

## Use the API

Here is an example taken from the ppp program. I expect this to be the format that will work.

```
#1/bin/sh
```

```
symbol=$( cat symbol.json )
body="{\"apAmountA\":$2,\"apAmountB\":$4,\"apCoinB\":{\"unAssetClass\":[$symbol,{\"unTokenName\":\"$symbol\"}]}"
echo $body
```

```
curl "http://localhost:9080/api/contract/instance/$(cat W$1.cid)/endpoint/add" \
--header 'Content-Type: application/json' \
--data-raw $body
```

This one is from the DBMigration document in the manual folder of `cardano-wallet`.

```
curl -vX PUT http://localhost:8090/v2/stake-pools/6f79ce9538fb79c9bb4ccd7a290eb58a878188b921
-H "Content-Type: application/json; charset=utf-8" \
-d '{
    "passphrase": "Secure Passphrase"
}'
```

```
curl -vX GET http://localhost:8090/v2/network/information
curl -vX GET http://localhost:8090/v2/network/parameters
curl -vX GET http://localhost:8090/v2/network/clock
```

Startup

```
cardano-wallet serve --port 8090 \
  --node-socket ../relay1/node.socket \
  --testnet testnet-byron-genesis.json \
  --database ./wallet-db
```

Submit transaction

```
bash $ curl -vX POST http://localhost:8090/v2/wallets/617963656a409b8a6828dc3a09001de22af904 \
      -H "Accept: application/json; charset=utf-8" \      -H
"Content-Type: application/json; charset=utf-8" \      -d '{
"payments": [      {      "address": "addr1qryq74hega5juqdl9n86697mwx0mfxu5hf02vuphnqfejtj3rkj
"amount": { "quantity": 1000000, "unit": "lovelace" }      }
], "metadata":{ "4": { "map": [ { "k": { "string": "key" },
"v": { "string": "value" } }, { "k": { "string": "14" }, "v": {
"int": 42 } } ] } }, "passphrase": "Secure Passphrase"      }'
```

For the implementation as a webpage, we can use Fetch. Here is an example.

```
var url = 'https://www.googleapis.com/geolocation/v1/geolocate?key=YOUR_KEY';
```

```

var data = {some: json};
fetch(url, {
  body: JSON.stringify(data),
  headers: {
    'dataType': 'json',
    'content-type': 'application/json'
  },
  method: 'POST',
  redirect: 'follow'
})
.then(response => {
  if (response.status === 200) {
    console.log(response.text());
  } else {
    throw new Error('Something went wrong on api server!');
  }
})
.catch(error => {
  console.error(error);
});

```

To find out more, use Mozilla. Here is a simple example.

```

fetch('http://example.com/movies.json')
  .then(response => response.json())
  .then(data => console.log(data));

```

Here is a much more complex example with notes.

```

// Example POST method implementation:
async function postData(url = '', data = {}) {
  // Default options are marked with *
  const response = await fetch(url, {
    method: 'POST', // *GET, POST, PUT, DELETE, etc.
    mode: 'cors', // no-cors, *cors, same-origin
    cache: 'no-cache', // *default, no-cache, reload, force-cache, only-if-cached
    credentials: 'same-origin', // include, *same-origin, omit
    headers: {
      'Content-Type': 'application/json'
      // 'Content-Type': 'application/x-www-form-urlencoded',
    },
    redirect: 'follow', // manual, *follow, error
    referrerPolicy: 'no-referrer', // no-referrer, *no-referrer-when-downgrade, origin, origin-when-cross-origin, unsafe-url
    body: JSON.stringify(data) // body data type must match "Content-Type" header
  });
  return response.json(); // parses JSON response into native JavaScript objects
}

```

```

postData('https://example.com/answer', { answer: 42 })
  .then(data => {
    console.log(data); // JSON data parsed by `data.json()` call
  });

```

Don't forget your credentials!

```

fetch('https://example.com', {
  credentials: 'include'
});

```

*// The calling script is on the origin 'https://example.com'*

```

fetch('https://example.com', {
  credentials: 'same-origin'
});

```

```

fetch('https://example.com', {
  credentials: 'omit'
})

```

What about JSON again?

```

const data = { username: 'example' };

fetch('https://example.com/profile', {
  method: 'POST', // or 'PUT'
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify(data),
})
  .then(response => response.json())
  .then(data => {
    console.log('Success:', data);
  })
  .catch((error) => {
    console.error('Error:', error);
  });

```

What do I do when I need to submit a transaction?

```

const formData = new FormData();
const fileField = document.querySelector('input[type="file"]');

formData.append('username', 'abc123');
formData.append('avatar', fileField.files[0]);

fetch('https://example.com/profile/avatar', {

```

```
    method: 'PUT',  
    body: formData  
  })  
  .then(response => response.json())  
  .then(result => {  
    console.log('Success:', result);  
  })  
  .catch(error => {  
    console.error('Error:', error);  
  });
```