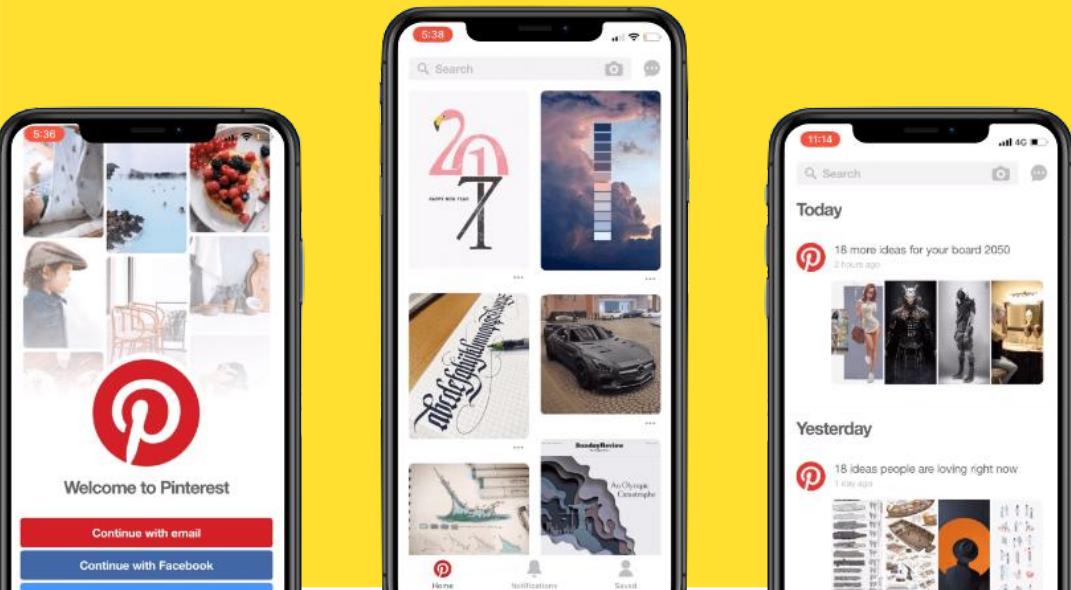




Pinterest Clone

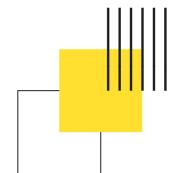
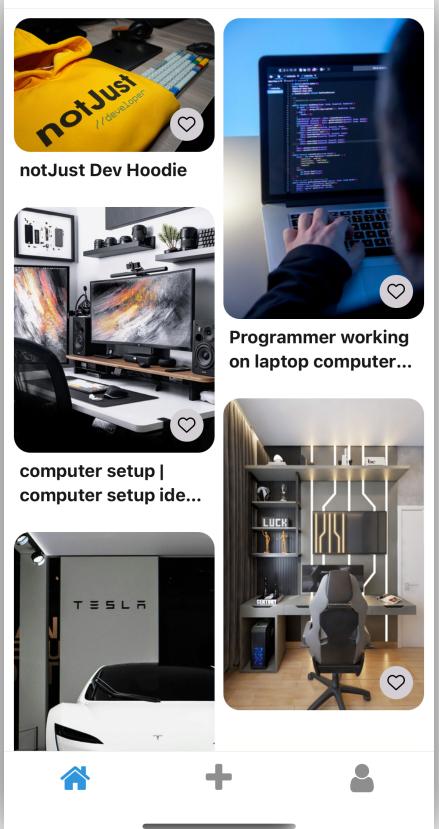


Beginner-friendly
React Native Tutorial



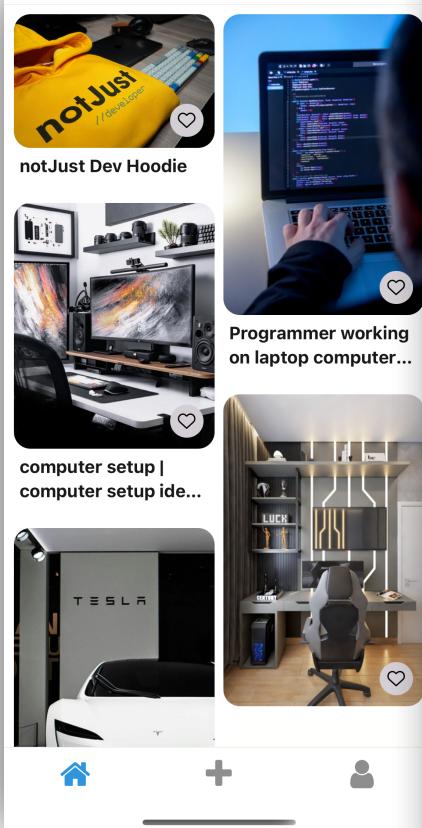
What will you build?

