

R6-generator-features

Rob Challen

19/10/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

```
codeSnip("java",filename="/home/terminological/Git/r6-generator-maven-plugin-test/src/main/java/uk/co/t
```

```
1 package uk.co.terminological.rjava.test;
2
3
4 import java.io.IOException;
5 import java.util.LinkedHashMap;
6 import java.util.Map;
7
8 import org.slf4j.Logger;
9 import org.slf4j.LoggerFactory;
10
11 import uk.co.terminological.rjava.RClass;
12 import uk.co.terminological.rjava.RConverter;
13 import uk.co.terminological.rjava.RMethod;
14 import uk.co.terminological.rjava.types.RCharacter;
15 import uk.co.terminological.rjava.types.RDataframe;
16 import uk.co.terminological.rjava.types.RNumeric;
17 import uk.co.terminological.rjava.types.RObject;
18
19 /**
20  * A test of the jsr223 templating
21  *
22  * this is a details comment
23  * @author given family email@example.com ORCIDID
24  *
25  */
26 @RClass(
27     imports = {"ggplot2","dplyr"},
28     suggests = {"roxygen2","devtools"}
29 )
30 public class FeatureTest {
31
32     String message;
33     static Logger log = LoggerFactory.getLogger(FeatureTest.class);
34
35     /**
36      * A maximum of one constructor of any signature can be used. </br>
37      *
38      * If different constructors are needed then they may be used but not
39      * included in the R Api (i.e. not annotated with @RMethod. </br>
40      *
41      * Static factory methods can be used instead.
42      * @param logMessage - a message which will be logged
43      */
44     @RMethod(examples = {
45         "J = testRapi::JavaApi$new()",
```

```

46         "minExample = J$FeatureTest$new('Hello, java constructor!')",
47     })
48     public FeatureTest(String logMessage) {
49         log.info(logMessage);
50         this.message = logMessage;
51     }
52
53
54     /**
55      * Description of a hello world function
56      * @return this java method returns a String
57      */
58     @RMethod(examples = {
59         "An example",
60         "Spans many lines"
61     })
62     public RCharacter doHelloWorld() {
63         return RConverter.convert("Hello world from Java!");
64     }
65
66     /**
67      * A fluent method which updates the message in this object, returning the
68      * same object. This is differentiated from factory methods which produce a new
69      * instance of the same class by checking to see if the returned Java object is in the
70      * same memory location as the calling Java object.
71      * @param message the message is a string
72      * @return this should return exactly the same R6 object.
73      */
74     @RMethod
75     public FeatureTest fluentSetMessage(RCharacter message) {
76         this.message = message.toString();
77         return this;
78     }
79
80     /**
81      * A fluent method description
82      * @param message the message is a string
83      * @return A MoreFeatureTest R6 reference
84      */
85     @RMethod
86     public MoreFeatureTest factoryMethod(RCharacter a, RCharacter b) {
87         return new MoreFeatureTest(a,b);
88     }
89
90     /**
91      * message description
92      * @return The message previously set or maybe null
93      */
94     @RMethod
95     public RCharacter getMessage() {
96         return RConverter.convert(message);
97     }
98
99     /**
100      * The doSum function description = it adds two numerics
101      * @param a the A parameter
102      * @param b the B parameter
103      * @return A+B of course
104      */
105     @RMethod
106     public RNumeric doSum(RNumeric a, RNumeric b) {
107         return RConverter.convert(a.get()+b.get());
108     }
109
110
111     /**
112      * Do sum 2 uses native ints rather than RNumerics
113      * It should throw an error if given something that cannot be an integer

