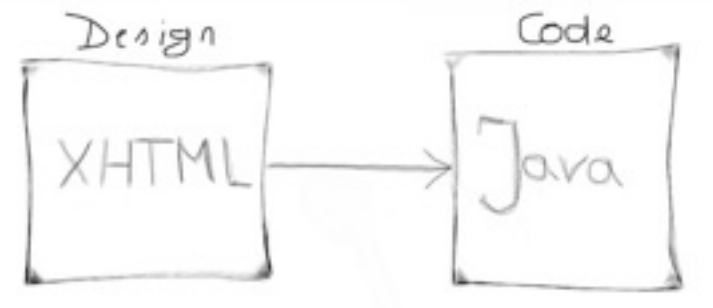
# JSXP Framework

Autor(en): Oliver Szymanski, Freiberufler, Source-Knights.com
David Tanzer, Freiberufler, davidtanzer.net

The JSXP framework is a web application framework that simplifies the use for developers. So it easy to learn and reduces development costs.

JSXP means use HTML (XHTML) as design, code **only** in Java.



1. Create your design.
2. Auto-generate Java code.
3. Code your view logic.
4. Update design with compile time safety.
5. Go on :)

You can find the distribution, documentation and the example web application with tutorials on http://www.jsxp.org.

## Features

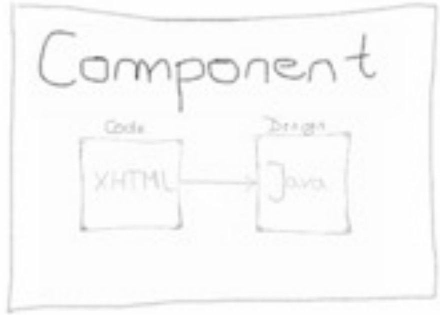
- **Separation of code and design**
  Design and code are totally separated. Java developers must not have knowledge of HTML.
  The design of your views is simple XML code (XHTML for example). In this XML code you specify the elements you want to access in your controller using a special id-attribute from the JSXP namespace. You can also define Variables - the Generator will create set-Methods for them. All the programming is done in Java-source-files. There is no such thing like JSP's expression language, which adds more possibilites for runtime errors. Also there are no special xml elements like in Wicket, which can not be generated using

HTML tools.
Enables html designers and java developers to work together on the same source base in an agile development process.

- **Compile-time safety**
  In the controllers for your views, all elements of the views are accessed using generated get-Methods. This ensures that you can only refer to elements which really exist in the XHTML file. When an element is re-named or deleted, the java compiler immediately reports an error. This means you do not have to wait until your application is running to see if all the referenced elements still exist (this would be the case with other web frameworks for java: You get a runtime error).

- **Simple XHTML design**
  JSXP allows your designer to create complete, fully functioning HTML pages. All the links can already refer the correct pages, and all the pages can contain mock-ups for navigation, header,... . This means the designs render nicely in the browser, and all the links are already clickable. Later you can programmatically apply view templates (which will remove all the mock-ups from your XHTML designs). This also means that you can work together with the designer on the same XHTML files, and all the changes made by the designer are directly reflected in your application.
  Your webapp is always testable / in a demo state and all pages can be previewed in a browser.
  You can exchange static content step by step by dynamic content.

- **Component-Orientation**
  JSXP supports creating components an composing your views from the components. There are powerful mechanisms for view-templating and element-templating. And they are just used via Java code, nothing hidden in the design. This allows us to have no special XML elements.



- **Human readable URLs**
  When the URL in your browser says "index.xhtml", you really see the index page in JSXP by default (of course this can be changed by the business logic of the index page, but this is discouraged). This even works in combination with redirects. This means that the URLs are not only bookmarkable by default, they are also human readable. Both things are rather difficult to achieve with other Java web frameworks.
  You can also access folders as URIs in the browser and the framework will resolve to an index view per default.

- **Server side state**
  JSXP manages your server side state. There are several different scopes which can contain your data, and the contents of these scopes are managed automatically. You only have to annotate a get- or set- method to manage an object in one of the scopes. For example, the following code will put a String into the session scope:
  `@SessionScope`

```
•   public String getSessionScopeString() {
•        return sessionScopeString;
•   }
•   public void setSessionScopeString(String
    sessionScopeString) {
•        this.sessionScopeString = sessionScopeString;
•   }
•
```

  You can also access the scopes in your Java code by calling the always accessable JSXP Context. For example you can use the following to access the flash scope in your code
  `Context.getContext().getScopes().getFlashScope();`

- **Ajax support**
  JSXP has built-in AJAX support and works well with other AJAX libraries, like JQuery or Prototype.

- **Internationalization**
  Internationalization is supported and easy to add to your web applications.

- **No special tools**
  Only one tool is needed: the integrated **ViewControllerGenerator**. You do not have to use special IDEs or plugins. You will get the full power of JSXP without any additional tool.