upstate.edu

Velocity Courses | Formats |SUNY Upstate Medical University

4-5 minutes

Acknowledgement

In the second edition of the tutorials, I included many interesting ideas from German Drulyk. If I am to be thanked for the information provided here, he should share part of the compliments. But he will share none of the blames.

Introduction

Many people think that Cascade Velocity is easy. Here is a list of tasks, ranked from very easy to very difficult, every one of which can be performed by writing pure Velocity code. How many of them can you perform?

1. Create and submit a format without executable code.
2. Given a Java object of any type, display the type (class) information of the object.
3. Select a single node using `$_XPathTool.selectNodes`.
4. Make a global variable created with `#set` inaccessible in a macro.

5. Implement a while structure using `#foreach`.
6. Assuming that the name of a global variable is unknown to a macro, write code for a macro to store the value of the global variable in another global variable by passing in the name of the first global variable.
7. Write four lines of code to stall Velocity.
8. Create a switch structure without using `#if`.
9. Check to see if an object x is of an instance of interface/class Y.
10. Use a `StringBuilder` or `StringBuffer` object to store strings to remove unnecessary whitespace characters.
11. Create a calendar of the current month.
12. Locate a format and execute the code therein.
13. Create an object out of the default constructor of a class.
14. Turn a format into an XML block (do not interpret this line literally).
15. Turn an XML block into a format (do not interpret this line literally).
16. Retrieve the code from a format.
17. Retrieve the code of a macro defined in a format.
18. Write code to generate code.
19. Parse an XML string and turn it into an `org.jdom.Element` object.
20. Given the variable `$currentPage` referencing a valid object, retrieve the id strings of the associated content type, metadata

set, configuration set, and data definition.

21. Query/modify the Cascade database.
22. Display the column names of a table in the Cascade database without selecting anything from the table.
23. Given a node of type `org.jdom.Element`, display all valid XPath expressions of the node and all its descendents.
24. Given an ID string like `54763ffe8b7ffe83552dce4f30290433`, retrieve the corresponding Cascade API object, which can be a page, a block, a folder, a symlink, a file, or a reference.
25. Given that a block and a format are attached to the same region of a page, write code in the format to locate the block, using the Locator Tool, without hard-coding the path and site name of the block in the format, so that we can access the metadata of the block; that is to say, the same code should work for any block from any site.
26. Reuse XSLT formats in Velocity.
27. Write a recursive macro to compute the nth term of a Fibonacci series.
28. Use Velocity to invoke an external PHP web service program to modify assets.
29. Write pure Velocity code to invoke web services without using any external programs.
30. Write pure Velocity code to read the contents of a JSP page installed in Cascade.

31. Given a data definition block, and further given that a certain structured data node in the block is the only instance of a field in the corresponding data definition, find out if the field is a multiple field.
32. Move a page from one folder to another folder.
33. Given a URL of a JPEG file, find the dimensions (width and height) of the image.
34. Given the path and site name of a JPEG file in Cascade, find the dimensions of the file.
35. Implement the one-region template master-site design.
36. Use Java trees to control presentation.
37. Implement the multiple-design master site.

Three Courses

Here I provide three Velocity courses. These courses will show you how to perform most of the tasks listed above.

- Introductory Course
- Intermediate Course
- Advanced Course

The entire third course is devoted to multiple designs, trees, and how the two components are tied together in the master site Brisk.

Exercises

I also provide some programming exercises here. Some of them

are relatively easy, and others can be challenging. Have fun!