Home Feed

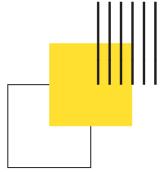


What will you build?

Home Feed

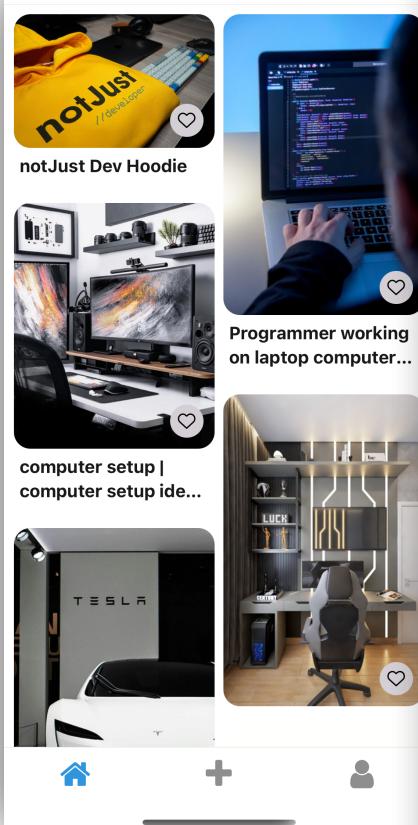


Pin Page



What will you build?

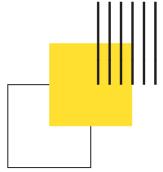
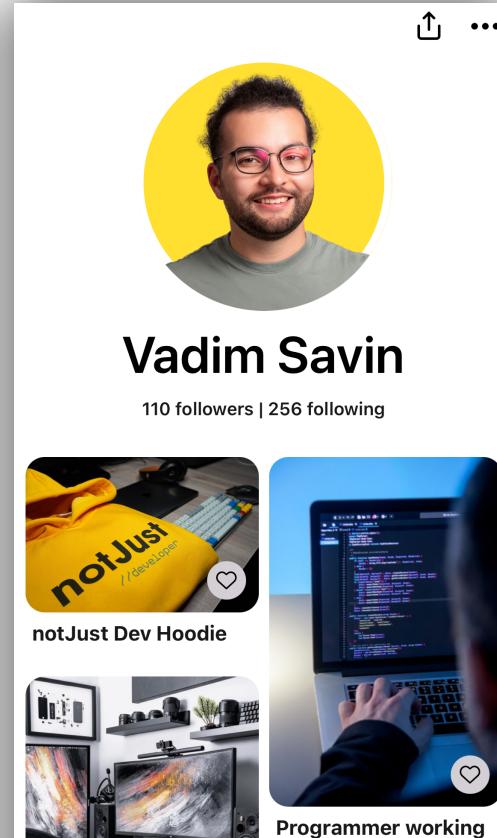
Home Feed



Pin Page

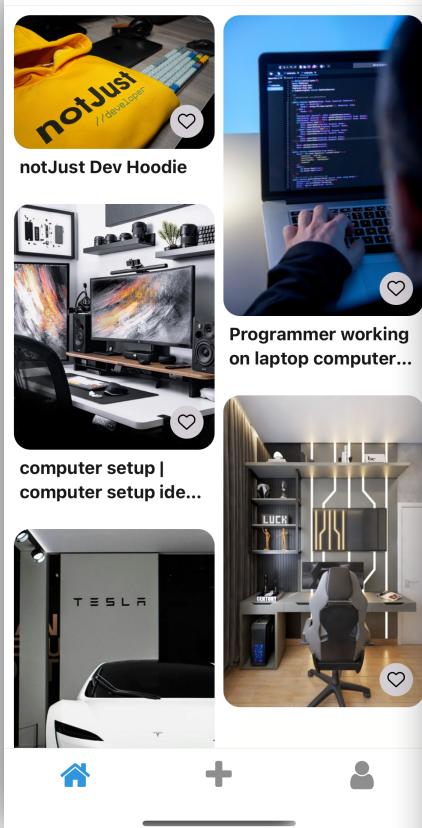


Profile Page



What will you build?

