

---

# **bos-mint Documentation**

***Release 0.0.1***

**PeerPlays Blockchain Standards Association**

**Jan 07, 2021**



---

## Contents:

---

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Install dependencies (as root/sudo) . . . . .	3
1.2	Install databases (as root/sudo) . . . . .	3
1.3	Install bos-mint (as user) . . . . .	3
1.4	Upgrading bos-mint (as user) . . . . .	4
1.5	Modify configuration . . . . .	4
1.6	Running bos-mint . . . . .	6
<b>2</b>	<b>bos_mint package</b>	<b>9</b>
2.1	Submodules . . . . .	9
2.1.1	bos_mint.assets module . . . . .	9
2.1.2	bos_mint.cli module . . . . .	9
2.1.3	bos_mint.forms module . . . . .	9
2.1.4	bos_mint.istring module . . . . .	9
2.1.5	bos_mint.models module . . . . .	9
2.1.6	bos_mint.node module . . . . .	9
2.1.7	bos_mint.tostring module . . . . .	9
2.1.8	bos_mint.utils module . . . . .	9
2.1.9	bos_mint.views module . . . . .	9
2.1.10	bos_mint.web module . . . . .	9
2.1.11	bos_mint.widgets module . . . . .	9
2.1.12	bos_mint.wrapper module . . . . .	9
2.2	Module contents . . . . .	9
<b>3</b>	<b>Indices and tables</b>	<b>11</b>



The purpose of these tools is to serve the witnesses and assist them with their regular ground work of managing events on the PeerPlays blockchain.



# CHAPTER 1

---

## Installation

---

### 1.1 Install dependencies (as root/sudo)

```
apt-get install libffi-dev libssl-dev python-dev python3-pip
pip3 install virtualenv
```

Note that virtualenv is a best practice for python, but installation can also be on a user/global level.

### 1.2 Install databases (as root/sudo)

MINT uses a local sqllite database which requires mysql setup (running a mysql server instance is not required). Assuming a Ubuntu 16.04. machine, please install

```
apt-get install libmysqlclient-dev
```

### 1.3 Install bos-mint (as user)

You can either install bos-mint via pypi / pip3 (production installation) or via git clone (debug installation). For production use install bos-auto via pip3 is recommended, but the git master branch is always the latest release as well, making both installations equivalent. Suggested is a seperate user

```
cd ~
mkdir bos-mint
cd bos-mint
# create virtual environment
virtualenv -p python3 env
# activate environment
source env/bin/activate
```

(continues on next page)

(continued from previous page)

```
# install bos-mint into virtual environment
pip3 install bos-mint
```

For debug use, checkout from github (master branch) and install dependencies manually

```
cd ~
# checkout from github
git checkout https://github.com/pbsa/bos-mint
cd bos-mint
# create virtual environment
virtualenv -p python3 env
# activate environment
source env/bin/activate
# install dependencies
pip3 install -r requirements.txt
```

BOS MINT is supposed to run in the virtual environment. Either activate it beforehand like shown above or run it directly in the env/bin folder.

## 1.4 Upgrading bos-mint (as user)

For production installation, upgrade to the latest version - including all dependencies - via

```
pip3 install --upgrade --upgrade-strategy eager bos-mint
```

For debug installation, pull latest master branch and upgrade dependencies manually

```
git pull
pip3 install -r requirements.txt --upgrade --upgrade-strategy eager
```

## 1.5 Modify configuration

We now need to configure bos-auto.

```
# basic mint configuration file
wget https://raw.githubusercontent.com/PBSA/bos-mint/master/config-example.yaml
mv config-example.yaml config-bos-mint.yaml
# modify config-bos-mint.yaml (add your own peerplays node and secret key)
```

Default config only requires as listed below:

```
# see bos_mint/config-defaults.yaml for explanation and all possible override values

secret_key: # enter some random string

allowed_assets:
  - BTF

connection:
  use: baxter

  baxter:
```

(continues on next page)



(continued from previous page)

```
node:
  - # enter your node
```

Possible override values are described below:

```
debug: False

project_name: MINT
project_sub_name: The BOS Manual Intervention Module
secret_key: # enter any random string

sql_database: "sqlite:///{{cwd}}/bookied-local.db"

connection:
  use: # enter your desired chain

  baxter:
    node:
      - # enter your node
    nobroadcast: False
    num_retries: 1

notifications:
  accountLessThanCoreInfo: 1000
  accountLessThanCoreWarning: 200

allowed_assets:
  - BTF
  - BTC
  - PPY
  - BTCTEST
  - TEST

allowed_transitions:
  EventStatus:
    create:
      - upcoming
    upcoming:
      - in_progress
      - finished
      - frozen
      - canceled
    in_progress:
      - finished
      - frozen
      - canceled
    finished:
      - canceled
    frozen:
      - upcoming
      - in_progress
      - frozen
      - canceled
      - finished
  BettingMarketGroupStatus:
    create:
      - upcoming
```

(continues on next page)

(continued from previous page)

```
upcoming:
  - closed
  - canceled
  - frozen
in_play:
  - frozen
  - closed
  - canceled
frozen:
  - closed
  - canceled
  - in_play
  - upcoming
closed:
  - graded
  - canceled
graded:
  - re_grading
  - settled
  - canceled
re_grading:
  - graded
  - canceled
BettingMarketStatus:
  create:
    - unresolved
  unresolved:
    - win
    - not_win
    - canceled
    - frozen
  frozen:
    - unresolved
    - win
    - not_win
    - canceled
  win:
    - not_win
    - canceled
  not_win:
    - canceled
    - win
```

## 1.6 Running bos-mint

To run MINT in debug mode use

```
bos-mint start --port 8001 --host localhost
```

The output that you see should contain

```
2018-05-18 11:56:04,754 INFO      : * Running on http://localhost:8001/ (Press_
↵CTRL+C to quit)
```

The above setup is basic and for development use. Going forward, a witness may want to deploy UWSGI with parallel workers for the endpoint.

MINT is purposely run on localhost to restrict outside access. Securing a python flask application from malicious break in attempts is tedious and would be an ongoing effort. Recommendation is to access it via a SSH tunnel or through VPN.

Example for SSH tunnel: Assume BOS MINT is running on a remote server accessible via 1.2.3.4 and you have login credentials via SSH (password or private key access). On the local machine that you will be using to access MINT via a web browser open the tunnel

```
ssh -f -N -L 8080:127.0.0.1:8001 yourusername@1.2.3.4
```

-f - send process to background -N - do not send commands (if you need open ssh connections only for tunneling) -L - port mapping (8080 port on your machine, 127.0.0.1:8001 - proxy to where MINT runs)

Now you can open mint in your browser using <http://localhost:8080> address.

After starting MINT use your favorite desktop browser to access it and you will be asked to enter your witness key that will be stored encrypted in the local peerplays wallet. Please note that MINT is not optimized for mobile use yet.



### 2.1 Submodules

2.1.1 `bos_mint.assets` module

2.1.2 `bos_mint.cli` module

2.1.3 `bos_mint.forms` module

2.1.4 `bos_mint.istring` module

2.1.5 `bos_mint.models` module

2.1.6 `bos_mint.node` module

2.1.7 `bos_mint.tostring` module

2.1.8 `bos_mint.utils` module

2.1.9 `bos_mint.views` module

2.1.10 `bos_mint.web` module

2.1.11 `bos_mint.widgets` module

2.1.12 `bos_mint.wrapper` module

### 2.2 Module contents



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`