

PebbleTemplates / pebble Public

<> **Code** Issues 25 Pull requests 2 Discussions Actions Security and

# Commit b3451c8



**ebussieres** authored on Dec 11, 2025 · ✓ 3/3 · Verified

[CVE-2025-1686 \(#715\)](#)

\* fix: [CVE-2025-1686](#)

- use only a ClasspathLoader by default
- Modify the `FileLoader` to use a mandatory sandboxed base directory parameter

\* chore: update version

\* feat: protect FileLoader against path traversal

\* chore: fix unit test to run on Windows

\* feat: simplify FileLoader and add some unit tests

\* Update FileLoader.java

\* Update FileLoader.java

**master** (#715) · v4.1.1 ... 4.1.0

1 parent [a5a38be](#) commit b3451c8

**18 files changed**

**+226 -152**

Top





























README.md

▼ docs

pom.xml

▼ src/orchid/resources

▼ changelog

-  v4\_0\_1.md
-  v4\_1\_0.md
- ▼  wiki/guide
  -  installation.md
- ▼  pebble-spring
  - ▼  pebble-legacy-spring-boot-starter
    -  pom.xml
  - ▼  pebble-spring-boot-starter
    -  pom.xml
  - ▼  pebble-spring6
    -  pom.xml
  - ▼  pebble-spring7
    -  pom.xml
    -  pom.xml
- ▼  pebble
  -  pom.xml
  - ▼  src
    - ▼  main/java/io/pebbletemplates/pebble
      -  PebbleEngine.java
    - ▼  loader
      -  FileLoader.java
    - ▼  utils
      -  PathUtils.java
    - ▼  test/java/io/pebbletemplates/pebble
      -  FileLoaderTest.java
      -  LoaderTest.java
      -  TestRelativePath.java
  -  pom.xml



Search within code



▼ README.md

```

@@ -6,6 +6,12 @@ includes integrated support for internationalization.
6 6
7 7 For more information please visit the [website](https://pebbletemplates.io).
8 8
9 + # Breaking changes in version 4.1.x
10 +
11 + - If you do not provide a custom Loader, Pebble will now use only a
    `ClasspathLoader` by default, same as the spring autoconfiguration.
12 + Before that, it would have used an instance of the `DelegatingLoader` which
    consists of a `ClasspathLoader` and a `FileLoader` behind the scenes to find
    your templates.
13 + - Modify the `FileLoader` to use a mandatory sandboxed base directory parameter.
14 +
9 15 # Breaking changes in version 4.0.x
10 16
11 17 - Use one of the following artifactId according to the spring boot version that
    you are using
  
```

▼ docs/pom.xml

```

@@ -5,7 +5,7 @@
5 5 <parent>
6 6 <groupId>io.pebbletemplates</groupId>
7 7 <artifactId>pebble-project</artifactId>
8 - <version>4.0.1-SNAPSHOT</version>
8 + <version>4.1.0-SNAPSHOT</version>
9 9 </parent>
10 10
11 11 <artifactId>docs</artifactId>
  
```

▼ docs/src/orchid/resources/changelog/v4\_0\_1.md

**Load Diff**

This file was deleted.

docs/src/orchid/resources/changelog/v4\_1\_0.md



```

... @@ -0,0 +1,17 @@
1 + ---
2 + version: '4.1.0'
3 + ---
4 +
5 + # BREAKING CHANGES
6 + - Modify the `FileLoader` to use a mandatory sandboxed base directory parameter.
  (715)
7 + - If you do not provide a custom Loader, Pebble will now use only a
  `ClasspathLoader` by default, same as the spring autoconfiguration. (715)
8 + Before that, it would have used an instance of the `DelegatingLoader` which
  consists of a `ClasspathLoader` and a `FileLoader` behind the scenes to find
  your templates.
9 +
10 + # New Features
11 + - Use a default existing format of `yyyy-MM-dd'T'HH:mm:ssZ` when using date
  filter with a string (677)
12 + - Look for exact method / field match when doing reflection. Look for method
  get/is/has if none match
13 + - Update some dependencies (709)
14 +
15 + # Bug Fixes
16 + - NaN must return false instead of throwing an exception (695)
17 + - [CVE-2025-1686](https://nvd.nist.gov/vuln/detail/CVE-2025-1686).

