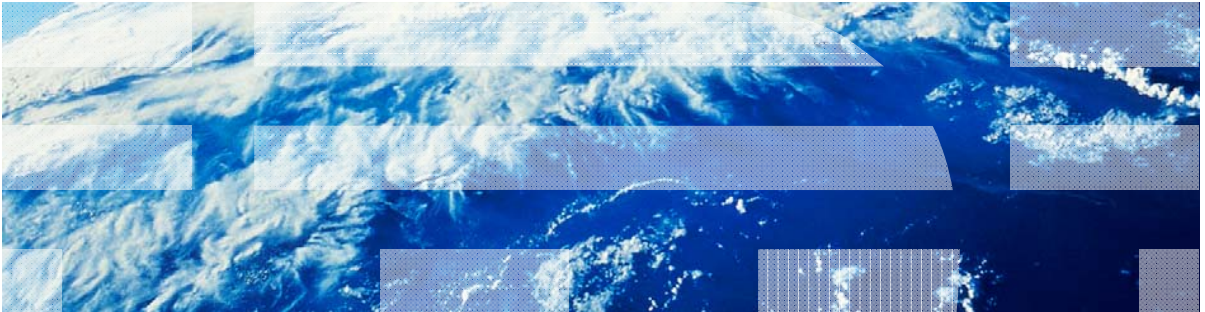


IBM Worklight V6.0.0 Getting Started

Debugging your applications



Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Worklight is a trademark or registered trademark of Worklight, an IBM Company. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)” at www.ibm.com/legal/copytrade.shtml.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Other company products or service names may be trademarks or service marks of others.
- This document may not be reproduced in whole or in part without the prior written permission of IBM.

About IBM®

- See <http://www.ibm.com/ibm/us/en/>

Agenda

- About
- Debugging on a desktop browser
- Debugging with IBM Worklight® debugger
- Testing the adapter procedures
- Debugging with Weinre
- Debugging with iOS Remote Web Inspector
- Debugging with WireShark
- Debugging with Mobile Browser Simulator

About Client-side Debugging

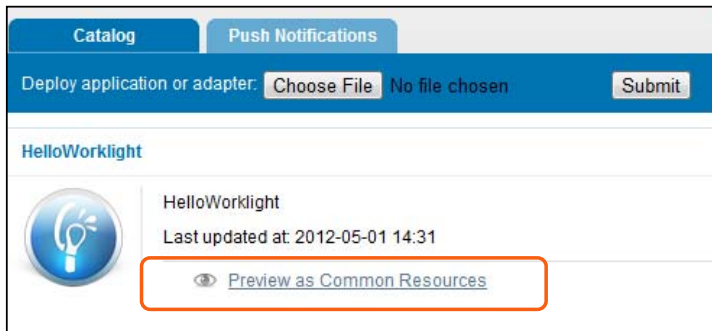
- Debugging is part of the application development process.
- Debugging is a process, which consists of finding the cause of defects in the application code and UI.
- IBM Worklight® applications consist of web-based resources and optional native code (such as Java™, Objective-C, and C#).
- You can debug native code by using standard tools that are provided by the platform SDK, such as: Xcode, Android LogCat/ADB, Microsoft Visual Studio.
- This module presents how to debug web-based resources.

Agenda

- About
- Debugging on a desktop browser
- Debugging with IBM Worklight debugger
- Testing the adapter procedures
- Debugging with Weinre
- Debugging with iOS Remote Web Inspector
- Debugging with WireShark
- Debugging with Mobile Browser Simulator

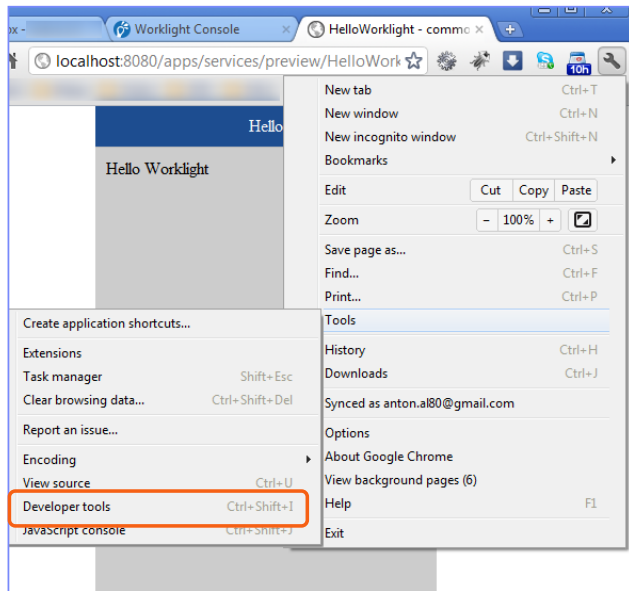
Debugging on a Desktop Browser

- Modern browsers, such as Chrome, Firefox, Safari, Opera, or Internet Explorer, provide an easy and convenient way to debug web apps.
- As seen in the previous modules, during development, you can preview applications in a desktop browser by using the Worklight Console.