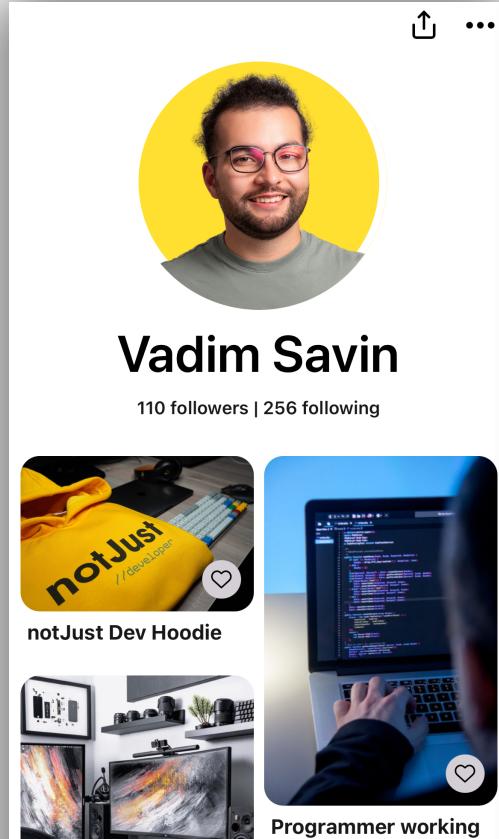
Home Feed



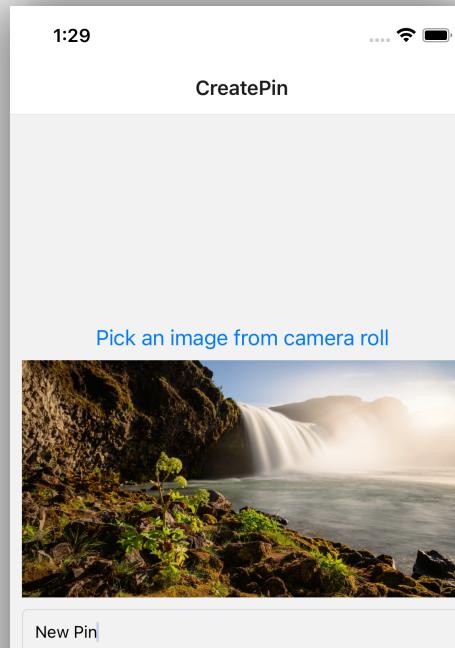
Pin Page



Profile Page



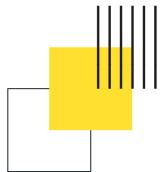
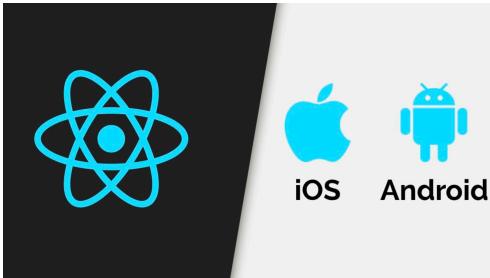
Upload a pin



