08

Registrando e gravando a informação

Agora vamos de fato registrar a informação no blockchain do Sawtooth. O código deve ficar parecido com a versão encontrada no endereço https://github.com/alberto-alura/curso-introducao-blockchain/releases/tag/4.4 (https://github.com/alberto-alura/curso-introducao-blockchain/releases/tag/4.4).

O seu arquivo `index.js` deve ficar parecido com o código abaixo.

```
var restify = require('restify');

const {registerBlockchain} = require('./sawtooth/client');
const processor = require('./sawtooth/processor');
const {VoteHandler} = require('./sawtooth/voteHandler');

processor(new VoteHandler());

function registerVote(req, res, next) {
  const voto = req.body;
  registerBlockchain(voto);

  res.send(200);
  next();
}

var server = restify.createServer();
server.use(restify.plugins.acceptParser(server.acceptable));
server.use(restify.plugins.bodyParser());

server.post('/register/vote', registerVote);

server.listen(8084, function() {
  console.log('%s listening at %s', server.name, server.url);
});
```

Um detalhe muito importante, algumas vezes, entre os restarts da nossa aplicação Javascript, o sawtooth rodando dentro do Docker simplesmente trava e para de processar nossos smart contratcts(handlers). Minha sugestão é que você mate a instância do docker e suba de novo. Para fazer isso, vá na aba do seu terminal rodando o sawtooth e execute a seguinte sequência de passos:

- `ctrl+c` para Linux ou Mac e `ctrl+d` para Windows.
- `docker-compose -f sawtooth-default.yaml down`
- `docker-compose -f sawtooth-default.yaml up`

Depois de subir o sawtooth, suba a nossa aplicação Javascript :).

Execute o processo para realizar um voto e depois acesse o endereço http://localhost:8008/blocks (http://localhost:8008/blocks) para verificar que um novo bloco foi adicionado!.