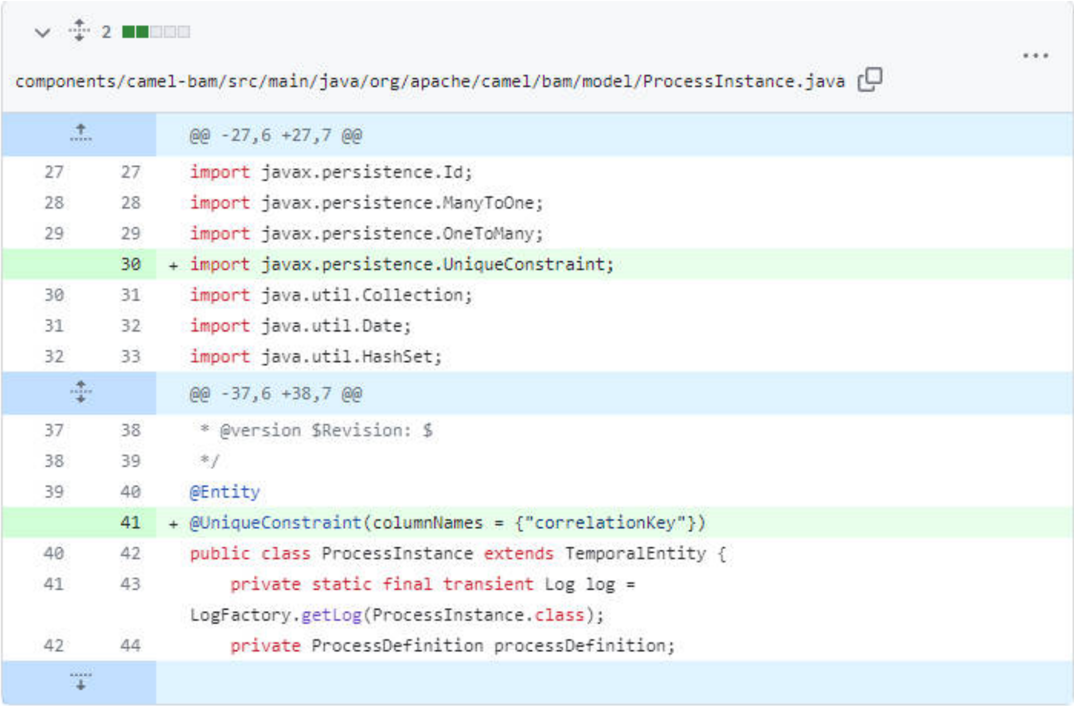
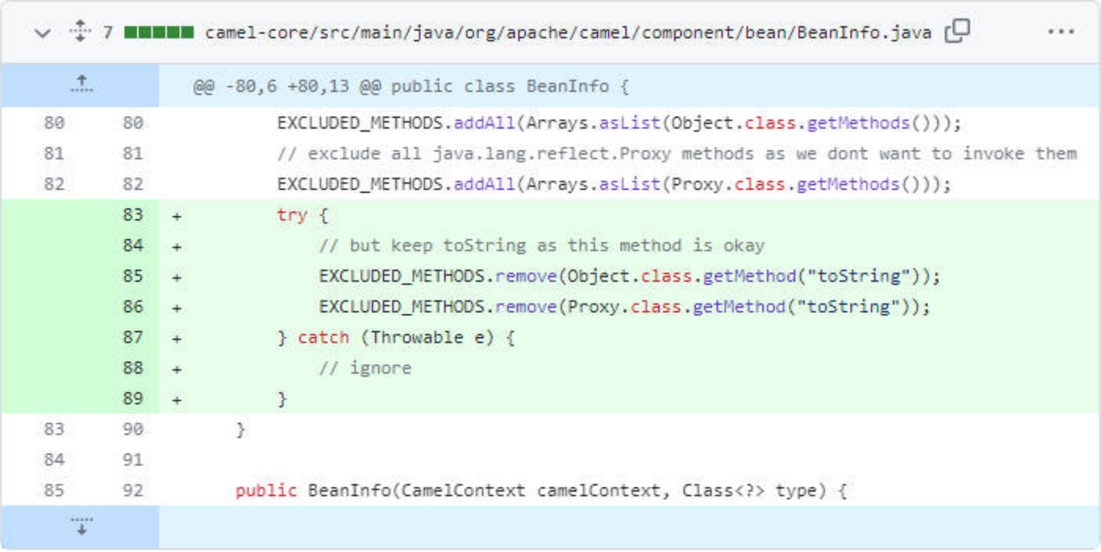
Figure 1,2,3 show the code changes of `ProcessInstance.java`, `BeanInfo.java`, and `CamelContextHelper.java`.



```
components/camel-bam/src/main/java/org/apache/camel/bam/model/ProcessInstance.java
```

```
@@ -27,6 +27,7 @@
  import javax.persistence.Id;
  import javax.persistence.ManyToOne;
  import javax.persistence.OneToMany;
+ import javax.persistence.UniqueConstraint;
  import java.util.Collection;
  import java.util.Date;
  import java.util.HashSet;
@@ -37,6 +38,7 @@
   * @version $Revision: $
   */
  @Entity
+ @UniqueConstraint(columnNames = {"correlationKey"})
  public class ProcessInstance extends TemporalEntity {
      private static final transient Log log =
  LogFactory.getLog(ProcessInstance.class);
      private ProcessDefinition processDefinition;
```

Fig. 1. Code change of `ProcessInstance.java`.

