

Integrate your BTC Wallet to Magic Eden

Learn how to integrate your BTC wallet into Magic Eden marketplace! [Suggest Edits](#)

This page is specifically designed for creators of other BTC wallets who want their wallet to be recognized on Magic Eden's Marketplace. Prepping your BTC wallet to be picked up by our site requires you to register your wallet with wallet standard. Find the instructions below.

Required dependencies:

Shell `npm i @exodus/bitcoin-wallet-standard-sats-connect @exodus/bitcoin-wallet-standard-satoshi`

1. Create a provider that conforms to the sats-connect

JSX provider.ts

```
import { decodeToken } from '@olistic/jsontokens'; import { hex } from '@scure/base'; import type { BitcoinProvider as SatsConnectBitcoinProvider, GetAddressResponse, SignTransactionResponse, } from 'sats-connect'; import { AddressPurposes } from 'sats-connect';
```

```
import type { RPC } from '../rpc'; import type { Account } from '../wallet';
```

```
export class BitcoinProvider implements SatsConnectBitcoinProvider { #rpc: RPC;
```

```
  constructor(rpc: RPC) {  
    this.#rpc = rpc;  
  }
```

```
  async connect(request: string): Promise<GetAddressResponse> {  
    if (!request) {  
      throw new Error('Invalid request.');
```

```
    }  
  
    const { payload } = decodeToken(request);  
    if (typeof payload !== 'string') {  
      throw new Error('Invalid request.');
```

```
    }  
  
    const { purposes } = payload as {  
      purposes?: AddressPurposes[];  
    };
```

```
    const accounts = await this.#rpc.callMethod<Account[]>('connect', [purposes]);
```

```
    return {  
      addresses: accounts.map(({ purpose, publicKey, address }) => ({  
        address,  
        publicKey: hex.encode(publicKey),  
        purpose: purpose === 'payment' ? AddressPurposes.PAYMENT : AddressPurposes.ORDINALS,  
      })),  
    };  
  }
```

```
  async signTransaction(request: string): Promise<SignTransactionResponse> {  
    throw new Error('Method not implemented.');
```

```
  }  
  
  async signMessage(request: string): Promise<string> {  
    throw new Error('Method not implemented.');
```

```
  }  
  
  async call(request: string): Promise<Record<string, any>> {  
    throw new Error('Method not implemented.');
```

2. Initialize the sats-connect provider through wallet-standard

```
JSX import { initialize as initializeSatsConnect } from '@exodus/bitcoin-wallet-standard-sats-connect'; import { initialize as initializeSatoshi } from '@exodus/bitcoin-wallet-standard-satoshi';
```

```
import { BitcoinProvider } from './provider';
```

```
const rpc = createWindowRPC(window); const provider = new BitcoinProvider(rpc);

// Temporary to allow debugging in the browser. Object.defineProperty(window, 'BitcoinProvider', { value: provider, });

/* * Initialize the standard wallet /

// Sats Connect Wallet initializeSatsConnect(provider); This will register the wallet with wallet standard and inject into
window.BitcoinProvider so that magiceden.io can detect your wallet Updated 9 days ago * Table of Contents * * Required dependencies: * * 1. Create a provider that conforms to the sats-connect * * 2. Initialize the sats-connect provider through wallet-standard
```