



# PocketBase Quick Start Guide

Learn how to:

- Install PocketBase
- Create users
- Access with JavaScript and Dart

The scripts and html files are in

```
$GITHUB/database/pocketbase/scripts.
```



# 1. Install PocketBase

---

Visit `https://pocketbase.io/docs/` to download the pocketbase binary.

- Check your architecture `$(uname -m)` and OS `$(uname -s)`.
- Download Linux if you use WSL2.
- Unzip and copy the binary in the `$HOME/pocketbase` directory.

## Option2: Run the `install_pockebase.sh` .

```
Download the binary:
```bash
os=$(uname -s | tr '[:upper:]' '[:lower:]')
arch=$(uname -m)
if [[ "$arch" == "x86_64" ]]; then
    arch="amd64"
elif [[ "$arch" == "aarch64" || "$arch" == "arm64" ]]; then
    arch="arm64"
fi

latest=$(curl -s https://api.github.com/repos/pocketbase/pocketbase/releases/latest | grep -E '"tag_name":' | sed -E 's/.*"([^"]+)".*/\1/')
version=$(echo "$latest" | sed 's/^v//')

if [[ -z "$latest" ]]; then
    echo "Error: Could not fetch latest PocketBase version tag."
    exit 1
fi

url="https://github.com/pocketbase/pocketbase/releases/download/${latest}/pocketbase_${version}_${os}_${arch}.zip"
dest="$HOME/pocketbase.zip"

echo "Download from: $url"

# Download with error check
wget -O "$dest" "$url"
if [[ $? -ne 0 ]]; then
    echo "Error: Download failed. Check your internet connection and the URL: $url"
    exit 1
fi

# Verify that download succeeded and is not a 404 HTML file
if ! file "$dest" | grep -qi zip; then
    echo "Error: Downloaded file is not a valid zip file. The file may not exist at the specified URL:"
    echo "$url"
    rm -f "$dest"
    exit 1
fi

# Unzip with error checking
unzip -d "$HOME/pocketbase" "$dest"
if [[ $? -ne 0 ]]; then
    echo "Error: Unzipping failed. The file may not be a valid ZIP archive:"
    echo "$dest"
    rm -f "$dest"
    exit 1
fi

rm "$dest"
```

Check where the pocketbase is stored:

- If you run the script, pocketbase is installed in \$HOME/pocketbase.
- In this example, we assume you installed pocketbase in the \$HOME/pocketbase directory.

```
POCKET_BASE_PATH=$HOME/pocketbase  
POCKET_BASE=$POCKET_BASE_PATH/pocketbase  
POCKETBASE_URL="http://127.0.0.1:8090"
```

Run the server: \$HOME is your home directory.

```
$HOME/pocketbase/pocketbase serve
```

or in the pocketbase directory

```
./pocketbase serve
```

It will start on: `http://127.0.0.1:8090`

```
2025/07/22 15:17:12 Server started at http://127.0.0.1:8090
└─ REST API:  http://127.0.0.1:8090/api/
└─ Dashboard: http://127.0.0.1:8090/_/
```



## 2. Creating Admin & Users (Script)

---

- Read `create_users.sh` in the `scripts` directory.
- Run the script to create the admin & user: this is the core part of the code.

```
POCKET_BASE_PATH=$HOME/pocketbase  
POCKET_BASE=$POCKET_BASE_PATH/pocketbase
```

```
$POCKET_BASE superuser upsert admin@example.com admin123456
```

```
curl -X POST http://127.0.0.1:8090/api/collections/users/records \  
  -H "Content-Type: application/json" \  
  -d '{"email":"user@example.com","password":"password","passwordConfirm":"password"}'
```

# create\_users.sh

---

- Read the output carefully.
  - Check the pocketbase version.
  - Check how to stop pocketbase.
  - Check pocketbase file locations.

=====

✅ Setup Complete!

=====

PocketBase Admin Panel: [http://127.0.0.1:8090/\\_/](http://127.0.0.1:8090/_/)  
Login with: admin@example.com / admin123456

- initialize or stop pocketbase
  - Delete `pb_data` in the `$POCKET_BASE_PATH` to initialize pocketbase.
  - Stop pocketbase using the `pkill -f pocketbase` command or `kill [PID_NUMBER]`
  - Run the script and check the output.



## 2B. Create Admin & Users (CLI or GUI)

---

### Option1: CLI

In the pocketbase directory, run command:

```
pocketbase superuser upsert admin@example.com admin123456
```

Check if you get the success message.

```
Successfully saved superuser "admin@example.com"!
```

Create user: `user@example.com` with a password `password` .

