EOSFACTORY

SMART-CONTRACT DEVELOPMENT & TESTING IDE

Presented by

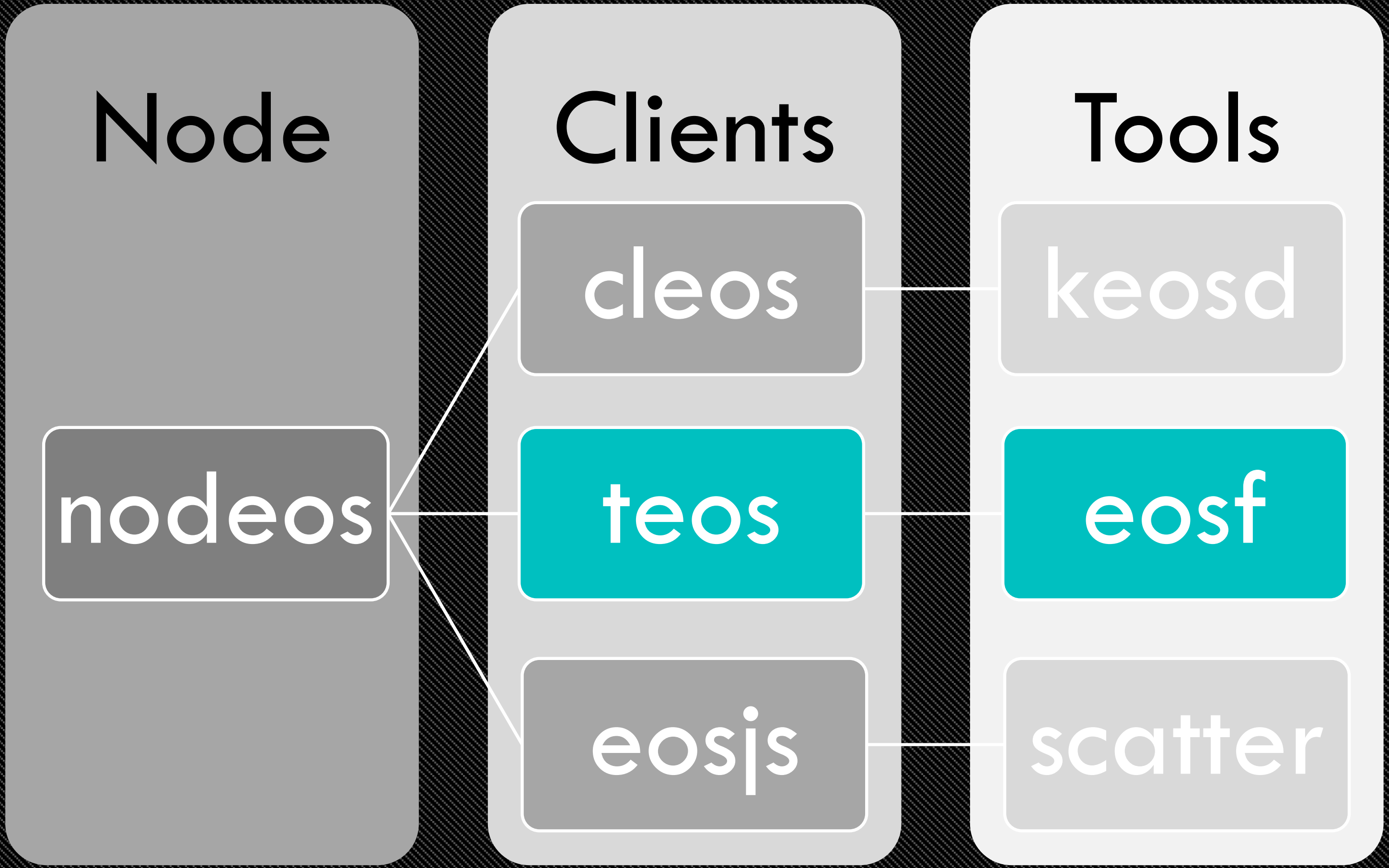
TOKENIKA

> “An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools, and a debugger.”

Wikipedia

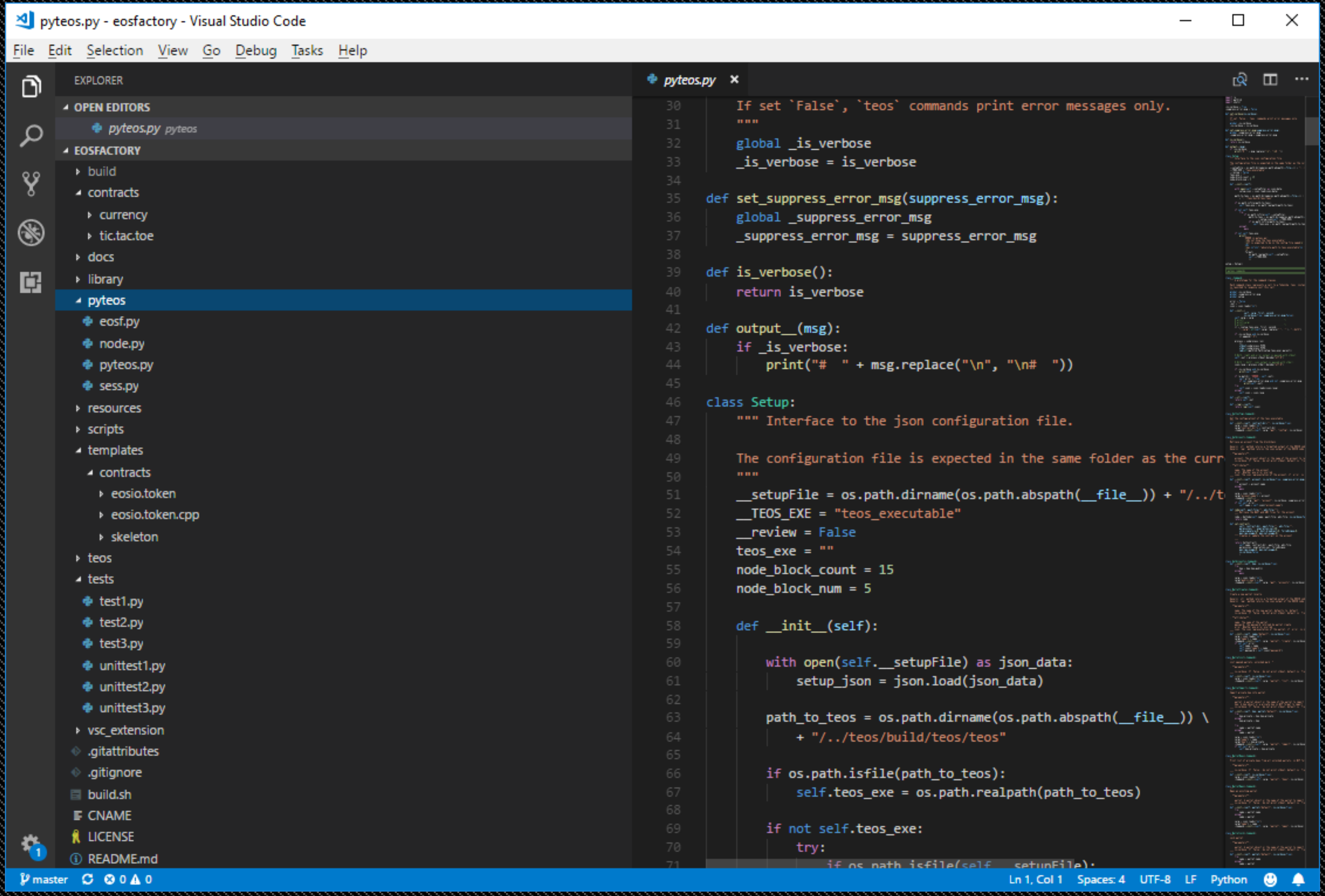
TOKENIKA

EOSIO INFRASTRUCURE



TOKENIKA

EOSFACTORY COMPONENTS

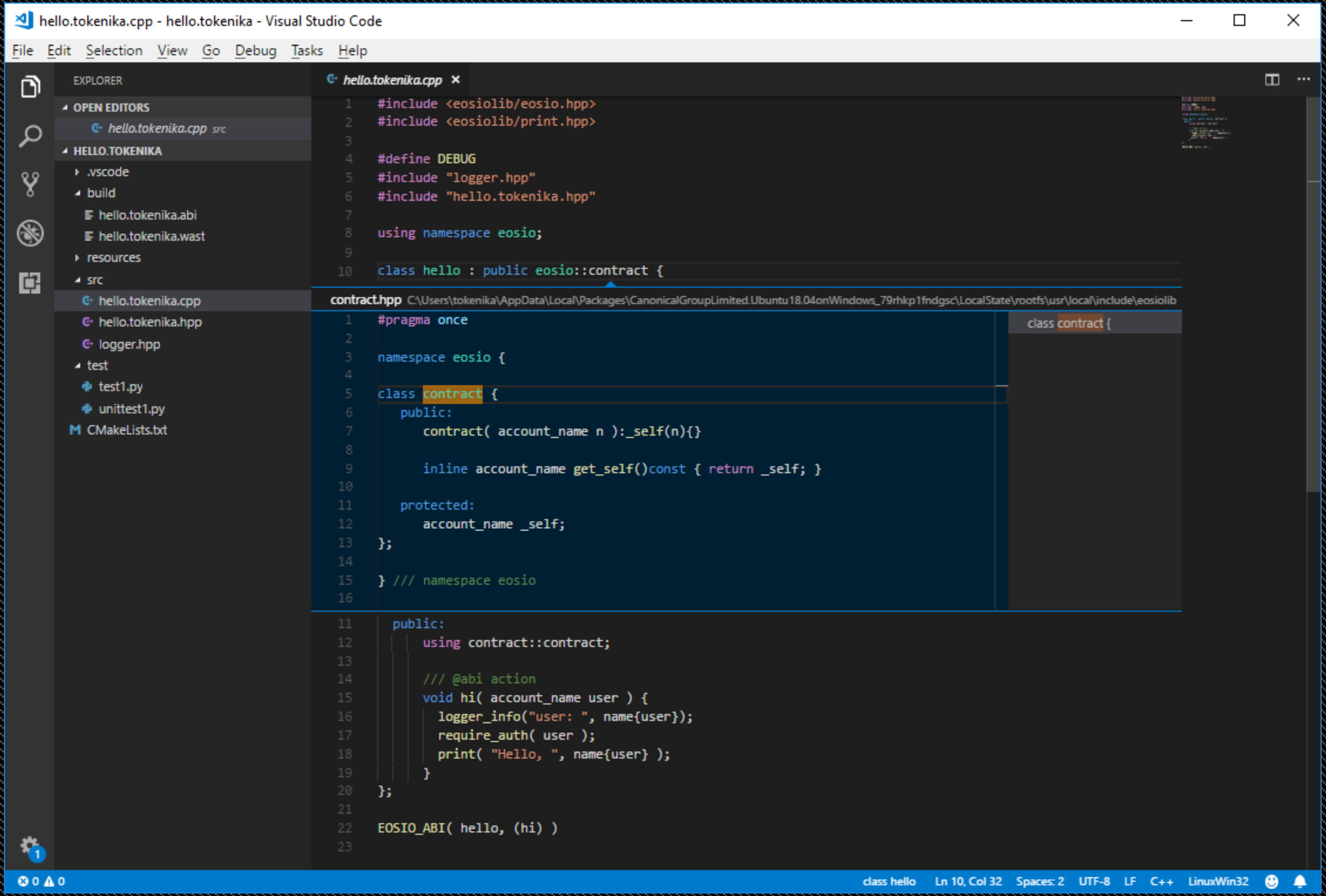


```python
    If set `False`, `teos` commands print error messages only.
    """
    global _is_verbose
    _is_verbose = is_verbose

def set_suppress_error_msg(suppress_error_msg):
    global _suppress_error_msg
    _suppress_error_msg = suppress_error_msg

def is_verbose():
    return is_verbose

def output__(msg):
    if _is_verbose:
        print("#  " + msg.replace("\n", "\n#  "))

class Setup:
    """ Interface to the json configuration file.

    The configuration file is expected in the same folder as the curr
    """
    __setupFile = os.path.dirname(os.path.abspath(__file__)) + "/../t
    __TEOS_EXE = "teos_executable"
    __review = False
    teos_exe = ""
    node_block_count = 15
    node_block_num = 5

    def __init__(self):

        with open(self.__setupFile) as json_data:
            setup_json = json.load(json_data)

        path_to_teos = os.path.dirname(os.path.abspath(__file__)) \
            + "/../teos/build/teos/teos"

        if os.path.isfile(path_to_teos):
            self.teos_exe = os.path.realpath(path_to_teos)

        if not self.teos_exe:
            try:
                if os.path.isfile(self.__setupFile):
```

