QTS App Manual

Version 2019-11-18

Content

[General information 3](#_Toc24957767)

[Compatibility 3](#_Toc24957768)

[Filesystem 3](#_Toc24957769)

[Folder Structure 3](#_Toc24957770)

[Setup 4](#_Toc24957771)

[Buglist 5](#_Toc24957772)

[Major 5](#_Toc24957773)

[Minor 5](#_Toc24957774)

[Upper Area 6](#_Toc24957775)

[Servicestatus 6](#_Toc24957776)

[ExecutorQueue 6](#_Toc24957777)

[Quick Menu 7](#_Toc24957778)

[General 7](#_Toc24957779)

[Service 7](#_Toc24957780)

[Upload 7](#_Toc24957781)

[General TABS 8](#_Toc24957782)

[TAB Schedule 8](#_Toc24957783)

[TAB Info 8](#_Toc24957784)

[TAB Config 8](#_Toc24957785)

[TAB Result 9](#_Toc24957786)

[TABs per Testcase 10](#_Toc24957787)

[http 10](#_Toc24957788)

[Web 10](#_Toc24957789)

[Ping 11](#_Toc24957790)

[YouTube 11](#_Toc24957791)

[Idle 11](#_Toc24957792)

[Changelog 12](#_Toc24957793)

# General information

## Compatibility

The App has been tested on a Nexus 5 running Android 6.0.1 and on a Moto G5 running Android 7, later Updated to Android 8.1.0. Tests on devices without Mobile network capabilities (No SIM-Socket à no IMEI) were not performed.

## Filesystem

The App will create the file structure it needs to Operate on the Primary (internal) storage of the device it is running on. It will also extract some resources from the APK-Bundle. A file containing the data for upload-testing will also be created, this will be 100 MB in size. Currently there is no check on availability of storage space, so the device might run out of free space.

Files may not be visible when connecting the device via USB, using a file explorer on the device is recommended. To copy test results it sometimes is necessary to copy/move the files from the QTS-Folder to a different folder on the SD-Card (e.g. /sdcard/copyme)

## Folder Structure

* sdcard
  + QTS
    - zipped
      * test Results in zip-format per Test
    - logs
      * upload logs
    - pcap
      * pcap trace per Test
    - screenshots
      * screenshot (jpeg) per Test
    - audio
      * audio recordings (empty for now) per Test
    - binaries
      * pcap binary
    - testdata
      * 100MB.bin, used for upload tests
    - qtsConfig.xml
      * permanent storage of config

# Setup

* Rooting a device is strongly recommended and mandatory to capture pcap files
  + Instructions will not be Provided here
  + **Rooting may destroy your Device, try at your own Risk**
  + Enable installation from unknown sources
* Install the app either via adb or by copying to the device SD-card
* Per default the app will use QUALIGON Servers as default, including:
  + Server to get Tasks (QTS-Portal)
  + Server to upload results to
  + Server used for http DL/UL tests
* To change the Server config
  + open the app
  + in the Tab config hit “Save”
  + **CLOSE the app**
  + Open the qtsConfig.xml (see chapter Folder Structure)
    - Change “qts.uploadURL”
    - Change “qts.taskURL”
  + Save the File
  + Start the App again
* Defaults for HTTP UL/DL in QTS-App cannot be changed
  + Defaults for QTS-Portal can be changed