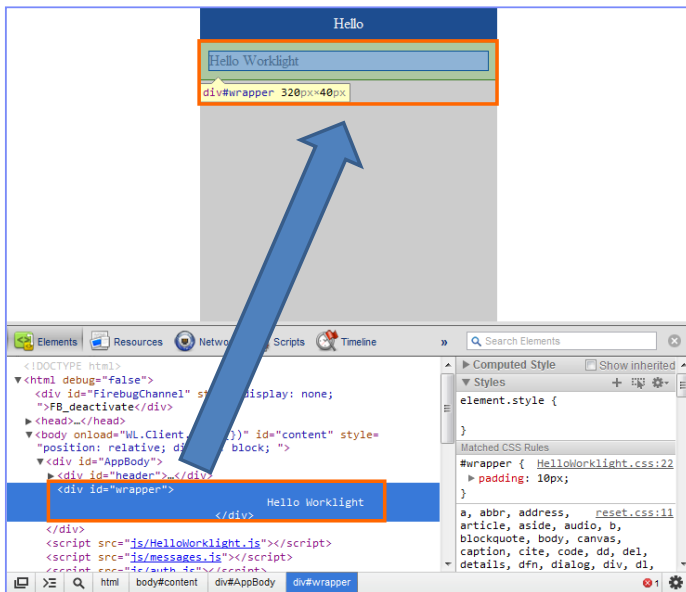
Debugging on a Desktop Browser

- There are many web tools for debugging on various desktop browsers.
- For example:
 - **FireBug** extension for Firefox
 - **Developer tools** for Chrome (press CTRL+SHIFT+I to start)
 - **Developer tools** for Internet Explorer (press F12 to start)
 - **Dragonfly** for Opera



Debugging on a Desktop Browser

- In early application development stages, use these tools to debug your application just like a regular website. You are not required to install them on a mobile device.



Agenda

- About
- Debugging on a desktop browser
- Debugging with IBM Worklight debugger
- Testing the adapter procedures
- Debugging with Weinre
- Debugging with iOS Remote Web Inspector
- Debugging with WireShark
- Debugging with Mobile Browser Simulator

Debugging with IBM Worklight Debugger

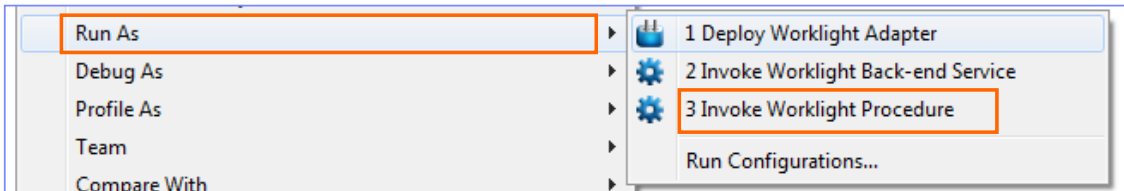
- IBM Worklight provides a `WL.Logger` object that you use to print log messages to the log for the environment used.
- Two of its methods are: **`WL.Logger.debug()`** and **`WL.Logger.error()`**
- **`WL.Logger.debug()`** and **`WL.Logger.error()`** APIs are multiplatform: the output destination changes according to the platform on which that application runs on – for example:
 - **Developer console** when it is running on a **desktop browser**
 - **LogCat** when it is running on Android device
 - **Visual Studio Output** when it is running on a **Windows Phone** device and **Windows 8** Apps
 - **Xcode console** when it is running on an **iOS** device
- `WL.Logger` contains more methods. See [the “WL.Logger methods” topic](#) in the IBM Worklight user documentation.

Agenda

- About
- Debugging on a desktop browser
- Debugging with IBM Worklight debugger
- **Testing the adapter procedures**
- Debugging with Weinre
- Debugging with iOS Remote Web Inspector
- Debugging with WireShark
- Debugging with Mobile Browser Simulator

Testing the Adapter Procedures

- It is possible to test adapter procedures by using the Worklight Studio.
- To test a procedure, click **Run As > Invoke Worklight Procedure**.