Submit

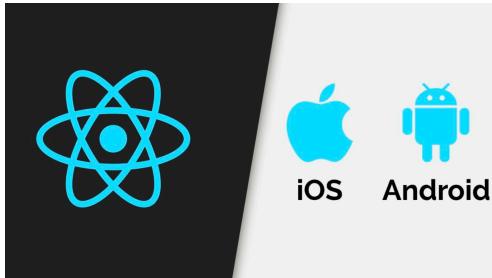


Today

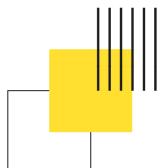




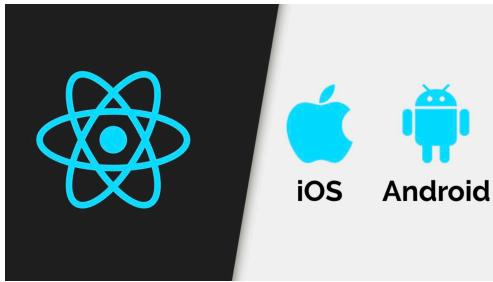
Today



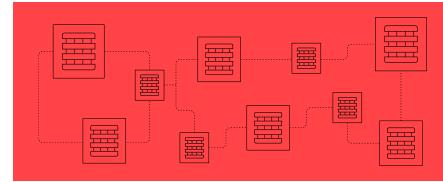
expo



Today

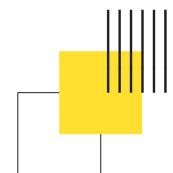


Next Friday
8th April

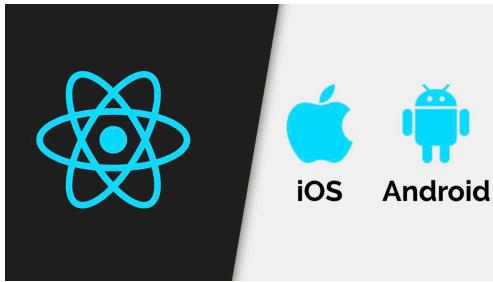


Serverless Backend

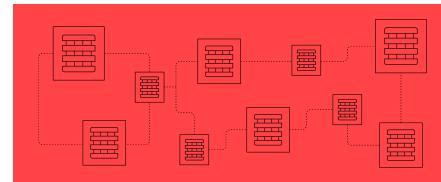
expo



Today



Next Friday
8th April

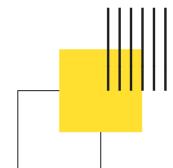


Serverless Backend

 Expo

 nHOST

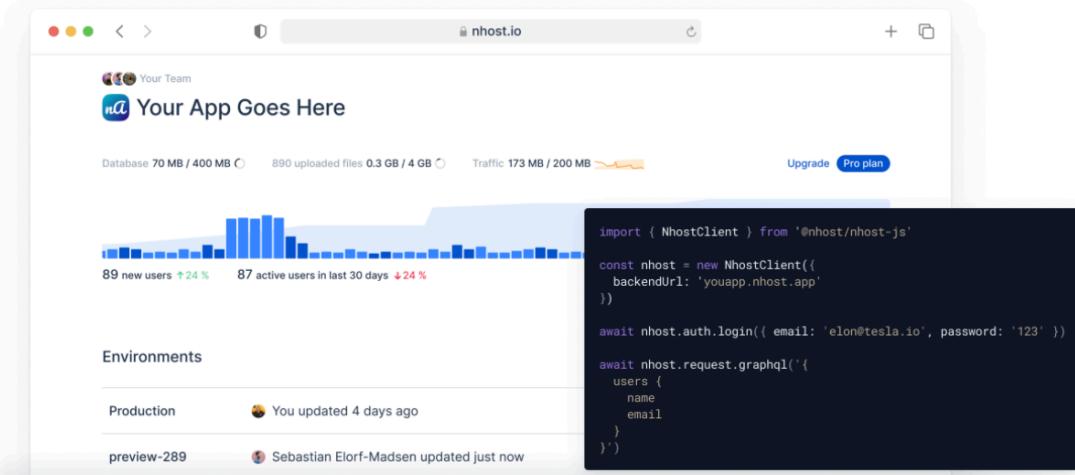
- Auth
- GraphQL API
- Storage



Serverless backend for web and mobile apps

Make backend easy. Never manage infrastructure.

✓ Real-time database ✓ GraphQL ✓ Authentication ✓ File storage ✓ Emails

[Sign up for Free](#)[Read more](#)

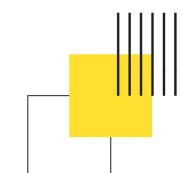
The screenshot shows the nhost.io dashboard. At the top, it displays 'Your Team' and 'Your App Goes Here'. Below this, there are three resource usage metrics: Database (70 MB / 400 MB), Files (890 uploaded files / 0.3 GB / 4 GB), and Traffic (173 MB / 200 MB). There are 'Upgrade' and 'Pro plan' buttons. A bar chart shows '89 new users ↑24%' and '87 active users in last 30 days ↓24%'. The 'Environments' section lists 'Production' (updated 4 days ago) and 'preview-289' (updated just now). On the right, a code editor window shows a snippet of NhostClient.js:

```
import { NhostClient } from '@nhost/nhost-js'

const nhost = new NhostClient({
  backendUrl: 'youapp.nhost.app'
})

await nhost.auth.login({ email: 'elon@tesla.io', password: '123' })

await nhost.request.graphql(`{
  users {
    name
    email
  }
}`)
```



Serverless backend for web and mobile apps

Make backend easy. Never manage infrastructure.