# Buglist

List of known bugs including estimated severity

## Major

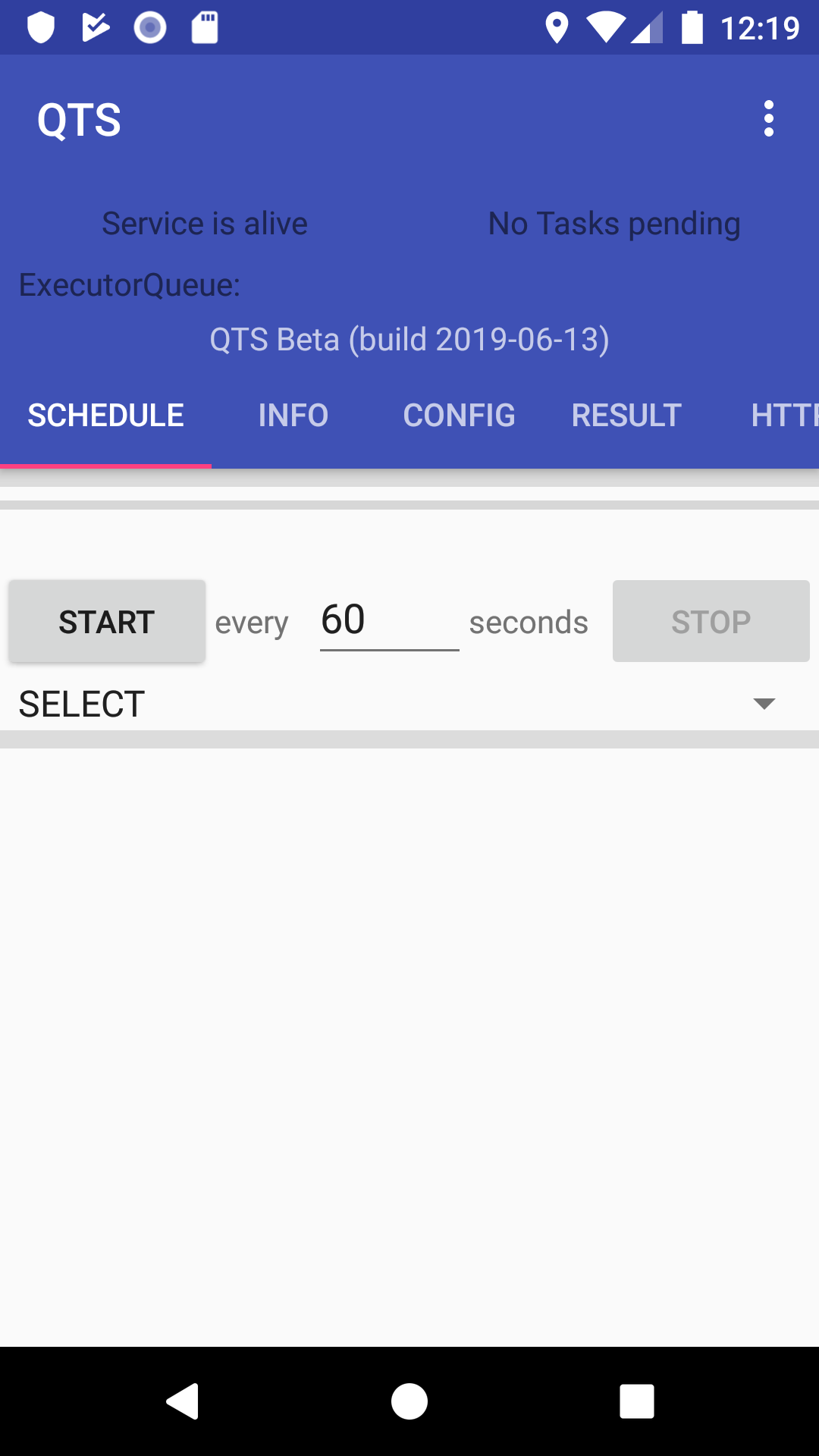
No known bugs preventing app usage

## Minor

Bugs representing unwanted behavior that are not greatly affecting the usability

* on first startup after installation when asking for permissions the app may crash
  + restarting the app solves the issue
* Testcase for Voice testing has been implemented 50%, but is not stable
  + TestCase unavailable in current App, as it is not vital
* There is currently no separate Service availability check on the Web Testcase
  + SNA is done implicitly during execution
  + Events for SNA are there
  + Separation between SNA/Transaction Phase unclear
* YouTube Testcase has different definition of playback for App/Portal
  + App: Start/Stop timestamp
  + Portal: playback duration
* Single/rare crashes of various App components
  + Mostly the App will restart all crashed components automatically
  + Minor data loss of results possible (single events)

# Upper Area



This area contains general information about the app such as App-Version, build date and what the services in the Background of the app are doing currently.