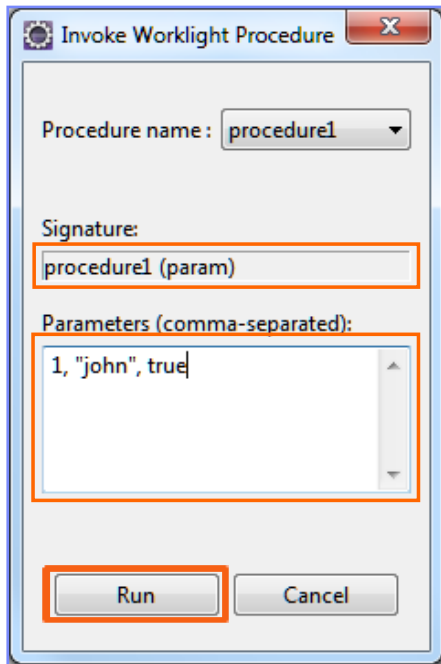


Testing the Adapter Procedures

- Select the procedure that you want to test.
- Enter the comma-separated procedure parameters.
- Click **Run**.
- Examine the result.

Invocation Result of procedure: 'getFeedsFiltered' from the Worklight Server:

```
{
  "items": [
    {
      "creator": "Danse Cesa",
      "link": "http://www.engadget.com/2011/10/30/telex-roadster-to-return-in-2014-3-series-f",
      "pubDate": "Sun, 30 Oct 2011 05:19:00 EDT",
      "title": "Telex Roadster to return in 2014, 3-Series fighter in the works"
    },
    {
      "creator": "Sean Buckley",
      "link": "http://www.engadget.com/2011/10/30/siri-port-now-talking-to-apple-servers-avoi",
      "pubDate": "Sun, 30 Oct 2011 03:19:00 EDT",
      "title": "Siri port now talking to Apple servers, avoiding Cydia"
    },
    {
      "creator": "Richard Lai",
      "link": "http://www.engadget.com/2011/10/30/lenovos-upcoming-lephone-s2-seen-running-an",
      "pubDate": "Sun, 30 Oct 2011 03:19:00 EDT",
      "title": "Lenovo's upcoming LePhone S2 seen running Android as well as Windows Phone, 1"
    },
    {
      "creator": "Darren Murph",
      "link": "http://www.engadget.com/2011/10/30/facebook-testing-trusted-friends-feature-th",
      "pubDate": "Sun, 30 Oct 2011 01:11:00 EDT",
      "title": "Facebook testing 'Trusted Friends' feature, the password unlock we hope you n"
    }
  ]
}
```



Agenda

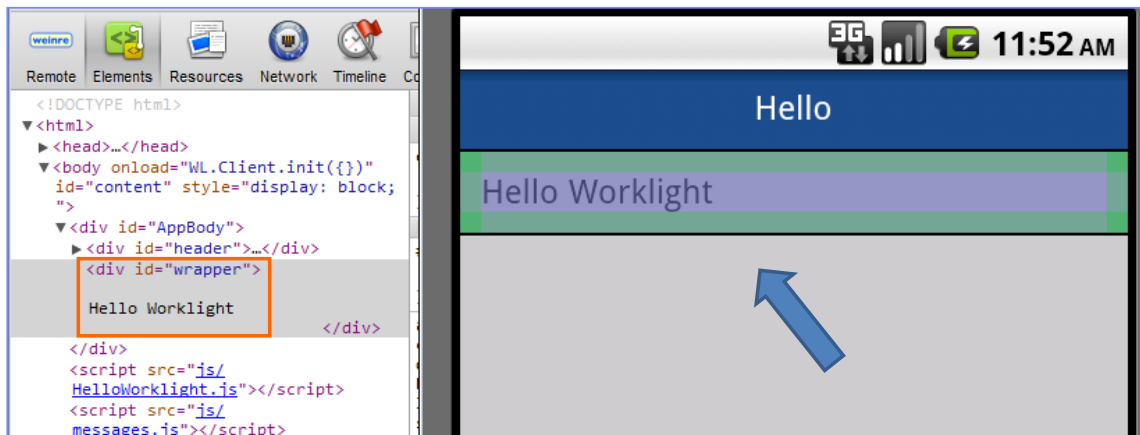
- About
- Debugging on a desktop browser
- Debugging with IBM Worklight debugger
- Testing the adapter procedures
- Debugging with Weinre
- Debugging with iOS Remote Web Inspector
- Debugging with WireShark
- Debugging with Mobile Browser Simulator

Debugging with Weinre

- Weinre stands for **Web Inspector Remote**
 - See <http://people.apache.org/~pmuellr/weinre/>
- Weinre is a debugger for web pages, like FireBug, except that it is designed to work remotely.
 - In particular, it allows debugging of web pages on mobile devices such as iPhone or Android.
- You can use Weinre to inspect and debug web resources such as HTML, JavaScript™, CSS, and Network traffic on a remote mobile handset.
- Weinre has a built-in JavaScript console that you can use as an extra powerful tool for web-based mobile application debugging.

