Writing A Velocity Pluglet

1. About This Document

One of the big improvements in Pavlov 1.1 is the leveraging of the Apache Jakarta Project's Velocity Template Engine to create templated Pluglets and Skins. This document describes how to write a Velocity Feedback Pluglet for Pavlov.

2. Step 1: Find The VelocityPluglet Directory

Let's assume you already have Pavlov 1.1B1 or later installed. Find the directory it was installed in. Now, there will be a subdirectory called `pluglets` and its subdirectory of `pluglets/velocity`. In the 1.1BX release, there are two subdirectories here called `aliengrades` and `navylife`. Every subdirectory of `pluglets/velocity` is scanned at startup for a file named `index.vm`. Those with a valid `index.vm` are accepted and added to the `Feedback Menu` in Pavlov. So, for example, on my machine, I have

```
/home/haus/pup/pluglets/velocity/aliengrades/index.vm
/home/haus/pup/pluglets/velocity/navylife/index.vm
```

let's take a look at one of these.

3. Step 2: Look at an Example

We open up the file `pluglets/velocity/aliengrades/index.vm` and see something like this:

```
<HTML>
<HEAD></HEAD>
<BODY BGCOLOR="black">

<TABLE HEIGHT="223" WIDTH="304">
<TR>
<TD HEIGHT="223" WIDTH="304">
#set ( $right = $ANSWER_EVENT.getNumberOfCorrectAnswers() )
#set ( $tot= $ANSWER_EVENT.getNumberOfAnswers() )
#set ( $perc  = 100 * $right/$tot )

#if ( !$perc )
  #set ($img ="logo.jpg")
```

```
14   #elseif ( $perc < 59 )
15     #set ($img ="f.jpg")
...stuff deleted...
16   #else
17      #set ($img ="aplus.jpg")
18   #end
19   <IMG SRC ="$BASE_URL$img">
20   </TD>
21   </TR>
22   </TABLE>
23
24   </BODY>
25   </HTML>
```

Let's make a few observations:

- This is a well formed HTML document (Java doesn't like bad HTML)
- It has some funny stuff with pound "#" signs (this is Velocity for "do something")
- It has some funny stuff with dollar "$" signs (this is Velocity for "value of something")
- `$ANSWER_EVENT` shows up a lot (Pavlov sends an `ANSWER_EVENT` to the `VelocityModel` every time the user answers a question. `ANSWER_EVENT` has a lot of fun information in it, see http://pavlov.sourceforge.net/api-docs/net/sourceforge/pavlov/event/AnswerEvent.html for all the goodies.
- On lines 8 and 9 we get the number of correct answers and number of total answers in the current quiz
- In line 10, we get `perc`, which is a percentage from 0 to 100 representing the score in the current quiz
- In line 12, we see if `$perc` exists. If it does not exist there have been no answers in this quiz, and we set `$img` to a default value
- In lines 14-18 we choose an image to display based on the value of $perc
- In line 19 we display the image.
- Note the `$BASE_URL` variable in line 19. This lets you use resources in the same directory as `index.vm`.
- The rest of the code just closes the table and document.
- Since we know about how big this HTML page is going to be, we set the table height and width. This makes life a little easier on Java. If we do not know the size, the Java components can figure it out on their own.

Note:

Java doesn't like some META tags, and it hates bad HTML. If you have problems debugging a VelocityPluglet, try to remove META tags and simplify the HTML.

4. Step 3: Use Advanced Velocity Features

The best source for information on advanced Velocity features is at Apache Jakarta Project's Velocity Site. There are some neat tricks on display there.

5. Step 4: Get More Information From Pavlov

Pavlov passes a great deal of information to your template using the AnswerEvent class. You can get information from the `AnswerEvent` by using it's methods. A few examples:

$ANSWER_EVENT.getQuestionData().getPercentage()	Percentage of correct answers over whole user history
$ANSWER_EVENT.getUser().getName()	User's login name.
$ANSWER_EVENT.getBookData().getTitle()	Title of book in use
$ANSWER_EVENT.getChapterData().getStrategy().getName()	Name of the question selection strategy in use
$ANSWER_EVENT.getChapterData().getQuizCollectionData().numQuizzes	Number of quizzes this user has taken on this chapter
$ANSWER_EVENT.getChapterData().getQuizCollectionData().totalAsked	Number of questions this user has answered over all quizzes on this chapter

The point here is not to list all the possible types of information you can glean from the system, but to illustrate that there is a lot to choose from.

6. Step 5: Publish

When you get a feedback pluglet that you like, send it to the Pavlov project. We'll need copyright information and have to filter out obscenities, but other than that, we'll be happy to make your pluglet available. That way, others can not only use your pluglet, but learn from it as well!