



FMT Reversi Android

by
[Francesco Benincasa](#)



Roadmap

- ▶ Intro
- ▶ The project
- ▶ Demo
- ▶ Let's dive into the code
- ▶ Conclusions



Introduction

- The aim of the project is to realize a Reversi implementation on Android platform:
 - Match on the same device
 - Match between two devices (each player on his own device)
- The game rules are based on [Reversi on Wikipedia](#) and [Federazione Nazionale Gioco Othello](#)



The project - what did we use?



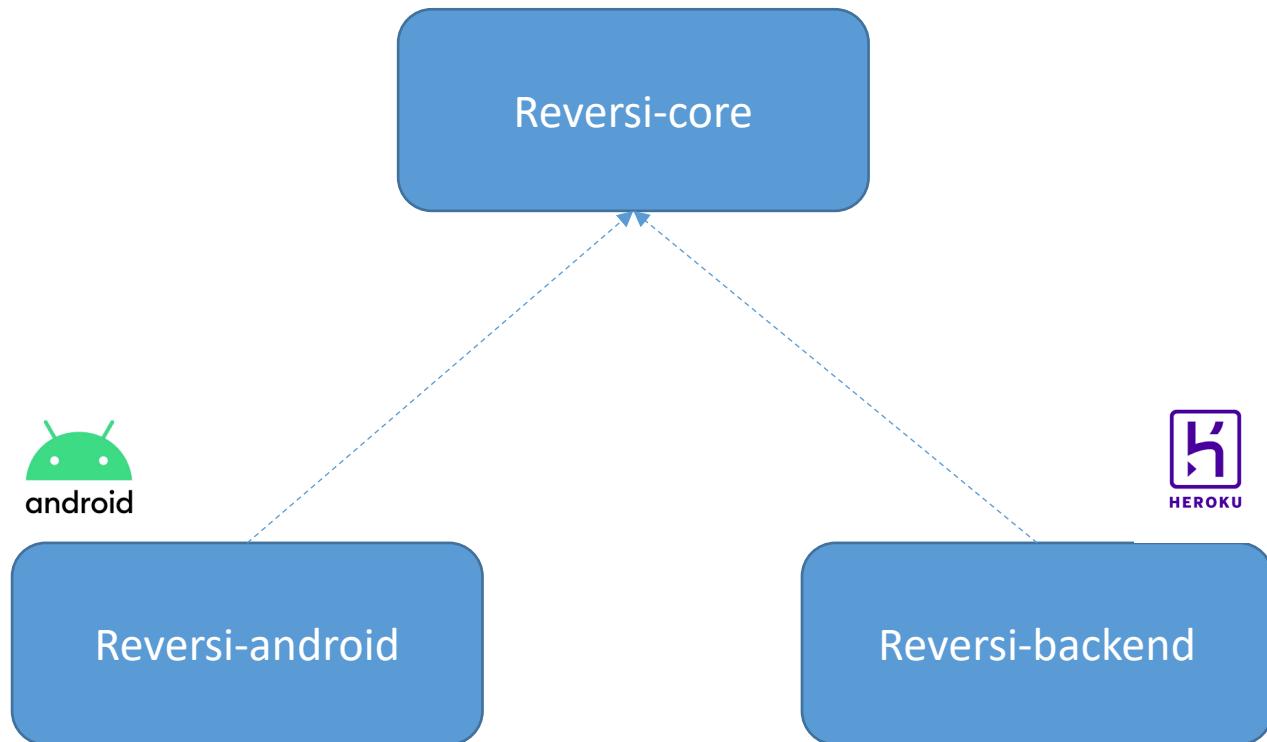
Maven[™]



ngrok



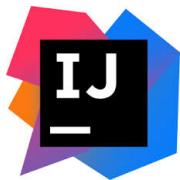
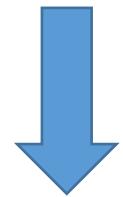
The project - modules





Build process

Android development



Backend development

A yellow downward-pointing arrow indicating the flow from Backend development to the ngrok URL.

ngrok

<https://836c360b507d.ngrok.io>



<https://fmt-reversi.herokuapp.com/>

Backend development

NGROK was used to expose local server on the web during development

The image shows a developer's workspace with two main components:

