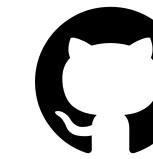


Server as a function with http4k



@dmitrykandalov



github.com/dkandalov



youtube.com/dkandalov

Your Server as a Function

Marius Eriksen

Twitter Inc.

marius@twitter.com

Abstract

Building server software in a large-scale setting, where systems exhibit a high degree of concurrency and environmental variability, is a challenging task to even the most experienced programmer. Efficiency, safety, and robustness are paramount—goals which have traditionally conflicted with modularity, reusability, and flexibility.

We describe three abstractions which combine to present a powerful programming model for building safe, modular, and efficient server software: Composable *futures* are used to relate concurrent, asynchronous actions; *services* and *filters* are specialized functions used for the modular composition of our complex server software.

Finally, we discuss our experiences using these abstractions and techniques throughout Twitter’s serving infrastructure.

Categories and Subject Descriptors D.1.1 [*Programming techniques*]: Applicative (Functional) Programming; D.1.3 [*Programming techniques*]: Concurrent Programming; D.1.3 [*Program-*

Services Systems boundaries are represented by asynchronous functions called *services*. They provide a symmetric and uniform API: the same abstraction represents both clients and servers.

Filters Application-agnostic concerns (e.g. timeouts, retries, authentication) are encapsulated by *filters* which compose to build services from multiple independent modules.

Server operations (e.g. acting on an incoming RPC or a timeout) are defined in a declarative fashion, relating the results of the (possibly many) subsequent sub-operations through the use of future combinators. Operations are phrased as *value transformations*, encouraging the use of immutable data structures and, we believe, enhancing correctness through simplicity of reasoning.

Operations describe *what* is computed; execution is handled separately. This frees the programmer from attending to the minutiae of setting up threads, ensuring pools and queues are sized cor-

Finagle

Finagle is an extensible RPC system for the JVM, used to construct high-concurrency servers. Finagle implements uniform client and server APIs for several protocols, and is designed for high performance and concurrency. Most of Finagle's code is protocol agnostic, simplifying the implementation of new protocols.

Finagle is written in Scala, but provides both Scala and Java idiomatic APIs.

- [GitHub project](#)
- [User's guide](#)
- [Developer's Guide](#)
- [API documentation](#)
- [Gitter channel](#)
- [Google group](#)
- [Blog](#)

Contributing

We feel that a welcoming community is important and we ask that you follow Twitter's [Open Source Code of Conduct](#) in all interactions with the community. Finagle is actively maintained by Twitter's infrastructure team, but we have [many external contributors](#) as well. The master branch represents the most recent published release while active development happens on the develop branch. Before endeavoring on large changes, please discuss them with the [Google groups](#) to receive feedback and suggestions. For all patches, please review our [contributing docs](#).



Search or jump to...

Pull requests Issues Marketplace Explore

Bell icon + More icon

twitter / finagle

Watch ▾

579

Unstar

7.8k

Fork

1.4k

Code

Issues 44

Pull requests 3

Actions

Security

Insights

develop ▾

6 branches

112 tags

Go to file

Add file ▾

Code ▾

About

A fault tolerant, protocol-agnostic RPC system

twitter.github.io/finagle

rpc distributed-systems
finagle http http2 thrift
mysql memcached redis
zipkin java scala

Readme

Apache-2.0 License

Releases 112

Finagle 20.10.0 Latest
19 days ago

+ 111 releases

Packages

 yufangong and jenkins	finagle/finagle-stats-core: Move	9710eec 2 days ago	7,268 commits
.github	Twitter OSS: Add ISSUE_TEMPLATE		4 years ago	
doc	finagle-core: Track offload task delays at a given rate		9 days ago	
finagle-base-http	[finagle-netty4-http] Extract test lib		2 months ago	
finagle-benchmark-thrift/src/...	Update Build files - deprecate [rz]globs (batch#9)		9 months ago	
finagle-benchmark	[finagle] declare dependencies explicitly		4 months ago	
finagle-core	finagle: Handle floating point errors when biasing for load i...		3 days ago	
finagle-example	finagle-example: Expand finagle binary helper function in ...		4 months ago	
finagle-exception	3rdparty: Upgrade scalatest to 3.1.2		4 months ago	
finagle-exp	finagle-core: Promote ThriftForwardingWarmUpFilter from ...		4 days ago	
finagle-grpc-context	[finagle-grpc-context] suppress CheckReturnValue		2 months ago	
finagle-http	Various core security ownership changes		last month	
finagle-http2	[csl targets] explicitly list out the runtime scoped depende...		3 months ago	
finagle-init	Corrected CSL team in PROJECT Files		3 months ago	

The Functional toolkit for Kotlin HTTP applications

http4k provides a simple and uniform way to serve, consume, and test HTTP services.