## Servicestatus

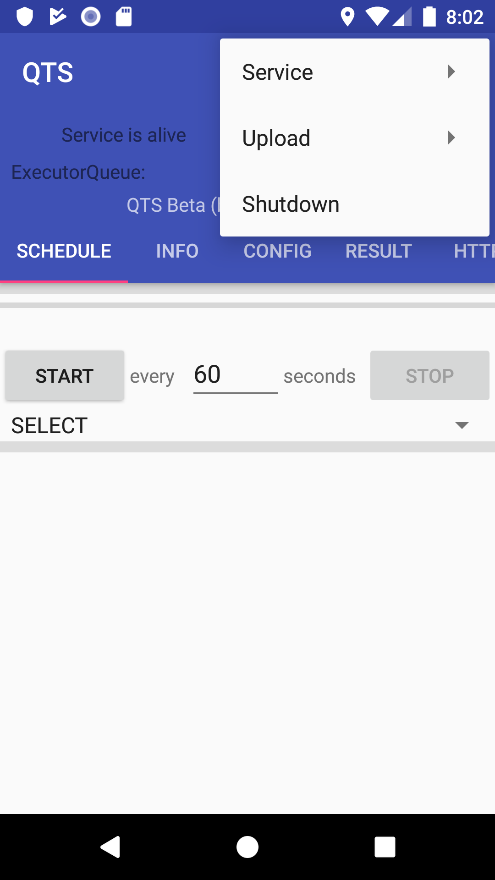
Either “Service is Alive” or “Service is Dead”, depending on if a heartbeat has been received in the last 3 seconds. This will always be “Service is Dead” upon start of the app and getting the app back to the front, as the first heartbeat has not been received

## ExecutorQueue

The number of pending tasks is displayed at the top, under this there is a list of tasks to be done from next on the left to last on the right. Tasks here include Testcases, but also system tasks like uploading Reports, fetching Tasks etc.

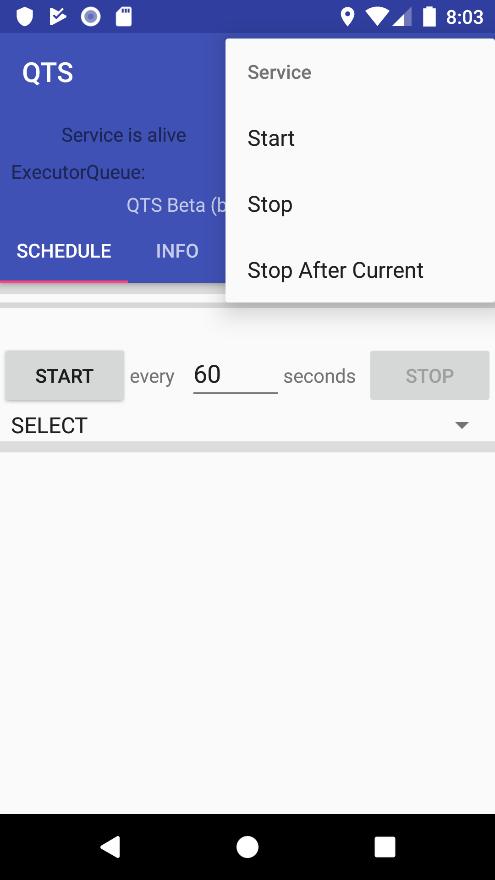
# Quick Menu

## General

The quick menu can ban accessed with the Symbol showing Three Dots in the top right corner of the app.

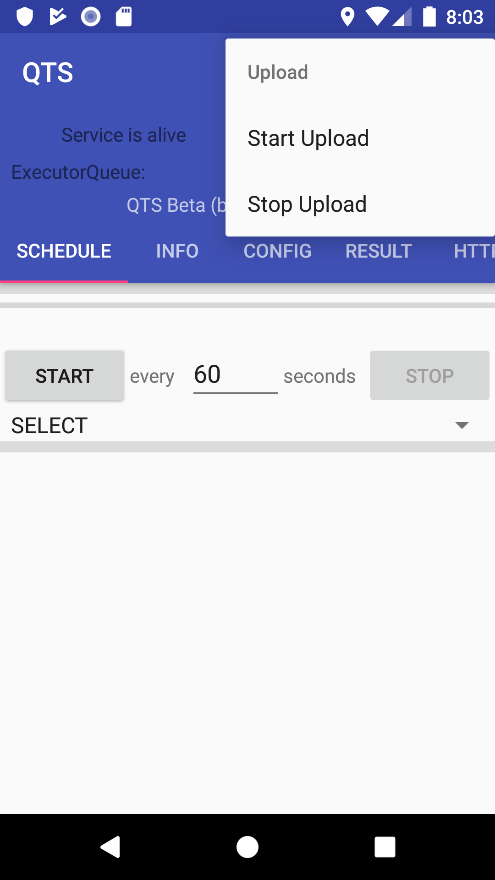
The Menu allows direct shutdown of the App and the background Service and further Control of the Service & Manual File Upload.