```

...orchid/resources/wiki/guide/installation.md



```

↑... @@ -61,18 +61,17 @@ finding your templates.
61 61
62 62 Pebble ships with the following loader implementations:
63 63
64 + - `DelegatingLoader`: Delegates responsibility to a collection of children
  loaders.
64 65 - `ClasspathLoader`: Uses a classloader to search the current classpath.
65 - - `FileLoader`: Finds templates using a filesystem path.
66 + - `FileLoader`: Finds templates using a filesystem path. Must provide a
  mandatory absolute base path.

```

```

66 67 - `ServletLoader`: Uses a servlet context to find the template. This is the
    recommended loader for use within an
67 68 application server but is not enabled by default.
68 69 - `Servlet5Loader`: Same as `ServletLoader`, but for Jakarta Servlet 5.0 or
    newer.
69 - - `StringLoader`: Considers the name of the template to be the contents of the
    template.
70 - - `DelegatingLoader`: Delegates responsibility to a collection of children
    loaders.
71 70 - `MemoryLoader`: Loader that supports inheritance and doesn't require a
    filesystem. This is useful for applications
71 + - `StringLoader`: Considers the name of the template to be the contents of the
    template. Should not be used in a production environment. It is primarily for
    testing and debugging. Many tags may not work when using this loader, such as
    "extends", "imports", etc.
72 72 that retrieve templates from a database for example.
73 73
74 - If you do not provide a custom Loader, Pebble will use an instance of the
    `DelegatingLoader` by default.
75 - This delegating loader will use a `ClasspathLoader` and a `FileLoader` behind
    the scenes to find your templates.
74 + If you do not provide a custom Loader, Pebble will use an instance of the
    `ClasspathLoader` by default.
76 75
77 76 ## Pebble Engine Settings
78 77
@@ -85,7 +84,7 @@ All the settings are set during the construction of the
`PebbleEngine` object.
85 84 | `tagCache` | An implementation of a ConcurrentMap cache that the Pebble engine
    will use for anchor('cache tag', 'cache'). | Default implementation is
    `ConcurrentMapTagCache` and another implementation based on Caffeine is
    available (`CaffeineTagCache`) |
86 85 | `defaultLocale` | The default locale which will be passed to each compiled
    template. The templates then use this locale for functions such as i18n, etc. A
    template can also be given a unique locale during evaluation. |
    `Locale.getDefault()` |
87 86 | `executorService` | An `ExecutorService` that allows the usage of some
    advanced multithreading features, such as the parallel tag. | `null` |
88 - | `loader` | An implementation of the `Loader` interface which is used to find
    templates. | An implementation of the `DelegatingLoader` which uses a

```

		<code>`ClassLoader`</code> and a <code>`FileLoader`</code> behind the scenes.
87	+	<code>`loader`</code>   An implementation of the <code>`Loader`</code> interface which is used to find templates.   An implementation of the <code>`ClassLoader`</code>
89	88	<code>`strictVariables`</code>   If set to true, Pebble will throw an exception if you try to access a variable or attribute that does not exist (or an attribute of a null variable). If set to false, your template will treat non-existing variables/attributes as null without ever skipping a beat.   <code>`false`</code>
90	89	<code>`methodAccessValidator`</code>   Pebble provides two implementations. <code>NoOpMethodAccessValidator</code> which do nothing and <code>BlacklistMethodAccessValidator</code> which checks that the method being called is not blacklisted.   <code>`BlacklistMethodAccessValidator`</code>
91	90	<code>`literalDecimalTreatedAsInteger`</code>   option for treating literal decimals as <code>`int`</code> . Otherwise it is <code>`long`</code> .   <code>`false`</code>

▼ ...g/pebble-legacy-spring-boot-starter/pom.xml		...
		@@ -4,7 +4,7 @@
4	4	<parent>
5	5	<artifactId>pebble-spring</artifactId>
6	6	<groupId>io.pebbletemplates</groupId>
7	-	<version>4.0.1-SNAPSHOT</version>
7	+	<version>4.1.0-SNAPSHOT</version>
8	8	</parent>
9	9	
10	10	<artifactId>pebble-legacy-spring-boot-starter</artifactId>

