

NS-CA3

ريحانه سادات شكوهي

(۱) دستورالعمل

-۱


Settings / Developer settings / NS-app

General

Optional features

Advanced

NS-app

 **sr-ssh** owns this application. [Transfer ownership](#)

You can list your application in the [GitHub Marketplace](#) so that other users can discover it. [List this application in the Marketplace](#)


0 users [Revoke all user tokens](#)

Client ID
b96bfbbba06a62ef2b3

Client secrets [Generate a new client secret](#)

You need a client secret to authenticate as the application to the API.

Application logo

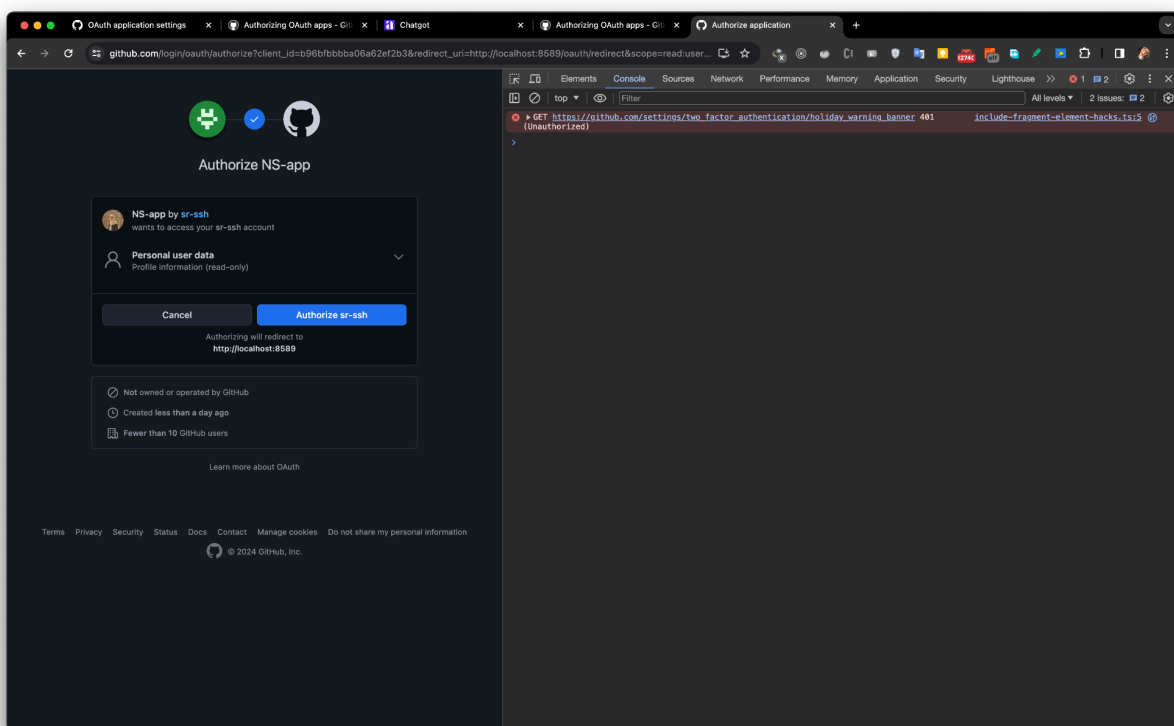
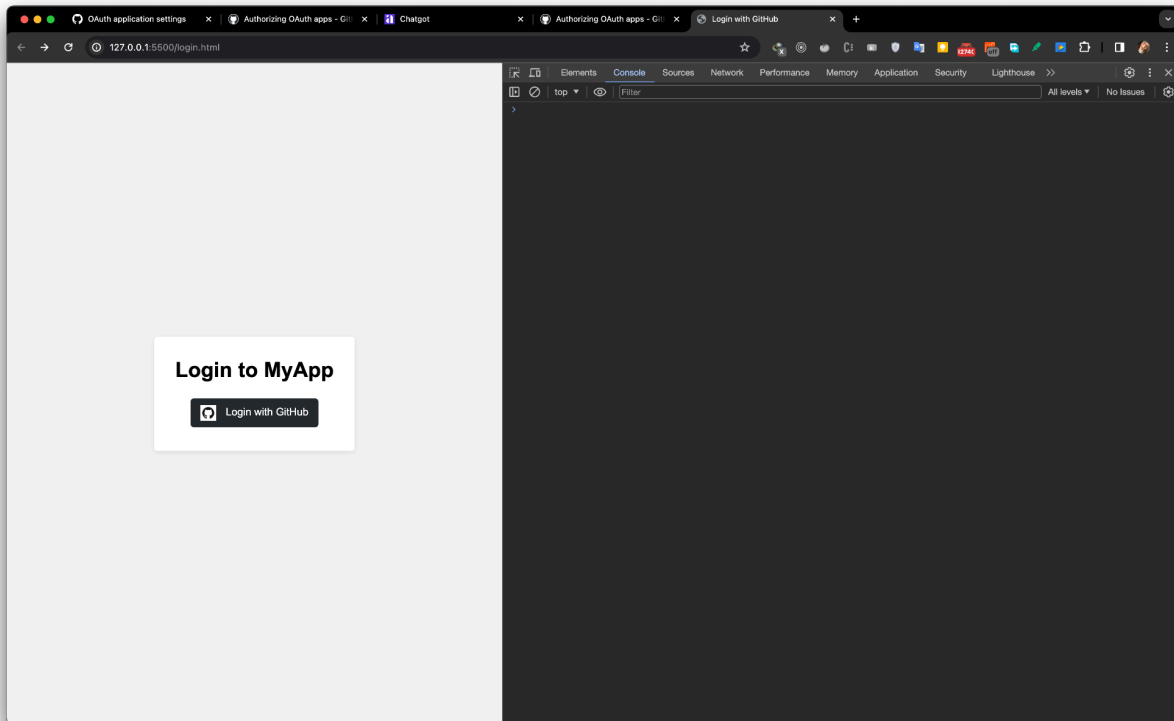