```
curl -X POST http://127.0.0.1:8090/api/collections/users/records \
-H "Content-Type: application/json" \
-d '{"email":"user@example.com","password":"password","passwordConfirm":"password"}'
```

```
{"avatar":"","collectionId":"_pb_users_auth_",
"collectionName":"users","created":"2025-07-24 13:30:29.539Z",
"emailVisibility":false,"id":"5thhma0uh5adv7","name":"","
"updated":"2025-07-24 13:30:29.539Z","verified":false}
```

## Option2: GUI

Open your browser:

```
http://127.0.0.1:8090/_/
```

- Create an **superuser** account
- Add fields: email and password
- Click "Create superuser and login" button

Use `admin@example.com` and `admin123456` for the test.

- Navigate to **Users** collection
- Click "+ New record" button
- Add fields: `email`, `password`
- Click the "Create" button

Use a user: `user@example.com` and a password: `password`  
for the test.



## 3. JavaScript(HTML) and Pocketbase API Example

---

pocketbase\_example.html

1. Install the JS SDK:

```
<script src="https://unpkg.com/pocketbase/dist/pocketbase.umd.js"></script>
```

2. Use the PocketBase SDK to display information

```
const pb = new PocketBase('http://127.0.0.1:8090');
```

3. Open the `pocketbase_user_fetch_example.html` in the browser, and check if you have the results (created/updated/id should be different from yours).

```
[
  {
    "avatar": "",
    "collectionId": "_pb_users_auth_",
    "collectionName": "users",
    "created": "2025-07-22 19:24:15.102Z",
    "email": "user@example.com",
    "emailVisibility": false,
    "id": "hvfwc925ov75lms",
    "updated": "2025-07-22 19:24:15.102Z",
    "username": "user"
  }
]
```

4. Open the `pocketbase_users.html` in the browser, and check if you have the results.

```
<body>
<h1>PocketBase Users</h1>
<ul id="userList"><li>Loading...</li></ul>

<script src="https://unpkg.com/pocketbase/dist/pocketbase.umd.js"></script>
<script>
  const pb = new PocketBase('http://127.0.0.1:8090');

  async function run() {
    try {
      // Authenticate with a valid user
      await pb.collection('users').authWithPassword('user@example.com', 'password');

      // Fetch users
      const users = await pb.collection('users').getFullList();

      const userList = document.getElementById('userList');
      userList.innerHTML = ''; // Clear loading text

      users.forEach(user => {
        const li = document.createElement('li');
        li.textContent = `ID: ${user.id}, Email: ${user.email}`;
        userList.appendChild(li);
      });
    } catch (err) {
      document.getElementById('userList').innerHTML = '<li>Error: ' + err.message + '</li>';
    }
  }
}
```

The ID should be different from yours.

PocketBase Users

ID: knbod29cvlwbxx6, Email: user@example.com



# Run `test_api_pocketbase.sh` (API test using `curl`)

---

- Check if you can run `curl` command before running the script.

```
🧪 Testing PocketBase Connection...
=====
1 Testing PocketBase health...
  ✅ PocketBase server is running

2 Testing admin panel access...
  ✅ Admin panel is accessible
  🌐 Access it at: http://127.0.0.1:8090/_/
  🔑 Login with: admin@example.com / admin123456
```

🔧 4. Make a student collection for tests.

## Option 1: `create_collection.sh` Script

---

- Read `create_collection.txt` about the two commands.
  - The first command retrieves the token.
  - The second command uses the token to create the `students` collection.
- Run `create_collection.sh` to automatically generate the `students` collection.

# Option 2: Manual Creation via Admin Panel

---


## 1. Access PocketBase Admin Panel

```
http://127.0.0.1:8090/_/  
Login: admin@example.com / admin123456
```

## 2. Create a Collection

- Click "New collection"
- Name:
- Type: Base collection
- Add these fields:
  - (Text, Required (Nonempty), Min: 1, Max: 100)
  - (Number, Required (Nonempty), Min: 0, Max: 150)
  - (Text, Required (Nonempty), Min: 1

### 3. Update the collection (if you need to update the collection later)

- Click the  (Edit Collection).
  - Update the fields from the Fields tab.
  - Add/Update the API rules from the API Rules tab.



## 5. Run tests in Dart

---

1. In the `scripts` directory: run the `test_dart_pocketbase.sh` script

- It installs dependencies.
- It runs `dart test`
- It runs `dart lib/main.dart`

```
> dart test
00:00 +14: All tests passed!
```

```
...
✓ Easy to switch implementations
```

```
> dart lib/main.dart
```



## You're Ready

---

- Pocketbase is now installed
- Users are set
- The student collection is made
- Pocketbase/API is OK
- Pocketbase/Dart is OK

Happy coding! 🎉

<https://pocketbase.io>