

Getting Started with SnappyJ

Welcome

Welcome to SnappyJ! SnappyJ is designed to allow you to reload sections of your page easily with browser history support, browser reload support, customizable loading indicators, and customizable error responses.

Installation

Installing SnappyJ takes a couple of simple steps.

- Place the lib/snappyj-<version>.jar in your <WebRoot>/WEB-INF/lib.
- Add the following entries into your <WebRoot>/WEB-INF/conf:

```
<servlet>
  <servlet-name>ServletForwarder</servlet-name>
  <servlet-class>com.getsnappy.w20.ServletForwarder</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ServletForwarder</servlet-name>
  <url-pattern>/servletForwarder</url-pattern>
</servlet-mapping>
```

- Place the js/snappyj-<version>.js anywhere inside your <WebRoot>.

Configuration

There is usually no configuration required. See the javadoc for the class `com.getsnappy.snappyj.Configuration` for more details

Introduction to SnappyJ

In order to understand how to use SnappyJ, we must introduce a couple terms, contexts and beans. Contexts represent sections of your page that are reloadable. Each context has an application associated with it. An application is what actually performs programming logic and outputs content. Dynamic contexts have the ability to be forwarded to any number of applications, while static contexts are permanently associated with a single application. Beans are like a bridge between client browsers and applications. All communication between your browser and reloadable applications are done using beans.

Using SnappyJ

In order to include SnappyJ within a specific page, perform the following:

- load the snappyj-<version>.js javascript file, e.g.:

```
<html>
```

```
<head>
  <title>Example Title</title>
  <script type="text/javascript" src="/snappyj-<version>.js"></script>
</head>
```

- create your beans and contexts
- instantiate a SnappyJEnvironment with these beans and contexts. See the javadoc for the class `com.getsnappy.snappyj.SnappyJEnvironment` for more details.
- use the `Context.include` or `Context.includeAsync` to include contexts within jsp pages, servlets, or other contexts.
- use `jsXXX` methods within script tags or javascript events.
- use `Bean.jsSet` to set bean values within script tags or javascript events.
- use `Context.jsReload` to reload a context asynchronously.
- use `DynamicContext.jsForward` to reload a context asynchronously using a different application.
- use `SnappyJEnvironment.jsSetTitle` to set the page title
- use `SnappyJEnvironment.jsAddToHistory` to add an entry in your browsers history with your current state.