

Sponsor Roles (SR)

Github Sponsors does not have a way to natively apply Discord roles to sponsors like Patreon does, that's what this service does

Proposed by Carter Himmel

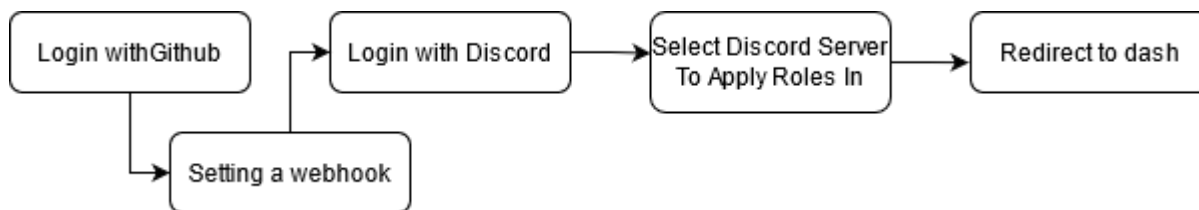
Terminology

creator - the person being sponsored

sponsor - the person sponsoring the creator

How?

Onboarding



Login with Github

Using the Github OAuth2 Flow (scopes **read:user** & **user:email**), creators will essentially link their Github account to their Sponsor Roles account. This is also one of two ways creators can log into the dashboard at a later date.

Setting a webhook

After ensuring the user is who they say they are, we'll provide the user with a payload URL and secret and also make sure they know to keep Content-Type set to **application/x-www-form-urlencoded**. Only after receiving the **ping** event from the webhook will we continue with the next step.

Login with Discord

Using the Discord OAuth2 Flow (scopes **email** and **guilds**), creators will essentially link their Discord account to their Sponsor Roles account. This allows us to ensure they are

who they say they are and allow us to fetch the guilds they have `MANAGE_GUILD` and `MANAGE_ROLES` in.

Select Discord server to Apply Roles In

In this step, the user will be able to select which server they'd like to apply roles to their sponsors in. Once clicking the server, they'll be redirected to Discord to add the bot to their server. After adding the bot to their server, they'll be redirected to the dashboard where they can specify which roles they'd like to apply to specific sponsor tiers. This is also one of two ways creators can log into the dashboard at a later date.

How to know which Discord user to apply roles to

In the `sponsorship` event payload, the sponsor's user object is sent.

One way to ensure the user is who they say they are is:

1. Authorization similar to Keybase -- creating a gist with data on the sponsor to prove who they are [example](#)
2. Require sponsors to log into the dashboard and link their Github and Discord

When we receive that event from Github, we can crossreference to what we have stored in our database and apply those roles to the user.

Handling new sponsors and tier changes

With the webhook we setup in the onboarding process, we'll receive the `sponsorship` event.

If a new sponsorship is created, the `action` field of that payload will be equal to `created`. On this event, we'll apply the roles if the user is in the server.

If the user changes their sponsorship tier, the `action` field of that payload will be equal to `tier_changed`. On this event, we'll remove roles associated with the old tier and add roles with the new tier.

We must also listen on the `GUILD_MEMBER_ADD` event from Discord to apply roles to a user upon joining the server.

Syncing

Sponsorship Tier Changes

Sponsors will likely want to create, delete or edit sponsorship tiers later down the road. To handle this, we need a cronjob to fetch sponsorship tiers every n minutes and ensure any tiers they've set roles for still exist.

Discord Roles

Roles being deleted

The creator or and admin of the server may delete roles. To handle this, we need a cronjob to fetch guild roles every `n` minutes and ensure they haven't deleted any configured roles. If they have, we'll remove that role from the configuration for that tier.

Role application falling out of sync

It's possible that we miss things. To handle this, we need a cronjob to fetch the current sponsors of the creator every `n` minutes and ensure they have their Discord roles.

Premium Tier

For a monthly fee, creators can:

- Specify more than one server to apply roles in
- Apply roles on more than `n` sponsorship tiers

Possible Domains

sponsorroles.com [namecheap](#)

sponsorrol.es [namecheap](#)

Short Domains

spnsrrol.es [namecheap](#)

spnrsrl.es [namecheap](#)

spnsrri.es [namecheap](#)

Notable Queries

Fetch a creators sponsors

```
{
  viewer {
    id
    databaseId
    email
  }
}
```

```

login
sponsorshipsAsSponsor(first: 100) {
  nodes {
    id
    tier {
      id
    }
    sponsorEntity {
      ... on User {
        id
        databaseId
        login
      }
    }
  }
}
createdAt
updatedAt
}

```

Fetch a creator's tiers

```

{
  viewer {
    id
    email
    login
    sponsorsListing {
      id
      tiers(first: 50) {
        nodes {
          id
          monthlyPriceInCents
          name
          createdAt
          updatedAt
        }
      }
    }
  }
}

```

```
}  
  createdAt  
  updatedAt  
}  
}
```