Drag & drop

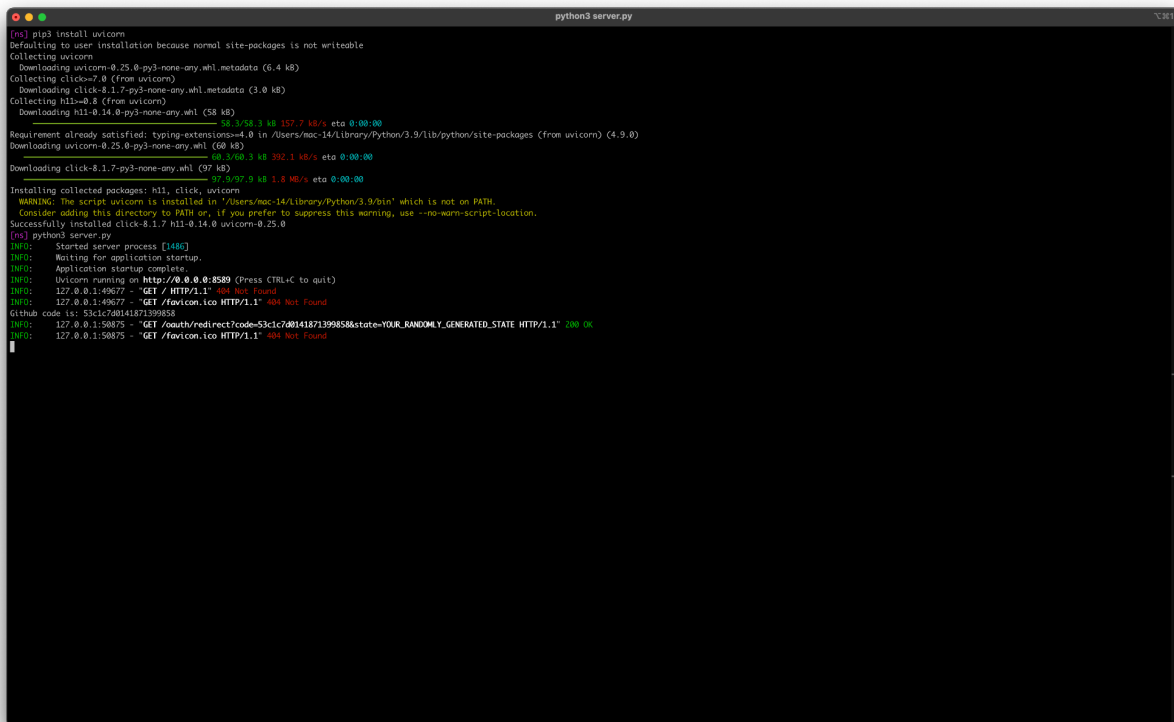
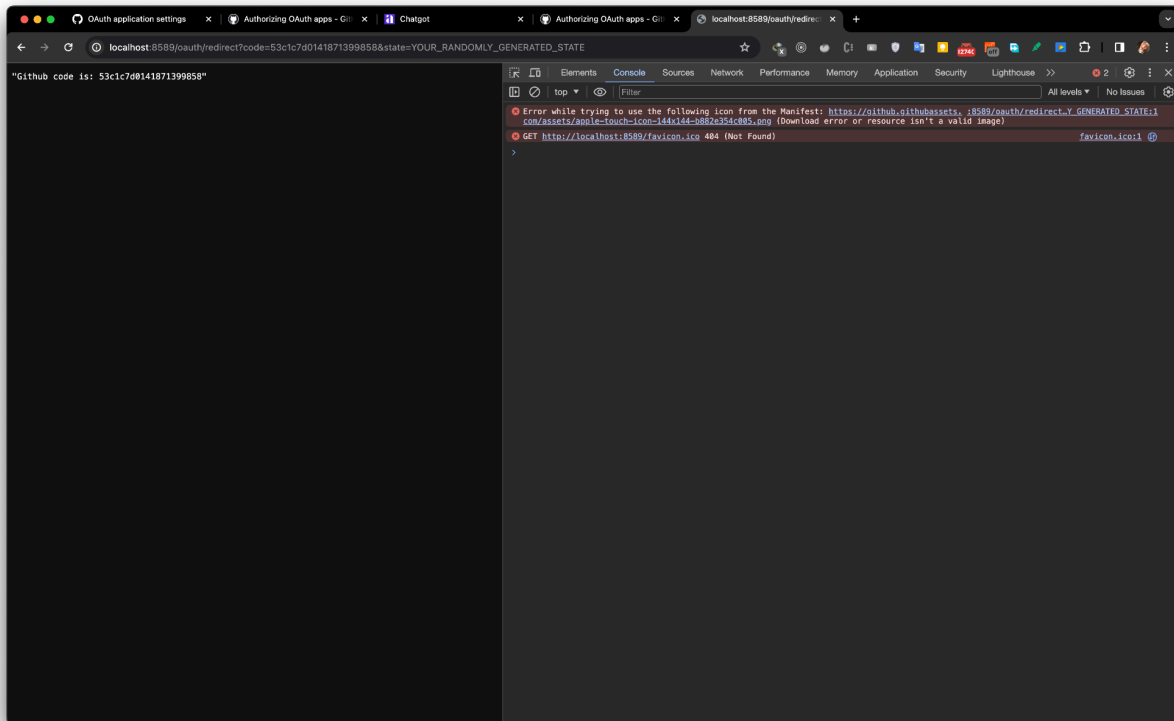
[Upload new logo](#)
You can also drag and drop a picture from your computer.