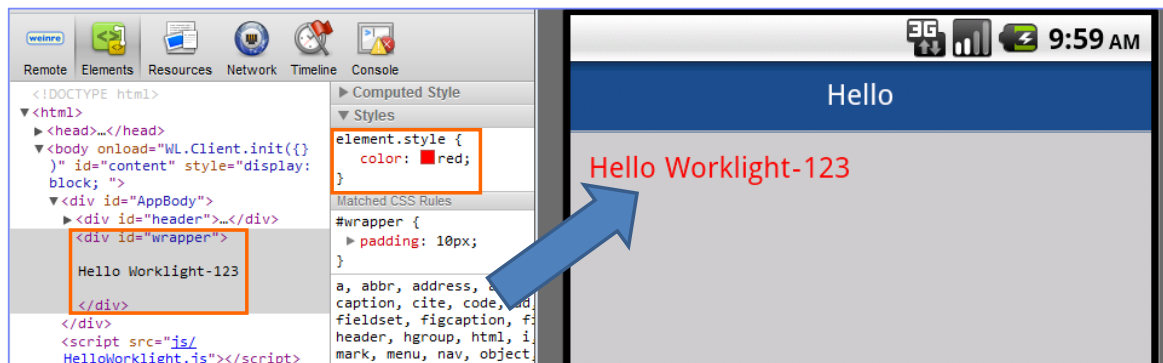
Debugging with Weinre

- Selecting an element in Weinre highlights it on the remote device.



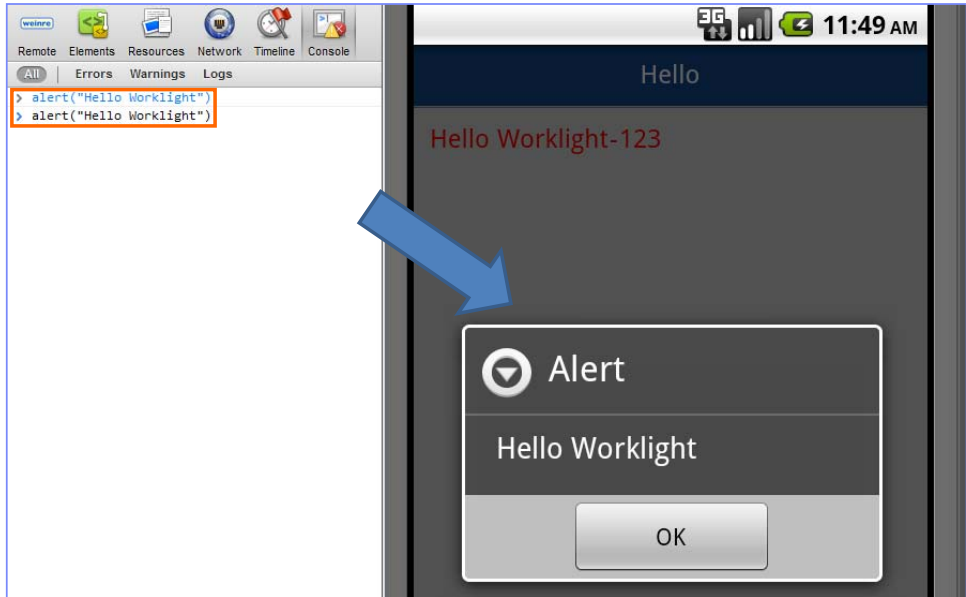
Debugging with Weinre

- You can change values and CSS properties as you debug.



