

---

Artículo

[Alberto Fuentes](#) · 24 dic, 2021 Lectura de 4 min

[Open Exchange](#)

## Probar los paquetes que desarrollamos con ZPM

¡Hola desarrolladores!

Como sabéis, es muy fácil publicar un [paquete en el repositorio de la Comunidad de Desarrolladores](#).

Pero, ¿cómo se puede probar antes de publicarlo? ¿Cómo se puede garantizar que todo lo que se introdujo en module.xml funciona correctamente?

¡Seguid leyendo las instrucciones que os detallo a continuación!

En este artículo, revisaremos paso a paso el proceso manual para probar el módulo ZPM.

Utilizaremos la plantilla [objectscript-rest-template](#) como ejemplo.

1. Clonar el proyecto con git clone y abrirlo en VSCode.

```
git clone git@github.com:intersystems-community/objectscript-rest-docker-template.git
```

2. Ejecutar el contenedor Docker con:

```
docker-compose up -d --build
```

3. Abrir un terminal en IRIS e iniciar zpm de la siguiente manera:

```
$ docker-compose exec iris iris session iris
USER>zn "IRISAPP"
```

```
IRISAPP>zpm
```

```
zpm:IRISAPP>
```

4. Cargar el paquete. Con este paso, verificaremos si los recursos que aparecen en la lista de module.xml realmente existen. La carpeta /irisdev/app se encuentra en la lista de docker-compose como mapeo de volúmenes y apunta a la carpeta repo.

```
zpm:IRISAPP>load /irisdev/app
```

```
[rest-template] Reload START
[rest-template] Reload SUCCESS
[rest-template] Module object refreshed.
[rest-template] Validate START
[rest-template] Validate SUCCESS
[rest-template] Compile START
[rest-template] Compile SUCCESS
[rest-template] Activate START
[rest-template] Configure START
```

```
[rest-template] Configure SUCCESS
[rest-template] Activate SUCCESS
zpm: IRISAPP>
```

5. Probar la compilación del paquete mediante el comando 'package-name package'. Comprobar que ninguno de los registros contiene errores

```
zpm:IRISAPP>rest-template package
```

### 6. Probar la publicación

Si no hay errores en el paquete, podemos publicarlo. Pero antes de publicarlo en Open Exchange e invitar a todos a que lo utilicen, merece la pena comprobar si todas las configuraciones y la propia instalación realmente se ejecutan como queremos.

Pero, ¿cómo hacerlo si el repositorio de la comunidad está inhabilitado para realizar publicaciones directas? La respuesta es: publicar el paquete en un repositorio de prueba y después desplegarlo con otro cliente zpm.

Podéis configurar vuestro propio repositorio (aquí tenéis cómo hacerlo) y cambiar el repositorio zpm mediante el comando:

```
zpm:IRISAPP>repo -n registry -r -url http://your-server:52773/registry/ -user username -pass password
```

Para hacerlo más sencillo aún y que no tengáis que montar vuestro propio repositorio, existe un repositorio zpm público de pruebas que podemos utilizar directamente:

```
zpm:IRISAPP>repo -n registry -r -url https://test.pm.community.intersystems.com/registry/ -user test -pass PassWord42
```

A continuación, llamar al comando publish:

```
zpm: IRISAPP>rest-template publish

[rest-template] Reload START
[rest-template] Reload SUCCESS
[rest-template] Module object refreshed.
[rest-template] Validate START
[rest-template] Validate SUCCESS
[rest-template] Compile START
[rest-template] Compile SUCCESS
[rest-template] Activate START
[rest-template] Configure START
[rest-template] Configure SUCCESS
[rest-template] Activate SUCCESS
[rest-template] Package START
Module exported to:
    /tmp/diraGF1Ui/rest-template-1.0.6/

Module package generated:
    /tmp/diraGF1Ui/rest-template-1.0.6.tgz
[rest-template] Package SUCCESS
[rest-template] Register START
[rest-template] Register SUCCESS
[rest-template] Publish START
```

```
[rest-template] Publish SUCCESS
zpm: IRISAPP>
```

Y ahora ya podemos comprobar si la aplicación se encuentra en el repositorio:

```
zpm: IRISAPP>search

registry https://test.pm.community.intersystems.com/registry/:
rest-template                               1.0.6
zpm: IRISAPP>
```

### 7. Probar el despliegue

Necesitaremos una instancia limpia de IRIS con ZPM, instalar el módulo y ver si todo funciona tal y como se planificó. ¿Dónde se puede conseguir instancia limpia de IRIS? - ¡Pues creamos un nuevo Docker :D!

Por ejemplo:

```
$ docker run --name my-iris -d --publish 9091:51773 --publish 9092:52773 intersystems
dc/irishealth-community:2020.2.0.196.0-zpm
```

Abrir terminal e iniciar zpm:

```
$ docker exec -it my-iris iris session IRIS
```

```
USER>zpm
```

```
zpm: USER>
```

Cambiar el repositorio zpm al de pruebas:

```
zpm: USER>repo -n registry -r -url https://test.pm.community.intersystems.com/registr
y/ -user test -pass Password42
```

```
pm: IRISAPP>search
```

```
registry https://test.pm.community.intersystems.com/registry/:
rest-template                               1.0.6
```

Instalar el paquete:

```
zpm: IRISAPP>install rest-template
```

```
[rest-template] Reload START
[rest-template] Reload SUCCESS
[rest-template] Module object refreshed.
[rest-template] Validate START
[rest-template] Validate SUCCESS
[rest-template] Compile START
[rest-template] Compile SUCCESS
[rest-template] Activate START
[rest-template] Configure START
[rest-template] Configure SUCCESS
[rest-template] Activate SUCCESS
```

zpm: IRISAPP

Comprobar si la aplicación funciona correctamente.

En nuestro caso, hay que abrir el navegador en `localhost:9092/crud/_spec` y comprobar que se instaló y funciona correctamente.



Tras esta prueba, se puede enviar la aplicación a Open Exchange y tener el módulo publicado en el repositorio oficial de la Comunidad [pm.community.intersystems.com](https://community.intersystems.com) y disponible para todo el mundo! No olvides marcar la casilla Publish in Package Manager ;)

Happy coding!

[#Despliegue](#) [#ObjectScript](#) [#InterSystems Package Manager \(IPM\)](#) [#InterSystems IRIS](#) [#Open Exchange](#)  
[Ir a la aplicación en InterSystems Open Exchange](#)

---

URL de fuente: <https://es.community.intersystems.com/post/probar-los-paquetes-que-desarrollamos-con-zpm>