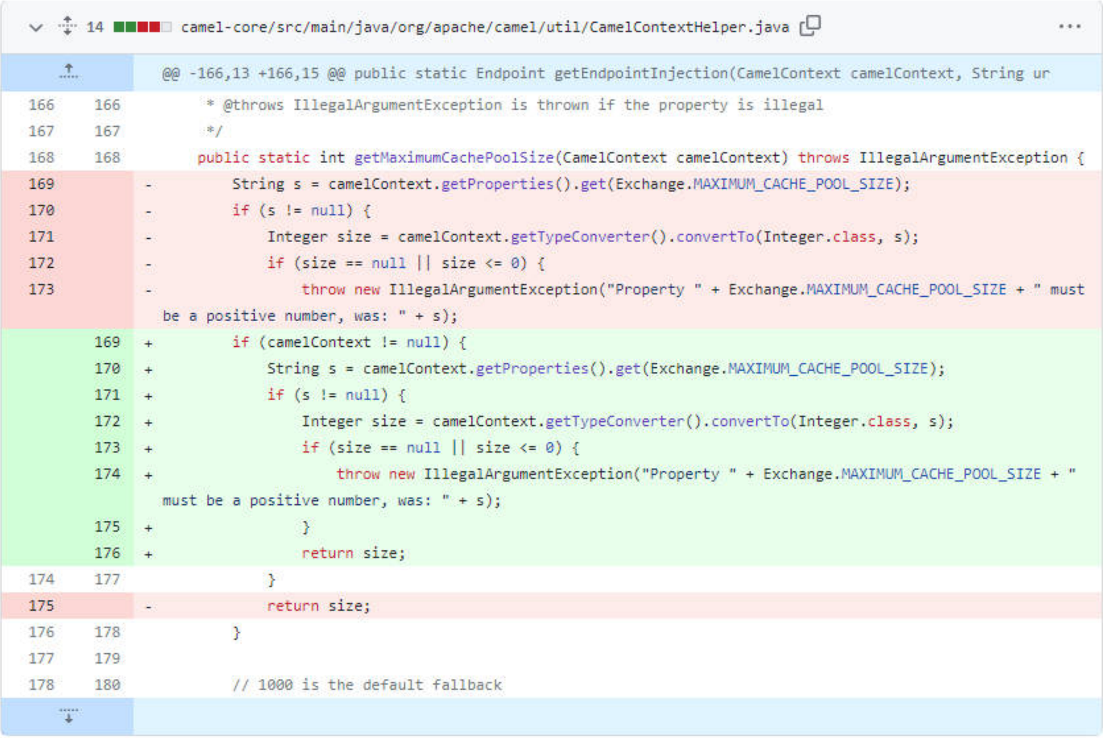


```
camel-core/src/main/java/org/apache/camel/component/bean/BeanInfo.java
```

```
@@ -80,6 +80,13 @@ public class BeanInfo {
              EXCLUDED_METHODS.addAll(Arrays.asList(Object.class.getMethods()));
              // exclude all java.lang.reflect.Proxy methods as we dont want to invoke them
              EXCLUDED_METHODS.addAll(Arrays.asList(Proxy.class.getMethods()));
+             try {
+                 // but keep toString as this method is okay
+                 EXCLUDED_METHODS.remove(Object.class.getMethod("toString"));
+                 EXCLUDED_METHODS.remove(Proxy.class.getMethod("toString"));
+             } catch (Throwable e) {
+                 // ignore
+             }
          }

          public BeanInfo(CamelContext camelContext, Class<?> type) {
```

Fig. 2. Code change of `BeanInfo.java`.



14 camel-core/src/main/java/org/apache/camel/util/CamelContextHelper.java

```
@@ -166,13 +166,15 @@ public static Endpoint getEndpointInjection(CamelContext camelContext, String ur
      * @throws IllegalArgumentException is thrown if the property is illegal
      */
     public static int getMaximumCachePoolSize(CamelContext camelContext) throws IllegalArgumentException {
-        String s = camelContext.getProperties().get(Exchange.MAXIMUM_CACHE_POOL_SIZE);
-        if (s != null) {
-            Integer size = camelContext.getTypeConverter().convertTo(Integer.class, s);
-            if (size == null || size <= 0) {
-                throw new IllegalArgumentException("Property " + Exchange.MAXIMUM_CACHE_POOL_SIZE + " must be a positive number, was: " + s);
+        if (camelContext != null) {
+            String s = camelContext.getProperties().get(Exchange.MAXIMUM_CACHE_POOL_SIZE);
+            if (s != null) {
+                Integer size = camelContext.getTypeConverter().convertTo(Integer.class, s);
+                if (size == null || size <= 0) {
+                    throw new IllegalArgumentException("Property " + Exchange.MAXIMUM_CACHE_POOL_SIZE + " must be a positive number, was: " + s);
+                }
+                return size;
             }
-            return size;
         }
 
         // 1000 is the default fallback
```

Fig. 3. Code change of `CamelContextHelper.java`.