✓ Real-time database ✓ GraphQL ✓ Authentication ✓ File storage ✓ Emails

[Sign up for Free](#)[Read more](#)

https://bit.ly/3iV2HIn

Your App Goes Here

Database 70 MB / 400 MB

890 uploaded files 0.3 GB / 4 GB

Traffic 173 MB / 200 MB

[Upgrade](#) [Pro plan](#)

89 new users +24%

87 active users in last 30 days -24%

Environments

Production

You updated 4 days ago

preview-289

Sebastian Elorff-Madsen updated just now

```
import ( NhostClient ) from '@nhost/nhost-js'

const nhost = new NhostClient({
  backendUrl: 'youapp.nhost.app'
})

await nhost.auth.login({ email: 'elon@tesla.io', password: '123' })

await nhost.request.graphql(`(
  users {
    name
    email
  }
)`)
```

Prerequisites

1. Asset Bundle (dummy data, PDF presentation):

||| <https://assets.notjust.dev/pinterest>

2. Expo environment setup

||| <http://bit.ly/expo-vadim>

notJust

//development



Let's get started



Let's get started

1. Initialise the expo project:

```
$ expo init Pinterest -t tabs
```

2. Open project

```
$ cd Pinterest
```

```
$ code .
```

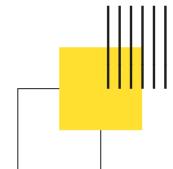
3. Run the development server

```
$ npm start
```

4. Run the app on device or emulator

```
$ npm run android
```

```
$ npm run iOS
```



Tab One



notJust

//development

Pin component



notJust Dev Hoodie

Programmer working
on laptop computer...computer setup |
computer setup ide...



Pin component



- Image
- Title
- Icon Button
- Extract to a separate component
- Props
- State

Image: <https://notjustdev-dummy.s3.us-east-2.amazonaws.com/pinterest/0.jpeg>



notJust Dev Hoodie

1:26

Tab One

notJust Dev Hoodie

computer setup | computer setup ide...

Programmer working on laptop computer...

computer setup | computer setup ide...

TESLA



Tab One



notJust

//development

Masonry layout



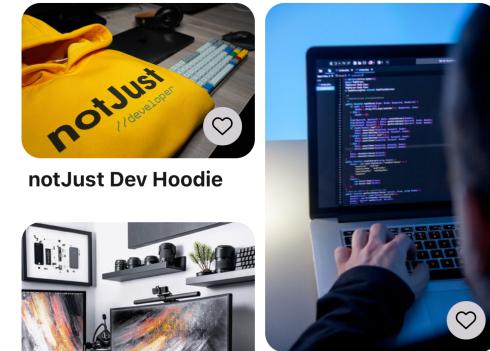
- 2 Column scrollable list



notJust Dev Hoodie



computer setup | computer setup ide...



Programmer working on laptop computer...



Pin Page



- Image



- Title



- Back button



Harley Davidson Sportster Iron
883 Custom ~ Rider & Helmet -
ModifiedX



Profile Screen

- ||| - Image
- || - Name
- || - Followers
- || - Pins
- || - Header icons
- |||



Vadim Savin

110 followers | 256 following



notJust Dev Hoodie



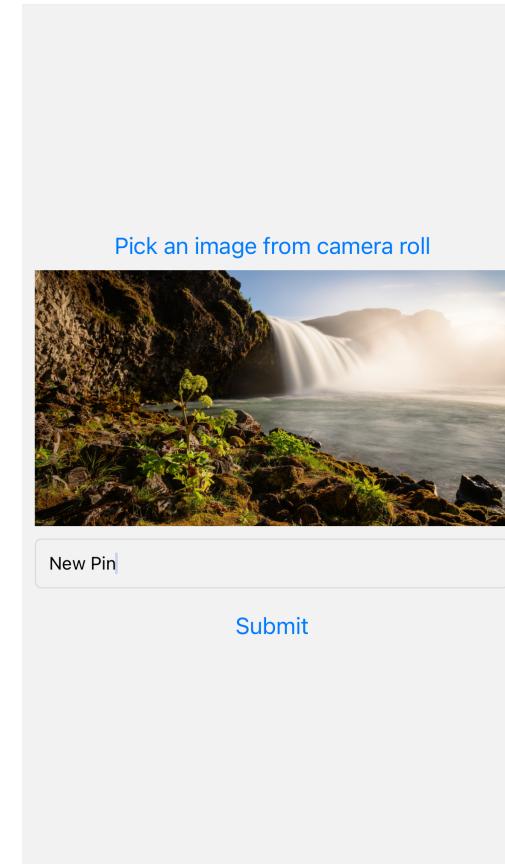
Programmer working on laptop computer...