MAIN FEATURES IN V1.0

1. Standard unit-testing framework
2. Support for user-defined workspaces
3. Support for debugging
4. Preliminary code verification
5. Visual Studio Code integration

EOSFACTORY INTELLISENSE



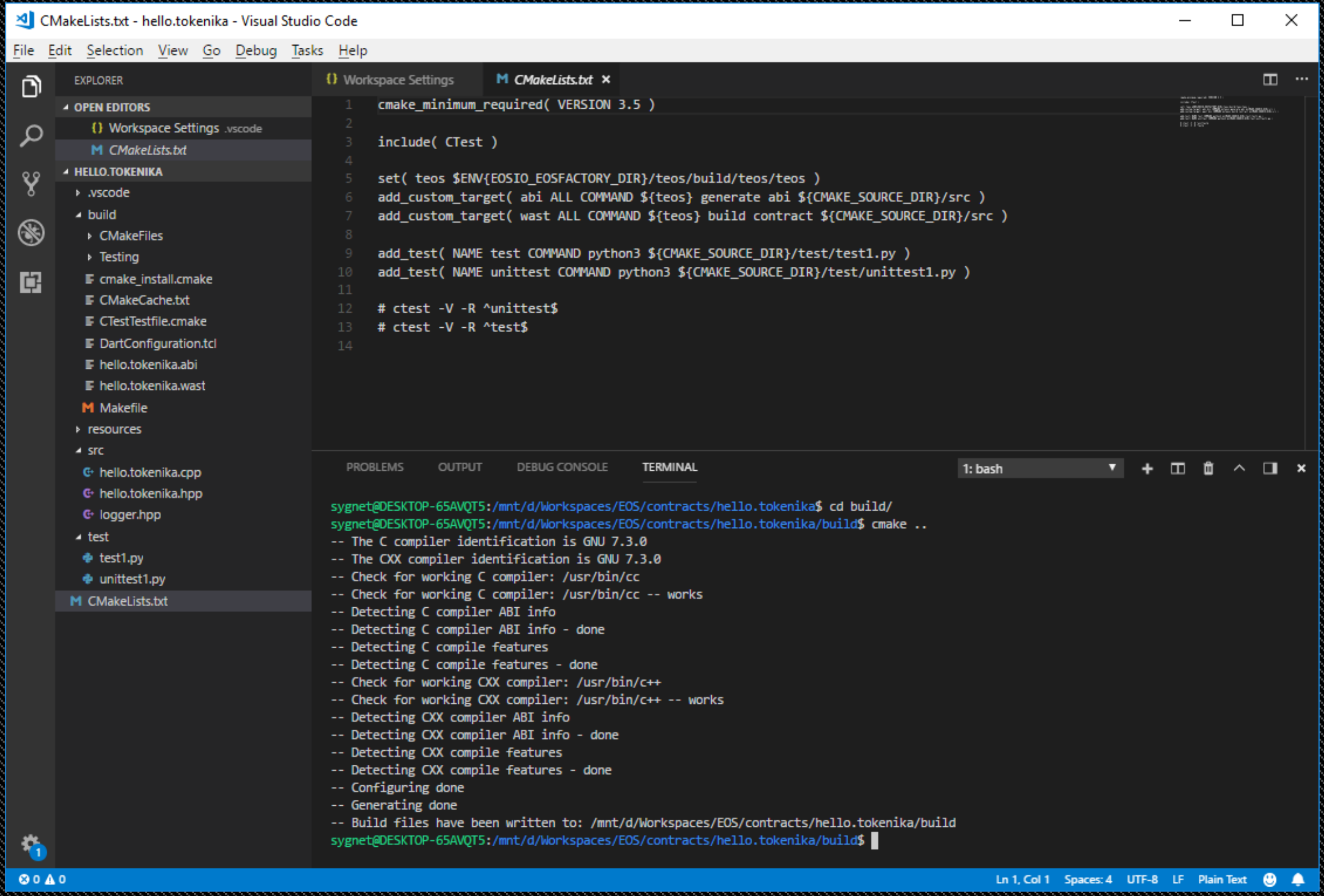
```
#include <eosiolib/eosio.hpp>
#include <eosiolib/print.hpp>

#define DEBUG
#include "logger.hpp"
#include "hello.tokenika.hpp"

using namespace eosio;

class hello : public eosio::contract {
```