[Quick Start](#)

HTTP as a Function

```
val app = { request: Request ->
    Response(OK)
        .body("Hello, ${request.query("name")}!")
}
```



```
app.asServer(Jetty(9000)).start()
```

```
val client = ApacheClient()

val request = Request(GET, "http://localhost:9000")
    .query("name", "John Doe")

val response: Response = client(request)
```



Search or jump to...

Pull requests Issues Marketplace Explore

[http4k / http4k](#)[Sponsor](#)[Watch](#) 29[Unstar](#) 1.5k[Fork](#) 136[Code](#)[Issues 21](#)[Pull requests 6](#)[Actions](#)[Projects](#)[Security](#)[Insights](#)[master](#)

9 branches

13 tags

[Go to file](#)[Add file](#)[Code](#)

About

The Functional toolkit for Kotlin HTTP applications. http4k provides a simple and uniform way to serve, consume, and test HTTP services.

[http4k.org](#)

kotlin http-client typesafe
testability tdd http-server
http http4k immutability

[Readme](#)

[Apache-2.0 License](#)

Releases 13

3.276.0 Latest

2 hours ago

+ 12 releases

daviddenton [skip travis]	checkout main repo!	✓ 1f4e517 16 minutes ago	4,900 commits
.github	[skip travis] checkout main repo!	16 minutes ago	
buildSrc	upgrade versions	2 hours ago	
gradle/wrapper	upgrade libs and gradle	21 days ago	
http4k-aws	tidying kotlin after analysis	10 days ago	
http4k-bom	added BOM module	now	
http4k-client	tidying kotlin after analysis	10 days ago	
http4k-cloudnative	tidying kotlin after analysis	10 days ago	
http4k-contract	Split out core modules for format and templating modules....	2 days ago	
http4k-core	Split out core modules for format and templating modules....	2 days ago	
http4k-format	add core to test deps	23 hours ago	
http4k-graphql	Graphql module added	4 hours ago	
http4k-incubator	Rename Matched to MatchedWithoutHandler	5 days ago	
http4k-jsonrpc	Split out core modules for format and templating modules....	2 days ago	



Search or jump to...

/ Pull requests Issues Marketplace Explore

Bell + ⚙️

http4k / http4k

Sponsor

Watch ▾

29

Unstar

1.5k

Fork

136

Code

Issues 21

Pull requests 6

Actions

Projects

Security

Insights

Pulse

Contributors

Community

Commits

Code frequency

Dependency graph

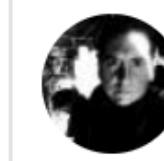
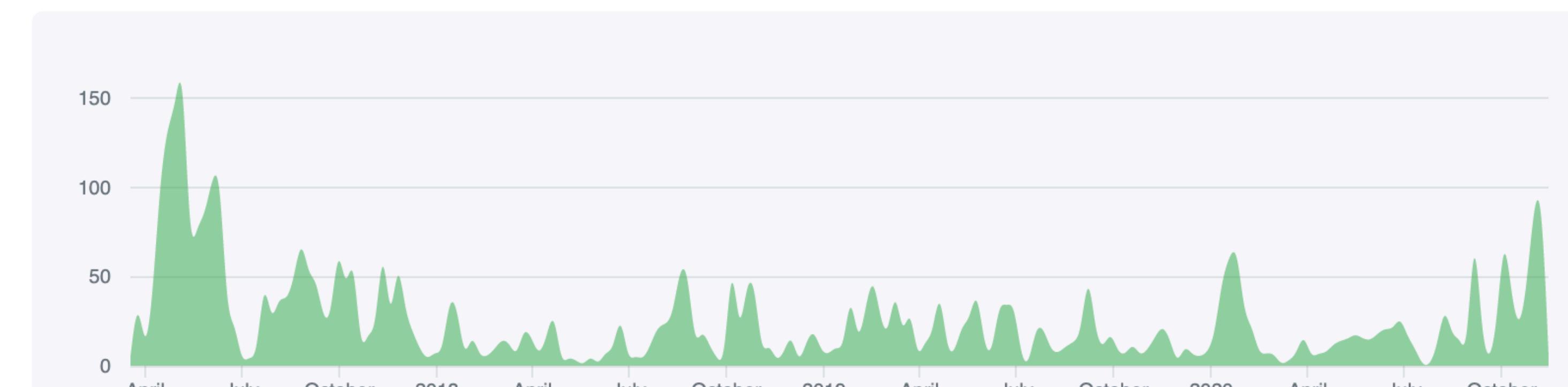
Network