▼ ...e-spring/pebble-spring-boot-starter/pom.xml		...
		@@ -4,7 +4,7 @@
4	4	<parent>
5	5	<artifactId>pebble-spring</artifactId>
6	6	<groupId>io.pebbletemplates</groupId>
7	-	<version>4.0.1-SNAPSHOT</version>
7	+	<version>4.1.0-SNAPSHOT</version>
8	8	</parent>
9	9	
10	10	<artifactId>pebble-spring-boot-starter</artifactId>

```
pebble-spring/pebble-spring6/pom.xml
```

↑		@@ -4,7 +4,7 @@
4	4	<parent>
5	5	<artifactId>pebble-spring</artifactId>
6	6	<groupId>io.pebbletemplates</groupId>
7	-	<version>4.0.1-SNAPSHOT</version>
7	+	<version>4.1.0-SNAPSHOT</version>
8	8	</parent>
9	9	
10	10	<artifactId>pebble-spring6</artifactId>
↓		

```
pebble-spring/pebble-spring7/pom.xml
```

↑		@@ -4,7 +4,7 @@
4	4	<parent>
5	5	<artifactId>pebble-spring</artifactId>
6	6	<groupId>io.pebbletemplates</groupId>
7	-	<version>4.0.1-SNAPSHOT</version>
7	+	<version>4.1.0-SNAPSHOT</version>
8	8	</parent>
9	9	
10	10	<artifactId>pebble-spring7</artifactId>
↓		

```
pebble-spring/pom.xml
```

↑		@@ -4,7 +4,7 @@
4	4	<parent>
5	5	<groupId>io.pebbletemplates</groupId>
6	6	<artifactId>pebble-project</artifactId>
7	-	<version>4.0.1-SNAPSHOT</version>
7	+	<version>4.1.0-SNAPSHOT</version>
8	8	</parent>
9	9	
10	10	<artifactId>pebble-spring</artifactId>
↓		

```
pebble/pom.xml
```

```

↑... @@ -3,7 +3,7 @@
3 3 <parent>
4 4 <groupId>io.pebbletemplates</groupId>
5 5 <artifactId>pebble-project</artifactId>
6 - <version>4.0.1-SNAPSHOT</version>
6 + <version>4.1.0-SNAPSHOT</version>
7 7 </parent>
8 8
9 9 <artifactId>pebble</artifactId>
↓...

```

```

▼ ...io/pebbletemplates/pebble/PebbleEngine.java ...
↑... @@ -9,45 +9,40 @@
9 9 package io.pebbletemplates.pebble;
10 10
11 11
12 + import
    io.pebbletemplates.pebble.attributes.methodaccess.BlacklistMethodAccessValidato
    r;
13 + import io.pebbletemplates.pebble.attributes.methodaccess.MethodAccessValidator;
12 14 import io.pebbletemplates.pebble.cache.CacheKey;
13 15 import io.pebbletemplates.pebble.cache.PebbleCache;
14 16 import io.pebbletemplates.pebble.cache.tag.ConcurrentMapTagCache;
15 17 import io.pebbletemplates.pebble.cache.tag.NoOpTagCache;
16 18 import io.pebbletemplates.pebble.cache.template.ConcurrentMapTemplateCache;
17 19 import io.pebbletemplates.pebble.cache.template.NoOpTemplateCache;
18 20 import io.pebbletemplates.pebble.error.LoaderException;
21 + import io.pebbletemplates.pebble.extension.*;
22 + import io.pebbletemplates.pebble.extension.escaper.EscapingStrategy;
19 23 import io.pebbletemplates.pebble.lexer.LexerImpl;
20 24 import io.pebbletemplates.pebble.lexer.Syntax;
21 25 import io.pebbletemplates.pebble.lexer.TokenStream;
26 + import io.pebbletemplates.pebble.loader.ClasspathLoader;
27 + import io.pebbletemplates.pebble.loader.Loader;
28 + import io.pebbletemplates.pebble.loader.StringLoader;
22 29 import io.pebbletemplates.pebble.node.RootNode;
23 30 import io.pebbletemplates.pebble.parser.Parser;
24 31 import io.pebbletemplates.pebble.parser.ParserImpl;
25 32 import io.pebbletemplates.pebble.parser.ParserOptions;

