VelocityTools2 Config

Configuration Overview

The VelocityTools2 support infrastructure exists to provide all your templates a common set of tools and data. This is inspired by the Pull MVC model, which deviates from the strict MVC purist approach out for the sake of convenience and clarity. The goal here is to provide template authors a common interface of data and functions across all templates (we call this a "toolbox"), whether they need all of those functions and data or not. This saves the template author from having to remember what keys were used where and makes it easy to drop a new template (i.e. View) into an app without having to modify the controller (which would typically involve creating a new action class). The degree to which this Pull MVC pattern violates MVC principles can (and should) vary widely depending on your needs and goals.

The default VelocityTools2 configuration does not include any "gross MVC offenders", as such things would be hard to generalize usefully. The default configuration primarily includes tools for manipulating values made available in the template's context by a controller and a few for accessing static resources.

However, it is likely that you will want to add your own data and tools to your VelocityTools2 configuration or at least want to change the default settings of the standard tools. To that end, configuration of your applications "toolbox(es)" can be done via XML, Java or properties. Different configurations can also be easily combined together.

There a few basic concepts to the configuration that it is useful to know. First, what you are creating a configuration for is a ToolboxFactory. This factory produces your toolbox(es) as needed by VelocityView or your own application. A factory can have any number of toolboxes, all distinguished by their scope property. There are three special scopes automatically recognized by VelocityTools2: "request", "application", and "session". The "session" scope is only relevant within a VelocityView app, but the other two may be useful anywhere.

When the "application" toolbox is requested, the ToolboxFactory will also include any "data" configured for it. These are unchanging, static values that are meant to be available to all templates in your application. You can configure any number of data for your application and the configuration supports both automatic and explicit type conversion (via Commons-BeanUtils converters) for these values (since XML and properties formats only allow string inputs).

Other things to know are that each toolbox can have any number of tools within it, and that "properties" may be added for any and all tools, toolboxes, and the factory itself. A "property" added to one of these also has all of the same type conversion support as the "data" values do. Properties set on a toolbox are made available to all tools within that toolbox and properties set for the factory itself are made available to all tools in your application.

Now on to the formats for specifying these things...

- XML
- Properties
- Java