How to create a decentralized application on Minima (MiniDapps)

Introduction

There is no “right” way to create decentralized applications on Minima. We leave it up to you, the developer, to choose your path and technology that you are most comfortable using. This guide will firstly explain how to create a basic MiniDapp and then how to build a MiniDapp that interacts with the Minima blockchain, using JavaScript and the Minima API.

Before we begin, let’s define **MiniDapps**. MiniDapps are essentially web applications that are served locally on a machine, they communicate over RPC (Remote Procedure Call) with a local Minima node and provide an easier user experience.

The MiniDapp System enables node runners to install, uninstall and run MiniDapps from their node.

It is likely more desirable and of interest to build dApps to be as decentralised as possible which is what Minima facilitates by having complete nodes on mobile devices.
Pure sovereignty over your data.
As developers we must be extra creative while developing MiniDapps where we must consider efficient methods to distribute and store data, lower power usage as much as possible and grant the SMOOTHEST user experience.

Let’s begin.

Pre-requisites

There are no special hardware requirements for building a MiniDapp.
Before you start to create MiniDapps, you will need several things:

- **Java** installed on your computer https://www.java.com/download/ie_manual.jsp (this is required for running Minima)

- A running **Minima node** (on the latest version) - see pdf provided.

- A **code editor** to write your code, we recommend VS code (https://code.visualstudio.com/download) but this is your own preference

- An understanding of how the Minima Protocol works by reading the **Learn** section https://docs.minima.global/docs/learn/networkoverview

Throughout this tutorial you should have Minima's Command Line Interface (CLI) open. If running Minima on a mobile device, this is the **Terminal** MiniDapp.

Accessing the MiniDapp Hub

To access your node's MiniDapp hub, your node will need to have mds enabled.

To check that your node has mds enabled, run the **mds** command in the Minima Command Line Interface (CLI).

You should see a password for accessing your MiniDapp hub, if the password value is *null*, mds is disabled and you should stop your node and restart it with the **-mdsenable** flag.

Example

mds

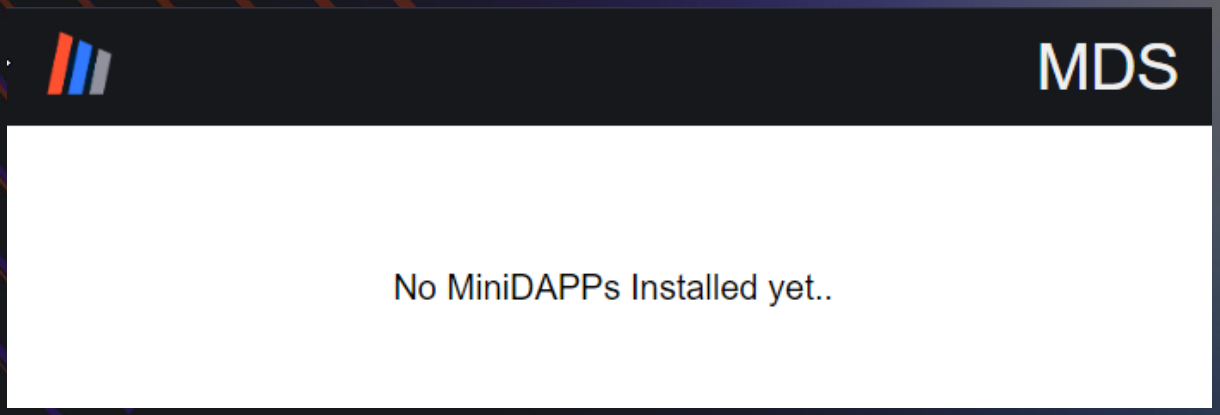

```
{
 "command":"mds",
 "status":true,
 "response":{
  "password":"H8NW-7NZ9-TV63",
  "minidapps":[]
 }
}
```


Each node runner has access to their MiniDapp hub via https://localhost:9003 or https://127.0.0.1:9003, you may be shown a security certificate warning, to which you can click on *Advanced, then Proceed.*
Note: This may be different depending on the browser and OS you are using, simply google it.