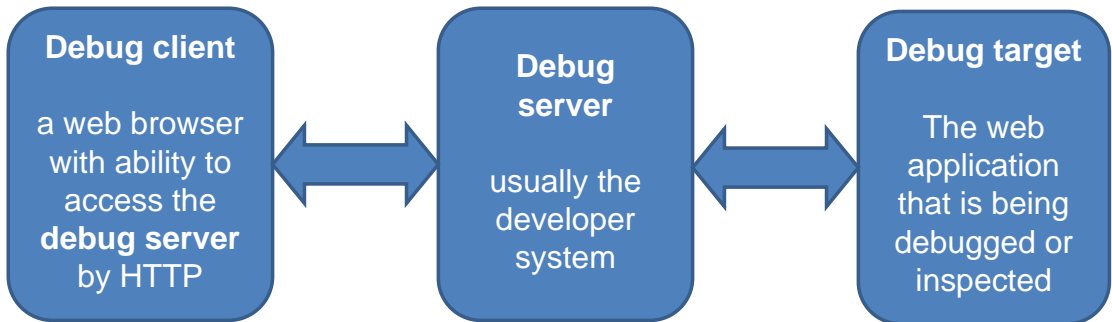
Debugging with Weinre

- You can execute custom JavaScript from a console.



Debugging with Weinre: Set-up

- To set up Weinre, first download it from:
 - <http://people.apache.org/~pmuellr/weinre/>
- The Weinre architecture includes:



- The debug server can run on either a Windows or Mac platform and requires a Java™ Virtual Machine (JVM) to be available.

Debugging with Weinre: Windows Set-up

- To run Weinre from the JAR file found in the downloaded package, run the following command from the command prompt:

```
java -jar weinre.jar --httpPort 8888 --boundHost -all-
```

- This command starts a Weinre server on port 8888:

```
C:\dev\weinre>java -jar weinre.jar --httpPort 8888 --boundHost -all-  
2011-11-07 14:08:17.993:INFO::jetty-7.x.y-SNAPSHOT  
2011-11-07 14:08:18.022:INFO::Started SelectChannelConnector@0.0.0.0:8888  
2011-11-07 14:08:18.023:INFO:weinre:HTTP server started at http://localhost:8888
```

- You must specify port 8888 because the default Weinre port 8080 collides with the one that is used by the Worklight Server.

Debugging with Weinre: Mac Set-up

- To configure the Weinre debug server, create a **`~/.weinre/server.properties`** file.
- Add the following lines to the created file:
 - `boundHost: -all-`
 - `httpPort: 8888`
- Start Weinre Server .app from the downloaded Mac Package.
- You must specify port 8888 because the default Weinre port collides with the one that is used by the Worklight Server.

Debugging with Weinre: Debug Target

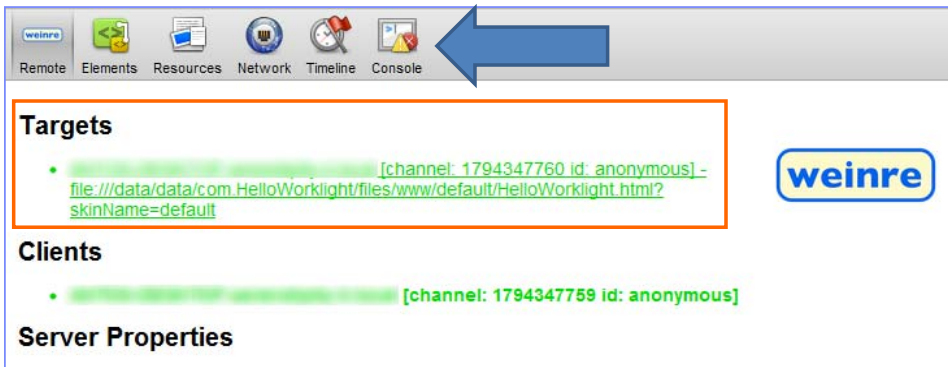
- Make sure that the Weinre server that you set up is accessible from the device that you are using for debugging.
- Add the following line to your web application (Debug Target):

```
<script src='http://192.168.53.40:8888/target/target-script-min.js'></script>
```

- Change the server URL to the one that you are working with.

Debugging with Weinre: Debug Client

- Browse to: <http://<your-server-url-or-ip>:8888/client/>
- Open the web application that you want to debug (the one to which you added a Weinre <script> element).
- Wait for the **Debug Target** to connect, and start debugging with Weinre.



