

goplay  Featured

goplay: Embed Go Playground on your Website



Ggicci T'ang

May 11, 2022 • 3 min read



```
package main

import "fmt"

func main() {
    fmt.Println("Hello, 世界")
}
```

Hello, 世界

Run



On this page

Demo

Demo 1. Hello World

Demo 2. Import third-party package from GitHub

How it works?

Set up a reverse proxy to go.dev

Use @ggicci/goplay

Hugo Shortcode goplay

Use goplay in your Ghost Blog

Who is using @ggicci/goplay?

This post is introducing a way to run Go snippets on websites. Which leveraged the official Go Playground service at <https://go.dev/play/>.

Demo

Let's take a quick look at the two demos below. You can click the **Run** button to see the output of the corresponding program. Click **Share** button to open the code in the official Go Playground to further edit/test the code on your hand.

Demo 1. Hello World

```
package main

func main() {
    println("hello world")
}
```

 Run  Share

Demo 2. Import third-party package from GitHub

```
package main

import (
    "fmt"
    "net/http"
    "net/http/httptest"

    "github.com/ggicci/httpin" // 3rd-party packages: will be auto-downloaded
    "github.com/justinas/alice"
)

type ListUsersInput struct {
    Page     int64 `in:"query=page;default=1"`
    PerPage  int64 `in:"query=per_page;default=20"`
    Token    string `in:"header=x-access-token;required"`
}

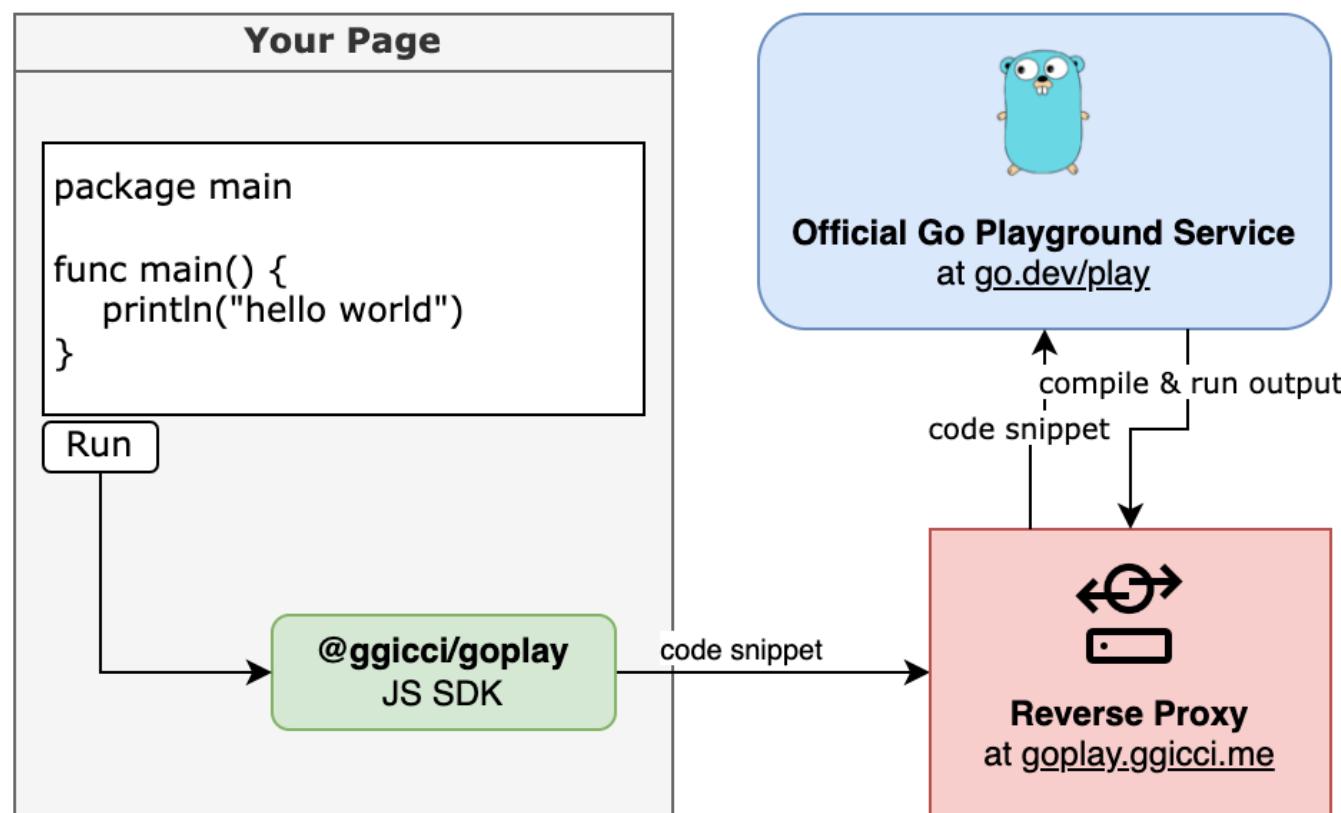
func ListUsers(rw http.ResponseWriter, r *http.Request) {
    input := r.Context().Value(httpin.Input).(*ListUsersInput)
    fmt.Printf("input: %#v\n", input)
    rw.WriteHeader(204)
}

func main() {
    r, _ := http.NewRequest("GET", "/?page=3", nil)
    r.Header.Set("X-Access-Token", "secret...")
    handler := alice.New(
        httpin.NewInput(ListUsersInput{}),
    ).ThenFunc(ListUsers)
    rw := httptest.NewRecorder()
    handler.ServeHTTP(rw, r)
}
```

```
input: &main.ListUsersInput{Page:3, PerPage:20, Token:"secret..."}  
Program exited.
```