contract.hpp C:\Users\tokenika\AppData\Local\Packages\CanonicalGroupLimited.Ubuntu18.04onWindows_79rhkp1fndgsc\LocalState\rootfs\usr\local\include\eosiolib

```
#pragma once

namespace eosio {

class contract {
   public:
      contract( account_name n ):_self(n){}

      inline account_name get_self()const { return _self; }

   protected:
      account_name _self;
};

} /// namespace eosio
```

```
  public:
      using contract::contract;

      /// @abi action
      void hi( account_name user ) {
        logger_info("user: ", name{user});
        require_auth( user );
        print( "Hello, ", name{user} );
      }
};

EOSIO_ABI( hello, (hi) )
```

EOSFACTORY CMAKE SUPPORT

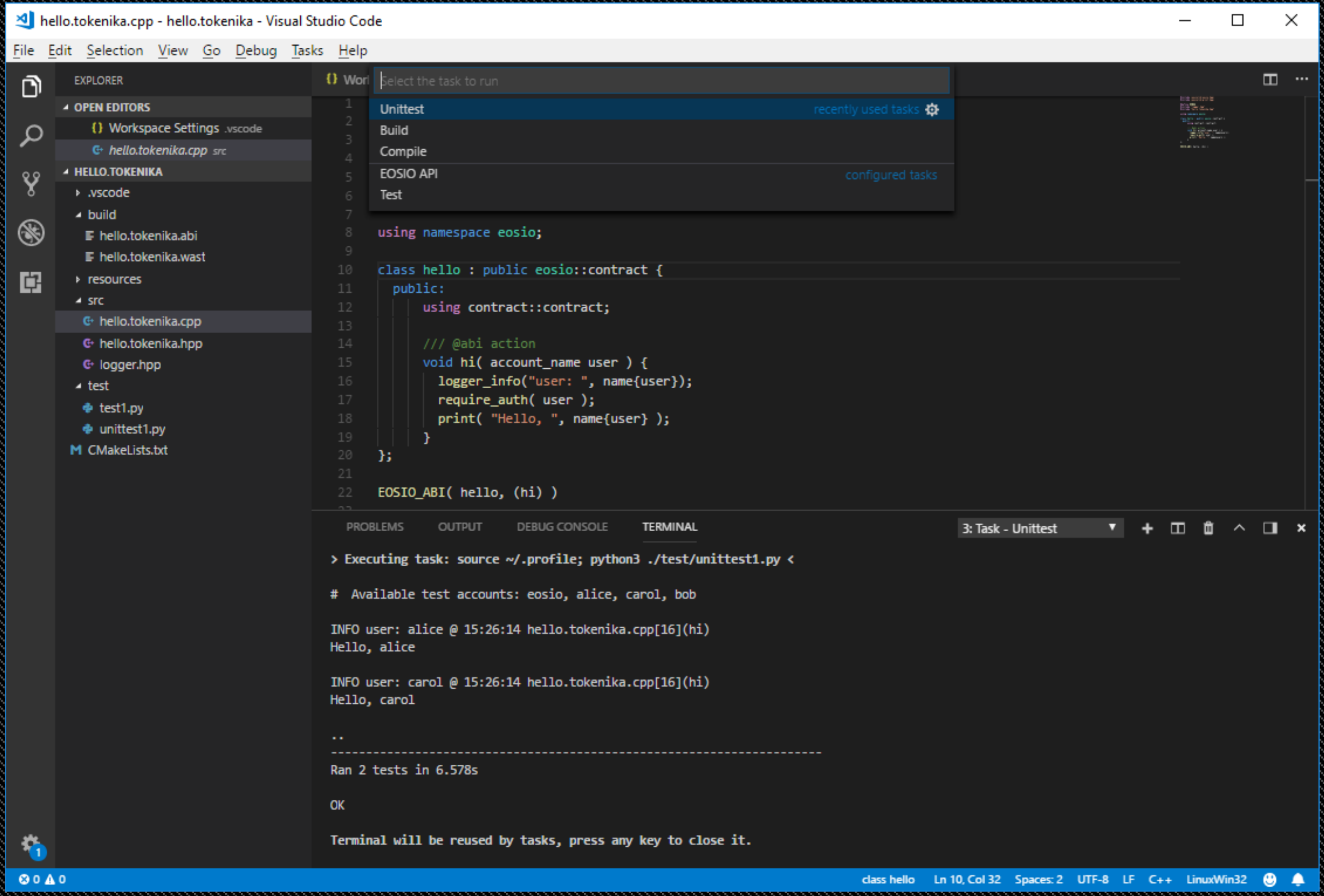


```
cmake_minimum_required( VERSION 3.5 )

include( CTest )

set( teos $ENV{EOSIO_EOSFACTORY_DIR}/teos/build/teos/teos )
add_custom_target( abi ALL COMMAND ${teos} generate abi ${CMAKE_SOURCE_DIR}/src )
add_custom_target( wast ALL COMMAND ${teos} build contract ${CMAKE_SOURCE_DIR}/src )

add_test( NAME test COMMAND python3 ${CMAKE_SOURCE_DIR}/test/test1.py )
add_test( NAME unittest COMMAND python3 ${CMAKE_SOURCE_DIR}/test/unittest1.py )

# ctest -V -R ^unittest$
# ctest -V -R ^test$
```

```
sygnet@DESKTOP-65AVQT5:/mnt/d/Workspaces/EOS/contracts/hello.tokenika$ cd build/
sygnet@DESKTOP-65AVQT5:/mnt/d/Workspaces/EOS/contracts/hello.tokenika/build$ cmake ..
-- The C compiler identification is GNU 7.3.0
-- The CXX compiler identification is GNU 7.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/Workspaces/EOS/contracts/hello.tokenika/build
sygnet@DESKTOP-65AVQT5:/mnt/d/Workspaces/EOS/contracts/hello.tokenika/build$
```