Application name *
NS-app
Something users will recognize and trust.

Homepage URL *
http://localhost:8589

-۳





```
~ (-zsh)
Last login: Wed Jan 10 09:18:18 on ttys000
[~] curl -X POST \
-H "Accept: application/json" \
-d "client_id=b96bfbbba06a62ef2b3&client_secret=5d03e9e44f8e70c40d90bb8dc7658bc698410511&code=53c1c7d0141871399858&redirect_uri=http://localhost:8589/oauth/redirect" \
https://github.com/login/oauth/access_token
```

```
~ (-zsh)
Last login: Wed Jan 10 09:18:18 on ttys000
[~] curl -X POST \
-H "Accept: application/json" \
-d "client_id=b96bfbbba06a62ef2b3&client_secret=5d03e9e44f8e70c40d90bb8dc7658bc698410511&code=53c1c7d0141871399858&redirect_uri=http://localhost:8589/oauth/redirect" \
https://github.com/login/oauth/access_token
{"access_token":"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","token_type":"bearer","scope":"read:user"}%
[~]
```

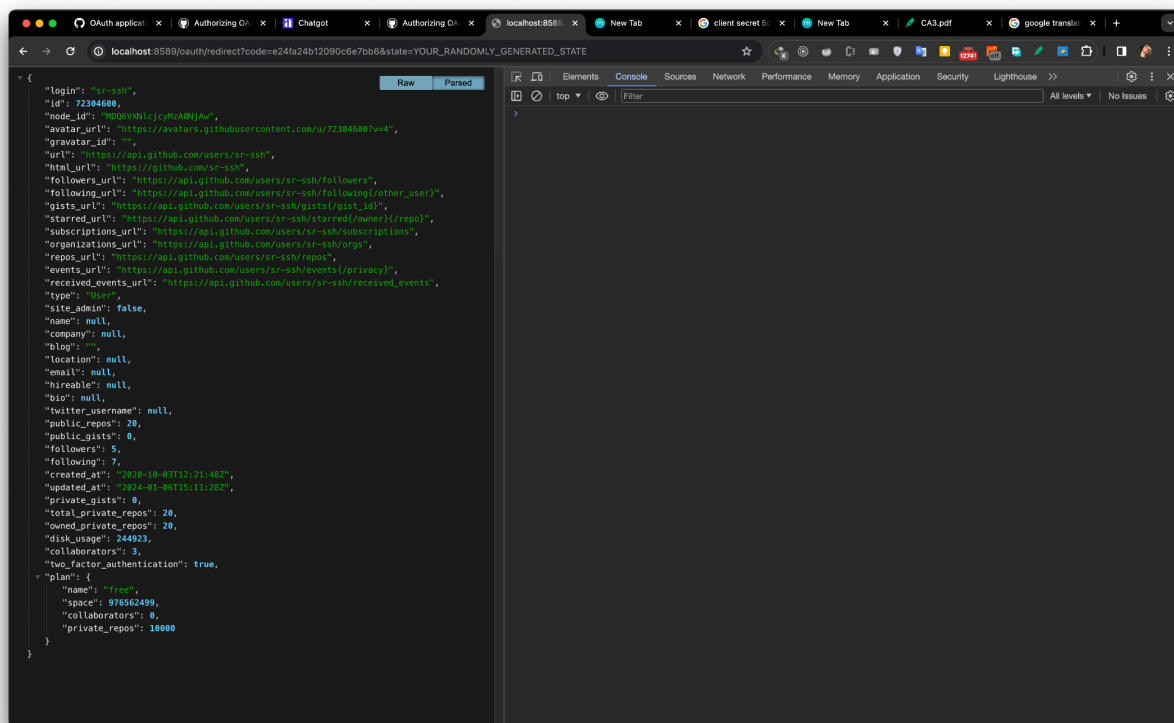
-6

```
Last login: Wed Jan 10 09:18:18 on ttys000
[~] curl -X POST \
-H "Accept: application/json" \
-d "client_id=b96bfbbba06a62ef2b3&client_secret=5d03e9e44f8e70c40d90bb8dc7658bc698410511&code=53c1c7d0141871399858&redirect_uri=http://localhost:8589/oauth/redirect" \
https://github.com/login/oauth/access_token
{"access_token":"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","token_type":"bearer","scope":"read:user"}%
[~] curl -H "Authorization: bearer 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" \
https://api.github.com/user
```

```

~ (-zsh)
Last login: Wed Jan 10 09:18:18 on ttys000
[~] curl -X POST \
-H "Accept: application/json" \
-d "client_id=b96bbfbbbba06a62ef2b3&client_secret=5d03e9e44f8e70c40d90bb8dc7658bc698410511&code=53c1c7d0141871399858&redirect_uri=http://localhost:8589/oauth/redirect" \
https://github.com/login/oauth/access_token
