

# 10 PRINT "Hello World!" Тестирование Social 20 GOTO 10



В данном разделе живет документация по тестированию соцкластера

- Тесты
  - Юнит-тесты
  - Функциональные тесты.

## Тесты

После каждого изменения в коде соц. сервера, необходимо запускать тесты на логику, базовые механизмы и т.д.

## Юнит-тесты

Не требуют поднятого соцсервера. Тесты запускаются из корня соц.кластера **Social\all\_test.py**

Для прохождения юнит-теста отдельных компонент (без работающего кластера, чисто с заглушками) запускаем /tests/unit.test.py (5 тестов должно пройти OK)

Данные тесты рекомендовано запускать перед каждым коммитом.

## Функциональные тесты.

Для полных кластерных тестов нужно поднять кластер и запустить его (см. Разворачивание кластера<sup>📄</sup>) и затем:

- для прохождения тестов кластера запускаем файлы тестов с названием типа "/test/\*.ftest.py"
- для тестирования matchmaking по http запускаем отдельным процессом mock\_matchmaker.py (например --port=8702; можно это делать через "/batch/ws\_MockMatchMaker.bat"), указываем тот же номер порта в конфиге координатора (COORDINATOR\_PVX\_MATCHMAKER), и запускаем async.load.py (можно это делать через "/batch/ws\_MockMmClient.bat")

⚠️ Для тестирования **standalone matchmaking** (БЕЗ запуска всего кластера и MySQL) достаточно:

1. запустить nginx (/tools/nginx/nginx.exe)
2. запустить standalone web server (с опциями "--coordinator=none --mc=none --sql=none", например с помощью батника /batch/standalone\_ws\_8801.bat)
3. запустить серверную заглушку mock\_matchmaker.py (например с помощью /batch/ws\_MockMatchMaker.bat)
4. запустить клиент async.load.py (можно через /batch/ws\_MockMmClient.bat)

