Librerías para desarrollo web con Go 35/36

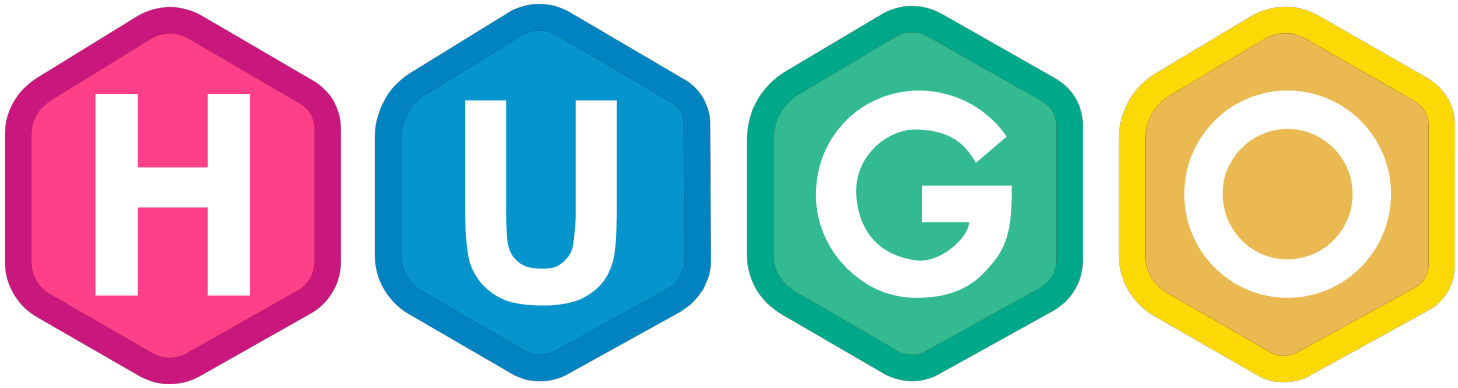
Curso Básico de Programación en Go

→|DATA SCIENCE CO...

Actualmente puedes usar Go para crear tu aplicación web sin necesidad de implementar librerías externas como sucede en muchos otros lenguajes. Aunque claro, esto tiene sus pros y sus contras. Pero al final del día todo dependerá del flujo de trabajo de tu equipo.

Sin embargo, te quiero compartir las librerías y frameworks de desarrllo web que se encuentran entre las más populares.

FrontEnd - Hugo



Hugo es el framework más popular para usar Go en el FrontEnd, claro no es que se utilice en FrontEnd específicamente sino que tiene su propio template para generar archivos estáticos (sin BackEnd, solo HTML, CSS y JavaScript).

Pero la mayor ventaja de Hugo, es que tienen Hugo Themes una tienda de templates donde muchos miembros de la comunidad suben sus temas para disponerlos al resto de la comunidad para su uso, solo debes prestar atención a la licencia. Pero encontrarás temas con todo tipo de features, desde uso para portafolio, compatibilidad con Google Anlytics y mucho más.

Y si lo combinas con Vercel podrás desplegar tu aplicación de forma gratuita.

BackEnd

Para el caso de BackEnd tenemos muchas opciones, las que te recomiendo probar son las siguientes:

Echo

Echo es el Framework que más he llegado a usar en las Apps en Go que he realizado. Esto porque reúne varias características:

- Es minimalista, evitando tener código que no vamos a usar
- Fácil de escalar y un excelente rendimiento al momento de compilar
- Tiene su sistema de enrutamiento, pudiendo incluso implementar middlewares (proceso de autenticación)
- Tiene su propia interfaz para implementar websockets (conexiones en tiempo real)
- Puedes usarlo tanto para crear APIs como páginas web completas.
- Es uno de los pocos frameworks que tiene su propia implementación de JWT (otro tipo de autenticación muy usado en APIs)

Gin-Gonic

Gin-Gonic Es un framework al que fácilmente le puedes agarrar cariño y esto es porque la forma como interactúas para crear Apps es muy intuitiva. De momento no es de los Frameworks que más destaque, pero tampoco se queda atrás. Ideal para tus primeros proyectos de BackEnd con Go.

Beego

beego_logo

Beego por bastante tiempo llegó a ser uno de los Frameworks más usados, no es que haya dejado de serlo, pero trajo a la comunidad 3 grandes features:

- Su propio ORM (Object Relational Mapping), básicamente son un conjunto de métodos para interactuar con la base de datos desde consulta a creación de tablas.
- Un CLI (Command Line Interface) llamado bee con el cual puedes generar proyecto ya sea para hacer una API como página web completa, empaquetar el proyecto listo para llevarlo a producción y hacer migraciones de bases de datos.
- Un admin, ¡sí un admin! Con el cual puedes ver estadísticas de las rutas de tu app, verificar el estado de la base de datos y ejecutar tareas de forma manual

Hoy en día Beego tiene muchos más features como por ejemplo crear el Dockerfile para que tu app se ejecute en Docker. Si vas a crear múltiples apps seguidamente, te recomiendo este Framework.

Revel

Revel

Revel es uno de los pesos pesados en desarrollo web para Go incluso fue conocido como el django de Go (django es el framework web más popular para Python) pero dejó de recibir actualizaciones por un tiempo significativo y también perdió un poco esa posición.

Hoy en día, Revel no solo ha llegado con nuevo look sino que también le han invertido a tener una de las documentaciones más completas incluso con apps de ejemplo.

Gorilla

Gorilla no es un Framework, ellos se definen como con herramientas para web. No tiene un ORM, tampoco un CLI, tampoco render de HTML, tampoco un admin. Entonces ¿por qué está en esta lista?

