

Debugging

Chad Michel

10/11/2016

```
public void ConfigureClientBinding(System.ServiceModel.Channels.Binding binding, Type contractType)
{
    if (binding is WS2007HttpBinding)
    {
        var bindingWS = binding as WS2007HttpBinding;

        // Use transport with message creds
        bindingWS.Security.Mode = SecurityMode.Transport;

        if (ServiceHelpersConfigSection.Settings.Endpoint(contractType).WindowsAuth)
            bindingWS.Security.Transport.ClientCredentialType = HttpClientCredentialType.Windows;
        else
            bindingWS.Security.Transport.ClientCredentialType = HttpClientCredentialType.None;

        MaxSetter.SetMaxes(bindingWS);
    }
}
```

If debugging is the process of removing bugs, then programming must be the process of putting them in.

- Edsger W. Dijkstra

Debugging is twice as hard as writing the code in the first place.
Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.

- Brian Kernighan

Pillars of a happy developer

- Great code
- Great team
- Great process
- Great quality

Why do we want happy devs?

Happy Dev is Productive Dev

Pillars of a happy developer

- Great code
- Great team
- Great process
- Great quality

What makes Great Code?

- Easy to navigate
- Easy to run
- Easy to test
- Easy to change
- Easy to debug

Why is debugging important for great code?

- Devs need ability to step thru code
- Devs need to be able to spelunk
- Devs need to be able to find bugs

Debug

- Almost all environments can be debugged
 - Windows
 - .NET
 - Java
 - Ruby
 - Python
 - JavaScript
 - Browser
 - node.js

