
Artículo

[Ricardo Paiva](#) · 17 jun, 2022 Lectura de 2 min

[Open Exchange](#)

Cómo implementar una aplicación de gran tamaño con el cliente ZPM y Config-API

¡Hola a todos!

En el artículo anterior, combinamos ZPM con Config-API para cargar una configuración en la instalación/carga de un módulo. Podría ser útil para aplicaciones pequeñas, pero no es conveniente para aplicaciones grandes.

Puede que necesites crear varios namespaces e instalar muchos módulos zpm en diferentes namespaces. Usar el documento de configuración en la instalación/carga de zpm es un problema para este caso especial. Además, incluso si solo necesitas un módulo y un namespace, el cliente ZPM está diseñado para que se instale en el namespace actual y no es posible cambiar a otro namespace durante la instalación. Por lo tanto, si necesitamos que se instale en un namespace específico, debe existir antes de utilizar el cliente ZPM.

Desde la versión Config-API 1.0.2, se añadió una sección ZPM al documento de configuración para ejecutar comandos.

Esta funcionalidad permite ejecutar varios comandos zpm en un namespace específico.

Para mostrar cómo funciona, utilizamos un documento de configuración mínimo con:

- 2 namespaces (NS1 y NS2)
- 2 bases de datos (DB1 y DB2)
- Instalación de 1 módulo en el namespace NS1 y 2 módulos en el namespace NS2.

```
{
  "Defaults": {
    "DBDATA1" : "${MGRDIR}irisapp1/",
    "DBDATA2" : "${MGRDIR}irisapp2/",
    "APPNS" : "IRISAPP"
  },
  "SYS.Databases": {
    "${DBDATA1}" : {},
    "${DBDATA2}" : {}
  },
  "Databases": {
    "DB1" : {
      "Directory" : "${DBDATA1}"
    },
    "DB2" : {
      "Directory" : "${DBDATA2}"
    }
  },
  "Namespaces": {
    "NS1": {
      "Globals": "DB1"
    },
    "NS2": {
      "Globals": "DB2"
    }
  }
},
```

```
"ZPM": {
  "NS1" : [
    {
      "install": "module-name-1"
    }
  ],
  "NS2" : [
    {
      "install": "module-name-2"
    },
    {
      "install": "module-name-3"
    }
  ]
}
```

**

En la sección ZPM, la primera clave es el namespace (se permite la sintaxis \${var}). Para cada namespace, se puede establecer una matriz de objetos con el par "command":"args".

Para realizar pruebas, recomiendo utilizar esta bifurcación de [intersystems-iris-dev-template](#). Puedes ver cómo funciona la plantilla observando estos archivos:

- [iris.script](#)
- y el documento de configuración [iris-config.json](#)

Espero que esta función pueda ayudar con la implementación.

[#DevOps](#) [#InterSystems](#) [IRIS](#)

[Ir a la aplicación en InterSystems Open Exchange](#)

URL de
fuente: <https://es.community.intersystems.com/post/c%C3%B3mo-implementar-una-aplicaci%C3%B3n-de-gran-tama%C3%B1o-con-el-cliente-zpm-y-config-api>