If running your node on a mobile device, you can interact with your node from a Desktop on the same WiFi as the mobile, to find the URL you need, go to the **Health** page.

You will see your MDS login page, enter the password shown when running ***mds*** from your node's CLI to access the MiniDapp Hub.

It will look something like this:



Building a basic MiniDapp

In this section, we walk through how to create a very simple MiniDapp that does not interact with the Minima blockchain.

You will learn how to

1. Create the MiniDapp config file
2. Give your MiniDapp an icon
3. Create a basic interface for your MiniDapp
4. Package your MiniDapp to load to your MiniDapp Hub
5. Install your MiniDapp on your node on Minima

Config file

The first step is to create a folder that your MiniDapp will live in. Call it **helloworld.**

In your code editor, create a new file called **dapp.conf** and save it into your **helloworld** folder.
The **dapp.conf** file is the configuration file for your MiniDapp and is where we provide the MiniDapp metadata in JSON format which will be required by the MiniDapp Hub later.

Metadata options:

name: Name of your MiniDapp
icon: name of icon file (located in the same folder)
version: MiniDapp version number
description: MiniDapp description
browser: internal or external (for mobile interface), depending on whether your MiniDapp should be launched within the app or in an external browser

Following the HelloWorld example, your dapp.conf file should look something like this:

```
{
"name":"Hello World",
"icon" :"favicon.ico",
"version" : "1.0",
"description": "My Hello World MiniDapp",
"browser": "internal"
}
```

Save it. **That's your config file done!**

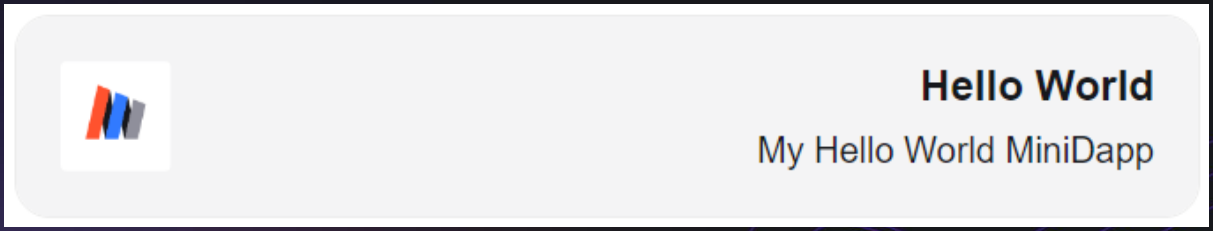
Icon

To include an icon for your MiniDapp, save an image as **favicon.ico** and add it to your helloworld folder with your dapp.conf file.

Your folder should now contain 2 files

1. dapp.conf
2. favicon.ico

Your config file and icon used will determine what is visible when your MiniDapp is loaded into the MiniDapp Hub, as shown below.



Basic interface for your MiniDapp

Next, we will create a simple webpage which will be the homepage of your MiniDapp.

Create and open an **index.html** file.

Copy and paste the following code, or alternatively, create your own basic html page. If you are comfortable with CSS styling you can also go ahead and create a .CSS file for your html file.

```
<html>

<head>
	<link rel="icon" type="image/x-icon" href="favicon.ico">
	<title>Hello World</title>
</head>
<body style="font-family:manrope;">
	<h1>Hello World!</h1>

	<p>This is my first MiniDapp!</p>

	<h3>Welcome to complete decentralization</h3>
	<p>
	Minima is a cooperative network that enables everyone to freely connect and prosper.<br></br>
	Click <a href="https://minima.global">here</a> to go to the Minima website.
	</p>

	<h3>Next Steps</h3>
	<p>
		Next, you will learn how to create a MiniDapp that interacts with the Minima Blockchain.
	</p>
	</body>

</html>
```

Save it.

Your folder should now contain 3 files.

1. dapp.conf
2. favicon.ico
3. index.html

If you also created a CSS file, add that to your helloworld folder.