Forks

Mar 19, 2017 – Nov 15, 2020

Contributions: Commits ▾

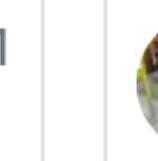
Contributions to master, excluding merge commits



daviddenton

3,985 commits 248,187 ++ 197,296 --

#1



s4nchez

656 commits 42,033 ++ 31,552 --

#2



[Main page](#)[Contents](#)[Featured content](#)[Current events](#)[Random article](#)[Donate to Wikipedia](#)[Wikipedia store](#)[Interaction](#)[Help](#)[About Wikipedia](#)[Community portal](#)[Recent changes](#)[Contact page](#)[Tools](#)[What links here](#)[Related changes](#)[Upload file](#)[Special pages](#)[Permanent link](#)[Page information](#)[Wikidata item](#)[Cite this page](#)[In other projects](#)[Wikimedia Commons](#)[Article](#) [Talk](#)[Read](#) [Edit](#) [View history](#)

Search Wikipedia



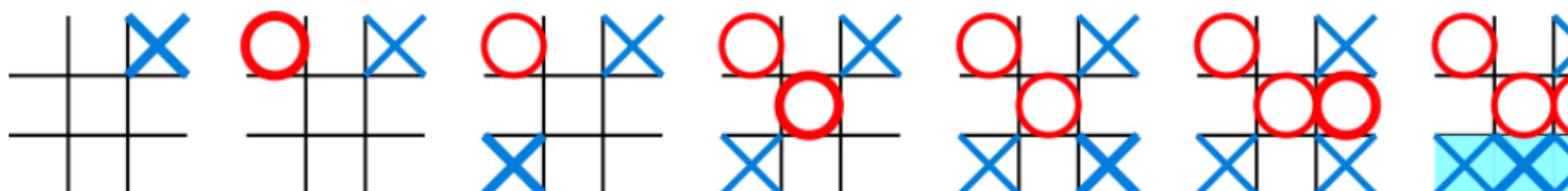
Tic-tac-toe

From Wikipedia, the free encyclopedia

"Tic Tac Toe" and *"Noughts and crosses"* redirect here. For other uses, see [Tic Tac Toe \(disambiguation\)](#) and [Noughts and crosses \(disambiguation\)](#).

Tic-tac-toe (American English), **noughts and crosses** (British English), or **Xs and Os** is a paper-and-pencil game for two players, *X* and *O*, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.

The following example game is won by the first player, *X*:



Players soon discover that the [best play](#) from both parties leads to a [draw](#). Hence, tic-tac-toe is most often played by young children, who often have not yet discovered the optimal strategy.

Because of the simplicity of tic-tac-toe, it is often used as a [pedagogical](#) tool for teaching the concepts of good [sportsmanship](#) and the branch of [artificial intelligence](#) that deals with the searching of [game trees](#). It is straightforward to write a [computer program](#) to play tic-tac-toe perfectly or to enumerate the 765 essentially different positions (the [state space complexity](#)) or the 26,830 possible games up to rotations and reflections (the [game tree complexity](#)) on this space.^[1]

The game can be generalized to an *m,n,k-game* in which two players alternate placing stones of their own color on an $m \times n$ board, with the goal of getting *k* of their own color in a row. Tic-tac-toe is the $(3, 3, 3)$ game.^[2] Heron's generalized tic-tac-toe is an even broader generalization of tic-tac-toe. It can also be