```

```

26      - import
          io.pebbletemplates.pebble.attributes.methodaccess.BlacklistMethodAccessValidato
          r;
27      - import io.pebbletemplates.pebble.attributes.methodaccess.MethodAccessValidator;
28      - import io.pebbletemplates.pebble.extension.escaper.EscapingStrategy;
29      - import io.pebbletemplates.pebble.loader.ClasspathLoader;
30      - import io.pebbletemplates.pebble.loader.DelegatingLoader;
31      - import io.pebbletemplates.pebble.loader.FileLoader;
32      - import io.pebbletemplates.pebble.loader.Loader;
33      - import io.pebbletemplates.pebble.loader.StringLoader;
34      - import io.pebbletemplates.pebble.extension.*;
35      33      import io.pebbletemplates.pebble.template.EvaluationOptions;
36      34      import io.pebbletemplates.pebble.template.PebbleTemplate;
37      35      import io.pebbletemplates.pebble.template.PebbleTemplateImpl;
38      36      + import io.pebbletemplates.pebble.utils.TypeUtils;
39      37      + import org.slf4j.Logger;
40      38      + import org.slf4j.LoggerFactory;
41      39
42      40      import java.io.IOException;
43      41      import java.io.Reader;
44      42      - import java.util.ArrayList;
45      43      - import java.util.List;
46      44      import java.util.Locale;
47      45      import java.util.concurrent.ExecutorService;
48      46      import java.util.function.Function;
49      47
50      48      - import io.pebbletemplates.pebble.utils.TypeUtils;
51      49      - import org.slf4j.Logger;
52      50      - import org.slf4j.LoggerFactory;
53      51      -
54      52      /**
55      53      * The main class used for compiling templates. The PebbleEngine is responsible
56      54      for delegating
57      55      * responsibility to the lexer, parser, compiler, and template cache.
58      56
59      57      @@@ -584,10 +579,7 @@ public PebbleEngine build() {
60      58
61      59      // default loader
62      60      if (this.loader == null) {
63      61
64      62      List<Loader<?>> defaultLoadingStrategies = new ArrayList<>();

```

```

588 - defaultLoadingStrategies.add(new ClasspathLoader());
589 - defaultLoadingStrategies.add(new FileLoader());
590 - this.loader = new DelegatingLoader(defaultLoadingStrategies);
582 + this.loader = new ClasspathLoader();
591 583     }
592 584
593 585     // default locale

```

...bbletemplates/pebble/loader/FileLoader.java

```

@@ -10,18 +10,12 @@
10 10
11 11     import io.pebbletemplates.pebble.error.LoaderException;
12 12     import io.pebbletemplates.pebble.utils.PathUtils;
13 -
14 13     import org.slf4j.Logger;
15 14     import org.slf4j.LoggerFactory;
16 15
17 - import java.io.BufferedReader;
18 - import java.io.File;
19 - import java.io.FileInputStream;
20 - import java.io.FileNotFoundException;
21 - import java.io.InputStream;
22 - import java.io.InputStreamReader;
23 - import java.io.Reader;
24 - import java.io.UnsupportedEncodingException;
16 + import java.io.*;
17 + import java.nio.file.Path;
18 + import java.nio.file.Paths;
25 19
26 20     /**
27 21     * This loader searches for a file located anywhere on the filesystem. It uses
    java.io.File to
@@ -34,69 +28,35 @@ public class FileLoader implements Loader<String> {
34 28     private static final Logger logger =
    LoggerFactory.getLogger(FileLoader.class);
35 29
36 30     private String prefix;
37 -
38 31     private String suffix;