## Service

This sub-Menu allows starting and stopping of the Background Service without affecting the app itself.

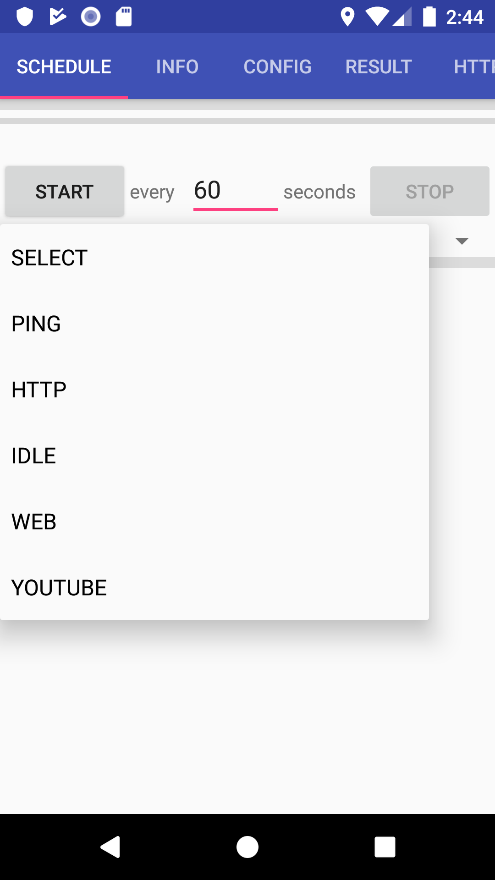
“Stop After Current” will stop the Background Service after the current Execution Queue has been processed.

## Upload

Manual file Upload may be started with “Start Upload”. This causes the Background Service to HALT all executions but uploading the Test Results. Each Upload-Task will try to Upload up to five Test Results.

By “Stop Upload” the Upload Mode will end after the current Upload Task has been completed. Normal Test Execution will continue afterwards.

# General TABS

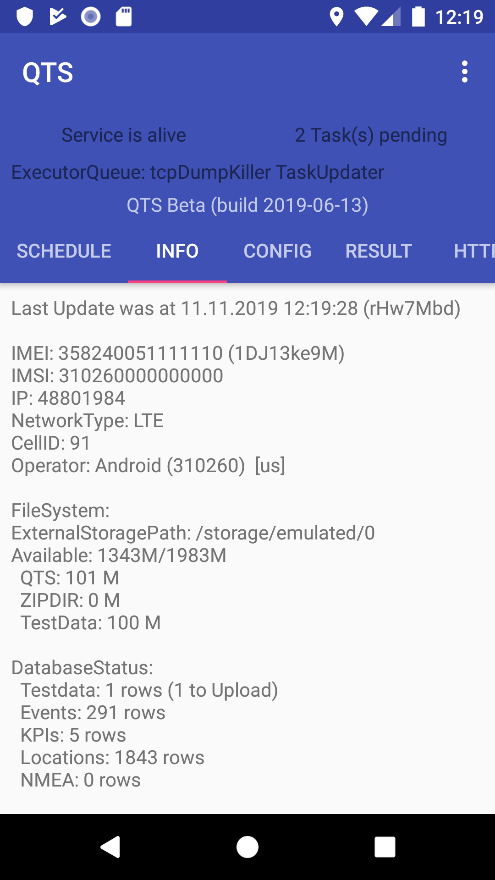
These Tabs contain general Information and Tasks not linked to a specific Testcase.

## TAB Schedule

This allows automatic start of a new test in a fixed interval. After a test is selected the parameters can be configured and a test can be scheduled for automatic execution.

Scheduling will end if the app is not in the foreground anymore or if the tab is changed.

If the tests execution time is longer than the scheduling interval the next test will be added to the execution Queue, so the Queue might get longer over time.



## TAB Info

Contains information about the Device, including when the information was last updated. For the Timestamp and the IMEI the Values are also converted to the short format which will be used when saving files to the Filesystem.

## TAB Config

This tab allows manipulation of some of the App parameters for testing.