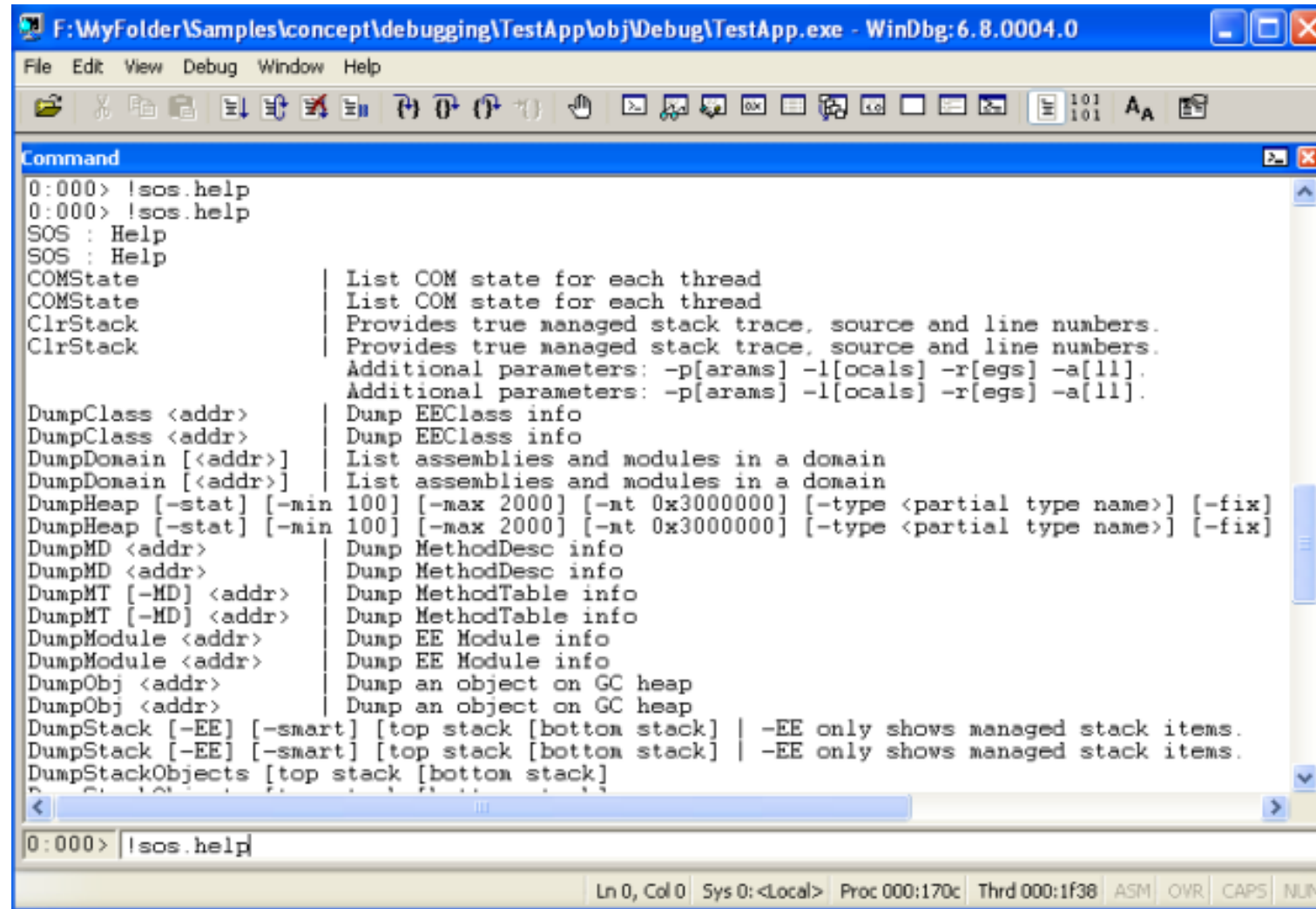
Pillars of a happy developer

- Great code
- Great team
- Great process
- Great quality

Desk checking

- Walk thru all code before submitting it
 - Use the debugger
- Steve McConnell – Code Complete