```

```
39 -
40 32     private String charset = "UTF-8";
41 33
42 34 +     public FileLoader(String prefix) {
43 35 +         this.setPrefix(prefix);
44 36 +     }
45 37 +
46 38     @Override
47 39     public Reader getReader(String templateName) {
48 40         // try to load File
49 41         InputStream is = null;
50 42         File file = this.getFile(templateName);
51 43         if (file.exists() && file.isFile()) {
52 44             try {
53 45                 is = new FileInputStream(file);
54 46             } catch (FileNotFoundException e) {
55 47             }
56 48         }
57 49         if (is == null) {
58 50             throw new LoaderException(null,
59 51                 "Could not find template \"" + templateName + "\"");
60 52         }
61 53         try {
62 54             InputStream is = new FileInputStream(file);
63 55             return new BufferedReader(new InputStreamReader(is, this.charset));
64 56         } catch (FileNotFoundException e) {
65 57             throw new LoaderException(e, String.format("Could not find template
66 58                 [prefix='%s', templateName='%s']", this.prefix, templateName));
67 59         } catch (UnsupportedEncodingException e) {
68 60             throw new LoaderException(e, String.format("Invalid charset '%s'",
69 61                 this.charset));
70 62         }
71 63         return null;
72 64     }
73 65     private File getFile(String templateName) {
74 66         // add the prefix and ensure the prefix ends with a separator character
```

```
69     -     StringBuilder path = new StringBuilder();
70     -     if (this.getPrefix() != null) {
71     -
72     -         path.append(this.getPrefix());
73     -
74     -         if (!this.getPrefix().endsWith(String.valueOf(File.separatorChar))) {
75     -             path.append(File.separatorChar);
76     -         }
77     -     }
78     -
79     52         templateName = templateName + (this.getSuffix() == null ? "" :
            this.getSuffix());
80     54
81     -     logger.trace("Looking for template in {}{}.", path.toString(),
            templateName);
82     55     +     Path path = Paths.get(this.getPrefix(), templateName);
83     56     +     logger.trace("Looking for template in {}.", path);
84     57
85     -     /*
86     -      * if template name contains path segments, move those segments into the
87     -      * path variable. The below technique needs to know the difference
88     -      * between the path and file name.
89     -      */
90     -     String[] pathSegments = PathUtils.PATH_SEPARATOR_REGEX.split(templateName);
91     -
92     -     if (pathSegments.length > 1) {
93     -         // file name is the last segment
94     -         templateName = pathSegments[pathSegments.length - 1];
95     -     }
96     -     for (int i = 0; i < (pathSegments.length - 1); i++) {
97     -         path.append(pathSegments[i]).append(File.separatorChar);
98     -     }
99     -
100    -     // try to load File
101    -     return new File(path.toString(), templateName);
102    58     +     this.checkIfDirectoryTraversal(templateName);
103    59     +     return path.toFile();
104    60     }
105    61
```

```
102 62      public String getSuffix() {
@@ -114,7 +74,17 @@ public String getPrefix() {
114 74
115 75      @Override
116 76      public void setPrefix(String prefix) {
117 -      this.prefix = prefix;
77 +      if (prefix == null) {
78 +          throw new LoaderException(null, "Prefix cannot be null");
79 +      }
80 +      String trimmedPrefix = prefix.trim();
81 +      if (trimmedPrefix.isEmpty()) {
82 +          throw new LoaderException(null, "Prefix cannot be empty");
83 +      }
84 +      if (!Paths.get(trimmedPrefix).isAbsolute()) {
85 +          throw new LoaderException(null, "Prefix must be an absolute path");
86 +      }
87 +      this.prefix = trimmedPrefix;
118 88      }
119 89
120 90      public String getCharset() {
@@ -140,4 +110,23 @@ public String createCacheKey(String templateName) {
140 110      public boolean resourceExists(String templateName) {
141 111          return this.getFile(templateName).exists();
142 112      }
113 +
114 +      private void checkIfDirectoryTraversal(String templateName) {
115 +          Path baseDirPath = Paths.get(prefix);
116 +          Path userPath = Paths.get(templateName);
117 +          if (userPath.isAbsolute()) {
118 +              throw new LoaderException(null, String.format("templateName '%s' must be
relative", templateName));
119 +          }
120 +
121 +          // Join the two paths together, then normalize so that any ".." elements
122 +          // in the userPath can remove parts of baseDirPath.
123 +          // (e.g. "/foo/bar/baz" + "../attack" -> "/foo/bar/attack")
124 +          Path resolvedPath = baseDirPath.resolve(userPath).normalize();
125 +
126 +          // Make sure the resulting path is still within the required directory.
127 +          // (In the example above, "/foo/bar/attack" is not.)
```

```

128 +     if (!resolvedPath.startsWith(baseDirPath)) {
129 +         throw new LoaderException(null, String.format("template is not in the
          base directory path [baseDir='%s', templateName='%s']", this.prefix,
          templateName));
130 +     }
131 + }
143 132 }

