Base Language - Bug #7089

sanitize p2j.oo method signature

02/01/2023 09:27 AM - Constantin Asofiei

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	50%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#2 - 02/01/2023 09:39 AM - Constantin Asofiei

In incremental conversion, it was found that a RequestBuilder.put method's LegacySignature was inconsistent with the method's real arguments.

This program can be used to check which methods have problems at the LegacySignature:

```
package testing;

import java.lang.reflect.Field;
import java.lang.reflect.Method;
import java.util.Set;
import java.util.TreeSet;

import org.reflections.Reflections;
import org.reflections.scanners.SubTypesScanner;

import com.goldencode.p2j.oo.lang._BaseObject_;
import com.goldencode.p2j.util.BaseDataType;
import com.goldencode.p2j.util.LegacyParameter;
import com.goldencode.p2j.util.LegacySignature;
import com.goldencode.p2j.util.OutputExtentParameter;
import com.goldencode.p2j.util.OutputTableHandle;

public class LegOOCheck
{
   public static void main(String[] args)
   {
      Reflections reflections = new Reflections("com.goldencode.", new SubTypesScanner(false));
      Set<Class<?>> classes = new TreeSet<>((c1, c2) -> c1.getName().compareTo(c2.getName()));
      classes.addAll(reflections.getSubTypesOf(_BaseObject_.class));

      for (Class<?> cls : classes)
      {
         Field[] fields = cls.getDeclaredFields();
         Method[] methods = cls.getDeclaredMethods();

         for (int i = 0; i < methods.length; i++)
         {
            Method m = methods[i];
            LegacySignature ls = m.getAnnotation(LegacySignature.class);
            if (ls == null)
            {
               continue;
            }

            Class<?>[] ptypes = m.getParameterTypes();

            LegacyParameter[] lsp = ls.parameters();
            if (lsp.length != ptypes.length)
            {
```

```
               System.out.println("1. " + m.toString());
            }

            for (int j = 0; j < lsp.length; j++)
            {
               LegacyParameter p1 = lsp[j];
               Class<?> p2 = ptypes[j];
               if (p2.isArray())
               {
                  p2 = p2.getComponentType();
               }
               if (p2 == OutputExtentParameter.class || p2 == OutputTableHandle.class || p2 == Object.class ||
p2 == String.class || p2 == long.class)
               {
                  continue;
               }

               if (!p1.type().equalsIgnoreCase(p2.getSimpleName()))
               {
                  System.out.println("2. " + j + " " + m.toString());
               }
               if (p2 == com.goldencode.p2j.util.object.class && p1.qualified().isEmpty())
               {
                  System.out.println("3. " + j + " " + m.toString());
               }
               if (!p1.qualified().isEmpty() && p2 != com.goldencode.p2j.util.object.class)
               {
                  System.out.println("4. " + j + " " + m.toString());
               }
            }

            int j = -1;
            Class<?> p2 = m.getReturnType();
            if (p2.isArray())
            {
               p2 = p2.getComponentType();
            }
            if (p2 == Object.class || p2 == BaseDataType.class)
            {
               continue;
            }
            if (!ls.returns().equalsIgnoreCase(p2.getSimpleName()))
            {
               System.out.println("2. " + j + " " + m.toString());
            }
            if (p2 == com.goldencode.p2j.util.object.class && ls.qualified().isEmpty())
            {
               System.out.println("3. " + j + " " + m.toString());
            }
            if (!ls.qualified().isEmpty() && p2 != com.goldencode.p2j.util.object.class)
            {
               System.out.println("4. " + j + " " + m.toString());
            }
         }

         for (int j = 0; j < fields.length; j++)
         {
            Field f = fields[j];
            LegacySignature ls = f.getAnnotation(LegacySignature.class);
            if (ls == null || ls.returns().isEmpty() || "void".equals(ls.returns()))
            {
               continue;
            }

            Class<?> p2 = f.getType();
            if (p2.isArray())
            {
               p2 = p2.getComponentType();
            }
            if (!ls.returns().equalsIgnoreCase(p2.getSimpleName()))
            {
               System.out.println("2. " + j + " " + f.toString());
            }
            if (p2 == com.goldencode.p2j.util.object.class && ls.qualified().isEmpty())
            {
```

```
                System.out.println("3. " + j + " " + f.toString());
            }
            if (!ls.qualified().isEmpty() && p2 != com.goldencode.p2j.util.object.class)
            {
                System.out.println("4. " + j + " " + f.toString());
            }
         }
      }
   }
}
```

The output of this program is:

```
3. -1 protected com.goldencode.p2j.util.object[] com.goldencode.p2j.oo.core.collections.Array.getValue()
2. 0 public void com.goldencode.p2j.oo.json.objectmodel.JsonArray.setLength(com.goldencode.p2j.util.int64)
2. 0 public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.logging.LogLevelEnum.getEnum(com.goldencode.p2j.util.int64)
2. 0 public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.UriSchemeEnum.getEnum(com.goldencode.p2j.util.int64)
2. 0 public void com.goldencode.p2j.oo.net.http.HttpRequest.setCookie(com.goldencode.p2j.util.object)
2. 0 public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.RequestBuilder.put(com.goldencode.p2j.util.object,com.goldencode.p2j.util.object)
4. 0 public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.RequestBuilder.put(com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)
2. 0 public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.RequestBuilder.options(com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)
4. 0 public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.RequestBuilder.options(com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)
4. 0 public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.net.http.RequestBuilder.patch(com.goldencode.p2j.util.character,com.goldencode.p2j.util.object)
2. 0 public static com.goldencode.p2j.util.object com.goldencode.p2j.oo.reflect.ParameterMode.getEnum(com.goldencode.p2j.util.int64)
```

OTOH, LegacySignature has duplicated code with the real Java method's signature... I think this can be reduced further, to infer certain annotation directly from the Java method.

#3 - 02/08/2023 02:08 PM - Constantin Asofiei

- % Done changed from 0 to 50

- Assignee set to Constantin Asofiei

- Status changed from New to WIP

I have the changes to solve the signature inconsistencies in p2j.oo, but I'll wait to create a branch after 6129c is merged to trunk.

The LegacySignature has duplicated code with the real Java method's signature.. part is not addressed yet.

#4 - 02/15/2023 05:21 AM - Constantin Asofiei

The current changes are in 7026a rev 14814.