

Running Scala in the Browser

Scala Days 2010

Wolfgang Kühn
wo.kuehn@enbw.com
EnBW Trading
13 April 2010

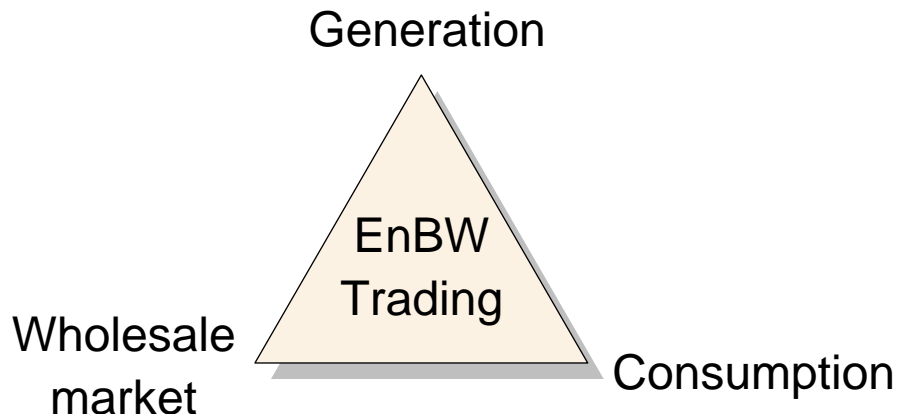


About the EnBW



- > EnBW Energie Baden-Württemberg AG
- > Some six million customers
- > More than 20,000 employees

- > Trading in electricity, gas, coal, oil and CO2
- > Power stations dispatching and fueling
- > Covering consumption by EnBW-Customers



Coding rich web client in Scala is possible now.