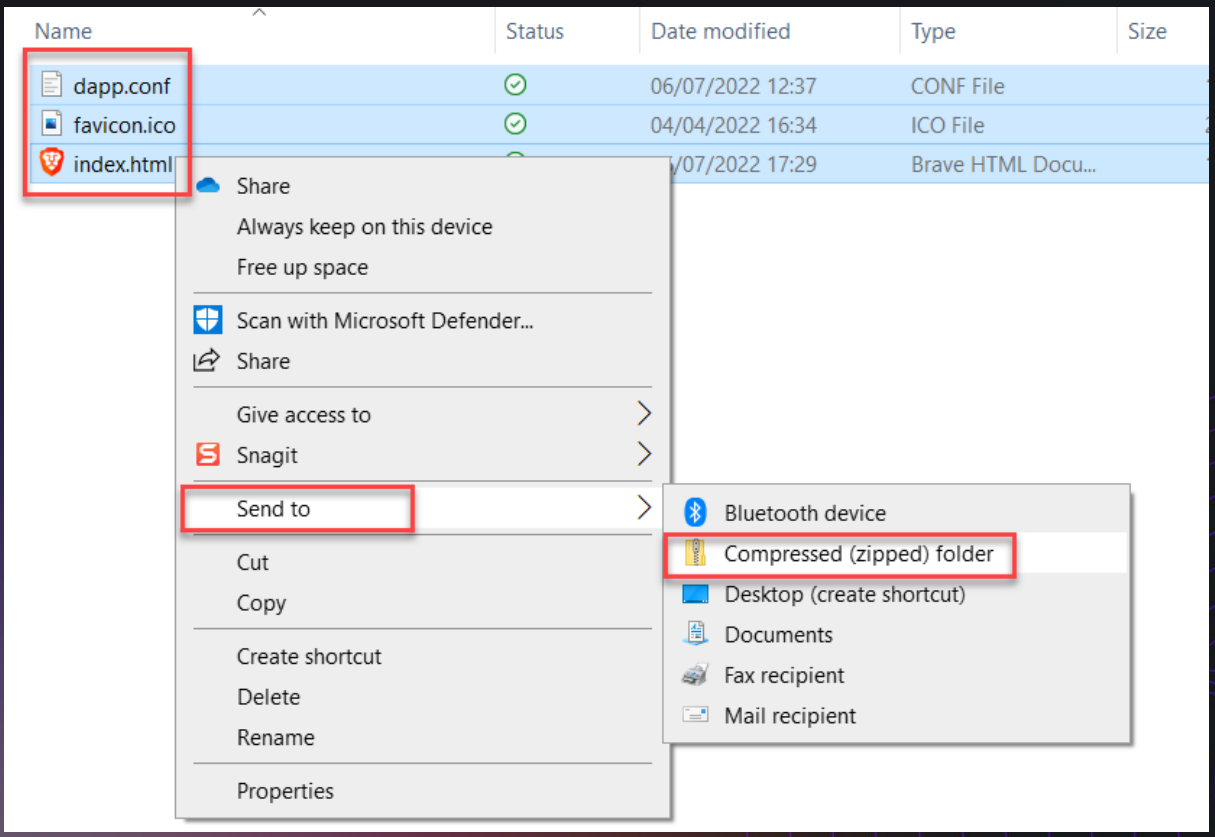
Next, we will see how the MiniDapp looks when installed onto a Minima node.

Packaging your MiniDapp

We now have a complete **helloworld** folder containing:

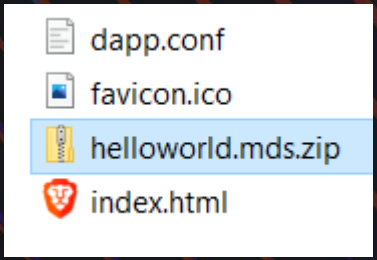
1. dapp.conf
2. favicon.ico
3. index.html
4. styling.css (optional)

Zip up the contents of this folder (not the folder itself)



Rename the folder to ***helloworld.mds.zip*** or if you are using a zip library through your cli run ***zip -r helloworld.mds.zip*** .

It should look something like this:



Installing your MiniDapp

The final stage is to install the MiniDapp onto your node.