TOKENIKA

EOSFACTORY UNIT TESTS

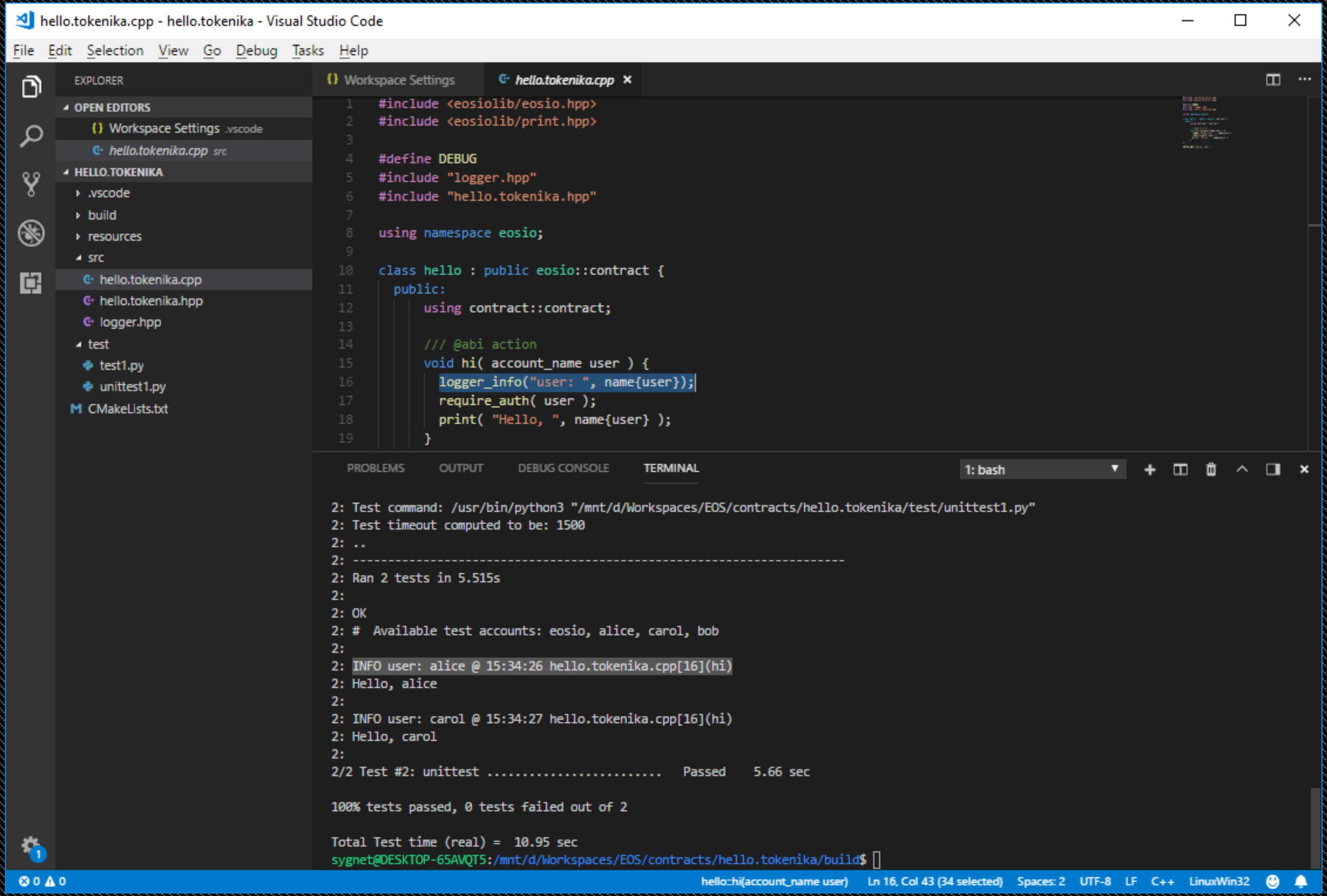


```
using namespace eosio;

class hello : public eosio::contract {
  public:
      using contract::contract;

      /// @abi action
      void hi( account_name user ) {
        logger_info("user: ", name{user});
        require_auth( user );
        print( "Hello, ", name{user} );
      }
};

EOSIO_ABI( hello, (hi) )
```

```
> Executing task: source ~/.profile; python3 ./test/unittest1.py <

#  Available test accounts: eosio, alice, carol, bob

INFO user: alice @ 15:26:14 hello.tokenika.cpp[16](hi)
Hello, alice

INFO user: carol @ 15:26:14 hello.tokenika.cpp[16](hi)
Hello, carol

..
----------------------------------------------------------------------
Ran 2 tests in 6.578s

OK

Terminal will be reused by tasks, press any key to close it.
```

TOKENIKA

EOSFACTORY DEBUGGING



```cpp
#include <eosiolib/eosio.hpp>
#include <eosiolib/print.hpp>

#define DEBUG
#include "logger.hpp"
#include "hello.tokenika.hpp"

using namespace eosio;

class hello : public eosio::contract {
  public:
      using contract::contract;

      /// @abi action
      void hi( account_name user ) {
        logger_info("user: ", name{user});
        require_auth( user );
        print( "Hello, ", name{user} );
      }
```

```
2: Test command: /usr/bin/python3 "/mnt/d/Workspaces/EOS/contracts/hello.tokenika/test/unittest1.py"
2: Test timeout computed to be: 1500
2: ..
2: ----------------------------------------------------------------------
2: Ran 2 tests in 5.515s
2:
2: OK
2: #  Available test accounts: eosio, alice, carol, bob
2:
2: INFO user: alice @ 15:34:26 hello.tokenika.cpp[16](hi)
2: Hello, alice
2:
2: INFO user: carol @ 15:34:27 hello.tokenika.cpp[16](hi)
2: Hello, carol
2:
2/2 Test #2: unittest .........................   Passed    5.66 sec

100% tests passed, 0 tests failed out of 2

Total Test time (real) =  10.95 sec
sygnet@DESKTOP-65AVQT5:/mnt/d/Workspaces/EOS/contracts/hello.tokenika/build$
```

TOKENIKA

THANK YOU

ANY QUESTIONS?

EOSFACTORY

eosfactory.io

github.com/tokenika/eosfactory

Presented by

TOKENIKA

tokenika.io

contact@tokenika.io