{"access_token":"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","token_type":"bearer","scope":"read:user"}
[~] curl -H "Authorization: bearer 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" \
https://api.github.com/user
{
  "login": "sr-ssh",
  "id": 72304600,
  "node_id": "MDQ6VXNlcjcyMzA0NjAw",
  "avatar_url": "https://avatars.githubusercontent.com/u/72304600?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/sr-ssh",
  "html_url": "https://github.com/sr-ssh",
  "followers_url": "https://api.github.com/users/sr-ssh/followers",
  "following_url": "https://api.github.com/users/sr-ssh/following{/other_user}",
  "gists_url": "https://api.github.com/users/sr-ssh/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/sr-ssh/starred{/owner}/{/repo}",
  "subscriptions_url": "https://api.github.com/users/sr-ssh/subscriptions",
  "organizations_url": "https://api.github.com/users/sr-ssh/orgs",
  "repos_url": "https://api.github.com/users/sr-ssh/repos",
  "events_url": "https://api.github.com/users/sr-ssh/events{/privacy}",
  "received_events_url": "https://api.github.com/users/sr-ssh/received_events",
  "type": "User",
  "site_admin": false,
  "name": null,
  "company": null,
  "blog": "",
  "location": null,
  "email": null,
  "hireable": null,
  "bio": null,
  "twitter_username": null,
  "public_repos": 20,
  "public_gists": 0,
  "followers": 5,
  "following": 7,
  "created_at": "2020-10-03T12:21:48Z",
  "updated_at": "2024-01-06T15:11:28Z",
  "private_gists": 0,
  "total_private_repos": 20,
  "owned_private_repos": 20,
  "disk_usage": 244923,
  "collaborators": 3,
  "two_factor_authentication": true,
  "plan": {
    "name": "free",
    "space": 976562499,
    "collaborators": 0,
    "private_repos": 10000
  }
}
[~]