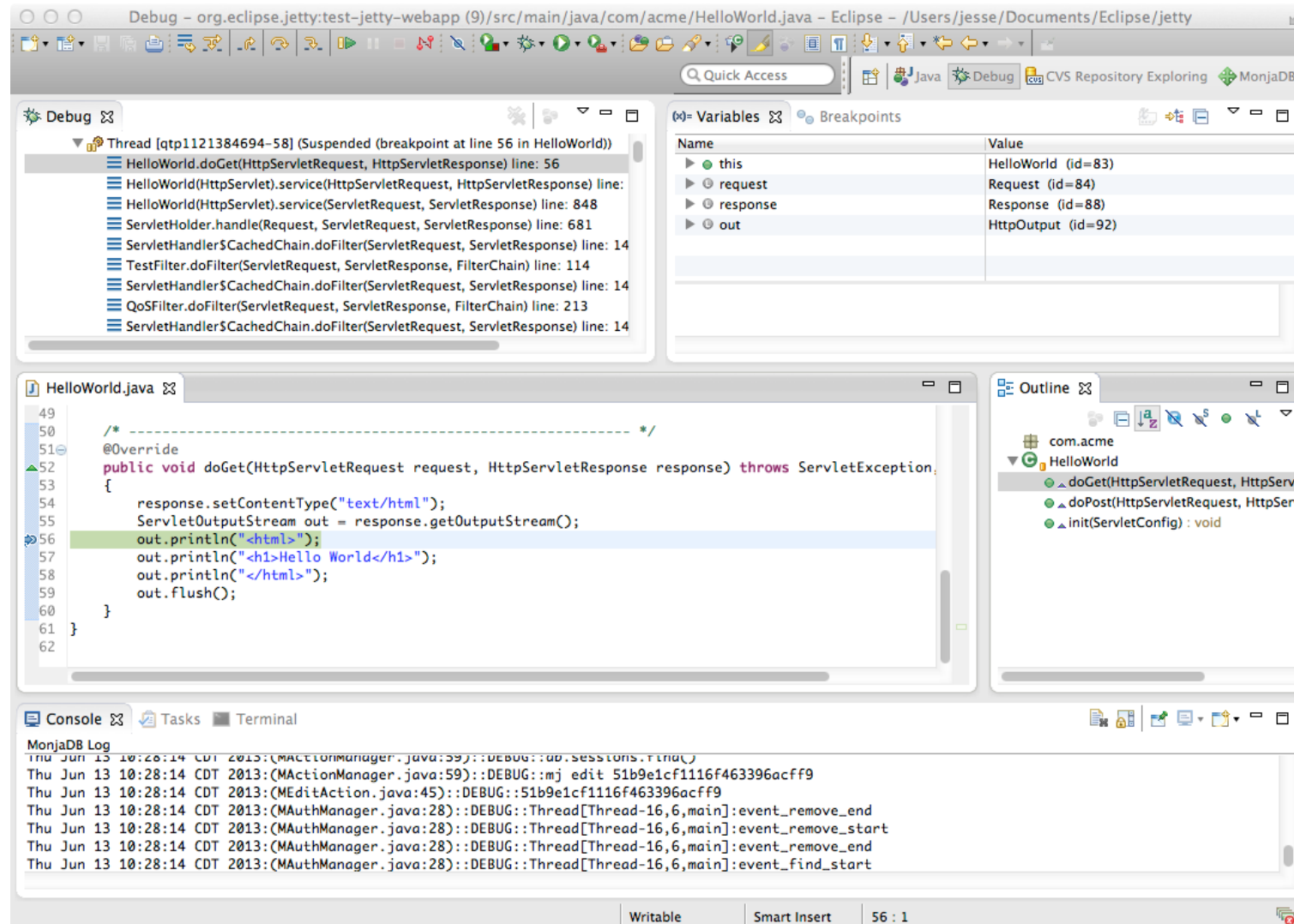
Examples - WinDbg



The screenshot shows the WinDbg 6.8.0004.0 interface. The title bar indicates the file path: F:\MyFolder\Samples\concept\debugging\TestApp\obj\Debug\TestApp.exe. The menu bar includes File, Edit, View, Debug, Window, and Help. The toolbar contains various icons for file operations, debugging, and viewing. The Command window is active, displaying the output of the !sos.help command. The output lists various SOS commands and their descriptions, including COMState, ClrStack, DumpClass, DumpDomain, DumpHeap, DumpMD, DumpMT, DumpModule, DumpObj, DumpStack, and DumpStackObjects. The status bar at the bottom shows: Ln 0, Col 0 Sys 0: <Local> Proc 000:170c Thrd 000:1f38 ASM OVR CAPS NUM.