Rich web client needs Scripting, i.e. JavaScript

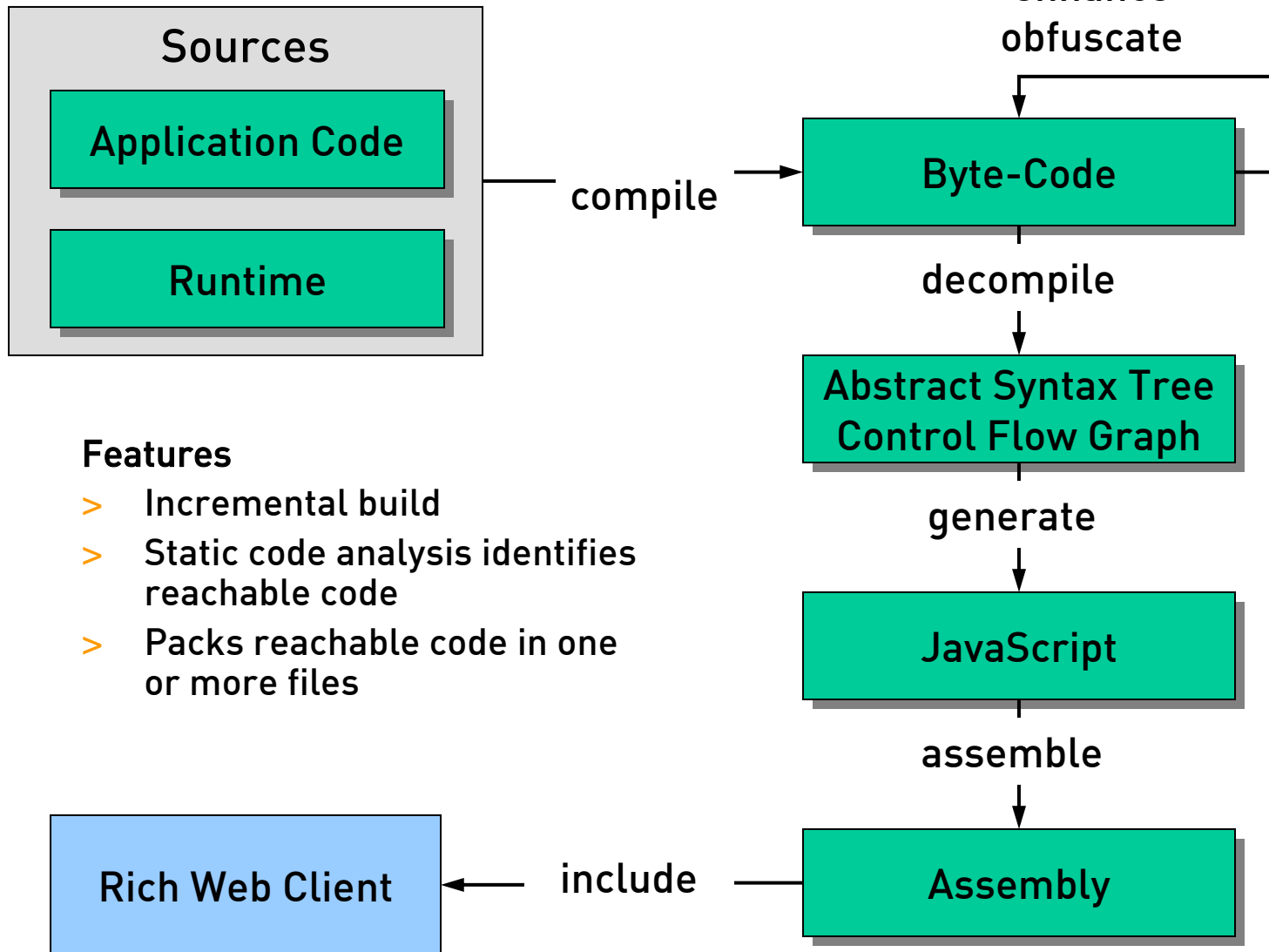
Topics of talk

- > Write client logic in Scala
- > Use Java Byte Code → JavaScript crosscompiler J2JS
- > Separate application logic and layout
- > Use standard interfaces
- > Gives non-trivial example

Related Frameworks

- > Lift
Runs User Interface in Scala on Server
- > GWT
Runs User Interface in Java on Client