```



(۲) سوالات

1. مزایای استفاده از روش Authorization Code Grant Type:

- این روش امنیت بیشتری نسبت به دیگر روش‌ها دارد زیرا توکن دسترسی مستقیماً به کاربر نهایی داده نمی‌شود.
- توکن Refresh Token ارائه می‌دهد که با آن می‌توان بدون نیاز به درخواست اطلاعات کاربر مجدداً توکن دسترسی را بدست آورد.
- بهترین گزینه برای برنامه‌هایی است که در دستگاه کاربر نصب می‌شوند زیرا از قرار دادن مشخصات حساس در دستگاه جلوگیری می‌کند.

2. ضعف‌های امنیتی مرتبط با استفاده از روش Client Credential Grant Type در نرم‌افزار تلفن همراه:

- از آنجایی که این روش نیاز به مشخصات کاربر (username و password) ندارد، در صورت به سرقت رفتن توکن، امکان تعیین هویت و دسترسی به منابع سرور برای مهاجم وجود دارد.

- این همچنین به این معنی است که اگر کلید مشتری (Client Key) و رمز مشتری (Client Secret) لو بروند، کنترل کامل برنامه در خطر است.

3. در مورد Access Token:

- Access Token می‌تواند در انواع شناخته شده‌ای مانند JWT (JSON Web Token) صادر شود که هم قابلیت Decode دارد و هم امکان تایید امضای دیجیتالی را فراهم می‌کند.
- JWTها به راحتی قابل Decode هستند ولی برای اینکه اطلاعات موجود واقعی و تغییر نیافته باشد، عملیات Verify نیاز است که مستلزم داشتن کلید عمومی مربوط به کلید خصوصی است که برای امضای توکن استفاده می‌شود.

4. یک ضعف امنیتی

یکی از ضعف‌های امنیتی رایجی که می‌تواند در برنامه‌هایی که با استفاده از OAuth از طریق GitHub برای احراز هویت کاربران استفاده می‌کنند، پیش آید، مربوط به ردیابی و مدیریت امن می‌باشد. در اینجا نکاتی برای افزایش امنیت برنامه آورده شده است:

1. فاش شدن Client Secret: اگر Client Secret شما فاش شود، مهاجمان می‌توانند به اطلاعات کاربری دسترسی پیدا کنند.

راه‌حل: اطمینان حاصل کنید که Client Secret هیچ‌گاه در مخزن کد (مثلاً GitHub) فاش نشده و در متغیرهای محیطی امن (Environment Variables) ذخیره شده‌اند.

2. Redirect URI های اشتباه: تعیین نامناسب Redirect URI می‌تواند به مهاجمان اجازه دهد تا Authorization Code را به دامنه‌های خود به جای برنامه شما هدایت کرده و توکن‌های OAuth را به دست آورند.

راه‌حل: تنها Redirect URI های مورد اعتماد و دقیق را در تنظیمات OAuth اعلام کنید.

3. سرقت Authorization Code: در حین تبادل Authorization Code با Access Token، اگر ارتباط امن نباشد، ممکن است Code به سرقت رود.

راه‌حل: استفاده همیشگی از HTTPS برای تمام ارتباطات شبکه که اطلاعات حساس را منتقل می‌کنند تا از انتقال امن اطلاعات اطمینان حاصل شود.

4. Cross-Site Request Forgery (CSRF): حملات CSRF می‌تواند در نظر گرفته شود که جایی که توکن‌ها و دسترسی‌ها بدون اطلاع کاربر و به صورت خودکار می‌توانند به سوءاستفاده منجر شوند.

راه‌حل: افزودن توکن CSRF در فرم‌های احراز هویت و سایر درخواست‌های مهم تا از درخواست‌های مشکوک جلوگیری شود.

5. پیکربندی ضعیف امنیتی: خطاهای پیکربندی می‌توانند در هر مرحله از استقرار برنامه رخ دهند و اغلب منجر به آسیب‌پذیری‌هایی می‌شوند که می‌توانند از طریق حمله‌های خودکار کشف و بهره‌برداری شوند.

راه‌حل: به کارگیری یک چک لیست امنیتی برای اطمینان از پیکربندی صحیح قوانین امنیتی، مانند سیاست‌های CORS، دسترسی‌های دیتابیس، و محدودیت دسترسی‌های API.