

Lección 3 - Configuración del servidor web

Tabla de contenido

- [Introducción](#)
- [Objetivos](#)
- [Desarrollo](#)
 - [Correr un proyecto en una ruta completa](#)
 - [Correr una aplicación en otro puerto](#)
 - [Levantando dos nuevos proyectos en diferentes puertos](#)
 - [Activar y desactivar el modo desarrollador](#)
- [Conclusión](#)
- [Conoce más del autor](#)

Introducción

Configurar el servidor web en Revel es fundamental para el desarrollo eficiente de aplicaciones web en Go. Revel proporciona herramientas y funcionalidades poderosas que permiten ejecutar, controlar y modificar fácilmente el entorno de desarrollo según las necesidades del proyecto. En esta lección, exploraremos cómo configurar el servidor web en Revel y aprovechar al máximo sus capacidades.

Objetivos

1. **Correr Proyectos en Diferentes Puertos:** Aprenderemos a ejecutar nuestros proyectos en puertos específicos y entenderemos cómo configurar el servidor para adaptarlo a distintos entornos de desarrollo.
 2. **Modificar la Configuración desde app.conf:** Exploraremos cómo modificar la configuración del servidor directamente desde el archivo `app.conf`, permitiendo ajustar aspectos como el puerto de escucha y otras opciones relacionadas con el servidor.
 3. **Aprovechar la Funcionalidad de Vigilancia:** Comprenderemos cómo utilizar la opción `watch` en Revel para supervisar los archivos del proyecto y reiniciar automáticamente la aplicación cuando se detecten cambios, facilitando el proceso de desarrollo y pruebas.
-

Desarrollo

Correr un proyecto en una ruta completa

Para correr un proyecto en una ruta completa, utilizamos los siguientes comandos:

```
revel new -a github.com/jersonmartinez/FrameworksWebEnGo/Revel/FirstApp
revel run -a github.com/jersonmartinez/FrameworksWebEnGo/Revel/FirstApp
```

Correr una aplicación en otro puerto

Si deseamos correr una aplicación en un puerto diferente, utilizamos el siguiente comando:

```
revel run -p 5000 RevelApplication/
```

También se puede configurar el puerto desde el fichero `./config/app.conf`. Recordemos que en un proyecto Revel, el archivo `app.conf` es un archivo de configuración que define diferentes aspectos de la aplicación, incluidos aspectos relacionados con el modo de desarrollo.

Modificando la siguiente variable:

```
# The port on which to listen.
http.port = 5000
```

Levantando dos nuevos proyectos en diferentes puertos

Para levantar dos nuevos proyectos en diferentes puertos, seguimos estos pasos:

```
# En una primera terminal
revel new -a FirstApplication
revel run -p 5000 FirstApplication

pwd

# En una segunda terminal
revel new -a SecondApplication
revel run -p 5001 SecondApplication
```

Modificar el contenido de salida de `FirstApplication` en la siguiente ruta
`./app/views/App/index.html`:

```
<h1>It works! - First App</h1>
```

Modificar el contenido de salida de `SecondApplication` en la siguiente ruta

```
./app/views/App/index.html:
```

```
<h1>It works! - Second App</h1>
```

Corre las aplicaciones en los diferentes puertos: `http://localhost:5000/` `http://localhost:5001/`

¿Te has preguntado si Revel puede recargar de forma automática el servidor si encuentra algún cambio desde el desarrollo? Por supuesto. Revel vigila su proyecto y permite la recarga en caliente de varios tipos de fuentes. Para eso buscamos la opción `watch`, en el fichero `./config/app.conf`.

```
watch = true
```

La opción `watch` en el archivo `app.conf` controla si Revel debe monitorear los archivos del proyecto para cambios y reiniciar automáticamente la aplicación cuando se detectan cambios.

Cuando el valor de `watch` está configurado como `true`, Revel estará en modo de "vigilancia" (watch mode). En este modo, Revel supervisa continuamente los archivos de la aplicación en busca de cambios. Si detecta que un archivo ha sido modificado, por ejemplo, un archivo de código fuente, un archivo de plantilla HTML o un archivo de configuración, Revel reiniciará automáticamente la aplicación para aplicar los cambios.

Esto es útil durante el desarrollo, ya que permite una experiencia de desarrollo más fluida y eficiente, ya que los desarrolladores pueden ver los cambios reflejados en la aplicación de inmediato sin tener que reiniciar manualmente el servidor.