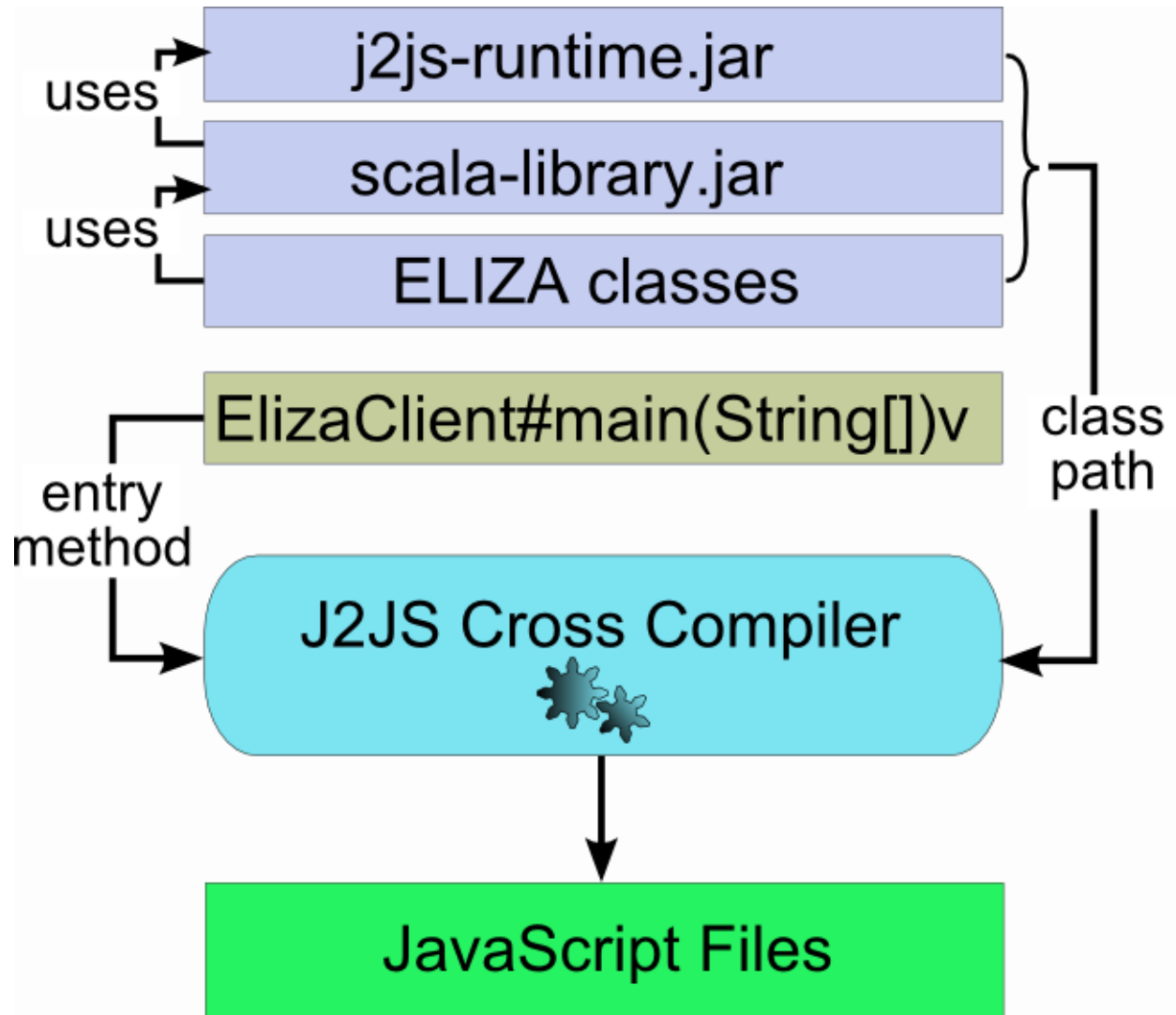
Cross Compilation Cycle



Features

- > Incremental build
- > Static code analysis identifies reachable code
- > Packs reachable code in one or more files

Cross compiler Input/Output



Cross compiled JavaScript methods are not for easy reading

```
_T.dM(
```

```
"<init>(name. pani tz. el i za. El i za$$anonfun$2, scal a. Li st, name. pani tz. el i za. Box)voi d",
```

```
8582, function(l1, l2, l3) {
```

```
  var _=j 2j s;
```

```
  var _0;
```

```
  _.ln=23;
```

```
  L1: {
```

```
    if (l1 != null) {
```

```
      break L1;
```

```
    }
```

```
    _0 = _.nl (559);
```

```
    _.iSp(_0, 559, 11, []);
```

```
    throw j 2j s. nul l SaveExcepti on(_0);
```

```
  }
```

```
  thi s["$outer"] = l1;
```

```
  thi s["key$1"] = l2;
```

```
  thi s["resp$1"] = l3;
```

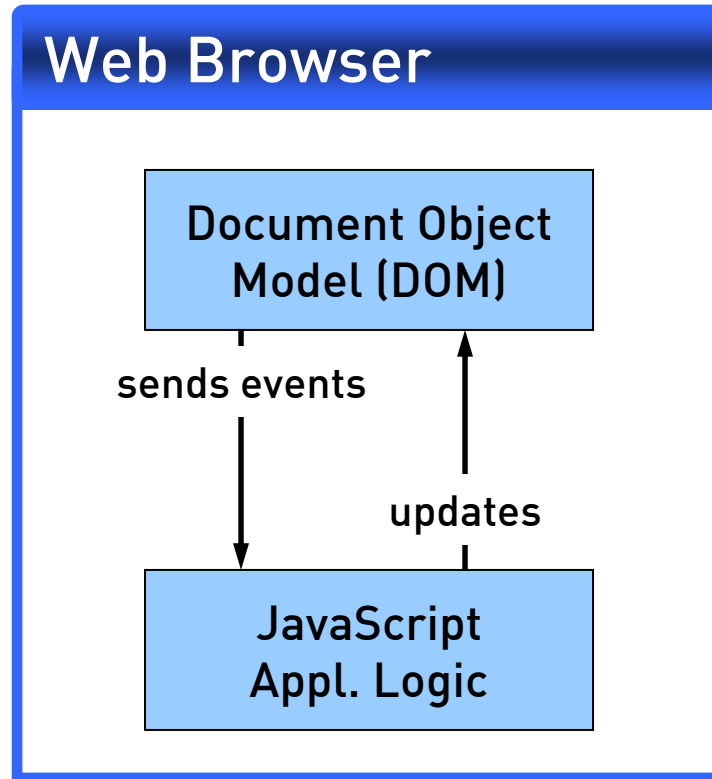
```
  _.iSp(thi s, 19, 11, []);
```

```
  _.iSt(1565, 1566, [thi s]);
```