```

▼ ...pebbletemplates/pebble/Utils/PathUtils.java ...

```

@@ -44,7 +44,7 @@ public static String resolveRelativePath(String
relativePath, String anchorPath,
44 44     return null;
45 45 }
46 46
47 - private static String sanitize(String path, char expectedSeparator) {
47 + public static String sanitize(String path, char expectedSeparator) {
48 48     return PATH_SEPARATOR_REGEX.matcher(path)
49 49
          .replaceAll(Matcher.quoteReplacement(String.valueOf(expectedSeparator)));
50 50 }

```

▼ .../pebbletemplates/pebble/FileLoaderTest.java ...

```

... @@ -0,0 +1,132 @@
1 + package io.pebbletemplates.pebble;
2 +
3 + import io.pebbletemplates.pebble.error.LoaderException;
4 + import io.pebbletemplates.pebble.error.PebbleException;
5 + import io.pebbletemplates.pebble.loader.FileLoader;
6 + import io.pebbletemplates.pebble.loader.Loader;
7 + import io.pebbletemplates.pebble.template.PebbleTemplate;
8 + import org.junit.jupiter.api.Test;
9 + import org.junit.jupiter.params.ParameterizedTest;
10 + import org.junit.jupiter.params.provider.ValueSource;
11 +
12 + import java.io.IOException;
13 + import java.io.StringWriter;
14 + import java.io.Writer;
15 + import java.net.URISyntaxException;