Installing onto a mobile device

To install the MiniDapp on your phone:

1. Send the zip file to your mobile device (via email or other method)
2. Download the MiniDapp zip file to your mobile device (do NOT try to open it at this stage)
3. Open the Minima App
4. From the Home page, click the + button
5. Find the MiniDapp in your mobile's file explorer (most likely the Downloads folder)
6. Once found, select the MiniDapp, the MiniDapp will install on your node
7. Click on your MiniDapp to open it
8. Congratulations! You have created your first MiniDapp!

Installing onto a desktop node

1. Run the command `mds action:install file:helloworld.mds.zip` from the Minima Command Line (including the path with the file if required.)

Example

```
mds action:install file:C:\Users\Downloads\HelloWorld.mds.zip
Minima @ 06/07/2022 12:49:44 [233.9 MB] : unzipping package into
C:\Users\minima\mds\web\0xCFAB5D98C98CED060F5FDD235CDA27AF
{
  "command":"mds",
  "params":{
    "action":"install",
    "file":"C:\\Users\\Downloads\\HelloWorld.mds.zip"
  },
  "status":true,
```

```
"response":{
  "installed":{
    "uid":"0xCFAB5D98C98CED060F5FDD235CDA27AF",
    "conf":{
      "name":"Hello World",
      "icon":"favicon.ico",
      "version":"1.0",
      "description":"My Hello World MiniDapp",
      "browser":"internal"
    }
  }
}
```


2. After installing/uninstalling a MiniDapp, you will be required to login to your MiniDapp Hub again. Run the `mds` command to retrieve your new password.
3. Login to your MiniDapp Hub (see accessing the MiniDapp Hub)
4. Click on your MiniDapp to open it
5. Congratulations! You have created your first MiniDapp!

Building a MiniDapp that interacts with Minima blockchain

***Note**: This tutorial uses Javascript to create a web application. If you are not using Javascript, you can still communicate with Minima using http to make secure RPC Get Requests.*

You can see which Minima commands are available by typing 'help' on your node's CLI.

Create a new folder for a new MiniDapp - **helloworld2.**

Using your helloworld MiniDapp as a template, add the following files to your new folder, changing the *name, version* and *description* in the config file as you wish.

1. dapp.conf
2. favicon.ico
3. index.html
4. styling.css (optional)

Next, we will go into how to use the MiniDapp System (MDS) and javascript to create a MiniDapp that interacts with the Minima blockchain.

Introducing mds.js

To simplify development, a script is provided which gives you access to an MDS object that allows you to access useful properties which we will look at next.
Download this here.

Add the **mds.js** file to your **helloworld2** folder.

Import the **mds.js** file into the head of your **index. html** file.

It should look something like the below:

```
<head>
    <!--   The MINIMA MDS Javascript Library -->
    <script type="text/javascript" src="mds.js"></script>
    <link rel="icon" type="image/x-icon" href="favicon.ico">
    <title>Hello World #2</title>
</head>
```

Add the **mds.js** file as a script inside your body and now have access to the MDS object.

Note: NPM pkg is due to be released in the future.

If you open the script file and have a look inside you will notice a few useful functions that exist on the **MDS** object.

1. **init** function which starts up Minima and a callback for all Minima messages.
2. **log** allows you to log some data to the console with consistent timestamp.
3. **cmd** allows you to run functions on the Minima command line.
4. **sql** allows you to run sql commands on the current MiniDapp's SQL database.
5. **form** allows you to retrieve parameters from your url.

Now that we know at a high level what each function does we can go ahead and talk about each in more detail.

Initialising MDS (MDS.init)

Before being able to use the Minima API, and run any commands you need to make sure you run **MDS.init** first.

init takes one parameter, the callback function which returns you a response object in JSON format.

That would look something like this:

```
MDS.init(function(event){});
```

Events

Minima Events can be listened for, so that users can be notified when specific on-chain events occur.

The following events exist:

inited: MDS has been initialised.

NEWBALANCE: The balance of the node has changed.