Create a pin Screen

- ||| - Image picker
- || - TextInput
- || - Submit button
- ||
- ||
- ||



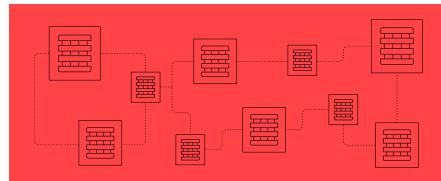


Today



 Expo

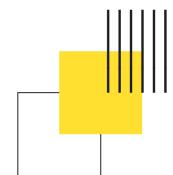
Next Friday
8th April



Serverless Backend



- Auth
- GraphQL API
- Storage



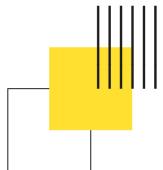
notJust

//development



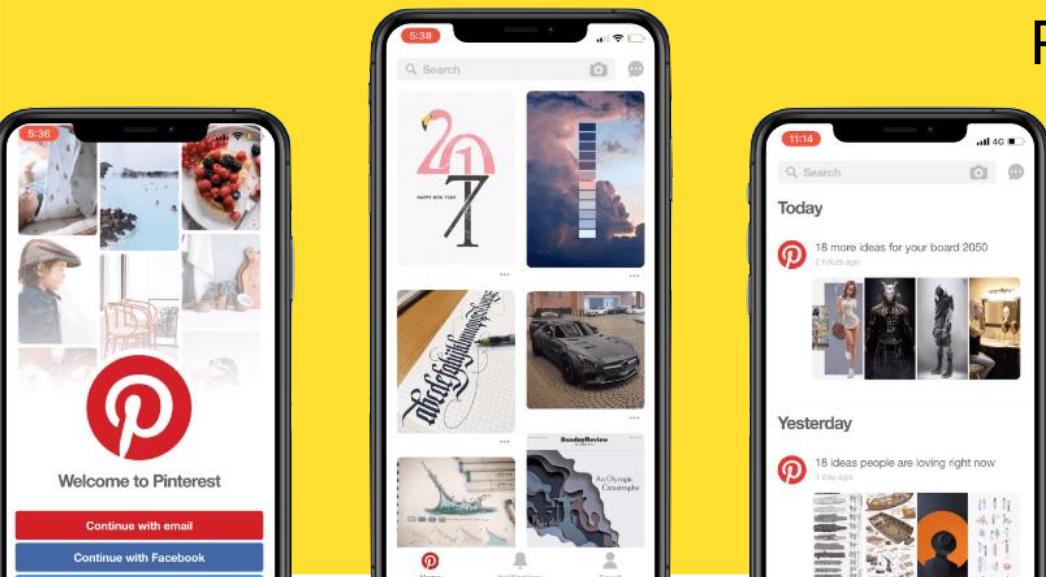
Q&A







Pinterest Clone



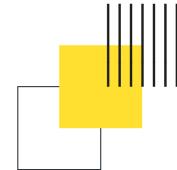
Part 2: Backend with Nhost
Beginner-friendly tutorial





Authentication

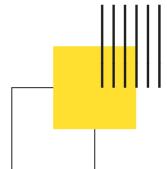
User accounts and user management





Database with GraphQL

Define the data model and query it with GraphQL





File Storage

Upload and manage files

Serverless backend for web and mobile apps

Make backend easy. Never manage infrastructure.

✓ Real-time database ✓ GraphQL ✓ Authentication ✓ File storage ✓ Emails

[Sign up for Free](#)[Read more](#)

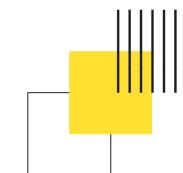
The screenshot shows the nhost.io dashboard. At the top, it displays 'Your Team' and 'Your App Goes Here'. Below this, there are three resource usage metrics: Database (70 MB / 400 MB), Files (890 uploaded files / 0.3 GB / 4 GB), and Traffic (173 MB / 200 MB). There are 'Upgrade' and 'Pro plan' buttons. A bar chart shows '89 new users ↑24%' and '87 active users in last 30 days ↓24%'. The 'Environments' section lists 'Production' (updated 4 days ago) and 'preview-289' (updated just now). On the right, a code editor window shows a snippet of Nhost.js code:

```
import { NhostClient } from '@nhost/nhost-js'

const nhost = new NhostClient({
  backendUrl: 'youapp.nhost.app'
})

await nhost.auth.login({ email: 'elon@tesla.io', password: '123' })

await nhost.request.graphql(`{
  users {
    name
    email
  }
}`)
```



Serverless backend for web and mobile apps

Make backend easy. Never manage infrastructure.