* RELOAD resets the config to the last saved state or defaults, depending on availability
* UPDATE reads the values from the GUI and uses the changed parameters, but does not save them to the filesystem. Thus the values will not be kept between restarts
* SAVE reads the parameters from the GUI and writes them to a properties File located in the QTS-Directory

## TAB Result

This tab shows information on the last executed tests, including:

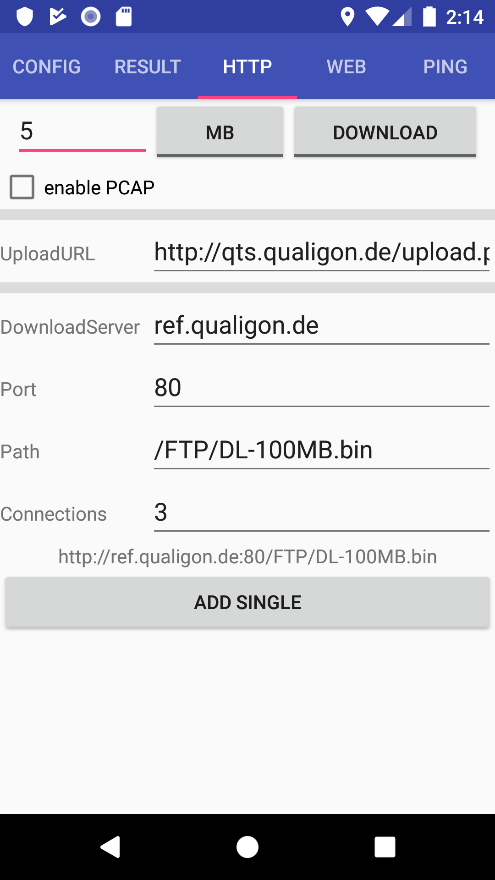
* the name used for saving
* the test type
* the time the Test started
* the last reported Event
* all KPIs

# TABs per Testcase

The Tab per testcase usually allows enabling/disabling of pcap and adding a single execution to the end of the Execution Queue. If no execution is pending currently execution will start immediately. Adding multiple executions in quick succession is valid.

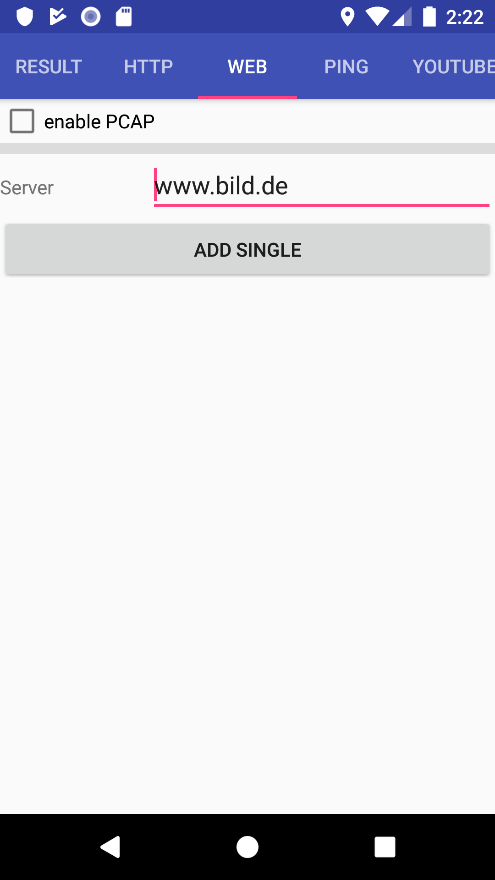
## http

This test performs a simple http upload or download with multiple connections in parallel. This is intended to test the available bandwidth.

* Set the limit (size/time)
* Set the direction of transfer (upload/download)
* Set the URL to upload to
* Set the parameters for the Download one by one
  + Base URL
  + Port
  + Path
* Set the number of parallel connections to use