NEWBLOCK: The chain tip has changed (i.e. a new block has been added to the chain).
JSON data returned: The TxPoW object of the last block is returned as a JSON Object.

MINING: Mining has started or ended.
JSON data returned: The TxPoW Object, true (if started)/false (if ended).

MINIMALOG: A new log message is available

Example use:

```
MDS.init(function(msg) {
 // inited means Minima API is ready to be used
Switch(msg.event)
case: "inited":

// do Minima dependent initializations

break;
case "newblock":
console.log(msg);
// newblock message data
break;
case:"mining":
// mining messages
console.log(msg);
// mining message data
break;
case:"newbalance":
console.log(msg);
// newbalance msg data
break;
...
            }
```

Commands (MDS.cmd)

Now that we have initialised Minima and MDS is ready to be used we can run useful commands to interface with Minima using the **cmd** function property.

Minima Terminal commands can be executed within JavaScript functions.

cmd takes two parameters, the name of the function you want to run and the callback function which returns you a response object in JSON format.

Ex.

```
MDS.cmd(minimaCommand,callback)
```

Eg.

```
MDS.cmd("balance", function(res){
console.log(res);
{
/**
  "command":"balance",
  "status":true,
  "response":[{
    "token":"Minima",
    "tokenid":"0x00",
    "confirmed":"0",
    "unconfirmed":"1000000000",
    "sendable":"0",
    "coins":"1",
    "total":"1000000000"
  }]
}
*/
});
```

For the full list of terminal commands, run help from the Minima Command Line.
Additional details can be found at:
https://docs.minima.global/docs/learn/scripting#terminal-commands

SQL (MDS.sql)

An instance of a MySQL database is available for every MiniDapp. You can create tables, insert, delete and query as you would any database using **MDS.sql.**

Under the hood is the H2 Java database engine, where you can do normal CRUD operations as you would. There may be some minor differences between MySQL and MySQL with H2. Read more here (https://www.h2database.com/html/main.html).

sql takes two parameters, the query string and a callback function.

Ex.

```
MDS.sql(sqlQuery, callback);
```

```
MDS.sql("CREATE TABLE IF NOT EXISTS CUSTOMTABLE values(...)",
function(res));
```

```
MDS.sql("INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...)", function(res));
```

```
MDS.sql("CREATE TABLE IF NOT EXISTS MESSAGES...", function(res));
```

Logs (MDS.log)

Log is a simple helper function that will output to the browser's console a time stamped message of your choice. This is useful for debugging.

Ex:

```
MDS.log("this is a time stamped message");
```

Forms (MDS.form)

MDS.form handles form submissions.
It has a property called **getParams** which will collect URL parameters.

Example

The example index.html file below provides a simple example of showing the latest block number and querying the node Status, using the Minima **status** command.

Copy and paste the code into your index. html file.

```
<html>
<head>
	<!--   The MINIMA MDS Javascript Library -->
	<script type="text/javascript" src="mds.js"></script>

	<link rel="icon" type="image/x-icon" href="favicon.ico">

	<title>Hello World</title>

</head>
<body>
<style></style>

<h2>Hello World!</h2>
<p>Current Minima Block Height: <span id="block-height"></span></p>
<p>Current Node Status: <span id="node-status"></span></p>

<button onclick='runStatus();'>CHECK STATUS</button>

<script type="text/javascript">

	function runStatus(){

		// run the Minima status command to return information about
the node's current state
		MDS.cmd("status", function(res) {

			//if the response status is true
			if (res.status) {
				//    convert the JSON object response to a
string and format it
				const nodeStatus = JSON.stringify(res.response, "
", '\t');

				// show the response on the page
				document.getElementById("node-status").innerText
= nodeStatus;
```

```
                //if the response status is false
                }else{

                      document.getElementById("node-status").innerText
= "Could not retrieve current node Status";

                }})
          }

          //Initialise web socket
          MDS.init(function(msg){
                if(msg.event == "inited") {
                      // READY TO RUN CMDS!
                      // run the Minima status command to return
information about the node's current state
                      MDS.cmd("status", function(res) {

                            if (res.status) {

                                  // get the block number from the
Status object returned
                                  const blockHeight =
res.response.chain.block;


document.getElementById("block-height").innerText = blockHeight;

                            }
                      })

                }else if(msg.event == "NEWBLOCK"){
                // the chain tip has changed

                }else if(msg.event == "NEWBALANCE"){
                // user's balance has changed

                }else if(msg.event == "MINING"){
                // mining has started or ended

                }else if(msg.event == "MINIMALOG"){
                // new Minima log message

                }else{
```

```
                }
            });


        </script>
</body>
</html>
```