```

```
16 + import java.nio.file.Paths;
17 +
18 + import static org.junit.jupiter.api.Assertions.assertEquals;
19 + import static org.junit.jupiter.api.Assertions.assertThrows;
20 +
21 + public class FileLoaderTest {
22 +
23 +     @Test
24 +     void testFileLoaderPrefixNull() {
25 +         assertThrows(LoaderException.class, () -> new FileLoader(null));
26 +     }
27 +
28 +     @Test
29 +     void testFileLoaderPrefixEmpty() {
30 +         assertThrows(LoaderException.class, () -> new FileLoader(" "));
31 +     }
32 +
33 +     @Test
34 +     void testFileLoaderPrefixRelativePath() {
35 +         assertThrows(LoaderException.class, () -> new FileLoader("../bar "));
36 +     }
37 +
38 +     @Test
39 +     void testFileLoader() throws PebbleException, IOException, URISyntaxException
40 +     {
41 +         String prefix =
42 +             Paths.get(this.getClass().getClassLoader().getResource("templates").toURI()).to
43 +             String();
44 +         Loader<?> loader = new FileLoader(prefix);
45 +         loader.setSuffix(".suffix");
46 +         PebbleEngine engine = new
47 +             PebbleEngine.Builder().loader(loader).strictVariables(false).build();
48 +         PebbleTemplate template1 = engine.getTemplate("template.loaderTest.peb");
49 +         Writer writer1 = new StringWriter();
50 +         template1.evaluate(writer1);
51 +         assertEquals("SUCCESS", writer1.toString());
52 +     }
53 +
54 +     @Test
```

```
51 + void testFileLoaderAbsoluteTemplateName() throws PebbleException,
    URISyntaxException {
52 +     String prefix =
    Paths.get(this.getClass().getClassLoader().getResource("templates").toURI()).to
    String();
53 +     Loader<?> loader = new FileLoader(prefix);
54 +     loader.setSuffix(".suffix");
55 +     PebbleEngine engine = new
    PebbleEngine.Builder().loader(loader).strictVariables(false).build();
56 +     assertThrows(LoaderException.class, () ->
    engine.getTemplate("/template.loaderTest.peb"));
57 + }
58 +
59 + @Test
60 + void testFileLoaderTemplateNameIsADirectory() throws PebbleException,
    URISyntaxException {
61 +     String prefix =
    Paths.get(this.getClass().getClassLoader().getResource("templates").toURI()).to
    String();
62 +     Loader<?> loader = new FileLoader(prefix);
63 +     PebbleEngine engine = new
    PebbleEngine.Builder().loader(loader).strictVariables(false).build();
64 +     assertThrows(LoaderException.class, () -> engine.getTemplate("loader"));
65 + }
66 +
67 + @Test
68 + void testFileLoaderRelativeTemplateName() throws PebbleException,
    IOException, URISyntaxException {
69 +     String prefix =
    Paths.get(this.getClass().getClassLoader().getResource("templates").toURI()).ge
    tParent().toString();
70 +     Loader<?> loader = new FileLoader(prefix);
71 +     loader.setSuffix(".suffix");
72 +     PebbleEngine engine = new
    PebbleEngine.Builder().loader(loader).strictVariables(false).build();
73 +     PebbleTemplate template1 =
    engine.getTemplate("templates/template.loaderTest.peb");
74 +     Writer writer1 = new StringWriter();
75 +     template1.evaluate(writer1);
76 +     assertEquals("SUCCESS", writer1.toString());
```

```
77 +   }
78 +
79 +   @Test
80 +   void testFileLoaderPathTraversal() throws PebbleException, URISyntaxException
81 +   {
82 +       String prefix =
83 +           Paths.get(this.getClass().getClassLoader().getResource("templates").toURI()).to
84 +               String();
85 +       Loader<?> loader = new FileLoader(prefix);
86 +       loader.setSuffix(".peb");
87 +       PebbleEngine engine = new
88 +           PebbleEngine.Builder().loader(loader).strictVariables(false).build();
89 +       assertThrows(LoaderException.class, () -> engine.getTemplate("../template-
90 +           tests/DoubleNestedIfStatement"));
91 +   }
92 +
93 +   @ParameterizedTest
94 +   @ValueSource(strings = {"%2e%2e%2f", "%2e%2e/", ":%2f", "%2e%2e%5c",
95 +       "%2e%2e\\", ":%5c", "%252e%252e%255c", ":%255c"})
96 +   void testFileLoaderPathTraversalEncoded(String relativePath) throws
97 +       URISyntaxException {
98 +       String prefix =
99 +           Paths.get(this.getClass().getClassLoader().getResource("templates").toURI()).to
100 +               String();
101 +       Loader<?> loader = new FileLoader(prefix);
102 +       loader.setCharset("foobar");
```

```
103 +     PebbleEngine engine = new
      PebbleEngine.Builder().loader(loader).strictVariables(false).build();
104 +     assertThrows(LoaderException.class, () ->
      engine.getTemplate("template.loaderTest.peb"));
105 + }
106 +
107 + /**
108 +  * Tests if relative includes work. Issue #162.
109 +  */
110 + @Test
111 + void testFileLoaderPathWithBackslash() throws IOException, URISyntaxException
      {
112 +     String prefix =
      Paths.get(this.getClass().getClassLoader().getResource("templates").toURI()).to
      String();
113 +     PebbleEngine pebble = new PebbleEngine.Builder().loader(new
      FileLoader(prefix)).build();
114 +     PebbleTemplate template =
      pebble.getTemplate("relativepath/subdirectory1/template.forwardslashes.peb".rep
      lace("/", "\\")); // ensure backslashes in all environments
115 +     Writer writer = new StringWriter();
116 +     template.evaluate(writer);
117 +     assertEquals("included", writer.toString());
118 + }
119 +
120 + /**
121 +  * Issue #162.
122 +  */
123 + @Test
124 + void testFileLoaderPathWithForwardSlash() throws IOException,
      URISyntaxException {
125 +     String prefix =
      Paths.get(this.getClass().getClassLoader().getResource("templates").toURI()).to
      String();
126 +     PebbleEngine pebble = new PebbleEngine.Builder().loader(new
      FileLoader(prefix)).build();
127 +     PebbleTemplate template =
      pebble.getTemplate("relativepath/subdirectory1/template.backwardslashes.peb");
128 +     Writer writer = new StringWriter();
129 +     template.evaluate(writer);
```

```

130 +     assertEquals("included", writer.toString());
131 + }
132 + }

```

▼ ...a/io/pebbletemplates/pebble/LoaderTest.java ...

