

# Ideas for Wicket 8.0

- [This page](#) lists ideas what can be improved for Wicket 8.0.
  - [Depend on Java 8](#)
  - [AppCache for Offline-Support](#)
  - [Better monitoring \( WICKET-6120 - Getting issue details... STATUS \)](#)
  - [Better SEO for stateful pages](#)
  - [Basic components for mobile devices based on HTML5](#)
  - [Server Push with HTTP2 \( WICKET-6194 - Getting issue details... STATUS \)](#)
  - [Update WicketTester assertions](#)
  - [New components](#)

## This page lists ideas what can be improved for Wicket 8.0.

These are only ideas. Some of them may not be implemented though. There will be a separate Roadmap page that will have only the ideas on which we agreed.

To be approved an idea has to be discussed first in dev@ mailing list. A patch and/or Pull Request increases the chance to get your idea in.

### Depend on Java 8

Make use of Java 8 features wherever it makes sense.

- `java.util.Optional` for `AjaxFallback**` [WICKET-6104 - Getting issue details... STATUS](#)
- new `DateTime` APIs for `wicket-datetime` [WICKET-6105 - Getting issue details... STATUS](#)
- `Lambda` API for `Listeners` (`onClick`, etc.) **Done. See `Lambdas.java`**

### AppCache for Offline-Support

- <http://www.html5rocks.com/en/tutorials/appcache/beginner/>
- <http://www.w3.org/TR/2011/WD-html5-20110525/offline.html>

See: [https://github.com/klopfdreh/wicket-components-playground/wiki/20.-HTML5-OfflineMode-Implementation-on-hold-\(with-ServiceWorker\)](https://github.com/klopfdreh/wicket-components-playground/wiki/20.-HTML5-OfflineMode-Implementation-on-hold-(with-ServiceWorker))

Also See: <https://github.com/wicketstuff/core/tree/master/offline-mode-parent> (Integration into Wicketstuff - with example)

Stays in `wicketstuff` repo as long as there is not enough browser support (<http://caniuse.com/#feat=serviceworkers>)

### Better monitoring ( [WICKET-6120 - Getting issue details... STATUS](#) )

Expose hooks for better monitoring.

For example an integration with something like <https://dropwizard.github.io/metrics/3.1.0/>

### Better SEO for stateful pages

Think of a way of encoding the page id so that it doesn't "pollute" the url.

Some applications currently need to use `NoVersionMapper` (e.g. Apache OpenMeetings) to avoid it.

### Basic components for mobile devices based on HTML5

Support for basic video / audio capture for example.

### Media Components ( [WICKET-6025 - Getting issue details... STATUS](#) )

Generic resource reference to handle with media files located at the file system of the server (out of the box) (also for Wicket 7)

### Server Push with HTTP2 ( [WICKET-6194 - Getting issue details... STATUS](#) )

We could provide *native* integration with different web containers to make use of their HTTP2 Push support, e.g. Tomcat 8.5+ provides *org.apache.catalina.core.ApplicationPushBuilder*.

Possible implementation is to add code similar to `HttpRequest.getPushBuilder().path(encodedUrl).push()` to `org.apache.wicket.markup.head.JavaScriptHeaderItem#internalRenderJavaScriptReference()`.

The problem is how to decide whether to push a resource or not, because the browser may already have it cached, so the push will be a waste of bandwidth (<https://http2.github.io/faq/#how-can-i-use-http2-server-push>).

Articles explaining possible solutions for this problem: <https://www.mnot.net/blog/2016/04/22/ideal-http>, <https://mariusgundersen.net/module-pusher/>

### **Status**

Tomcat: <https://github.com/apache/wicket/tree/master/wicket-experimental/wicket-http2/wicket-http2-tomcat> (Mailing list request: <https://www.mail-archive.com/users@tomcat.apache.org/msg121670.html>)

Jetty: <https://github.com/apache/wicket/tree/master/wicket-experimental/wicket-http2/wicket-http2-jetty>

Undertow: <https://github.com/apache/wicket/tree/master/wicket-experimental/wicket-http2/wicket-http2-undertow>

### **Implementation**

<https://github.com/apache/wicket/tree/master/wicket-experimental/wicket-http2>

### **Note**

When the official servlet 4.0 API has been finished and all servers are using it this line of code: <https://github.com/apache/wicket/blob/master/wicket-experimental/wicket-http2/wicket-http2-core/src/main/java/org/apache/wicket/http2/markup/head/PushHeaderItem.java#L91> has to be changed to the standard interface

## **Update WicketTester assertions**

Take inspiration from JUnit 5 and improve WicketTester & Co.

## **New components**

### **StreamView**

A component similar to ListView but using `java.util.stream.Stream` instead of `java.util.List` as a model object.

The idea is to be as lazy as possible. At render time the Stream will be "collected".