Packaging your MiniDapp

We now have a complete **helloworld2** folder containing:

1. dapp.conf
2. favicon.ico
3. index.html
4. styling.css (optional)
5. mds.js

Zip up the contents of this folder (not the folder itself)

Rename the folder to ***helloworld2.mds.zip*** or if you are using a zip library through your CLI run ***zip -r helloworld2.mds.zip .***

Installing your MiniDapp

Please refer to the Installing your MiniDapp section.

Alternative Examples

The **HelloWorld2** MiniDapp can also be downloaded from https://github.com/minima-global/innovation-challenge/tree/main/Resources

Minima's official MiniDapps can be downloaded from https://github.com/minima-global/Minima/tree/dev-spartacus/mds/store/files

MiniDapp Stores (only usable on Android)

MiniDapp stores are a collection of MiniDapps that can be shared with users.
Users who install a MiniDapp store on their Android node can choose to download and install any MiniDapps within the store.

Creating a MiniDapp Store

To create a MiniDapp store, create a .json file linking to your list of MiniDapps.

This .json file can be hosted on a server, this could be a raspberry pi with a static IP.

Example of the Minima Store

```
{
	"name":"Minima Global",
	"description":"Official Minima DAPPs",
"icon":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/minima_logo.png",
	"version":"1.0",

	"dapps":[

{
"file":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/files/news-1.0.mds.zip",
"icon":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/icons/news.png",
"name":"News Feed",
"description": "Keep up to date with Minima News",
"version" : "1.0"
	},
	{
"file":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/files/block-0.1.5.mds.zip",
"icon":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/icons/block.png",
"name":"Block",
"description": "Explore the Minima blockchain",
"version" : "0.1.5"
	},
	{
"file":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/files/wallet-0.1.5.mds.zip",
"icon":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/icons/wallet.png",
"name":"Wallet",
```

```
"description": "Official Minima wallet",
"version" : "0.1.5"
      },
      {
"file":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/files/terminal-1.91.mds.zip",
"icon":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/icons/terminal.png",
"name":"Terminal",
"description": "Terminal CLI for Minima",
"version" : "1.91"
      },
      {
"file":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/files/helpdocs-0.1.1.mds.zip",
"icon":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/icons/helpdocs.png",
            "name":"Help Docs",
            "description": "Help Docs",
            "version" : "0.1.1"
      },
{
"file":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/files/maxsolo-1.83.mds.zip",
"icon":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/icons/maxsolo.png",
            "name":"MaxSolo",
            "description": "P2P chat app running over Maxima",
            "version" : "1.83"
      },
{
"file":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/files/scriptide-1.7.mds.zip",
"icon":"https://raw.githubusercontent.com/minima-global/Minima/dev-spartacus/mds
/store/icons/scriptide.png",
            "name":"Script IDE",
            "description": "Minima Script IDE",
            "version" : "1.7"
            }
      ]
}
```

Adding a MiniDapp Store to an Android node

To add a store to an Android node, go to the DAPP Store in the Minima Android app.

Click on the + button and paste in the URL to the MiniDapp store

Next Steps

The next step is to learn how to write Minima scripts and smart contracts.

Please use the following resources:
Scripting
https://docs.minima.global/docs/learn/scripting

Transaction Tutorial
Documentation including a Transaction Tutorial can be found in the Docs MiniDapp
https://github.com/minima-global/Minima/blob/dev-spartacus/mds/store/files/helpdocs-0.1.1.mds.zip

Alternatively, use Transaction Tutorial v0.81 provided.