- Java IDE (Left):** A screenshot of an IDE (IntelliJ IDEA) showing the code for `WebPathConstants.java`. The code defines several static final strings for URLs related to user moves, ready status, and match topics.
- ngrok Interface (Right):** A screenshot of the ngrok web interface. The URL `https://katesapp.ngrok.io` is displayed in the browser. The page content reads "Welcome to Kate's Site! It's currently under development...". Below this, a command prompt shows the user running `./ngrok http 3000` and connecting via ngrok. To the right, session details are listed:
 - Session Status:** online
 - Account:** Kate Libby (Plan: Pro)
 - Web Interface:** <http://127.0.0.1:4040>
 - Forwarding:** <http://katesapp.ngrok.io> -> localhost:3000
 - Forwarding:** <https://katesapp.ngrok.io> -> localhost:3000

Backend development

To deploy backend server, github, travis and heroku were used

The image displays three screenshots illustrating the deployment process for the `xcesco/reversi-backend` repository:

- Github Repository View:** Shows the repository structure with files like `.mvn/wrapper`, `docs`, `src`, `.gitignore`, `.travis.yml`, `HELP.md`, `LICENSE`, `README.md`, `mvnw`, `mvnw.cmd`, `pom.xml`, `reversi-backend.iml`, and `system.properties`. It also shows a recent commit by `xcesco` and a Travis CI build status.
- Heroku App Overview:** Shows the Heroku application dashboard for `xcesco/reversi-backend`. It indicates the app is running on the `master` branch. The "Latest activity" section shows deployment logs from `xcesco@gmail.com` on July 3, 2016, at various times, all of which succeeded.
- Travis CI Build Log:** Provides a detailed view of the Travis CI build log for the `master` branch. The log shows a successful build (#50) that passed, ran for 5 minutes and 39 seconds, and was completed a day ago. It lists commits, branches, and the developer involved.



Android development

- ▶ View model & live data
- ▶ Shared preferences
- ▶ Navigation
- ▶ Intent
- ▶ Retrofit
- ▶ Websocket & STOMP
- ▶ Recycler View
- ▶ Jackson
- ▶ RX java
- ▶ Timber
- ▶ Firebase
- ▶ Dagger2



websocket navigation timber model_view rx java live_data firebase jackson intent dagger2 view retrofit recycler-view room view_binding