Tic-tac-toe

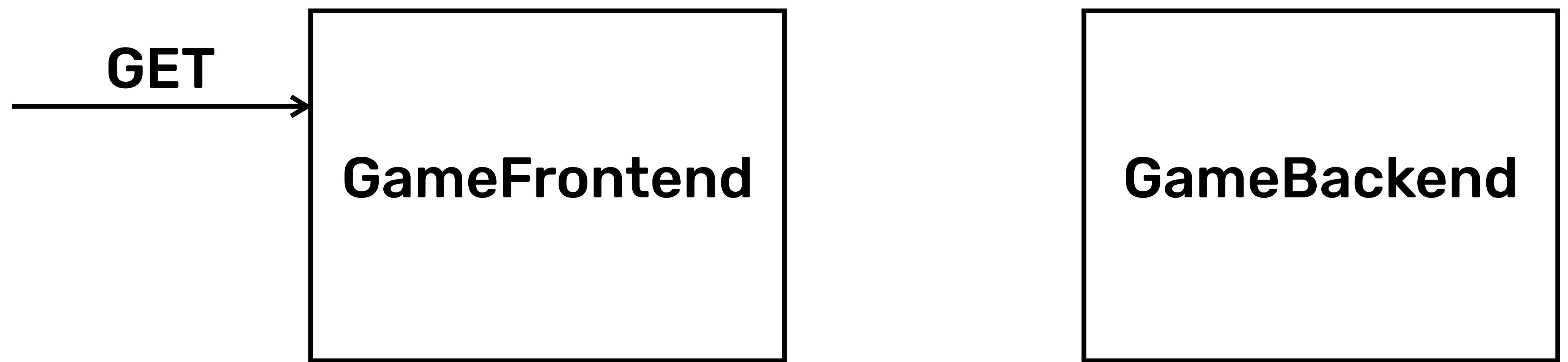


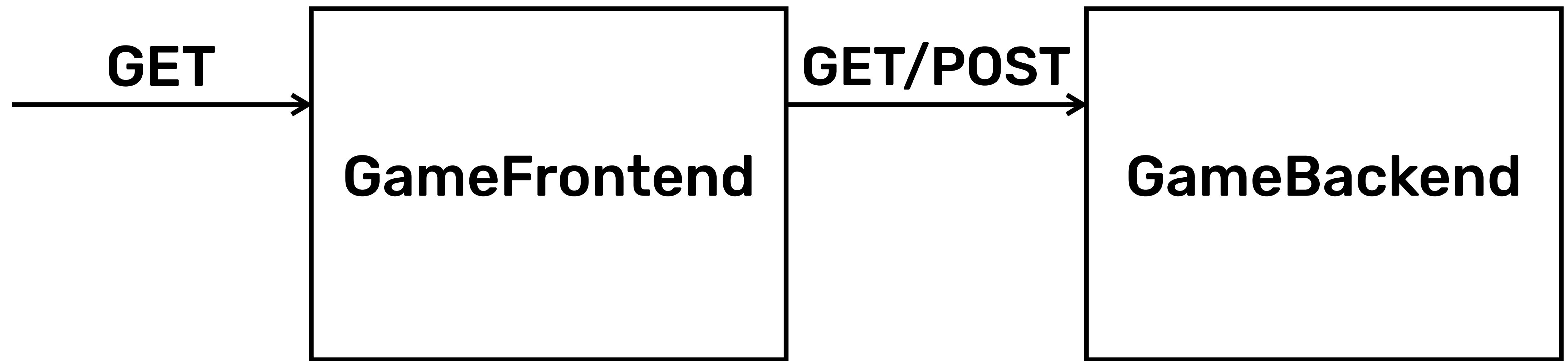
A completed game of Tic-tac-toe

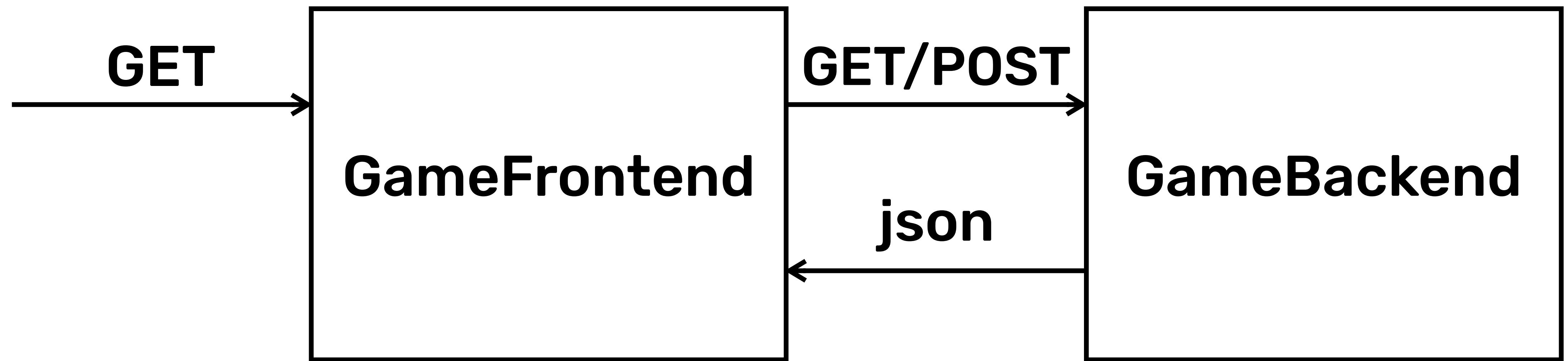
Genre(s)	Paper-and-pencil game
Players	2
Setup time	Minimal
Playing time	~1 minute
Random chance	None
Skill(s) required	Strategy , tactics, observation
Synonym(s)	Noughts and crosses Xs and Os

