Scala.js workshop

- https://github.com/oyvindberg/scalajs-workshop

Øyvind Raddum Berg
@olvindberg

Stein Kåre Skytteren
@skskytteren

Who are we?

What we will cover today

- Very quick intro
- A few limitations to keep in mind
- Building / development
- Look at a few libraries for typed Html, Dom, Ajax
- Hack a bit on an example project

What is Scala.js?

- Write Scala, compile, run in browser
- Full support for the entire Scala language*
- Full javascript interop
- Fast enough (0.9x - 4x times slower than javascript),
- Small enough (hundreds of kilobytes)
- Production-ready since February 2015
- 1.0 scheduled for release summer 2017

What *is* Scala.js?

- Primarily: An output target for the Scala compiler
- An optimising linker (using Google Closure Compiler)
- A Javalib - reimplemented functionality from Java
- A Scalalib - reimplemented functionality from Scala
- A runtime library
- An sbt plugin

Scala.js limitations

- No Java support (source or binary)
- No reflection
- Browser-sandbox (no file system, etc)
- Slightly differing semantics
    - beware of Floats and Longs,
    - Equality tests and toString for primitives
    - Exceptions from VM (ClassCastException, ArithmeticException, etc)

How?

```
Option("Hello") match {
  case Some(msg) ⇒ dom.window.alert(msg)
  case None      ⇒ dom.window.alert("No Message")
}
```

```
var x1 = $m_s_Option$().apply__O__s_Option("Hello");
if ($is_s_Some(x1)) {
  var x2 = $as_s_Some(x1);
  var msg = $as_T(x2.x$2);
  $m_Lorg_scalajs_dom_package$
().window__Lorg_scalajs_dom_raw_Window().alert(msg)
} else {
  var x = $m_s_None$();
  if ((x === x1)) {
    $m_Lorg_scalajs_dom_package$
().window__Lorg_scalajs_dom_raw_Window().alert("No Message")
  } else {
    throw new $c_s_MatchError().init___O(x1)
  }
};
```

Anatomy of a Scala.js project

- Scala.js sbt plugin
- CrossProject
- fastOptJS/fullOptJS
- Dependency handling
- Packaging of Javascript dependencies
- Testing on Node.js

Guided tour of libraries

- scala-js-dom
- scalatags
- autowire
- scalacss?

Assignements

1. Extend the application to show metadata. Last changed? file size? Right now it's pretty bare bones

2. Add support for showing content of files. Such basic functionality missing!. Can you make it happen?

3. Search for files

Our take

- Solves **a lot** of headaches for web dev
- Having access to big parts of the Scala ecosystem is **awesome**
- Young frontend ecosystem, still very influenced by javascript, no “one, true way”
- Interplay between ide, sbt, plugins, compilers and dependencies is complex