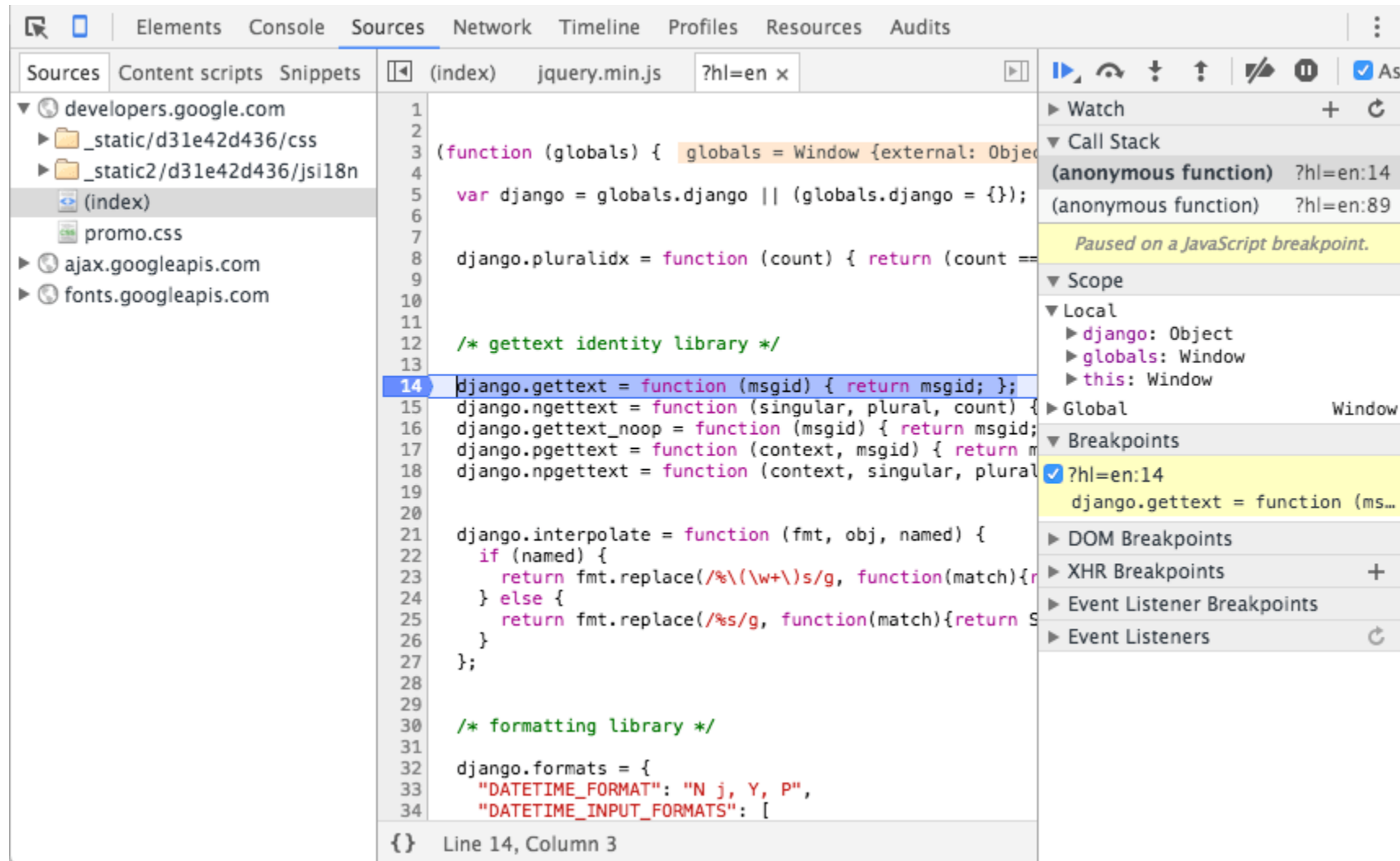
```
F:\MyFolder\Samples\concept\debugging\TestApp\obj\Debug\TestApp.exe - WinDbg:6.8.0004.0
File Edit View Debug Window Help
0:000> !sos.help
0:000> !sos.help
SOS : Help
SOS : Help
COMState          | List COM state for each thread
COMState          | List COM state for each thread
ClrStack          | Provides true managed stack trace, source and line numbers.
ClrStack          | Provides true managed stack trace, source and line numbers.
                  | Additional parameters: -p[arams] -l[ocals] -r[egs] -a[ll].
                  | Additional parameters: -p[arams] -l[ocals] -r[egs] -a[ll].
DumpClass <addr>   | Dump EEClass info
DumpClass <addr>   | Dump EEClass info
DumpDomain [<addr>] | List assemblies and modules in a domain
DumpDomain [<addr>] | List assemblies and modules in a domain
DumpHeap [-stat] [-min 100] [-max 2000] [-nt 0x30000000] [-type <partial type name>] [-fix]
DumpHeap [-stat] [-min 100] [-max 2000] [-nt 0x30000000] [-type <partial type name>] [-fix]
DumpMD <addr>      | Dump MethodDesc info
DumpMD <addr>      | Dump MethodDesc info
DumpMT [-MD] <addr> | Dump MethodTable info
DumpMT [-MD] <addr> | Dump MethodTable info
DumpModule <addr>   | Dump EE Module info
DumpModule <addr>   | Dump EE Module info
DumpObj <addr>      | Dump an object on GC heap
DumpObj <addr>      | Dump an object on GC heap
DumpStack [-EE] [-smart] [top stack [bottom stack]] | -EE only shows managed stack items.
DumpStack [-EE] [-smart] [top stack [bottom stack]] | -EE only shows managed stack items.
DumpStackObjects [top stack [bottom stack]]
0:000> !sos.help
Ln 0, Col 0 Sys 0: <Local> Proc 000:170c Thrd 000:1f38 ASM OVR CAPS NUM
```