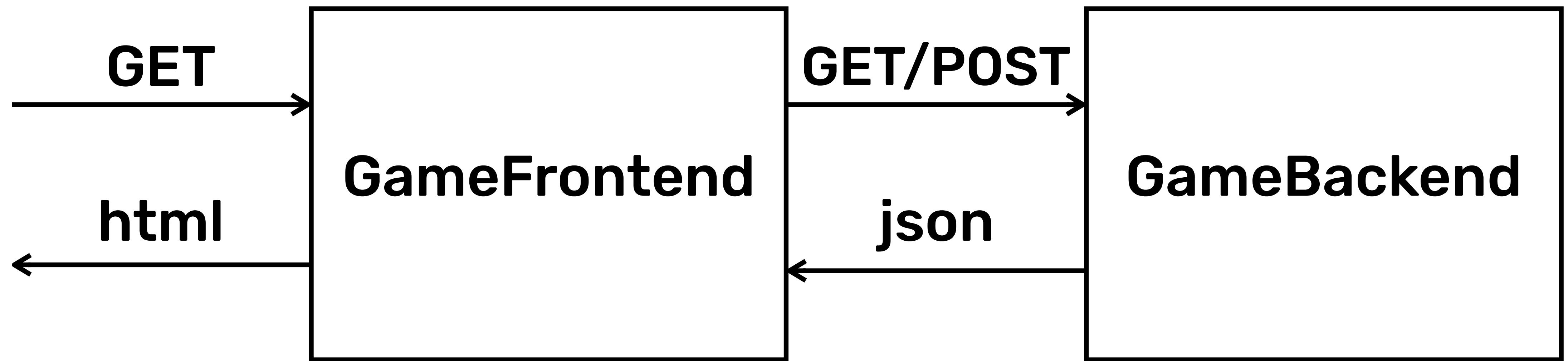
GameFrontend

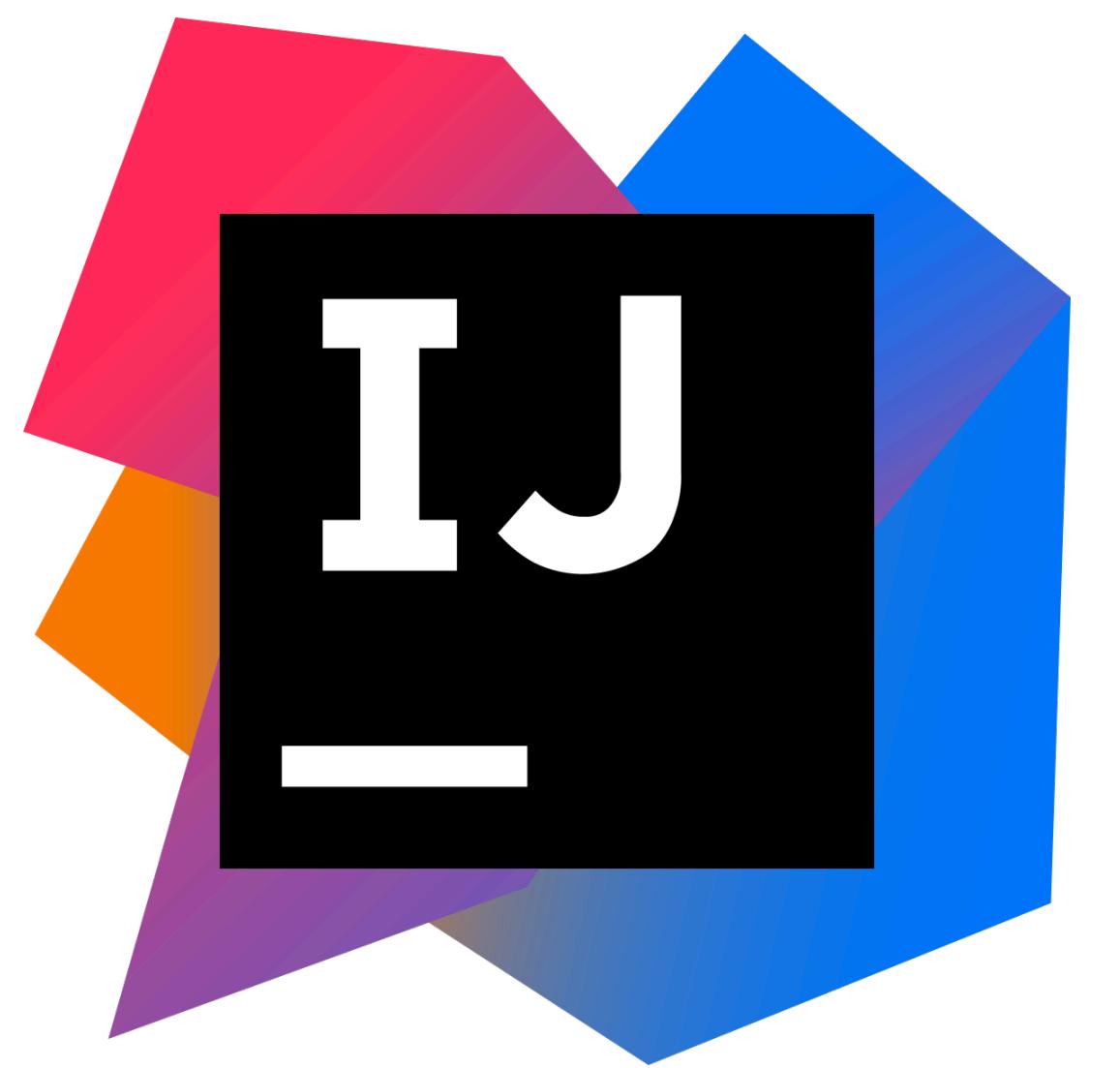
GameBackend













There are
more http4k
things...

- more Filters

- more Filters
- more Clients

- more Filters
- more Clients
- more Servers

- more Filters
- more Clients
- more Servers
- more json/xml libs

- more Filters
- more Clients
- more Servers