Debugging with Weinre

- You can find more information, including installation manual, running instructions and educational videos, at the Weinre home page.
 - See <http://people.apache.org/~pmuellr/weinre/>

Agenda

- About
- Debugging on a desktop browser
- Debugging with IBM Worklight debugger
- Testing the adapter procedures
- Debugging with Weinre
- Debugging with iOS Remote Web Inspector
- Debugging with WireShark
- Debugging with Mobile Browser Simulator

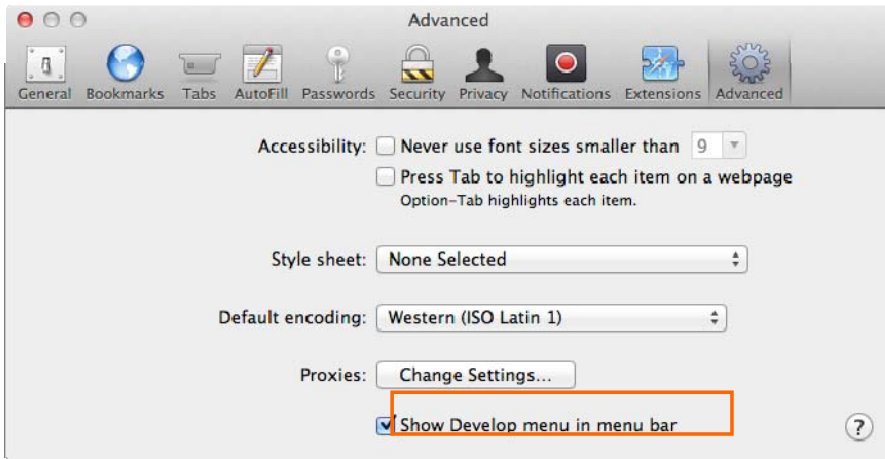
Debugging with iOS Remote Web Inspector

- In iOS6, Apple introduced a remote web inspector for debugging web applications on iOS devices.
- To enable it, go to **Settings > Safari > Advanced** and turn the Web Inspector **ON**.



Debugging with iOS Remote Web Inspector

- Connect the iOS device to a Mac. You must have Safari version 6.0 or higher.
- In Safari **Preferences > Advanced**, select the **“Show Develop menu in menu bar”** check box.



Debugging with iOS Remote Web Inspector

- Go to Safari > Develop > your iDevice ID > your application HTML file.
- Now you can inspect the DOM. You can modify CSS, and run JavaScript commands, just like in the desktop inspector.

The screenshot displays the Web Inspector interface for an iPad2. The left pane shows the DOM tree with the following HTML structure:

```

<!DOCTYPE html>
<!-- Licensed Materials - Property of IBM 5725-G92 (C) Copyright IBM Corp. 2006, 2012. All Rights Reserved. US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp. -->
<html>
  <head></head>
  <body id="content" onload="WL.Client.Init({})" style="display: block;">
    <div class="tab" id="feedTab" style="display: none;">
      <div class="tab" id="feedDescriptionTab" style="display: none;">
        <div class="tab" id="aboutTab" style="display: none;">
          </div>
          <script src="js/worklightStarter.js"></script>
          <script src="js/Utils.js"></script>
          <script src="js/messages.js"></script>
        </body>
      </html>
    
```

The right pane shows the remote browser view of the application, displaying a news feed with various articles. The interface includes a toolbar at the top with navigation and development tools icons.

Agenda

- About
- Debugging on a desktop browser
- Debugging with IBM Worklight debugger
- Testing the adapter procedures
- Debugging with Weinre
- Debugging with iOS Remote Web Inspector
- **Debugging with WireShark**
- Debugging with Mobile Browser Simulator

Debugging with Wireshark (1 of 4)

- Wireshark is a network protocol analyzer that you can use to see what happens on your network at a microscopic level.
- With Wireshark, you can filter the range of traffic that you want to follow.

The screenshot shows the Wireshark interface with a capture filter set to `http && tcp.port == 10080`. The packet list pane displays the following data:

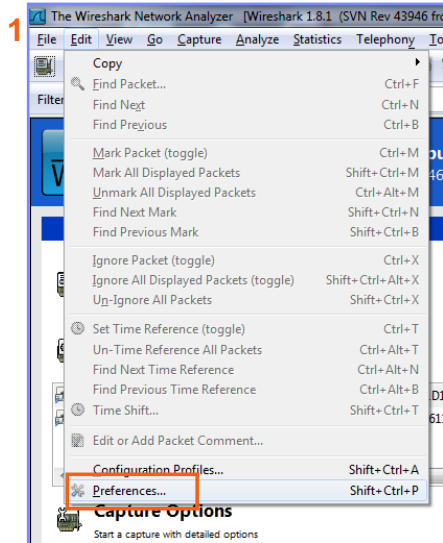
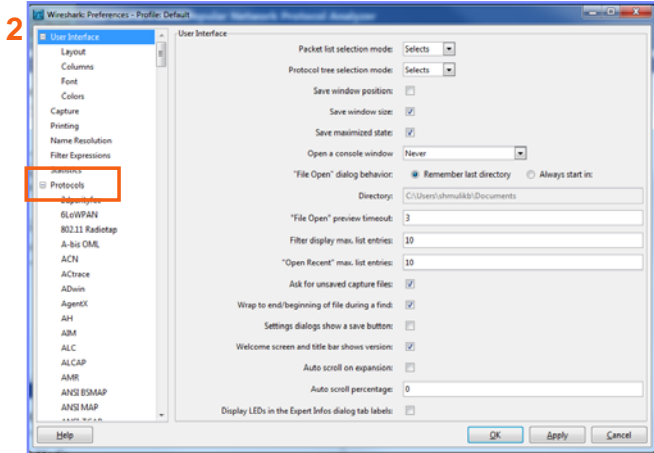
No.	Time	Source	Destination	Protocol	Length	Info
11007	300.268344	9.148.225.225	9.148.23.122	HTTP	518	HTTP/1.1 403 Forbidden (application/json)
11228	349.498153	9.148.23.122	9.148.225.225	HTTP	482	GET /zzz/apps/services/api/zzz/android/an
11230	349.507158	9.148.225.225	9.148.23.122	HTTP	71	HTTP/1.1 403 Forbidden (text/html)
11282	358.387361	9.148.23.122	9.148.225.225	HTTP	385	POST /pushNotificationsProject/apps/servi
11288	358.405062	9.148.225.225	9.148.23.122	HTTP	523	HTTP/1.1 200 OK
11806	459.279102	9.148.23.122	9.148.225.225	HTTP	447	GET /pushNotificationsProject/apps/servi
11807	459.281675	9.148.225.225	9.148.23.122	HTTP	407	HTTP/1.1 200 OK (text/html)
11810	459.322210	9.148.23.122	9.148.225.225	HTTP	188	POST /pushNotificationsProject/apps/servi
11814	459.331383	9.148.225.225	9.148.23.122	HTTP	343	HTTP/1.1 200 OK
12019	467.480230	9.148.23.122	9.148.225.225	HTTP	94	POST /pushNotificationsProject/apps/servi
12021	467.343988	9.148.225.225	9.148.23.122	HTTP	417	HTTP/1.1 200 OK (application/json)
12026	467.625773	9.148.23.122	9.148.225.225	HTTP	188	POST /pushNotificationsProject/apps/servi
12028	467.628632	9.148.225.225	9.148.23.122	HTTP	621	HTTP/1.1 401 unauthorized (application/j
12031	467.686034	9.148.23.122	9.148.225.225	HTTP	190	POST /pushNotificationsProject/apps/servi
12034	467.715583	9.148.225.225	9.148.23.122	HTTP	2230	HTTP/1.1 200 OK (application/json)

The packet details pane for the selected packet (No. 12034) shows:

- Frame 10706: 385 bytes on wire (3080 bits), 385 bytes captured (3080 bits) on interface 1
- Ethernet II, Src: Cisco_Bf:16:3f (00:22:55:bf:16:3f), Dst: wistroni_ce:b2:ca (f0:de:f1:ce:b2:ca)
- Internet Protocol Version 4, Src: 9.148.23.122 (9.148.23.122), Dst: 9.148.225.225 (9.148.225.225)
- Transmission Control Protocol, Src Port: 56781 (56781), Dst Port: amanda (10080), Seq: 751, Ack: 1, Len: 319
- [2 Reassembled TCP Segments (1069 bytes): #10705(750), #10706(319)]
- Hypertext Transfer Protocol
- Line-based text data: application/x-www-form-urlencoded

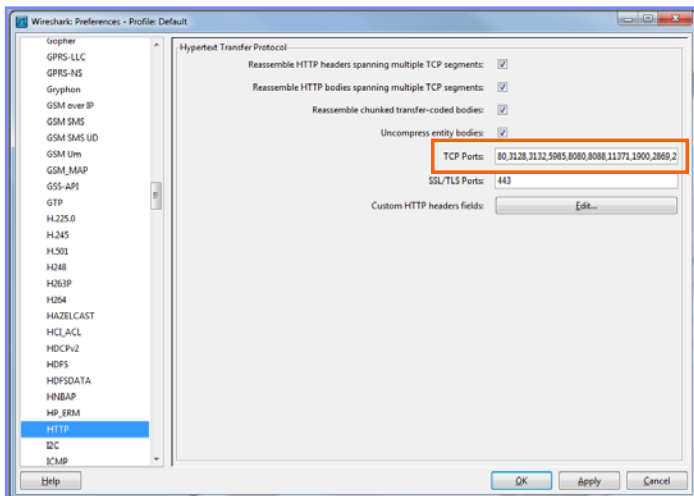
Debugging with Wireshark (2 of 4)

- Open **Edit > Preferences**.
- Expand **Protocols**.