Examples - Eclipse



<http://www.eclipse.org/jetty/documentation/9.4.x/debugging-with-eclipse.html>

Examples – Chrome (JavaScript)



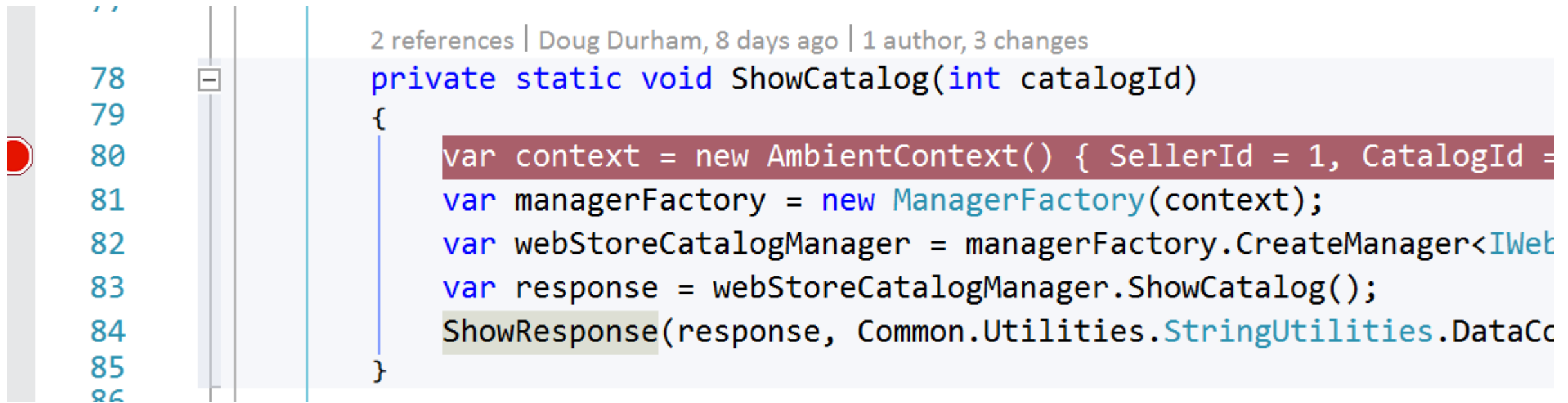
<https://developers.google.com/web/tools/chrome-devtools/images/sources.png>

Common debugging Features

- Breakpoints
- Step into
- Step over
- Watch variables

Breakpoints

- F9 in VS



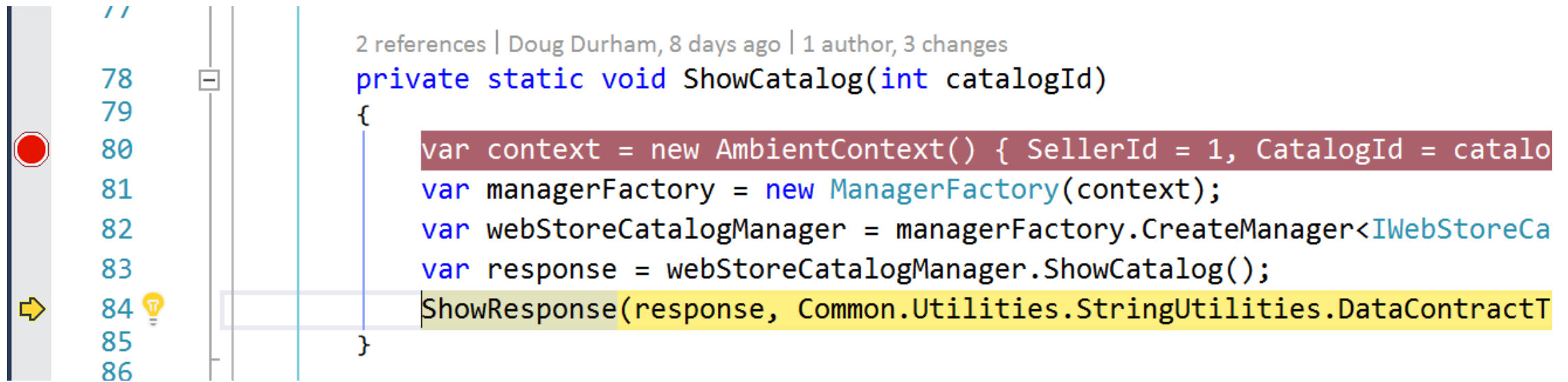
```
78  
79  
80  
81  
82  
83  
84  
85  
86
```