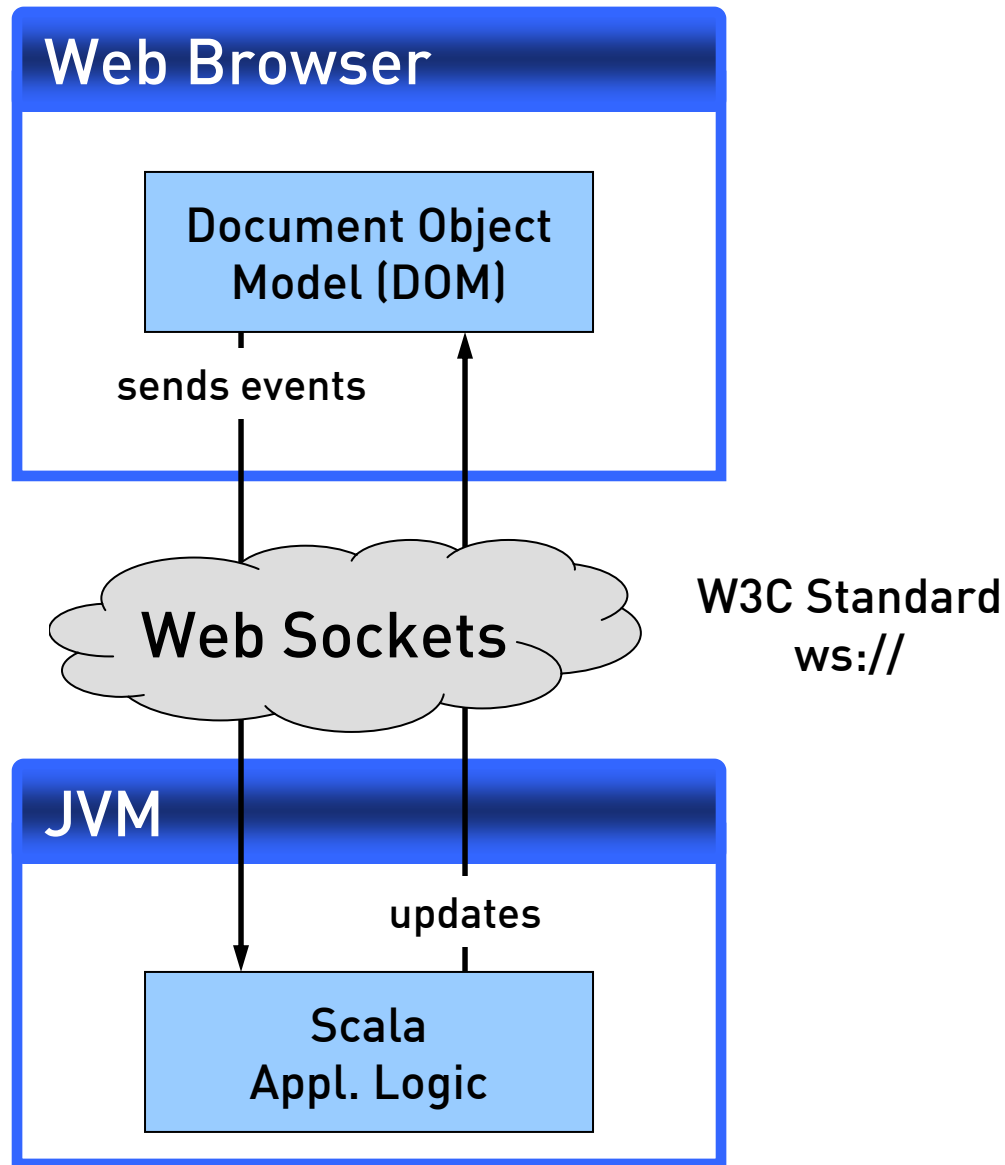
```
});
```

Signature

Production mode is deployed



Developer mode is for debugging and testing



en.wikipedia.org/wiki/ELIZA

ELIZA is a computer program and an early example of primitive natural language processing.

ELIZA operated by processing users' responses ... mostly rephrasing the user's statements as questions.

Panitz, S. E. 2006.

Grundlagen der Kuenstlichen Intelligenz.