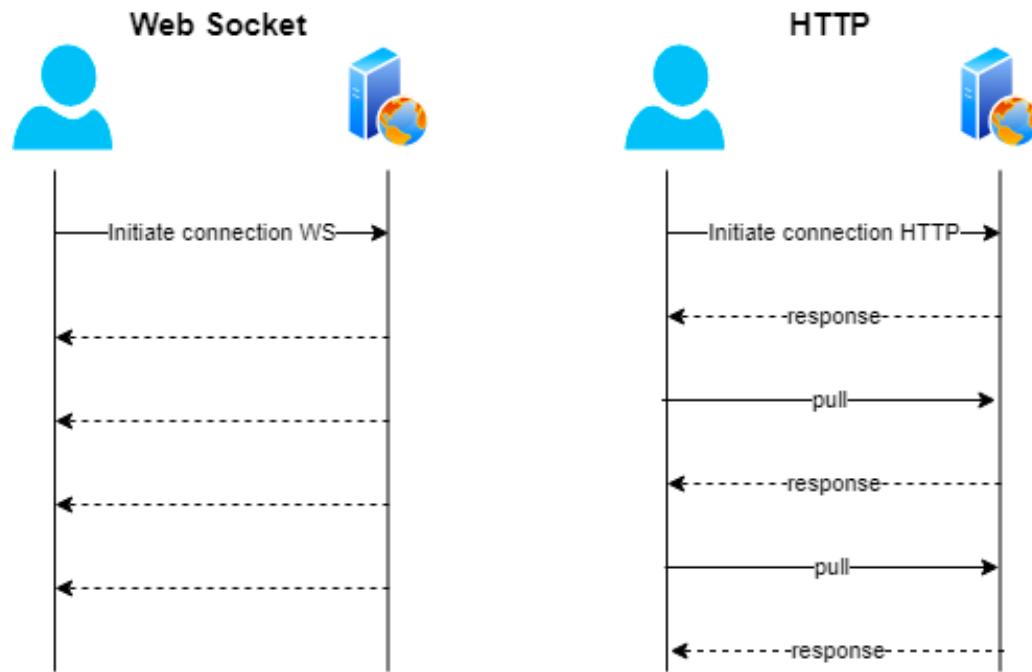
About Android development

- ▶ Retrofit: <https://square.github.io/retrofit/>
- ▶ Okhttp: <https://square.github.io/okhttp/>
- ▶ Websocket & STOMP protocol: <https://github.com/NaikSoftware/StompProtocolAndroid>
- ▶ Jackson: <https://github.com/FasterXML/jackson>
- ▶ ViewBinding: <https://developer.android.com/topic/libraries/view-binding#java>
- ▶ Dagger: <https://github.com/google/dagger>
- ▶ Navigation component: <https://developer.android.com/guide/navigation/navigation-pass-data>
- ▶ ModelView & Live data components:
<https://developer.android.com/topic/libraries/architecture/livedata>
- ▶ Timber: <https://github.com/JakeWharton/timber>
- ▶ Firebase
<https://firebase.google.com/docs/crashlytics/get-started?authuser=0&platform=android>

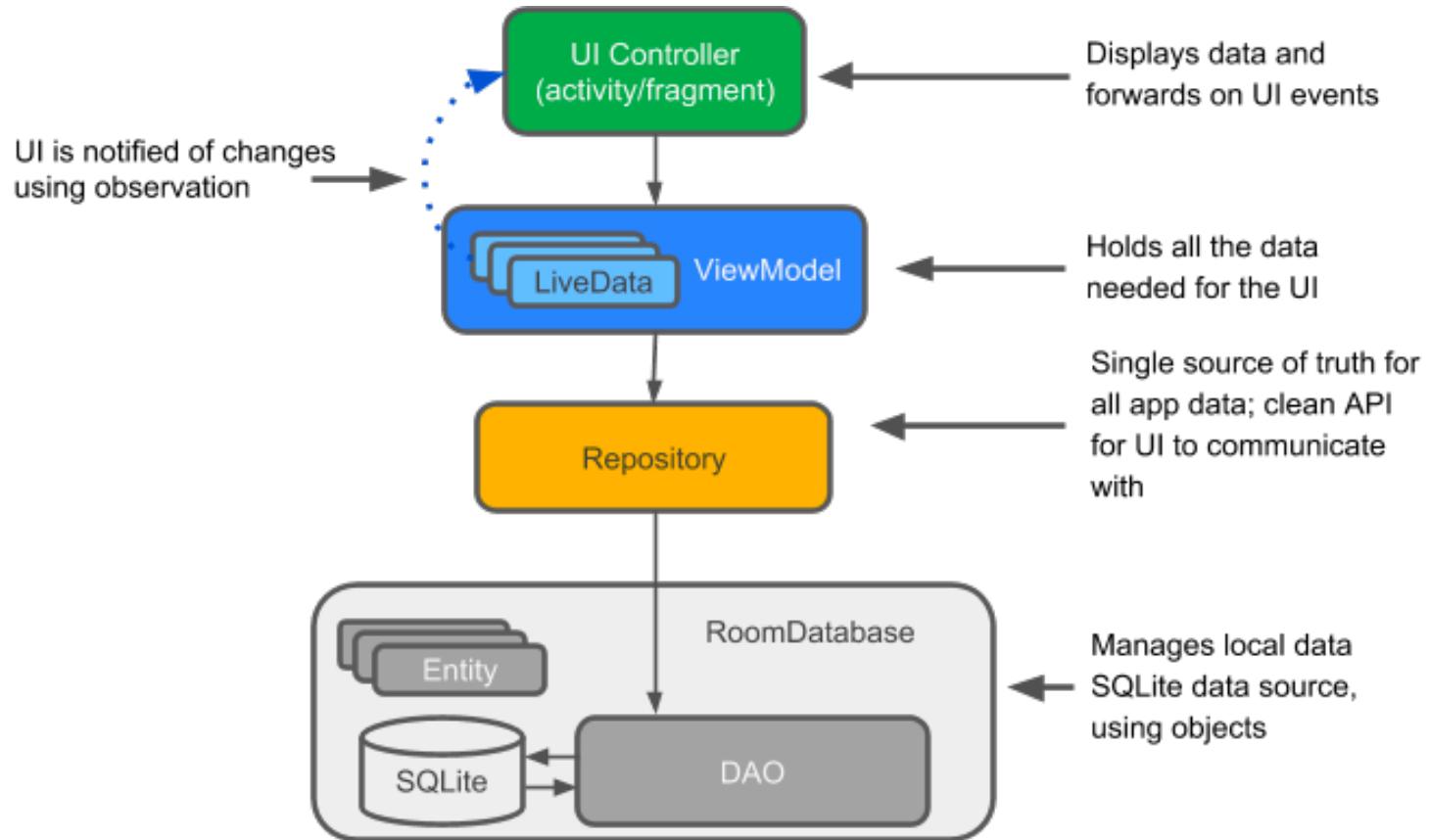


Why WebSocket?

- ▶ Game server contains match status
- ▶ Network matches required full duplex communication between clients and server.
- ▶ HTTP is based on pull approach (client calls server).
- ▶ Network game communication required fast full duplex communication (no create every time a new connection).
- ▶ The choice was Simple (or Streaming) Text Oriented Message Protocol (STOMP) on WebSocket
- ▶ In WebSocket communication, client and server communicate in async way

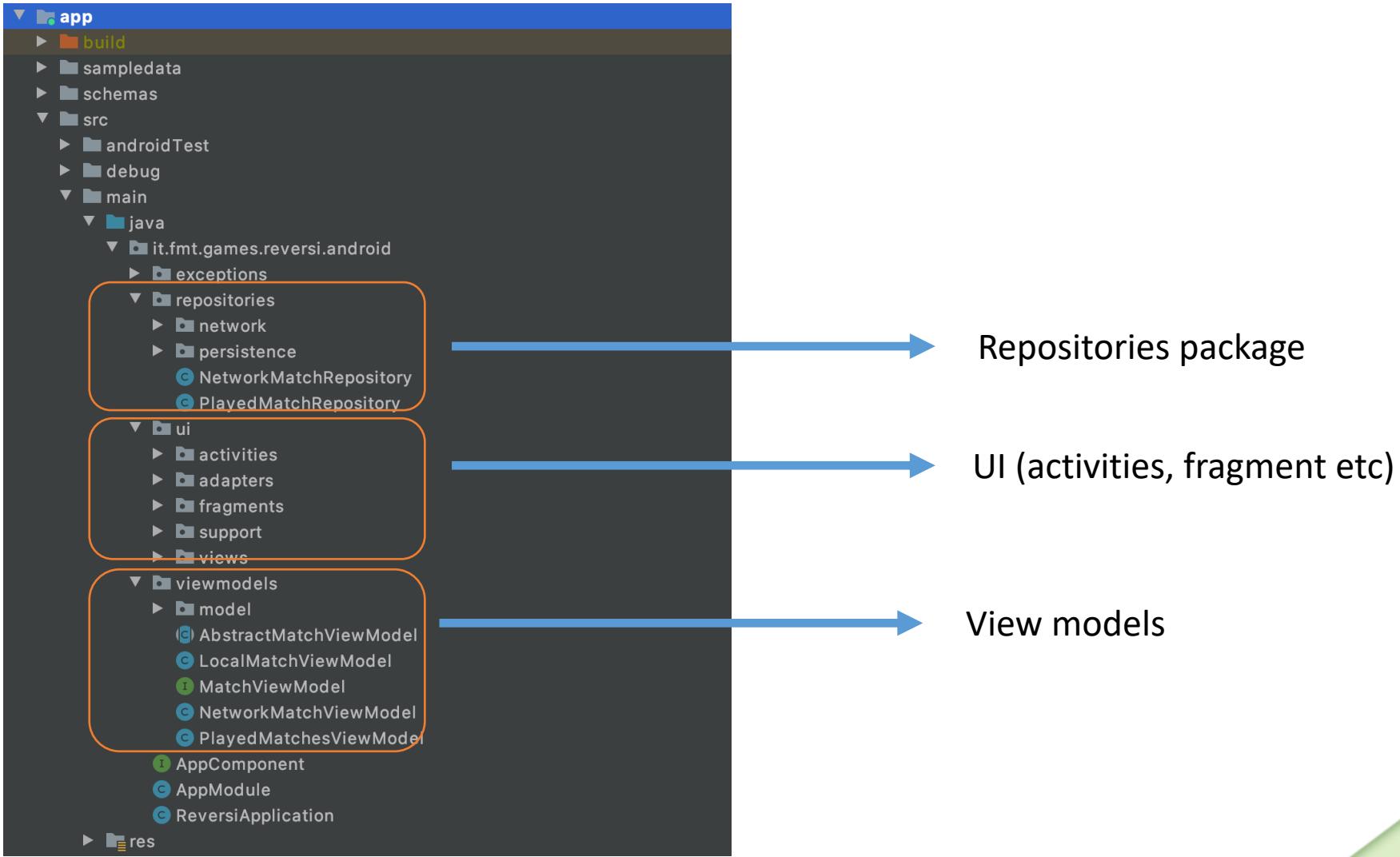


Why Live Data and View Model?

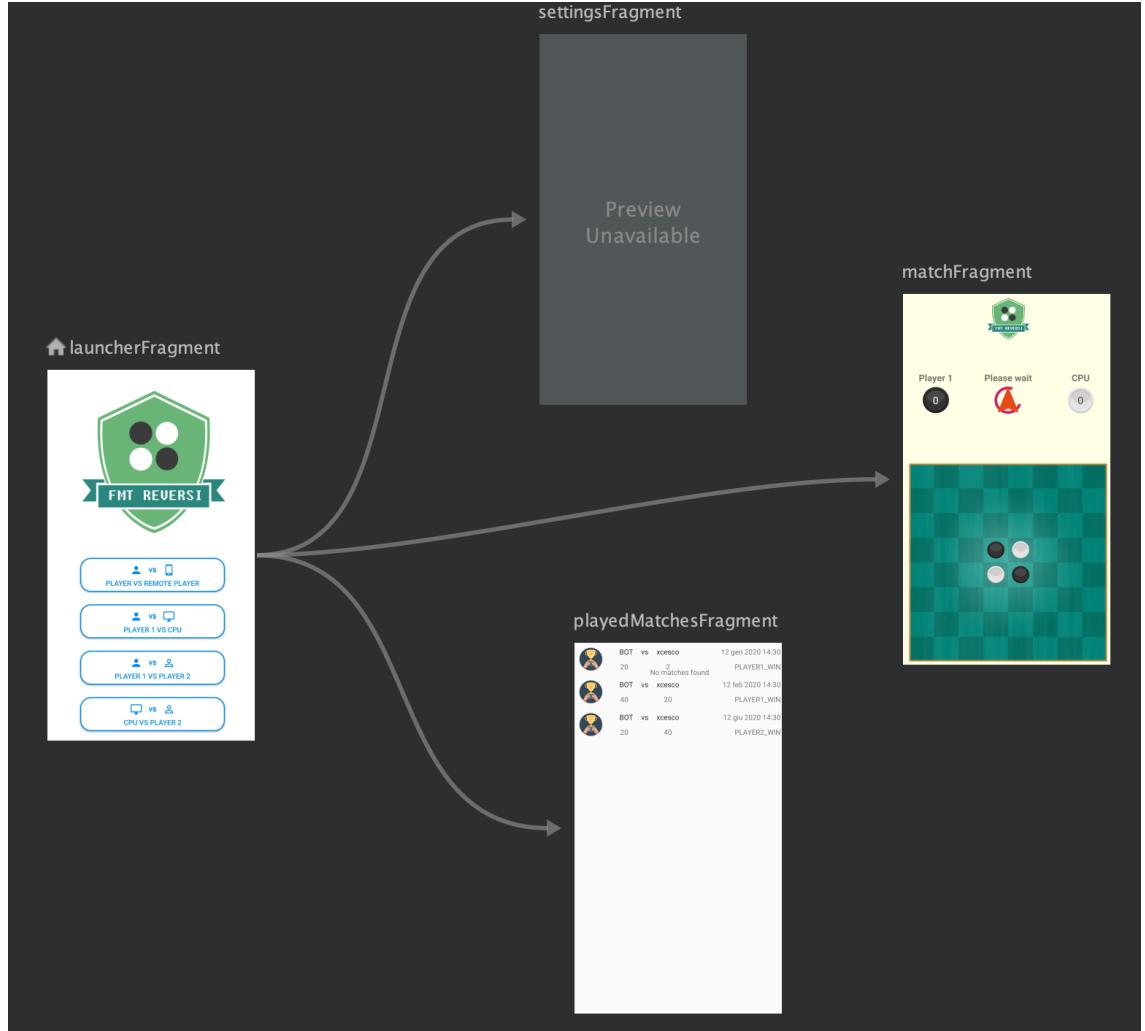


- ▶ Live data and view models allow to manage model changes on UI respecting UI components lifecycle

Source code organization

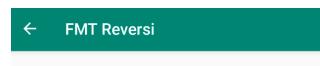


Navigation between fragments



1 activity, 4 fragments, 1 navigation graph

Navigation between fragments



Network
Preferences for network matches

Network player name
Cesco

Local
Preference for local match

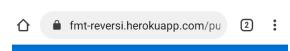
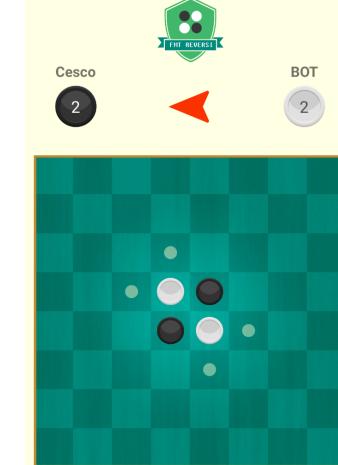
Player 1 name
Player 1

Player 2 name
Player 2

CPU Type
Random choice



	Cesco	vs	BOT	06 lug 2020 12:04
39	25			PLAYER1 WIN
	BOT	vs	Cesco	06 lug 2020 11:58
29	35			PLAYER2 WIN
	Cesco	vs	BOT	06 lug 2020 11:50
37	27			PLAYER1 WIN
	BOT	vs	Cesco	05 lug 2020 02:40
10	0			PLAYER1 WIN
	BOT	vs	Cesco	05 lug 2020 02:39
4	0			PLAYER1 WIN
	BOT	vs	Cesco	05 lug 2020 02:32
4	0			PLAYER1 WIN
	BOT	vs	Cesco	05 lug 2020 02:30
5	0			PLAYER1 WIN
	BOT	vs	Cesco	05 lug 2020 02:24
4	0			PLAYER1 WIN
	Cesco	vs	BOT	05 lug 2020 01:47
0	2			PLAYER2 WIN



FMT Reversi Server is running!
Here are some links to help you get started:
[Swagger UI](#)



Reversi FMT API
This page is created using springdocs - a library for OpenAPI 3 with spring boot.
Terms of service
Apache 2.0

Servers
<https://fmt-reversi.herokuapp.com/> - Generated server url

match-controller
GET /api/v1/public/matches
users-controller
DELETE /api/v1/public/users
GET /api/v1/public/users
GET /api/v1/public/users/{uuid}/match
PUT /api/v1/public/users/{uuid}/ready
PATCH /api/v1/public/users/{uuid}/not-ready
POST /api/v1/public/users

Demo



It's time to play!

Android version available on [Google Play Store](#)

<https://play.google.com/store/apps/details?id=it.fmt.games.reversi.android>



Network match demo on <https://youtu.be/RUfBwd1IXWg>





Source code on GitHub

- ▶ [FMT Reversi Android source code on Github](#)
- ▶ [FMT Reversi Backend source code on Github](#)
- ▶ [FMT Reversi source code on Github](#)



Conclusions

- ▶ FMT Reversi can be improved:
 - ▶ Support for other platform (web)
 - ▶ Improved IA for CPU players
 - ▶ Google play game service
 - ▶ PS4 version cooming soon!
- ▶ Any question?

Thanks!