Pues en `gorilla` se enfocaron en una cosa hasta hacerla muy bien, estos son los WebSockets (para mantener conexiones en tiempo real por ejemplo un chat, algo en lo que Go destaca y por mucho). Con el paso del tiempo han agregado más features, como por ejemplo manejos de cookies, enrutador, etc.

Puedes desarrollar una web completa usando las librerías nativas de Go y con `gorilla` tienes esos extras que te ahorran implementación. Han hecho tan bien el código que incluso framworks web lo usan en su código sobre todo la parte de websockets.

Buffalo

Buffalo es otro de los pesos pesados en frameworks para web con Go. Hoy en día, es uno de los más completos tiene CLI, ORM, manejo de cookies, middlewares, templates para

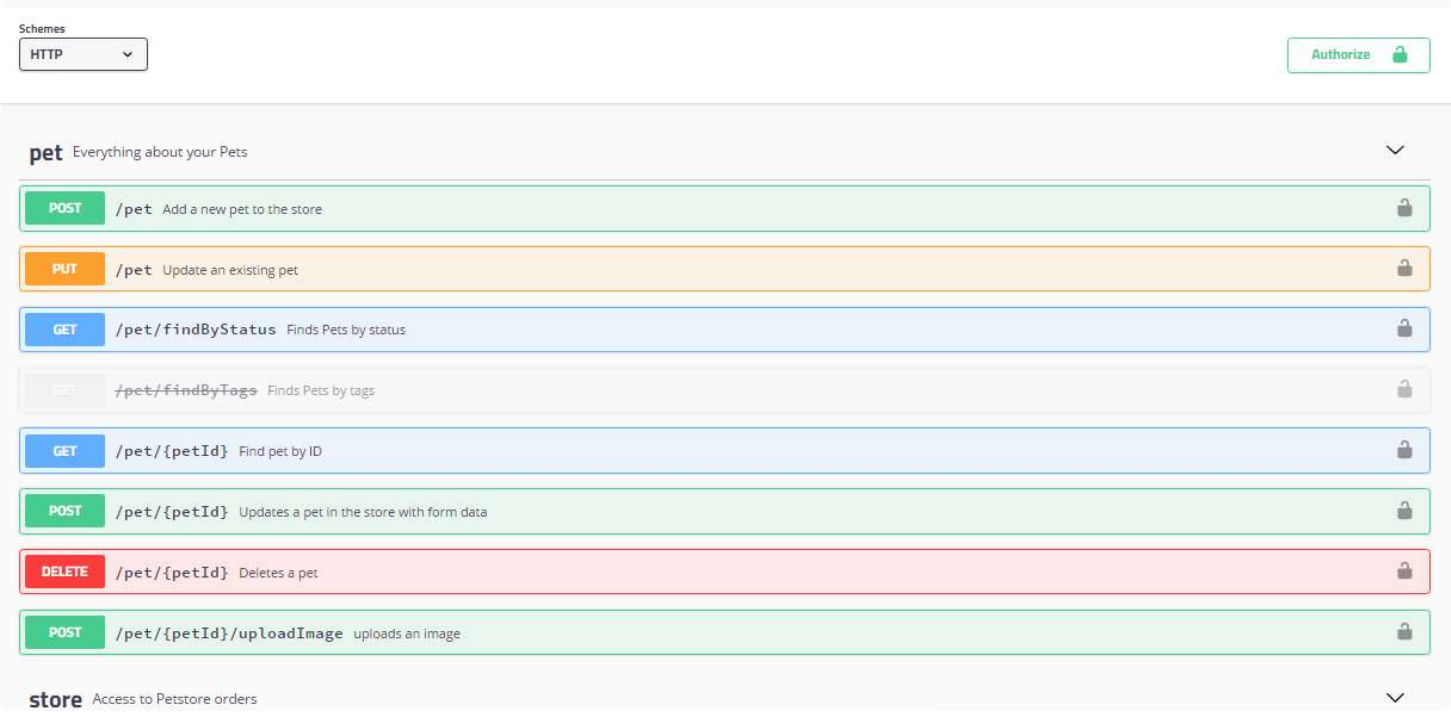
render de HTML, empaquetador con Docker e incluso puedes hacer despliegue a Heroku, Digital Ocean y Azure con el mismo CLI de Buffalo. Junto a otros features adicionales.

Sin embargo, el que sea más completo no significa que sea fácil de implementarlo. Es fácilmente recomendable para aquellas aplicaciones que necesite ser robustas desde el principio, mi sugerencia acá es que no seas el único(ca) desarrollador(ra) en el equipo, ya que el mantenimiento es tan pesado como lo es la librería.

Extra

Como programdor(ra) en más de una ocasión debes crear documentación de la API que has desarrollado, pues no suele ser de las tareas más divertidas al desarrollar. Pero si te dijera que puedes crear documentación en la medida que desarrollas ¿me creerías?

Digamos que sí me crees, ¿me seguirías creyendo si te dijera que generaría toda una interfaz parecida a esta?



Esto se pone más interesante, ahora imagina que esa documentación autogenerada además maneja autenticación e incluso es interactiva. Puedes verlo por ti mismo(ma) en

este swagger de ejemplo.

Allí es donde entra swaggo

swaggo es la forma como podemos implementar swagger en nuestro código en Go. De los frameworks que te he compartido acá, es compatible con Gin-Gonic, echo y búffalo. Beego, por su parte, está incluido en su CLI.

Como punto adicional, dependiendo de la librería los comentarios para hacer la documentación en `swagger` pueden generarse de forma automática o bien requiere una cierta intervención manual.