<http://www.cs.hs-rm.de/~panitz/ki/skript.pdf>

Show ELIZA online in browser(s)



The screenshot shows a web browser window with the title "Scala Demo". The address bar contains the file path: `file:///D:/workspace/j2js-scala-demo/target/j2js-scala-demo-0.3/Eliz`. The main content area displays the following:

Welcome to ELIZA, your personal psycho therapist

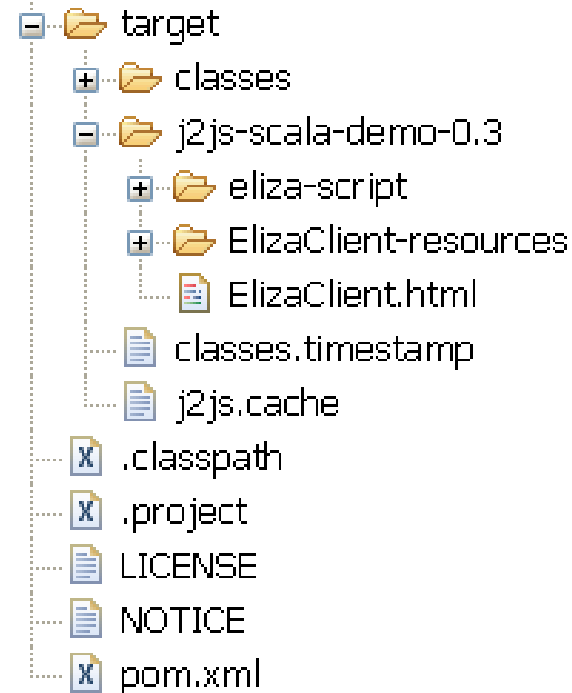
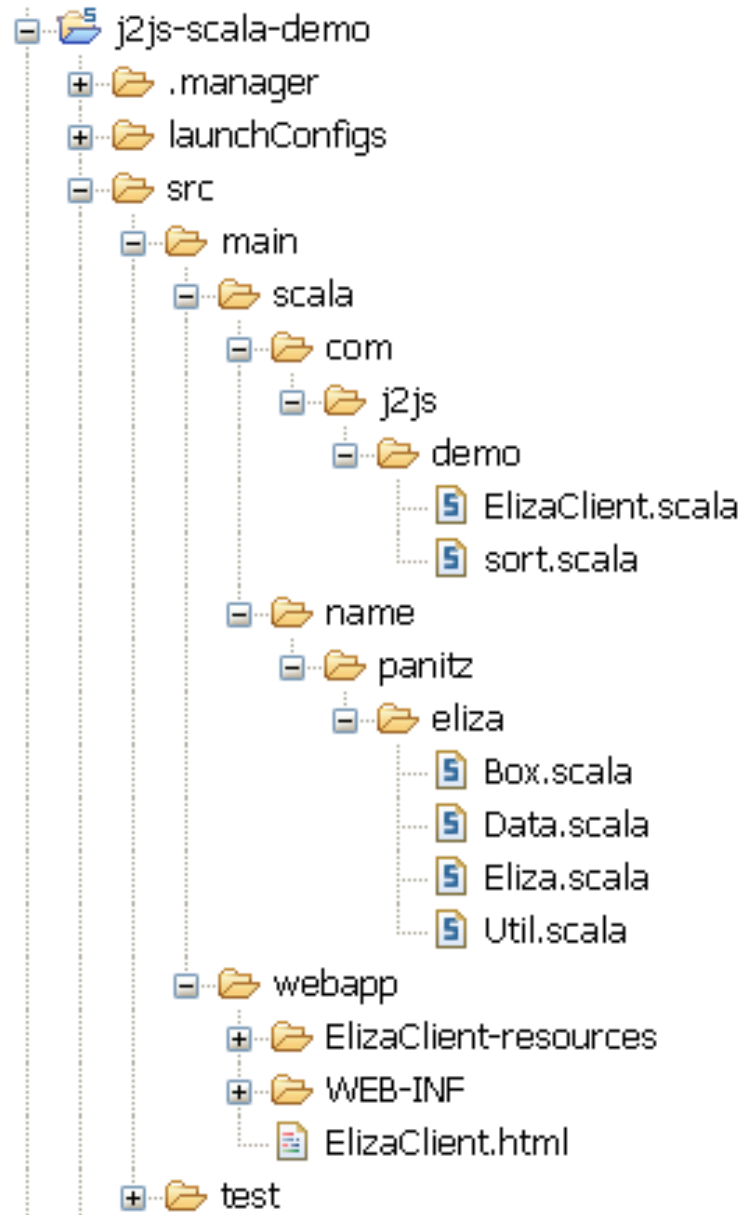
Hi!

