2. Get Credentials

This page explains how you get access, what credentials mean, and what still has to be approved before rewards can be issued.

Sandbox vs production

Environment	Purpose	What you can do
Sandbox	Build and test your integration	Create keys, make signed API calls, test donation links, test reward flows if a sandbox token pool is provisioned
Production	Issue real rewards	Requires approval, production credentials, and a production token pool

Self-service credentials

If you have a SIR Giving account with access to partner keys, creating your first API key automatically creates a Partner record linked to your user.

Use:

```
POST /v1/partner/keys
Authorization: Bearer <your user JWT>
Content-Type: application/json

{
  "name": "Sandbox Integration",
  "environment": "sandbox",
  "keyType": "secret"
}
```

The response includes:

```
{
  "id": "key_...",
  "partnerId": "...",
```

```
  "name": "Sandbox Integration",
  "environment": "sandbox",
  "publicKey": "pk_test_...",
  "secretKey": "sk_test_...",
  "hmacSecret": "64-character-hex-string"
}
```

Store `secretKey` and `hmacSecret` immediately. They are not shown again.

Production approval

Production rewards require approval because SIR tokens have real value and every reward draws from a token pool.

Before production launch, SIR Giving needs:

Item	Why it matters
Partner name and legal entity	Account and compliance record
Contact email	Operational alerts and key expiry notices
Use case summary	Confirms acceptable use and token economics
Expected monthly action volume	Sets rate limits and token pool size
Action types	Confirms what events you will submit
Stakeholder model	Confirms who receives rewards
Webhook URL	Lets SIR Giving send reward lifecycle events

After approval, SIR Giving provides or enables:

- Production API credentials.
- A production token pool.
- Campaign or reward rules, if needed.
- Production webhook configuration.

What each key is for

Key	Where it belongs	Use it for
`pk_test_...` or `pk_live_...`	Browser or mobile app	Donation links, public config, organization lookup

Key	Where it belongs	Use it for
`sk_test_...` or `sk_live_...`	Backend only	Users, actions, campaigns, token pools, webhooks, dashboard data
`hmacSecret`	Backend secret manager	Signing server-to-server requests

Check your partner record

After creating a key, verify your Partner record:

```
GET /v1/partner/keys/partner-info
Authorization: Bearer <your user JWT>
```

You should see your partner ID, name, slug, status, enabled features, and rate limit.

Check your keys

```
GET /v1/partner/keys
Authorization: Bearer <your user JWT>
```

This lists keys but does not return secret values.

Go-live checklist

Before switching to production:

- You have a `pk_live_...` and `sk_live_...` key.
- Your backend signs requests with the production `hmacSecret`.
- You have an active production token pool.
- Your webhook endpoint is reachable from the public internet.
- Your webhook handler verifies `X-SIR-Signature`.
- Your action requests use stable, unique `idempotencyKey` values.
- You have tested retries and rate limit handling.

Revision #5
Created 2026-05-10 09:11:11 UTC by krtin shet
Updated 2026-05-30 13:31:56 UTC by krtin shet