2 references | Doug Durham, 8 days ago | 1 author, 3 changes

```
private static void ShowCatalog(int catalogId)  
{  
    var context = new AmbientContext() { SellerId = 1, CatalogId =  
    var managerFactory = new ManagerFactory(context);  
    var webStoreCatalogManager = managerFactory.CreateManager<IWeb  
    var response = webStoreCatalogManager.ShowCatalog();  
    ShowResponse(response, Common.Utilities.StringUtilities.DataCo  
}
```

Step over

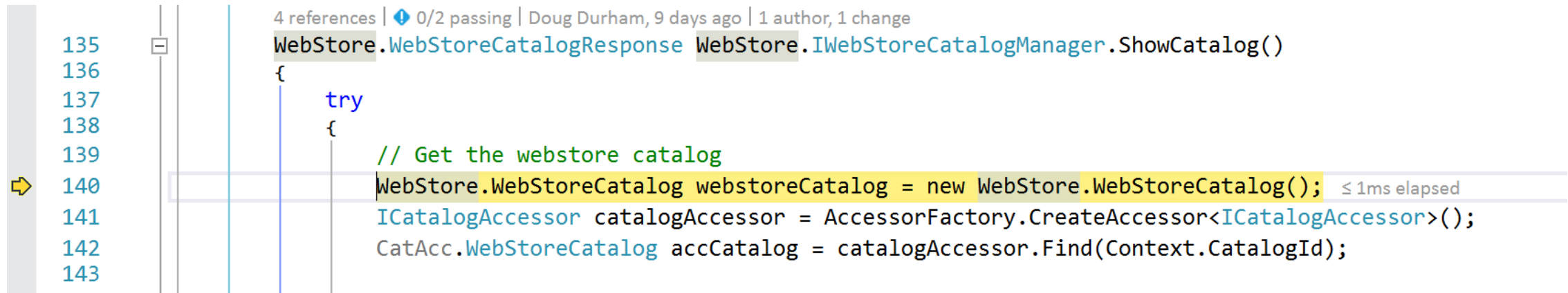
- F10 in VS



```
//  
78  
79  
80  
81  
82  
83  
84  
85  
86  
  
2 references | Doug Durham, 8 days ago | 1 author, 3 changes  
private static void ShowCatalog(int catalogId)  
{  
    var context = new AmbientContext() { SellerId = 1, CatalogId = catalogId };  
    var managerFactory = new ManagerFactory(context);  
    var webStoreCatalogManager = managerFactory.CreateManager<IWebStoreCatalogManager>(context);  
    var response = webStoreCatalogManager.ShowCatalog();  
    ShowResponse(response, Common.Utilities.StringUtilities.DataContractToJson(response));  
}
```