How do you...please state your problem.

Tell

The interface includes a cartoon dog character on the left and a cartoon woman character on the right. The text boxes are styled to look like speech bubbles or input fields.

Maven Project Layout



Eliza online

Scala Demo

file:///D:/workspace/j2js-scala-demo/target/j2js-scala-demo-0.3/Eliz

Welcome to ELIZA, your personal psycho therapist

Hi!

`<textarea id="QUESTION"/>`



How do you...please state your problem.

`<textarea id="ANSWER"/>`

`<button id="TELL"/>`

Tell



Scala Client: Imports

```
package com.j2js.demo;

import _root_.javax.Global
import org.w3c.dom.Element
import org.w3c.dom.events.Event
import org.w3c.dom.events.EventTarget
import org.w3c.dom.events.EventListener
import org.w3c.dom.events.KeyboardEvent
import org.w3c.dom.html2.HTMLTextAreaElement
import org.w3c.dom.html2.HTMLInputElement
```

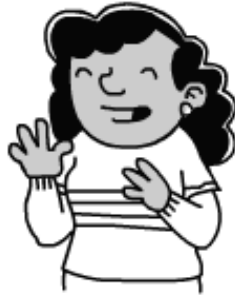
Scala Client: DOM Binding

```
object ElizaClient {  
  
  // Initialize the global context.  
  Global.init("ElizaClient")  
  
  // Short cut to access a DOM element by id.  
  def $(id: String) = Global.document.getElementById(id)  
  
  val questionElem = $(„QUESTION“).asInstanceOf[HTMLTextAreaElement]  
  val answerElem   = $(„ANSWER“).asInstanceOf[HTMLTextAreaElement]  
  val button       = $(„TELL“)  
  val eliza        = new name.pani tz. el i za. El i za()  
  
  // Entry method.  
  def main(args: Array[String]) {  
    setEventHandler(button, "click", answerToQuestion);  
  }  
  
  def answerToQuestion(evt: Event) {  
    val question = questionElem.getValue()  
    val answer   = eliza.eval(question)  
    answerElem.setValue(answer)  
  }  
}
```