✓ Real-time database ✓ GraphQL ✓ Authentication ✓ File storage ✓ Emails

[Sign up for Free](#)[Read more](#)

https://bit.ly/3iV2HIn

Your App Goes Here

Database 70 MB / 400 MB

890 uploaded files 0.3 GB / 4 GB

Traffic 173 MB / 200 MB

[Upgrade](#) [Pro plan](#)

89 new users ↑24%

87 active users in last 30 days ↓24%

Environments

Production

You updated 4 days ago

preview-289

Sebastian Elorff-Madsen updated just now

```
import ( NhostClient ) from '@nhost/nhost-js'

const nhost = new NhostClient({
  backendUrl: 'youapp.nhost.app'
})

await nhost.auth.login({ email: 'elon@tesla.io', password: '123' })

await nhost.request.graphql(`(
  users {
    name
    email
  }
)`)
```

Prerequisites

1. Asset Bundle (dummy data, PDF presentation):

||| <https://assets.notjust.dev/pinterest>

2. Nhost account (Free, no credit card required):

||| <https://bit.ly/3iV2Hln>



Starting point

1. Part 1

<https://youtu.be/Fc062FgrAJ0>



The source code from part 1 is in the Asset bundle:

<https://assets.notjust.dev/pinterest>



notJust

//development



Let's get started



Who am I

- Full stack developer for >7yrs
- Founded multiple startups
- ex Amazon SDE
- Passionate about coding and building impactful startups.

I AM **notJust**
//developer





//development

Create our first app on Nhost

- ||| - Create an account
- ||| - Create the app
- ||| - Create the first user
- |||
- |||
- |||
- |||
- |||

The screenshot shows the Nhost.io dashboard. At the top, it says "Serverless backend for web and mobile apps" and "Make backend easy. Never manage infrastructure." Below this are links for "Real-time database", "GraphQL", "Authentication", "File storage", and "Emails". There are two buttons: "Sign up for Free" and "Read more". The main area displays real-time metrics: Database 70 MB / 400 MB, 890 uploaded files 0.3 GB / 4 GB, and Traffic 173 MB / 200 MB. It also shows 89 new users (+24%) and 87 active users in last 30 days (-24%). A section titled "Environments" lists "Production" and "preview-289", with a note that "Production" was updated 4 days ago and "preview-289" was updated just now. On the right, there is a code editor window showing a snippet of GraphQL code:

```
import { NhostClient } from '@nhost/nhost-js'
const nhost = new NhostClient({
  backendUrl: 'youapp.nhost.app'
})

await nhost.auth.login({ email: 'elon@tesla.io', password: '' })

await nhost.request.graphql(`{
  users {
    name
    email
  }
}`)
```



Create our first Database table

- Open Hasura
 - Add a new table in the public folder
 - Fields: title, image, user_id
 - Add a foreign key
 - Add some data

The screenshot shows the Hasura Data Manager interface. The top navigation bar includes the Hasura logo, version v2.2.0, and Cloud, Help buttons. The left sidebar, titled "Data Manager", lists databases: default (selected), auth, public (expanded), pins (selected), and storage. A "Manage" button is located at the top of the sidebar. The main area displays the "pins" database with tabs for Browse Rows, Insert Row, Modify, Relationships, and Permissions. A "Filter" section with dropdowns for column, operator (eq), and value is present. Below it are "Run query" and "Export data" buttons. The main content is a table with columns id, created_at, updated_at, title, description, and file_id. Each row contains edit, delete, and copy icons. The table lists 10 rows of pin data.

	id	created_at	updated_at	title	description	file_id
<input type="checkbox"/>	f11a66b8-5309-47ed-9244-b5bbe53c2f63	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 1	Pin 1 description	pin1
<input type="checkbox"/>	b0ce0fda-7c0b-4708-b149-5cdf466c9b6b	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 2	Pin 2 description	pin2
<input type="checkbox"/>	a8c271fb-7323-4b3e-97d7-292ec3c5bd5e	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 3	Pin 3 description	pin3
<input type="checkbox"/>	51dd0e17-6961-4ef0-af9c-ad271144e301	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 4	Pin 4 description	pin4
<input type="checkbox"/>	6096cae7-b11b-4530-923d-21bb088de3ad	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 5	Pin 5 description	pin5
<input type="checkbox"/>	4b50b37d-ce4b-4c38-a2ba-0b5534212a73	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 6	Pin 6 description	pin6
<input type="checkbox"/>	82fb9e88-ef0d-4a33-ad46-578f304d4ef1	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 7	Pin 7 description	pin7
<input type="checkbox"/>	876b4b4c-6585-410e-99ae-060eb4a8da9f	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 8	Pin 8 description	pin8
<input type="checkbox"/>	a082c5f4-83de-4438-93f2-8bf882577735	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 9	Pin 9 description	pin9
<input type="checkbox"/>	6a290b8b-4e88-406c-82f5-513359b6c064	2022-07-12T10:00:00Z	2022-07-12T10:00:00Z	Pin 10	Pin 10 description	pin10

prev 50 rows



Query our database

```
query MyQuery {  
  pins {  
    id  
    created_at  
    image  
    title  
    user_id  
  }  
}
```

