

# TRUFFLE UNIVERSITY

# CREATING TRUFFLE PLUGINS

By Rosco Kalis



# ABOUT ME

- Graduate Student at University of Amsterdam
- High level contract language for Bitcoin Cash (CashScript)
- truffle-assertions
- truffle-plugin-verify



# CONTENTS

- What are Truffle plugins?
- How do Truffle plugins work?
- Creating our own Truffle plugin



# WHAT ARE PLUGINS?

1. Install the plugin from NPM.

```
npm install --save-dev truffle-plugin-hello
```

2. Add a `plugins` section to your Truffle config.

```
module.exports = {
  /* ... rest of truffle-config */

  plugins: [
    "truffle-plugin-hello"
  ]
}
```

3. Run the command

```
$ truffle run hello
Hello, World!
```



# WHAT PLUGINS EXIST?

- truffle-plugin-verify

```
truffle run verify SmartContractName [--network networkName]
```

- truffle-security

```
truffle run verify [options] [solidity-file[:contract-name] [solidity-file[:contract-name] ...]]
```



# HOW DO PLUGINS WORK?

1. Implement the command as a Node module with a function as its default export.

   Example: `hello.js`

```
module.exports = async (config) => {
  if (config.help) {
    console.log(`Usage: truffle run hello [name]`);
    done(null, [], []);
    return;
  }

  let name = config._.length > 1 ? config._[1] : 'World!';
  console.log(`Hello, ${name}`);
}
```

2. Define a `truffle-plugin.json` file to specify the command. Example: `truffle-plugin.json`

```
{
  "commands": {
    "hello": "hello.js"
  }
}
```



# THE CONFIG OBJECT

```
Config {
  // Information about the CLI
  truffle_directory: string, // Truffle executable location
  working_directory: string, // CWD
  _: string[], // Raw positional command line arguments

  // Any additional command line flags / arguments
  myString: string // --myString hello
  myint: number // --myInt 5

  // Information included in the truffle-config.js file
  networks: { rinkeby: object, ropsten: object },
  compilers: { solc: object, vyper: {} },
  plugins: string[],
  api_keys: { etherscan: string },
  /* ... any other fields defined in truffle-config.js */

  // Network config
  network: string, // --network rinkeby|ropsten|etc
  network_id: number // Current network id
  network_config: object // Current network config (from truffle-config.js)

  // Directory information
  build_directory: string, // ./build
  contracts_directory: string, // ./contracts
  contracts_build_directory: string, // ./build/contracts
  migrations_directory: string, // ./migrations
  migrations_file_extension_regexp: RegExp,
  test_directory: string, // ./test
  test_file_extension_regexp: RegExp,
}
```



# PLUGIN CREATION WORKSHOP

- Plugin for storing artifacts
  - Reads a contract's artifact file
  - Sends it to a MongoDB collection URL
  - Outputs the number of stored artifacts for the contract
- https://github.com/rkalis/truffle-university-workshop



# FURTHER READING

- https://www.trufflesuite.com/docs/truffle/getting-started/writing-external-scripts
- https://github.com/rkalis/awesome-truffle-plugins
- https://kalis.me/verify-truffle-smart-contracts-etherscan/