

PublicFieldUberspect

JUnit tests can be found [here](#).

```
/*
 * Copyright 2003-2004 The Apache Software Foundation.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.apache.velocity.tools.generic.introspection;

import java.lang.reflect.Array;
import java.lang.reflect.Field;

import org.apache.velocity.util.introspection.Info;
import org.apache.velocity.util.introspection.UberspectImpl;
import org.apache.velocity.util.introspection.VelPropertyGet;
import org.apache.velocity.util.introspection.VelPropertySet;

/**
 * Uberspect implementation that exposes public fields.
 * Also exposes the explicit "length" field of arrays.
 *
 * <p>To use, tell Velocity to use this class for introspection
 * by adding the following to your velocity.properties:<br />
 *
 * <code>
 * runtime.introspector.uberspect = org.apache.velocity.tools.generic.introspection.PublicFieldUberspect
 * </code>
 * </p>
 *
 * @author <a href="mailto:shinobu@ieee.org">Shinobu Kawai</a>
 * @version $Id: $
 */
public class PublicFieldUberspect extends UberspectImpl
{
    /**
     * Default constructor.
     */
    public PublicFieldUberspect()
    {
    }

    /**
     * Property getter - returns VelPropertyGet appropos for #set($foo = $bar.woogie).
     * <br />
     * Returns a special {@link VelPropertyGet} for the <code>length</code> property of arrays.
     * Otherwise tries the regular routine. If a getter was not found,
     * returns a {@link VelPropertyGet} that gets from public fields.
     *
     * @param obj the object
     * @param identifier the name of the property
     * @param i a bunch of information.
     * @return a valid <code>VelPropertyGet</code>, if it was found.
     * @throws Exception failed to create a valid <code>VelPropertyGet</code>.
     */
    public VelPropertyGet getPropertyGet(Object obj, String identifier, Info i)
        throws Exception
    {
    }
}
```

```

{
    Class clazz = obj.getClass();
    boolean isArray = clazz.isArray();
    boolean isLength = identifier.equals("length");
    if (isArray && isLength)
    {
        return new ArrayLengthGetter();
    }

    VelPropertyGet getter = super.getPropertyGet(obj, identifier, i);
    // there is no clean way to see if super succeeded
    // @see http://issues.apache.org/bugzilla/show_bug.cgi?id=31742
    try
    {
        getter.getMethodName();
        return getter;
    }
    catch (NullPointerException notFound)
    {
    }

    Field field = obj.getClass().getField(identifier);
    if (field != null)
    {
        return new PublicFieldGetter(field);
    }

    return null;
}

/**
 * Property setter - returns VelPropertySet appropos for #set($foo.bar = "geir").
 * <br />
 * First tries the regular routine. If a setter was not found,
 * returns a {@link VelPropertySet} that sets to public fields.
 *
 * @param obj the object
 * @param identifier the name of the property
 * @param arg the value to set to the property
 * @param i a bunch of information.
 * @return a valid <code>VelPropertySet</code>, if it was found.
 * @throws Exception failed to create a valid <code>VelPropertySet</code>.
 */
public VelPropertySet getPropertySet(Object obj, String identifier,
    Object arg, Info i) throws Exception
{
    VelPropertySet setter = super.getPropertySet(obj, identifier, arg, i);
    if (setter != null)
    {
        return setter;
    }

    Field field = obj.getClass().getField(identifier);
    if (field != null)
    {
        return new PublicFieldSetter(field);
    }

    return null;
}

/**
 * Implementation of {@link VelPropertyGet} that gets from public fields.
 *
 * @author <a href="mailto:shinobu@ieee.org">Shinobu Kawai</a>
 * @version $Id: $
 */
protected class PublicFieldGetter implements VelPropertyGet
{
    /** The <code>Field</code> object representing the property. */
    private Field field = null;

```

```

/**
 * Constructor.
 *
 * @param field The <code>Field</code> object representing the property.
 */
public PublicFieldGetter(Field field)
{
    this.field = field;
}

/**
 * Returns the value of the public field.
 *
 * @param o the object
 * @return the value
 * @throws Exception failed to get the value from the object
 */
public Object invoke(Object o) throws Exception
{
    return this.field.get(o);
}

/**
 * This class is cacheable, so it returns <code>true</code>.
 *
 * @return <code>true</code>.
 */
public boolean isCacheable()
{
    return true;
}

/**
 * Returns <code>"public field getter"</code>, since there is no method.
 *
 * @return <code>"public field getter"</code>
 */
public String getMethodName()
{
    return "public field getter";
}
}

/**
 * Implementation of {@link VelPropertyGet} that gets length from arrays.
 *
 * @author <a href="mailto:shinobu@ieee.org">Shinobu Kawai</a>
 * @version $Id: $
 */
protected class ArrayLengthGetter implements VelPropertyGet
{
    /**
     * Constructor.
     */
    public ArrayLengthGetter()
    {
    }

    /**
     * Returns the length of the array.
     *
     * @param o the array
     * @return the length
     * @throws Exception failed to get the length from the array
     */
    public Object invoke(Object o) throws Exception
    {
        // Thanks to Eric Fxler for this refactor.
        return new Integer(Array.getLength(o));
    }
}

```

```

/**
 * This class is cacheable, so it returns <code>true</code>.
 *
 * @return <code>true</code>.
 */
public boolean isCacheable()
{
    return true;
}

/**
 * Returns <code>"array length getter"</code>, since there is no method.
 *
 * @return <code>"array length getter"</code>
 */
public String getMethodName()
{
    return "array length getter";
}
}

/**
 * Implementation of {@link VelPropertySet} that sets to public fields.
 *
 * @author <a href="mailto:shinobu@ieee.org">Shinobu Kawai</a>
 * @version $Id: $
 */
protected class PublicFieldSetter implements VelPropertySet
{
    /** The <code>Field</code> object representing the property. */
    private Field field = null;

    /**
     * Constructor.
     *
     * @param field The <code>Field</code> object representing the property.
     */
    public PublicFieldSetter(Field field)
    {
        this.field = field;
    }

    /**
     * Sets the value to the public field.
     *
     * @param o the object
     * @param value the value to set
     * @return always <code>null</code>
     * @throws Exception failed to set the value to the object
     */
    public Object invoke(Object o, Object value) throws Exception
    {
        this.field.set(o, value);
        return null;
    }

    /**
     * This class is cacheable, so it returns <code>true</code>.
     *
     * @return <code>true</code>.
     */
    public boolean isCacheable()
    {
        return true;
    }

    /**
     * Returns <code>"public field setter"</code>, since there is no method.
     *
     * @return <code>"public field setter"</code>
     */

```

```
        */  
    public String getMethodName()  
    {  
        return "public field setter";  
    }  
}
```