FreeMarker

Benefits of MVC

- Data assembled by controllers can be presented by different views
- Site can be reskinned without touching the application logic
- Application logic can be re-factored without touching the presentation
- UI and Application teams can work with maximum independence

What is FreeMarker

- An all-purpose Java template engine for generating text output based on templates
- Not specifically designed to generate web pages, so text generation is independent of the servlet application architecture

FreeMarker Features

- Separation of application and presentation logic is enforced through:
  - Syntactic limitations of the template language
  - Exposure settings on the template data
- Only FTL is allowed in templates
  - No Java code
- Template language
  - Clean, simple, and consistent
  - Just powerful enough
    - Loops, conditionals, local variables, macros, object property access, convenient built-in functions
- String capture
  - Controller can use the engine to generate a string without relinquishing control of the request
- Flexible and powerful API
- Thorough documentation
  - API documentation
  - Template developer documentation
- Active user mailing list

Two ways to use FreeMarker in VIVO

- Generate an entire page in FreeMarker
  - This is the direction the application is moving in; transition is still incomplete
- Generate a string for use outside a FreeMarker page
  - In the transition from JSPs, generate page headers, footers, menus, etc. using the same FreeMarker templates used in generating whole pages
  - Email messages
  - Ajax responses

Using FreeMarker

```
public class AboutController extends FreemarkerHttpServlet {

        private static final String TEMPLATE_DEFAULT = "about.ftl";

    @Override
    protected ResponseValues processRequest(VitroRequest vreq) {
        ApplicationBean application = vreq.getAppBean();

        Map<String, Object> body = new HashMap<String, Object>();
        body.put("aboutText", application.getAboutText());
        body.put("acknowledgeText", application.getAcknowledgeText());

        return new TemplateResponseValues(TEMPLATE_DEFAULT, body);
    }

    @Override
    protected String getTitle(String siteName, VitroRequest vreq) {
            return "About " + siteName;
    }

}
```

- processRequest() can return a different type of object to trigger a forward, redirect, or error
- Override FreemarkerHttpServlet.doGet()
    - Use methods defined in FreemarkerHttpServlet as needed to generate a page

Resources

- FreeMarker site: http://freemarker.org/
- Online documentation: http://freemarker.org/docs/index.html
- API documentation: http://freemarker.org/docs/api/index.html
- Mailing list: https://sourceforge.net/mail/?group_id=794

Tools

- JBoss Tools FreeMarker IDE for Eclipse
    - http://www.jboss.org/tools/download
    - Nightly build
        - http://download.jboss.org/jbosstools/updates/nightly/trunk
        - Works with Eclipse 3.6 / 3.7
- vTextmate FreeMarker bundle
    - http://code.google.com/p/textmate-freemarker-bundle/