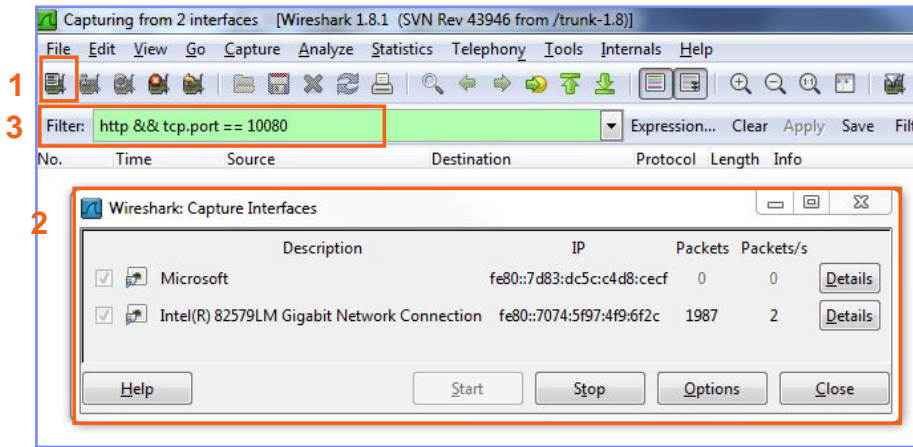
Debugging with Wireshark (3 of 4)

- Select **HTTP** and add port 10080 to the **TCP Ports** list.



Debugging with Wireshark (4 of 4)

1. Click the left button (list the available capture interfaces...)
2. Select the capture interface that you want to follow.
3. Create a filter (protocol && TCP Port).



Agenda

- About
- Debugging on a desktop browser
- Debugging with IBM Worklight debugger
- Testing the adapter procedures
- Debugging with Weinre
- Debugging with iOS Remote Web Inspector
- Debugging with Mobile Browser Simulator

Debugging with the mobile browser simulator

- You can also use the IBM Worklight Mobile Browser Simulator to preview and debug Worklight applications in Android, iPhone, and iPad environments.
- The Mobile Browser Simulator also simulates Cordova APIs, so you can test these APIs without having to use an emulator, simulator, or device.
- Notice that this simulation **does not replace** tests on a physical device.

Notices

- Permission for the use of these publications is granted subject to these terms and conditions.
- This information was developed for products and services offered in the U.S.A.
- IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
- IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
 - IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.
- For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:
 - Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan
- **The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.** Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.
- This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.
- Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
- IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
- Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:
 - IBM Corporation
Dept F6, Bldg 1
294 Route 100
Somers NY 10589-3216
USA

- Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.
- The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.
- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

- This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
- Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:
 - © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp., enter the year or years... All rights reserved.

Privacy Policy Considerations

- IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.
- Depending upon the configurations deployed, this Software Offering may use session cookies that collect session information (generated by the application server). These cookies contain no personally identifiable information and are required for session management. Additionally, persistent cookies may be randomly generated to recognize and manage anonymous users. These cookies also contain no personally identifiable information and are required.
- If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent. For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy>; and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details>; the sections entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Support and comments

- For the entire IBM Worklight documentation set, training material and online forums where you can post questions, see the IBM website at:
 - <http://www.ibm.com/mobile-docs>
- **Support**
 - Software Subscription and Support (also referred to as Software Maintenance) is included with licenses purchased through Passport Advantage and Passport Advantage Express. For additional information about the International Passport Advantage Agreement and the IBM International Passport Advantage Express Agreement, visit the Passport Advantage website at:
 - <http://www.ibm.com/software/passportadvantage>
 - If you have a Software Subscription and Support in effect, IBM provides you assistance for your routine, short duration installation and usage (how-to) questions, and code-related questions. For additional details, consult your IBM Software Support Handbook at:
 - <http://www.ibm.com/support/handbook>
- **Comments**
 - We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.
 - For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.
 - When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state.
 - Thank you for your support.
 - Submit your comments in the IBM Worklight Developer Edition support community at:
 - <https://www.ibm.com/developerworks/mobile/worklight/connect.html>
 - If you would like a response from IBM, please provide the following information:
 - Name
 - Address
 - Company or Organization
 - Phone No.
 - Email address

Thank You

