



App Platform

Step by Step

Building a simple but complete application

<https://www.digitalocean.com>

Agenda

- Installing the cli
 - <https://docs.digitalocean.com/reference/doctl/how-to/install/>
- Develop the Front-End
 - Provision a static site
- Implement the Api
 - Provision a node backed
- Setup storage
 - Provision a database

Get your token

- Install the `doctl` cli
- Click on **API** on Menu
- Click on **Generate New Token**
- Click on **Copy**
- Type `doctl auth init` and paste

The screenshot shows the DigitalOcean 'Applications & API' interface. On the left, a dark sidebar contains a menu with 'API' highlighted. A red arrow labeled '1' points to this menu item. The main panel shows 'Personal access tokens' with a 'Generate New Token' button. A red arrow labeled '2' points to this button. A modal window titled 'New personal access token' is open, showing a 'Token name' field with 'macbookari' and 'Select scopes' with 'Read (default)' and 'Write (optional)' checked. A red arrow labeled '3' points to the 'Generate Token' button in the modal. Below the modal, a table lists the generated token. A red arrow labeled '4' points to the 'Copy' link next to the token value.

Name	Scope
macbookari	READ WRITE
377c7c0c161210369879e2327e5fd50b5681cad864cde6b2df3082a951e0a2a1 Copy	

MANAGE

Apps **NEW**

Droplets

Kubernetes

Volumes

Databases

Spaces

Container Registry

Images

Networking

Monitoring

Settings

Billing

API

Applications & API

Tokens/Keys OAuth Applications Access

Personal access tokens

Tokens you have generated to access the [DigitalOcean API](#).

Name	Scope	Created
msciabarra		

Generate New Token

New personal access token

Token name

Enter token name
macbookari

Select scopes

☒ Read (default)

☒ Write (optional)

Read our [personal access token documentation](#) for more information on scopes.

Generate Token

Name

Scope

macbookari

READ

WRITE

377c7c0c161210369879e2327e5fd50b5681cad864cde6b2df3082a951e0a2a1 Co

Creating a frontend

- An example using Svelte

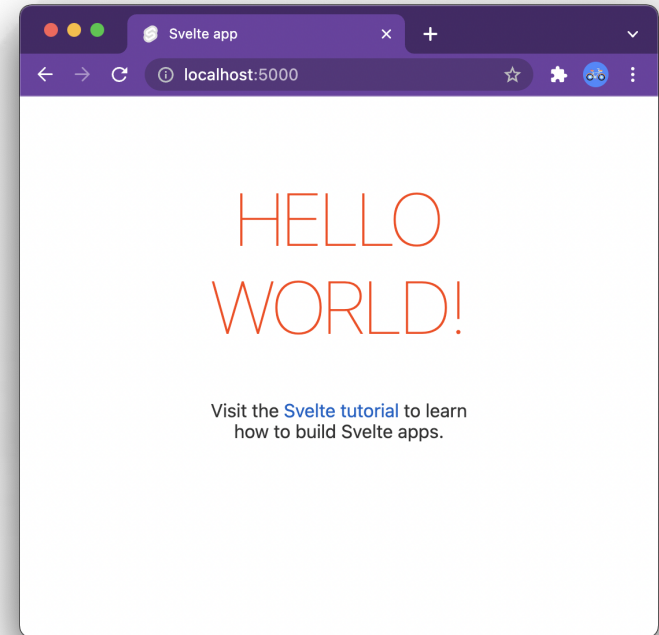
```
npx degit  
sveltejs/template frontend
```

- Install it

```
cd frontend && npm install
```

- Run in development mode

```
npm run dev
```



Deployment: `.do/app.yaml`

```
name: tutorial-app-platform
static_sites:
- name: frontend
  github:
    repo: sciabarrado/tutorial-app-platform
    branch: main
    deploy_on_push: true
  build_command: npm run build
  source_dir: frontend
  routes:
  - path: /
```

- Deploy with: `doctl app create --spec .do/app.yaml`

Exercise: frontend

```
# creating the frontend
npx degit sveltejs/template frontend
cd frontend && npm install
npm run dev
# deploying the frontend
mkdir .do
cp conf/app-1.yaml .do/app.yaml
doctl app create --spec .do/app.yaml
# monitoring
ID=$(doctl app list | awk '/tutorial-app-platform/ { print $1}')
echo $ID
doctl app logs $ID
```

Updating

```
# update  
ID=$(doctl app list | awk '/tutorial-app-platform/ { print $1}')
```

```
doctl app update $ID --spec .do/app.yaml
```