```

```

114     * @param a the A parameter
115     * @param b the B parameter
116     * @return A+B of course
117     */
118     @RMethod
119     public int doSum2(int a, int b) {
120         return a+b;
121     }
122
123
124     @RMethod
125     public String objectAsParameter(MoreFeatureTest otherObj) {
126         return otherObj.toString();
127     }
128
129     /**
130     * Consumes a data frame and logs its length
131     * @param dataframe
132     */
133     @RMethod
134     public void doSomethingWithDataFrame(RDataframe dataframe) {
135         log.info("dataframe length: "+dataframe.nrow());
136     }
137
138     /**
139     * Creates a basic dataframe and returns it
140     */
141     @RMethod
142     public RDataframe generateDataframe() {
143         RDataframe out = new RDataframe();
144         for (int i=0; i<10; i++) {
145             Map<String,Object> tmp = new LinkedHashMap<String,Object>();
146             tmp.put("index", i);
147             tmp.put("value", 10-i);
148             out.addRow(tmp);
149         }
150         return out;
151     }
152
153
154     /**
155     * Static methods are also supported. These are accessed through the
156     * root of the R api.
157     * @param message
158     */
159     @RMethod(examples = {
160         "J = testRapi::JavaApi$new()",
161         "J$FeatureTest$demoStatic('Ola, el mundo')",
162     })
163     public static void demoStatic(String message) {
164         log.info(message);
165     }
166
167
168     /**
169     * A copy of the ggplot2::diamonds dataframe serialised into java, using
170     * RObject.writeRDS, saved within the jar file of the package, and exposed here
171     * using RObject.readRDS.
172     */
173     @RMethod(examples = {
174         "J = testRapi::JavaApi$new()",
175         "J$FeatureTest$diamonds()",
176     })
177     public static RDataframe diamonds() throws IOException {
178         return RObject.readRDS(RDataframe.class, FeatureTest.class.getResourceAsStream("/diamonds.ser"));
179     }
180 }

```

```
codeSnip("XML",filename="/home/terminological/Git/r6-generator-maven-plugin-test/pom.xml")#,starts = c(
```

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4
5   <properties>
6     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
7     <maven.compiler.source>1.8</maven.compiler.source>
8     <maven.compiler.target>1.8</maven.compiler.target>
9     <r6.version>master-SNAPSHOT</r6.version>
10  </properties>
11
12  <groupId>com.github.terminological</groupId>
13  <artifactId>r6-generator-maven-plugin-test</artifactId>
14  <version>${r6.version}</version>
15  <packaging>jar</packaging>
16
17  <name>R6 Generator Maven Plugin Test</name>
18
19  <dependencies>
20    <dependency>
21      <groupId>com.github.terminological</groupId>
22      <artifactId>r6-generator-runtime</artifactId>
23      <version>${r6.version}</version>
24    </dependency>
25  </dependencies>
26
27  <!-- Resolve runtime library on github -->
28  <repositories>
29    <repository>
30      <id>jitpack.io</id>
31      <url>https://jitpack.io</url>
32    </repository>
33  </repositories>
34
35  <!-- Resolve maven plugin on github -->
36  <pluginRepositories>
37    <pluginRepository>
38      <id>jitpack.io</id>
39      <url>https://jitpack.io</url>
40    </pluginRepository>
41  </pluginRepositories>
42
43  <build>
44    <plugins>
45      <plugin>
46        <artifactId>maven-compiler-plugin</artifactId>
47        <version>3.1</version>
48        <configuration>
49          <source>${maven.compiler.source}</source>
50          <target>${maven.compiler.target}</target>
51        </configuration>
52      </plugin>
```

```

53     <plugin>
54       <groupId>com.github.terminological</groupId>
55       <artifactId>r6-generator-maven-plugin</artifactId>
56       <version>${r6.version}</version>
57       <configuration>
58         <packageData>
59           <title>A test library</title>
60           <version>0.01</version>
61           <debug>true</debug>
62           <rjavaOpts>
63             <rjavaOpt>-Xmx256M</rjavaOpt>
64           </rjavaOpts>
65           <packageName>testRapi</packageName>
66           <license>MIT</license>
67           <description>An optional long description of the package</description>
68           <maintainerName>test forename</maintainerName>
69           <maintainerFamilyName>optional surname</maintainerFamilyName>
70           <maintainerEmail>test@example.com</maintainerEmail>
71         </packageData>
72         <outputDirectory>${project.basedir}/r-library</outputDirectory>
73       </configuration>
74       <executions>
75         <execution>
76           <id>generate-r-library</id>
77           <goals>
78             <goal>generate-r-library</goal>
79           </goals>
80         </execution>
81       </executions>
82     </plugin>
83   </plugins>
84 </build>
85 </project>

```

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this: