

serezhka /
java-airplay-lib

<> Code

Issues 6

Pull requests

Actions

Projects

Security

Insights

This library is intended to easily create AirPlay2 servers acting like Apple TV

MIT license

183 stars 38 forks 21 watching Branches Activity Tags

Public repository

3 Branches

6 Tags

Go to file

Go to file

+

Add file

Code

serezhka

Update README.md

0269ad0 · 2 years ago

.github/workflows	publish package attempt	2 years ago
src	rtsp audio stream info	2 years ago
.gitignore	publish package attempt	2 years ago
LICENSE	Initial commit	5 years ago
README.md	Update README.md	2 years ago
build.gradle	publish package attempt	2 years ago
jitpack.yml	Initial commit	5 years ago
settings.gradle	fatJar gradle task	5 years ago

README

MIT license

java-airplay-lib

build

no status

JitPack

1.0.5

views

7980

license

MIT

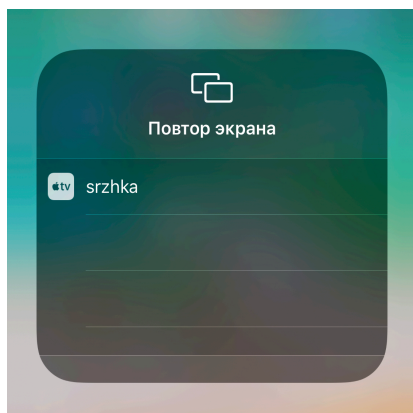
14.12.2022: Check out new java-airplay project: <https://github.com/serezhka/java-airplay>

This library is intended to easily create AirPlay2 servers acting like Apple TV. Tested with iPhone X (iOS 14.0.1)

How to use?

- Add java-airplay-lib [dependency](#) to your project
- Make your server discoverable by [Bonjour](#)

```
String serverName = "@srzhka";
int airPlayPort = 5001;
int airTunesPort = 7001;
AirPlayBonjour airPlayBonjour = new AirPlayBonjour(serverName);
airPlayBonjour.start(airPlayPort, airTunesPort);
...
airPlayBonjour.stop();
```



- Listen airTunesPort and handle RTSP requests. Pass request content bytes to the library and respond with provided content bytes.

RTSP GET | POST



```
String uri = ...
byte[] requestContent = ...
switch (uri) {
    case "/info": {
        airPlay.info(.. byte output stream ..);
        // RTSP OK + provided bytes
    }
    case "/pair-setup": {
        airPlay.pairSetup(.. byte output stream ..);
        // RTSP OK + provided bytes
    }
    case "/pair-verify": {
        airPlay.pairVerify(.. requestContent input stream ..,
            .. byte output stream ..);
        // RTSP OK + provided bytes
    }
    case "/fp-setup": {
        airPlay.fairPlaySetup(.. requestContent input stream ..,
            .. byte output stream ..);
        // RTSP OK + provided bytes
    }
    case "/feedback": {
        // RTSP OK
    }
}
```

RTSP SETUP

```
airPlay.rtspSetup(.. requestContent input stream ..,
    .. byte output stream .., int videoDataPort, int videoEventPort,
    int videoTimingPort, int audioDataPort, int audioControlPort);
// RTSP OK + provided bytes

if (airPlay.isFairPlayVideoDecryptorReady()) {
    // start listening video data on videoDataPort
}
```

```
if (airPlay.isFairPlayAudioDecryptorReady()) {
    // start listening audio data on audioDataPort
}
```

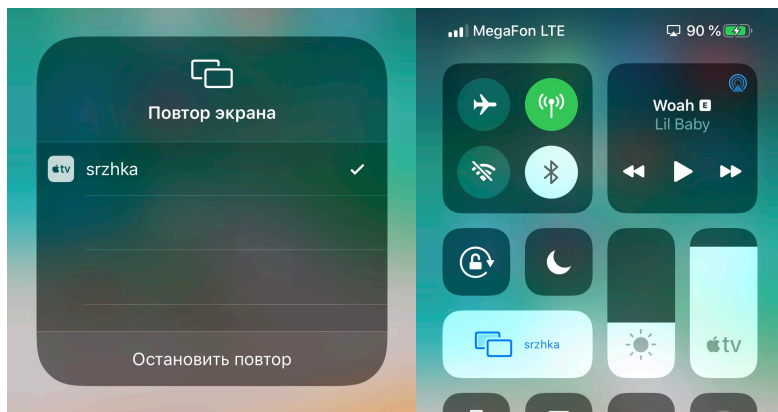
RTSP GET_PARAMETER, RECORD, SET_PARAMETER, TEARDOWN

...

DECRYPT MIRROR DATA

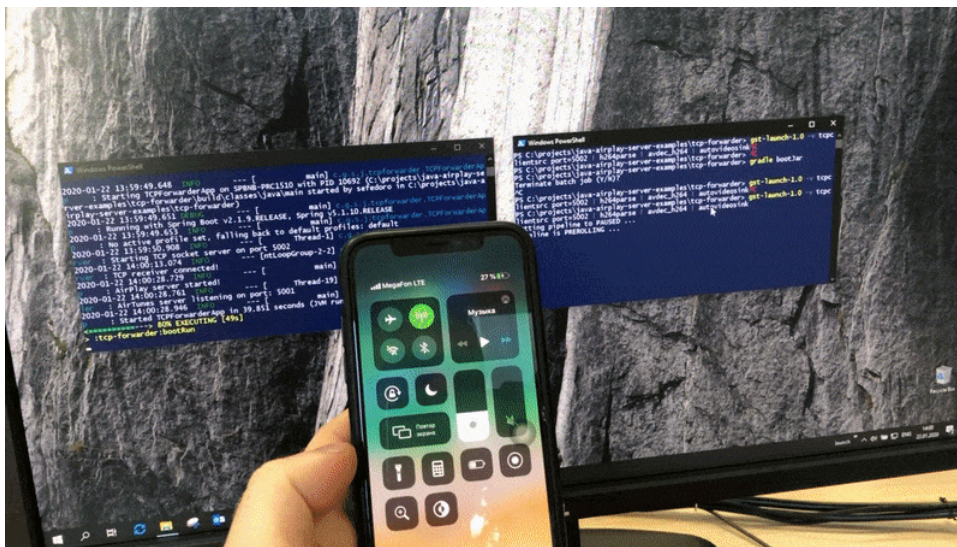
```
airPlay.decryptVideo(byte[] video);

airPlay.decryptAudio(byte[] audio, int audioLength);
```



Example server

[java-airplay-server](#) with Netty



Links


[Analysis of AirPlay2 Technology](#)

Info

Inspired by many other open source projects analyzing AirPlay2 protocol. Special thanks to OmgHax.c's author



Releases 6

 **java-airplay-lib:1.0.5** Latest
on Jun 9, 2022

[+ 5 releases](#)

Packages 1

 **com.github.serezhka.java-airplay-lib**

Languages

● Java 100.0%