Scala Client: W3C Event Handling

```
// Set event handler for specified element.
def setEventHandler(elem: Element,
                   evtType: String,
                   handler: Event => Unit) {
  // Wrap handler function in w3c EventListener.
  val listener = new EventListener {
    override def handleEvent(evt: Event) = handler(evt)
  }
  elem.asInstanceOf[EventTarget]
    .addEventListener(evtType, listener, false);
}
}
```

Debugging Example in Development Mode



The screenshot displays an IDE interface during a debugging session. The top toolbar includes icons for 'Debug', 'Suspend', 'Resume', 'Step Over', 'Step Into', 'Step Out', and 'Run'. The 'Debug' console shows a list of threads: 'Thread [qtp33341602-13] (Running)', 'Thread [DestroyJavaVM] (Running)', and 'Thread [Thread-9] (Suspended)'. The 'Variables' window shows two variables: 'question' and 'answer', with the value 'How do you...please state your problem.' displayed below. The source code editor shows the following code:

```
33  
34 def answerToQuestion(evt: Event) {  
35     val question = questionElem.getValue()  
36     val answer = eliza.eval(question)  
37     answerElem.setValue(answer)  
38 }
```

A red arrow points from the 'Tell' button to the breakpoint at line 37.

ELIZA in Numbers

Number of compiled classes	179
Number of assembled methods	978
Size of JavaScript assembly	229 45 kB
Time to cross compile	20 sec
Number of Calls per Click	14326

Number of calls per click is to massive



```
953 java.lang.String#length()int
889 java.lang.String#charAt(int)char
695 java.lang.StringBuffer#append(char)java.lang.StringBuffer
695 java.lang.String#valueOf(char)java.lang.String
695 java.lang.StringBuffer#append(java.lang.String)java.lang.StringBuffer
624 java.lang.Object#<init>()void
494 scala.collection.mutable.ListBuffer#scala.collection.mutable.ListBuffer$$$start()scala.List
295 scala.collection#head()java.lang.Object
295 scala.collection#hd()java.lang.Object
294 scala.PartialFunction$class#init$(scala.PartialFunction)void
294 scala.Iterable$class#init$(scala.Iterable)void
294 scala.Collection$class#init$(scala.Collection)void
294 scala.Seq$class#init$(scala.Seq)void
255 scala.collection#tail()scala.List
255 scala.collection#tl()scala.List
201 scala.Product$class#init$(scala.Product)void
141 scala.collection.mutable.ListBuffer#<init>()void
141 scala.collection.mutable.ConeableCollection$class#init$(scala.collection.mutable.ConeableCollection)void
141 scala.collection.mutable.Buffer$class#init$(scala.collection.mutable.Buffer)void
141 scala.runtime.BoxedArray$AnyIterator#init$(int)int
141 scala.runtime.BoxedArray$AnyIterator#scala.runtime.BoxedArray$AnyIterator$$$outer()scala.runtime.BoxedArray
141 scala.runtime.BoxedObjectArray#valueOf()java.lang.Object[]
140 scala.collection.mutable.ListBuffer#toList()scala.List
140 scala.collection.mutable.ListBuffer#exported_$eq(bool)void
135 scala.runtime.BoxesRunTime#boxToBoolean(bool)java.lang.Boolean
135 scala.runtime.BoxesRunTime#unboxToBoolean(java.lang.Object)bool
135 java.lang.Boolean#booleanValue()bool
106 scala.collection#<init>(java.lang.Object, scala.List)void
106 scala.List#<init>()void
101 scala.StringBuilder#valueOf()char[]
...
```

Outlook

- > Improve Scala tooling
 - Coverage
 - Quality
- > Client widgets written in Scala
- > Reduce footprint of Scala runtime
- > Formulate best practises for Scala on the Client
 - Elegant Data Binding ... the Scala way
 - Remoting (Client/Server communication)
 - Play with embedded XML

Feedback

Thank you for your attention.

Any questions or feedback?

Further information

- > <http://www.j2js.com/scala-demo/index.html>
- > wo.kuehn@enbw.com