Tapestry5ManipulateClassesAtRuntime

in my applications many times i use pages with the grid component and i am too lazy 😉 to write in every page a setter and getter methode for grid's row parameter. so i help my self with the following classes and code segments...

the annotation ...

```
/**
 * generates setter/getter for a grid bean.
 *
 * @author <a href="mailto:shomburg@hsofttec.com">shomburg</a>
 * @version $Id: GridRowBean.java 41 2007-11-25 18:54:58Z shomburg $
 */
@Target(FIELD)
@Retention(RUNTIME)
@Documented
public @interface GridRowBean
{
}
```

the worker, it adds the setter and getter for the grid bean.

```
/**
 * add setter/getter methodes.
 *
 * @author <a href="mailto:shomburg@hsofttec.com">shomburg</a>
 * @version $Id: GridRowBeanWorker.java 41 2007-11-25 18:54:58Z shomburg $
 */
public class GridRowBeanWorker implements ComponentClassTransformWorker
{
    /**
     * Invoked to perform a transformation on an as-yet unloaded component class, represented by the
     * {@link org.apache.tapestry.services.ClassTransformation} instance. In some cases,
     * the worker may make changes to the
     * component model -- for example, a worker that deals with parameters may update the model to
     * reflect those parameters.
     */
    public void transform(ClassTransformation transformation, MutableComponentModel model)
    {
        List<String> names = transformation.findFieldsWithAnnotation(GridRowBean.class);

        if (names.isEmpty())
            return;

        for (String name : names)
        {
            addGridRowBeanGetter(transformation, name);
            addGridRowBeanSetter(transformation, name);
        }
    }

    private void addGridRowBeanGetter(ClassTransformation transformation, String fieldName)
    {
        String fieldType = transformation.getFieldType(fieldName);
        String methodName = "get" + StringUtils.capitalize(InternalUtils.stripMemberPrefix(fieldName));

        TransformMethodSignature sig =
                      new TransformMethodSignature(Modifier.PUBLIC, fieldType, methodName, null, null);

        BodyBuilder builder = new BodyBuilder();
        builder.begin();
        builder.addln("return %s;", fieldName);
        builder.end();

        transformation.addMethod(sig, builder.toString());
    }

    private void addGridRowBeanSetter(ClassTransformation transformation, String fieldName)
    {
        String fieldType = transformation.getFieldType(fieldName);
        String methodName = "set" + StringUtils.capitalize(InternalUtils.stripMemberPrefix(fieldName));

        TransformMethodSignature sig =
                   new TransformMethodSignature(Modifier.PUBLIC, "void",
                                methodName, new String[]{fieldType}, null);

        BodyBuilder builder = new BodyBuilder();
        builder.begin();
        builder.addln("%s = $1;", fieldName);
        builder.end();

        transformation.addMethod(sig, builder.toString());
    }
}
```

... contribute the new worker to Tapestry's ComponentClassTransformWorker ...

```
/**
 * Adds a number of standard component class transform workers:
 * <ul>
 * <li>GridRowBean -- generates setter/getter for a grid bean</li>
 * </ul>
 */
public static void contributeComponentClassTransformWorker(
                            OrderedConfiguration<ComponentClassTransformWorker> configuration)
{
    configuration.add("GridRowBean", new GridRowBeanWorker(), "after:Inject*");
}
```

... mark the bean _client with @GridRowBean annotation ...

```
@GridRowBean
private Client _clientRow;

/**
 * get the grid data from datasource.
 *
 * @return grid data from datasource
 */
public GridDataSource getGridSource()
{
    ClientDAO entityDAO = (ClientDAO) getDefaultDAO(Client.class);
    return new HibernateDataSource(entityDAO, "FROM Client");
}
```

... tell the grid component wich property should use for the row parameter ...

```
<table t:type="Grid" model="gridModel" source="gridSource" row="clientRow">

    <t:parameter name="recIdCell">
        <span t:type="core5/link/EditAction" context="clientRow.recId"/>
    </t:parameter>

    <t:parameter name="addressCityCell">
        ${clientRow.address.country.isoSign}
        ${clientRow.address.zip}
        ${clientRow.address.city}
    </t:parameter>

</table>
```