GraphiQL Prettify History Explorer Code Exporter REST Derive action Analyze

```
1  query MyQuery {  
2  pins {  
3    id  
4    created_at  
5    image  
6    title  
7    user_id  
8  }  
9 }  
10
```

QUERY VARIABLES

```
1
```

```
{  
  "data": {  
    "pins": [  
      {  
        "id": "f11a66b8-5309-47ed-9244-b5bbe53c2f63",  
        "created_at": "2022-04-07T18:21:45.24869+00:00",  
        "image": "https://notjustdev-dummy.s3.us-east-  
2.amazonaws.com/pinterest/0.jpeg",  
        "title": "notJust Dev Hoodie",  
        "user_id": "a6e3b288-01e4-43a3-aa81-a845a039855c"  
      },  
      {  
        "id": "b0ce0fda-7c0b-4708-b149-5cdf466c9b6b",  
        "created_at": "2022-04-07T18:22:02.846128+00:00",  
      }  
    ]  
  }  
}  
1
```

RESPONSE TIME 403 ms RESPONSE SIZE 5730 bytes

Install Nhost in our RN project

||| - Install Nhost client

```
npm install @nhost/react
```

||| - Install a storage library

```
expo install expo-secure-store
```

Initialise Nhost

```
import * as SecureStore from "expo-secure-store";
import { NhostClient } from "@nhost/react";
window = undefined;

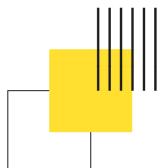
const nhost = new NhostClient({
  backendUrl: "https://<your app id>.nhost.run",
  clientStorageType: "expo-secure-storage",
  clientStorage: SecureStore,
});

<NhostReactProvider nhost={nhost}>
  ...
</NhostReactProvider>
```

Query data

```
const fetchPins = async () => {
  const { data, error } = await nhost.graphql.request(`query {
    pins {
      id
      image
      created_at
      title
      user_id
    }
  }
`);
};

useEffect(() => { fetchPins(); }, []);
```



Permissions

- ||| In the previous section, you could fetch the todos because the **admin** role is enabled by default when using Hasura Console. When building your applications, you want to define permissions using **roles** that your users can assume when making requests.
- | | Hasura supports role-based access control. You create rules for each role, table, and operation (select, insert, update and delete) that can check dynamic session variables, like user ID.

Relationships

- ||| Create the relationship between Pin <>> User.
- || This will allow us to query the info about the user who posted the pin
- |
- ||
- |||

Authentication

- ||| 1. Import the Auth screens
from the asset bundle
- ||
- ||
- ||
- |||

Sign up

```
const { error } = await nhost.auth.signUp({
  email,
  password,
  options: {
    displayName: name,
  },
});
if (error) {
  Alert.alert("Error", error.message);
} else {
  navigation.navigate("Sign in");
}
```

Sign in

```
const response = await nhost.auth.signIn({  
    email,  
    password,  
});  
  
if (response.error) {  
    Alert.alert("error", response.error.message);  
}  
  
|||
```

Sign out

```
||| nhost.auth.signOut();
```



Protected routes

||| Protect routes from un-authenticated users

```
const { isLoading, isAuthenticated } = useAuthenticationStatus();
```

What else is possible?

- ||| - Account management:
 - || - reset/forgot password
 - || - change email,
- || - OAuth Providers (Google, Github, FB, LinkedIn, Discord, and more)
- || - Passwordless Auth (with magic links and with SMS OTP)
- || <https://docs.nhost.io/reference/sdk/authentication>
- ||

notJust

//development



Storage



Permissions

- To upload a file, a user must have the right permission to the `storage.files` table.
-
-
-
-
-

Upload a file

```
const uri = Platform.OS === "ios" ? image.replace("file://", "") : image;  
const uploadResponse = await nhost.storage.upload({  
  file: {  
    name: name,  
    type: "image/png",  
    uri,  
  },  
});
```

Displaying an image

```
const { presignedUrl, error } = await nhost.storage.getPresignedUrl({  
    fileld: image,  
});
```

Check Storage docs for more

||| <https://docs.nhost.io/reference/sdk/storage>



notJust

//development



Q&A