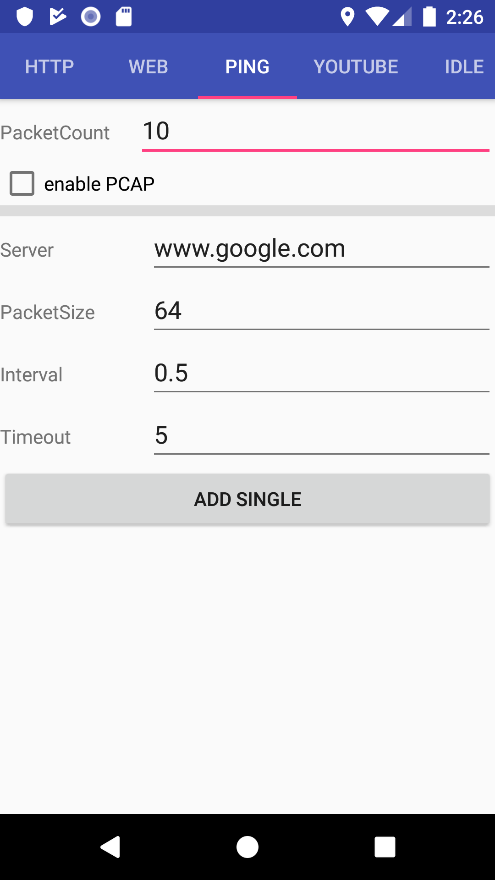
## Web

This test renders a website in the background and takes a screenshot on completion.

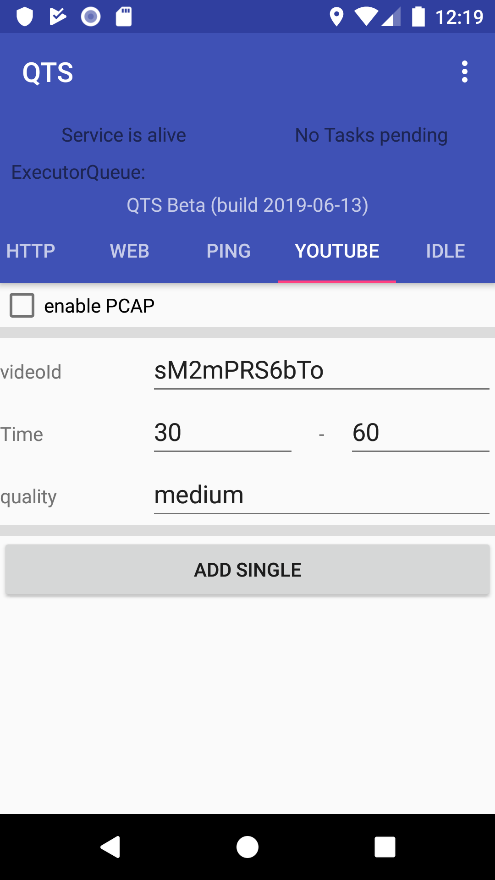
* set the URL to load

## Ping

This test is intended to test if a server can be reached

* number of ping-packets to send
* Server (hostname or IP) to ping
* Size per packet
* Interval between packets
* Timeout per ping

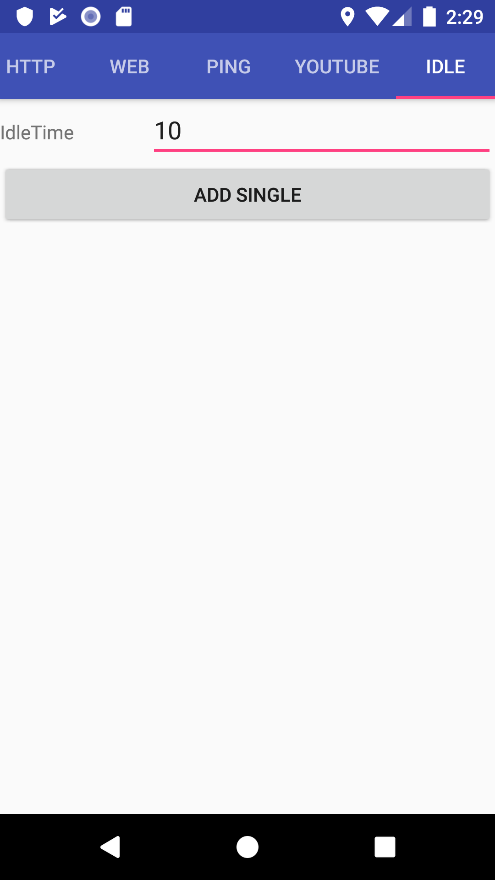
## YouTube

This test tries to play a YouTube-video in the background. Audio and Video will not be visible, for verification a screenshot of the videos preview will be taken.

* videoId to play
* start and end of playback
* requested quality (might be overridden by YouTube API)

## Idle

This is a small test to be performed between the “real” tests. A small waitingtime in between assures the last test has completely ended.

* time to do nothing

# Changelog

2019-11-18 Marcel Grotmann added list of known bugs, Chapter Setup

2019-11-13 Marcel Grotmann Initial release