Por otro lado, si el valor de `watch` se establece como `false`, Revel no supervisará los cambios en los archivos y la aplicación no se reiniciará automáticamente cuando se realicen modificaciones en el código fuente u otros archivos relacionados con la aplicación. Esto puede ser útil en entornos de producción donde no se desean reinicios automáticos de la aplicación.

Lo cambiamos a `false` y comprobamos que no funciona el monitoreo, ni reinicia en automático la aplicación con los nuevos cambios.

Al cambiar el valor de `watch` en el archivo `app.conf`, estás controlando si Revel debe o no supervisar los archivos del proyecto y reiniciar automáticamente la aplicación cuando se realicen cambios, lo que afecta directamente la experiencia de desarrollo y el comportamiento de la aplicación en diferentes entornos.

Más abajo encontramos algunas variantes de la funcionalidad de la opción `watch`.

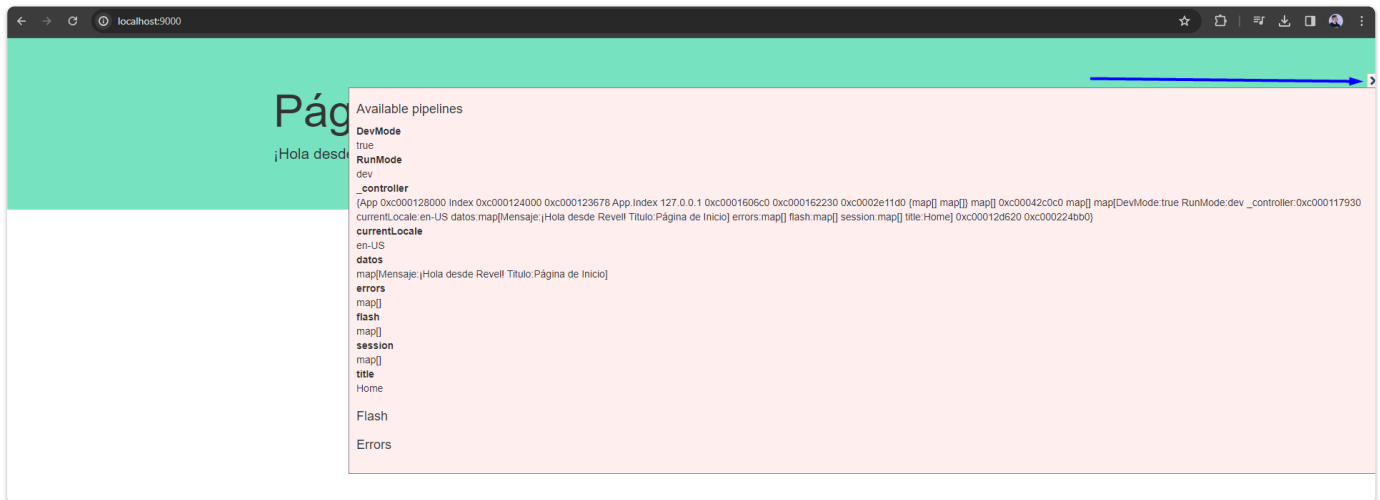
```
watch.mode = eager | normal
```

Si `watch.mode = "eager"`, el servidor empieza a recompilar la aplicación cada vez que cambian los archivos de la aplicación.

Si `watch.mode = "normal"`, el servidor recompila con una petición, por ejemplo, una actualización del navegador.

Activar y desactivar el modo desarrollador

Por omisión, esta opción se encuentra activada y lo podemos comprobar mediante este botón que aparece en la web:



En el mismo fichero de configuración, está la opción `mode.dev`:

```
mode.dev = false
```

Si la volvemos a activar, aparecerá el botón.

Conclusión

Al comprender estas configuraciones, podemos adaptar el entorno de desarrollo según nuestras necesidades específicas.

¡Continúa explorando y configurando el servidor web en Revel para aprovechar al máximo este poderoso framework de desarrollo web en Go!

Conoce más del autor

¡Encuéntrame en las siguientes redes sociales para estar al tanto de mis proyectos y actividades!

 Red Social	 Enlace
 Página web	jersonmartinez.com
 LinkedIn	Jerson Martínez - DevOps Engineer
 Canales de YouTube	DevOpsea Side Master
 GitHub	Perfil en GitHub
 Twitter (X)	@antoniomorenosm