↑ ... @@ -15,7 +15,6 @@

```

15 15     import org.junit.jupiter.api.Test;
16 16
17 17     import java.io.*;
18 18     - import java.net.URISyntaxException;
19 18     import java.net.URL;
20 19     import java.net.URLClassLoader;
21 20     import java.util.ArrayList;

```

↓ ... @@ -69,19 +68,6 @@ void testClassLoaderLoaderWithNestedTemplateInJar() throws  
↑ PebbleException, IOEx

```

69 68
70 69     }
71 70
72 72     - @Test
73 73     - void testFileLoader() throws PebbleException, IOException, URISyntaxException
74 74     {
75 75         Loader<?> loader = new FileLoader();
76 76         loader.setSuffix(".suffix");
77 76         PebbleEngine engine = new
78 76         PebbleEngine.Builder().loader(loader).strictVariables(false).build();
79 76         URL url = this.getClass().getResource("/templates/template.loaderTest.peb");
80 76         PebbleTemplate template1 = engine.getTemplate(new
81 76         File(url.toURI()).getPath());
82 76         Writer writer1 = new StringWriter();
83 76         template1.evaluate(writer1);
84 76         assertEquals("SUCCESS", writer1.toString());
85 76
86 76     }
87 76
88 76     @Test
89 76     void testDelegatingLoader() throws PebbleException, IOException {
90 76         List<Loader<?>> loaders = new ArrayList<>();

```

↓

```

...ebbletemplates/pebble/TestRelativePath.java
... @@ -1,17 +1,12 @@
1 1 package io.pebbletemplates.pebble;
2 2
3 3 import io.pebbletemplates.pebble.error.PebbleException;
4 - import io.pebbletemplates.pebble.loader.FileLoader;
5 4 import io.pebbletemplates.pebble.template.PebbleTemplate;
6 -
7 5 import org.junit.jupiter.api.Test;
8 6
9 - import java.io.File;
10 7 import java.io.IOException;
11 8 import java.io.StringWriter;
12 9 import java.io.Writer;
13 - import java.net.URISyntaxException;
14 - import java.net.URL;
15 10
16 11 import static org.junit.jupiter.api.Assertions.assertEquals;
17 12
... @@ -65,38 +60,4 @@ void testRelativeImports() throws PebbleException,
... IOException {
65 60 assertEquals("<input name=\"company\" value=\"forcorp\" type=\"text\" />",
66 61 writer.toString().replaceAll("\\r?\\n", "").replace("\\t", ""));
67 62 }
68 -
69 - /**
70 - * Tests if relative includes work. Issue #162.
71 - */
72 - @Test
73 - void testPathWithBackslashesWithRelativePathWithForwardSlashes()
74 - throws PebbleException, IOException, URISyntaxException {
75 - PebbleEngine pebble = new PebbleEngine.Builder().loader(new
FileLoader()).build();
76 - URL url = this.getClass()
77 -
.getResource("/templates/relativepath/subdirectory1/template.forwardslashes.peb
");
78 - PebbleTemplate template = pebble
79 - .getTemplate(new File(url.toURI()).getPath())

```

```

80     .replace("/", "\\"); // ensure backslashes in all environments
81     Writer writer = new StringWriter();
82     template.evaluate(writer);
83     assertEquals("included", writer.toString());
84 }
85
86 /**
87  * Issue #162.
88  */
89 @Test
90 void testPathWithForwardSlashesWithRelativePathWithBackwardSlashes()
91     throws PebbleException, IOException, URISyntaxException {
92     PebbleEngine pebble = new PebbleEngine.Builder().loader(new
93     FileLoader()).build();
94     URL url = this.getClass()
95     .getResource("/templates/relativepath/subdirectory1/template.backwardslashes.pe
96     b");
97     PebbleTemplate template = pebble
98     .getTemplate(new File(url.toURI()).getPath()
99     .replace("\\", "/")); // ensure forward slashes in all environments
100    Writer writer = new StringWriter();
101    template.evaluate(writer);
102    assertEquals("included", writer.toString());
103 }

```

▼ pom.xml

...

↑

@@ -3,7 +3,7 @@

3	3	<modelVersion>4.0.0</modelVersion>
4	4	<groupId>io.pebbletemplates</groupId>
5	5	<artifactId>pebble-project</artifactId>
6	-	<version>4.0.1-SNAPSHOT</version>
6	+	<version>4.1.0-SNAPSHOT</version>
7	7	
8	8	<packaging>pom</packaging>
9	9	

↓

Comments 0



Please [sign in](#) to comment.