 Run  Share

How it works?



Since we were not able to make AJAX requests directly to `go.dev` because of CORS issue. I setted up a **reverse proxy** `goplay.gdicci.me`, which works as a downstream of `go.dev`. It is responsible to make requests on behalf of the clients to `go.dev` and return the response back to the clients.

I also implemented a simple JS SDK `@ggicci/goplay` to easily interact with the go playground service. Use it on your page to connect to your own reverse proxy and run go snippets.

Set up a reverse proxy to `go.dev`

with Caddy:

```

goplay.gdicci.me {
    header {
        # https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers#cors
        access-control-allow-origin     "https://ggicci.me"
        access-control-allow-headers   "content-type"
        access-control-allow-methods   "OPTIONS, HEAD, GET, POST"
    }

    # only proxy the playground API of go.dev/play
    route /* {
        reverse_proxy https://go.dev {
            header_up Host go.dev
        }
    }

    respond 404
}

```

Use `@ggicci/goplay`

The dead simple SDK just implemented a class `GoPlayProxy` with 3 primary APIs:

```
class GoPlayProxy {
    // CTOR: specify a proxy url to connect
    constructor(public proxyUrl: string = '/goplay') {}
    // compile source code and get JSON response
    public async compile(sourceCode: string, goVersion?: CompilerVersion): Promise<Com
    // same as `compile`, but renders the JSON response as an `HTMLElement` into the s
    public async renderCompile(container: HTMLElement, sourceCode: string, goVersion?::
    // get the share URL of the source code
    public async share(sourceCode: string, goVersion?: CompilerVersion): Promise<strin
}
```

Visit <https://github.com/ggicci/goplay> for more details.

Hugo Shortcode `goplay`

If you are using `hugo` to generate your website. This section should be helpful to you.

By using the `goplay` shortcode, you can easily turn your Go code block into a Go playground. Take the following example, in your markdown file, just wrap the code block between `{% goplay %}` and `{% /goplay %}`.

`## Sample Code`

```
{% goplay %}

```go
package main

func main() {
 println("hello world")
}
```

{% /goplay %}
```

The latest code of `goplay - hugo shortcode` used by this site (former version) can be found [here](#). Save it under your hugo directory correspondingly. Typically it is `${YourHugoSiteRoot}/layouts/shortcodes/goplay.html`.

Use `goplay` in your Ghost Blog

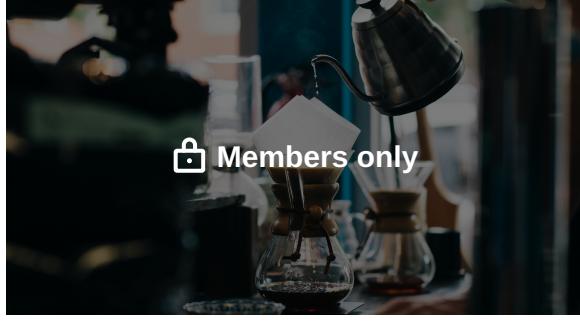
My site is now using [Ghost](#). Thanks to its **Code Injection** function (go to your blog settings and find it there), we can add our custom JavaScript code to drive `goplay` on our website.

The script I'm using on this website can be found here:

<https://gist.github.com/ggicci/c85b4665e959a10fbe0c97b33f44eb0>

Who is using @ggicci/goplay ?

- [httpin Documentation](#)
- add yours here by commenting below :)



```
API (ListAppUsers): GET /apps/:app/users
GET /apps/zoom/users?is_member=true&age_range=18&age_range=30
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

type ListAppUsersInput struct {
    AppName    string `in:"path=app"`           // "zoom"
    IsMember   bool   `in:"query=is_member"`     // true
    AgeRange  []int   `in:"query=age_range"`     // [18, 30]
    Authorization string `in:"header=authorization"` // "Bearer..."
}
```

喝咖啡的好处与坏处

Jan 16, 2023

Postgres ID Generator with Safe JSON Numbers

Before reading this post, I recommend you to read A Better ID Generator For...

Decode HTTP Query Params into a Struct in Go

Use httpin package to decode HTTP request params into a struct in Go.

GGICCI © 2024

[Sign up](#) · [中文](#)

Powered by Ghost