Artículo
Mathew Lambert · Jun 25, 2020 Lectura de 3 min

Proyecciones de clase y clases para proyecciones

El propósito de este artículo es reforzar el perfil de un procedimiento muy potente que ha estado disponible desde hace mucho tiempo para nosotros, y abrir un debate sobre las maneras en que puede utilizarse (o explotarse).

Puedes leer más información sobre el el mecanismo aquí. En resumen, cuando definas una clase puedes utilizar la palabra clave Projection para referirte a una o más clases de proyección. Una clase de proyección permite implementar métodos que se llaman desde puntos clave en el ciclo de vida de tu clase.

Una clase de proyección debe contener %Projection.AbstractProjection y normalmente implementará al menos uno de los siguientes métodos:

- CreateProjection
- RemoveProjection

Se llamará a CreateProjection una vez que tu clase haya sido compilada. Se llamará a RemoveProjection justo antes de que tu clase se compile nuevamente, o justo antes de que se elimine.

Creo que este procedimiento originalmente se utilizó para generar archivos Java, que implementaron una proyección en Java para una clase de Caché. Desde entonces, este procedimiento se esta usando ampliamente y se esta volviendo cada vez más sofisticado. En la versión 2015.2 conté veinticuatro %-clases que provienen de %Projection.AbstractProjection.

Además de la forma en que InterSystems utiliza el procedimiento, también observé que este se aprovecha de otras maneras. Aquí hay un par de ejemplos:

- UMLExplorer se envía como un archivo XML que contiene cuatro clases. Una de ellas es una clase para proyecciones llamada ClassExplorer.WebAppInstaller, la cual se proyecta ingeniosamente a sí misma:

```
Class ClassExplorer.WebAppInstaller Extends %Projection.AbstractProjection
{

Projection Reference As WebAppInstaller;
...
```

Por lo tanto, cuando esta clase se compila, se ejecuta su método CreateProjection, realizando cualesquiera sean los pasos que el desarrollador codificó ahí. En este caso añade una aplicación web llamada /CacheExplorer, pero podría hacer cualquier cosa que permitan los permisos de la persona que compiló la clase.

- Un sitio que usa la herramienta Deltanji para administrar el código fuente creó una clase para proyectar utilidades. Cada vez que activan una pieza de trabajo que requiere de algunos pasos para instalarse y ejecutarse en un namespace objetivo, ellos implementan esos pasos en una clase de distribución (D), que tiene una proyección que hace referencia a su clase para proyectar utilidades (P). Entonces, ellos hacen un paquete (D) con la pieza de trabajo. Cuando (D) se carga y compila en un namespace objetivo, el método CreateProjection de (P) es llamado automáticamente y se le pasa el nombre de la clase (D), lo que le permite llamar a los métodos de (D)

Si has visto otras maneras de utilizar la proyección, o si tú mismo has diseñado una nueva forma de usarla,

compártela como un comentario en esta publicación

Una última reflexión que me gustaría compartir. Este procedimiento significa que probablemente deberíamos pensarlo dos veces antes de realizar una compilación durante la importación de un archivo XML, en especial si no estamos seguros de que podemos confiar en su contenido. El alcance que tiene una proyección maliciosa de clase es grande.

#Modelo de datos de objetos #Caché