Step Into

- F11 in VS

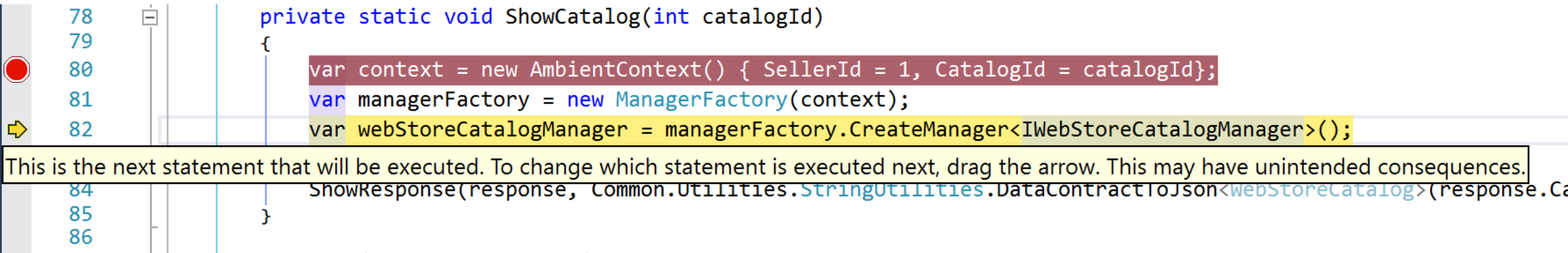


```
135 WebStore.WebStoreCatalogResponse WebStore.IWebStoreCatalogManager.ShowCatalog()  
136 {  
137     try  
138     {  
139         // Get the webstore catalog  
140         WebStore.WebStoreCatalog webstoreCatalog = new WebStore.WebStoreCatalog(); ≤ 1ms elapsed  
141         ICatalogAccessor catalogAccessor = AccessorFactory.CreateAccessor<ICatalogAccessor>();  
142         CatAcc.WebStoreCatalog accCatalog = catalogAccessor.Find(Context.CatalogId);  
143     }
```

Step Out

- Shift + F11

Move current executing line



The screenshot shows a code editor with a method `ShowCatalog`. Line 82 is highlighted in yellow, indicating it is the next statement to be executed. A tooltip at the bottom explains that this is the next statement to be executed and that dragging the arrow can change the execution order, which may have unintended consequences.

```
78 private static void ShowCatalog(int catalogId)
79 {
80     var context = new AmbientContext() { SellerId = 1, CatalogId = catalogId};
81     var managerFactory = new ManagerFactory(context);
82     var webStoreCatalogManager = managerFactory.CreateManager<IWebStoreCatalogManager>();
84     ShowResponse(response, Common.Utilities.StringUtilities.DataContractToJson<WebStoreCatalog>(response.Catalog));
85 }
86
```





This is the next statement that will be executed. To change which statement is executed next, drag the arrow. This may have unintended consequences.

Call stacks

Call Stack	
	Name
➔	DPLRef.eCommerce.Accessors.dll!DPLRef.eCommerce.Accessors.Catalog.CatalogAccessor.Find(int catalogId) Line 24
	DPLRef.eCommerce.Managers.dll!DPLRef.eCommerce.Managers.Catalog.CatalogManager.DPLRef.eCommerce.Contracts.WebStore.Catalog.I
	DPLRef.eCommerce.Client.WebStore.exe!DPLRef.eCommerce.Client.WebStore.Program.ShowCatalog(int catalogId) Line 83
	DPLRef.eCommerce.Client.WebStore.exe!DPLRef.eCommerce.Client.WebStore.Program.Main() Line 25
	[External Code]

Watch

Watch 1

Name	Value
◀  model	{ Catalog = {DPLRef.eCommerce.Accessors.Models.Catalog}, SellerName =
▶  Catalog	{DPLRef.eCommerce.Accessors.Models.Catalog}
 IsApproved	false
 SellerName	"TEST_SELLER"

In line - Watch

```
''  
.FirstOrDefault();
```

```
if (model != null) ≤ 21ms elapsed
```

```
{
```

```
var ca
```

```
DTOMap
```

```
catalog
```

```
catalog
```

```
return catalog;
```

model	{ Catalog = {DPLRef.eCommerce.Accessors.Models.Catalog}, SellerName = "TEST"
▶ Catalog	{DPLRef.eCommerce.Accessors.Models.Catalog}
IsApproved	false
SellerName	"TEST_SELLER"

Uses for debugging

- Find a bug
- Desk checking
- Spelunking...