



How to migrate and modernize a Java applet to HTML5 <http://www.cincheo.fr>

🔄 14 commits	🌿 1 branch	📦 0 releases	👤 1 contributor
--------------	------------	--------------	-----------------

Branch: **master** [New pull request](#) [Find file](#)

Clone or download

 **renaudpawlak** removed useless ref
Latest commit ad124e7 Jul 3, 2016

📁 src/main/java/applet	add comments	Jun 20, 2016
📁 target	adjustments	Jun 20, 2016
📁 webapp	removed useless ref	Jul 3, 2016
📄 README.md	more info	Jun 20, 2016
📄 pom.xml	adjustments	Jun 20, 2016

📖 README.md

Migrating a Java Applet to HTML5

This project is a proof of concept demonstrating that it is possible to migrate an existing Java Applet to a true HTML5 application. The migrated applet is a simple [calculator applet](#) that was found on the Internet.

Why?

Soon, [Java Applets will be deprecated](#) (read the [official post from Oracle](#)) and Chrome does not support applets anymore.

Applets can be migrated to Java WebStart applications, but they would still rely on outdated technology and users may hit many accessibility and maintainability issues.

Here, we show that it is possible to run an applet in any browser as a pure JavaScript program. We even show that it is possible to truly modernize it by taking advantage of available JavaScript libraries.

How does it work?

We use [JSweet](#) and the [J4TS](#) library to transpile the Java code to JavaScript code. For each class or method of the Applet/AWT API used by the calculator program, we provide an implementation in JSweet (i.e. in JavaScript but with the Java language). Nowadays, most Java APIs have a JavaScript counterpart, so it would be possible to provide an implementation for most Java libraries by wrapping existing JavaScript libraries.

This migration technique has several advantages:

- The Java code is compiled "as is" to plain JavaScript code that can run in any browser.
- There is no VM and no emulation.
 - It means no overhead compared to JavaScript.
 - Also, JavaScript APIs can be accessed directly, which is important for actual modernization.
- The modernized applet is not only better looking, but the code is also more maintainable (view/logic separation).
- One can switch its code base to the generated TypeScript or JavaScript, but can also remain in Java.
- This technique can also apply to plain AWT/Swing and SWT desktop applications.

For more details, check out the [video](#).

How to launch

Just clone the project and open the applet version you want to try with your favorite browser.

- [webapp/index.html](#) to open the original applet (only in Firefox and you must configure Java to allow local applets).
- [webapp/index-no-applet.html](#) to open the JavaScript version of the applet compiled with JSweet (any browser).
- [webapp/index-no-applet-modernized.html](#) to open a modernized version separating the applet's logic and the HTML view.
- [index-no-applet-modernized-bootstrap.html](#) to open a better-looking HTML5 version using the Bootstrap framework.

How to build

You need to have Node.js and Maven installed.

```
> mvn clean generate-sources
```