- more json/xml libs
- more template engines

- approval tests

- approval tests
- contract tests

- approval tests
- contract tests
- server fakes

- approval tests
- contract tests
 - server fakes
 - record/replay

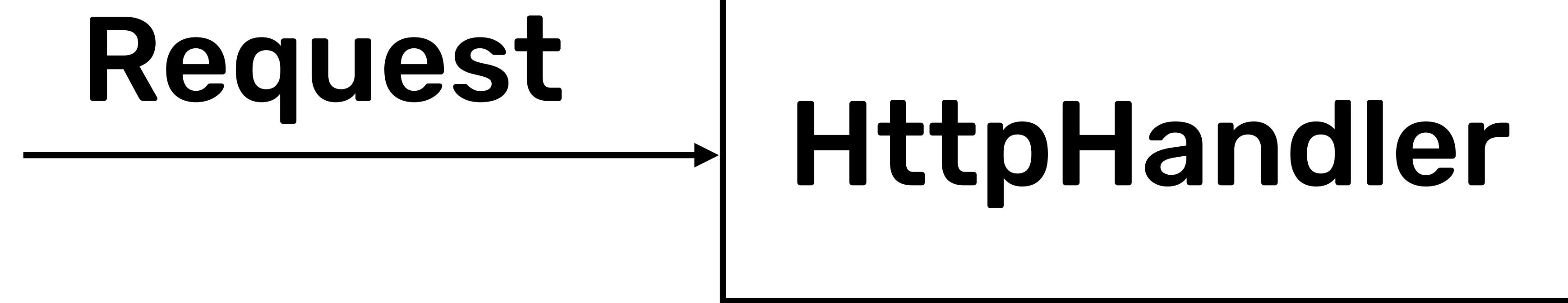
- approval tests
- contract tests
 - server fakes
 - record/replay
 - chaos testing

Summary

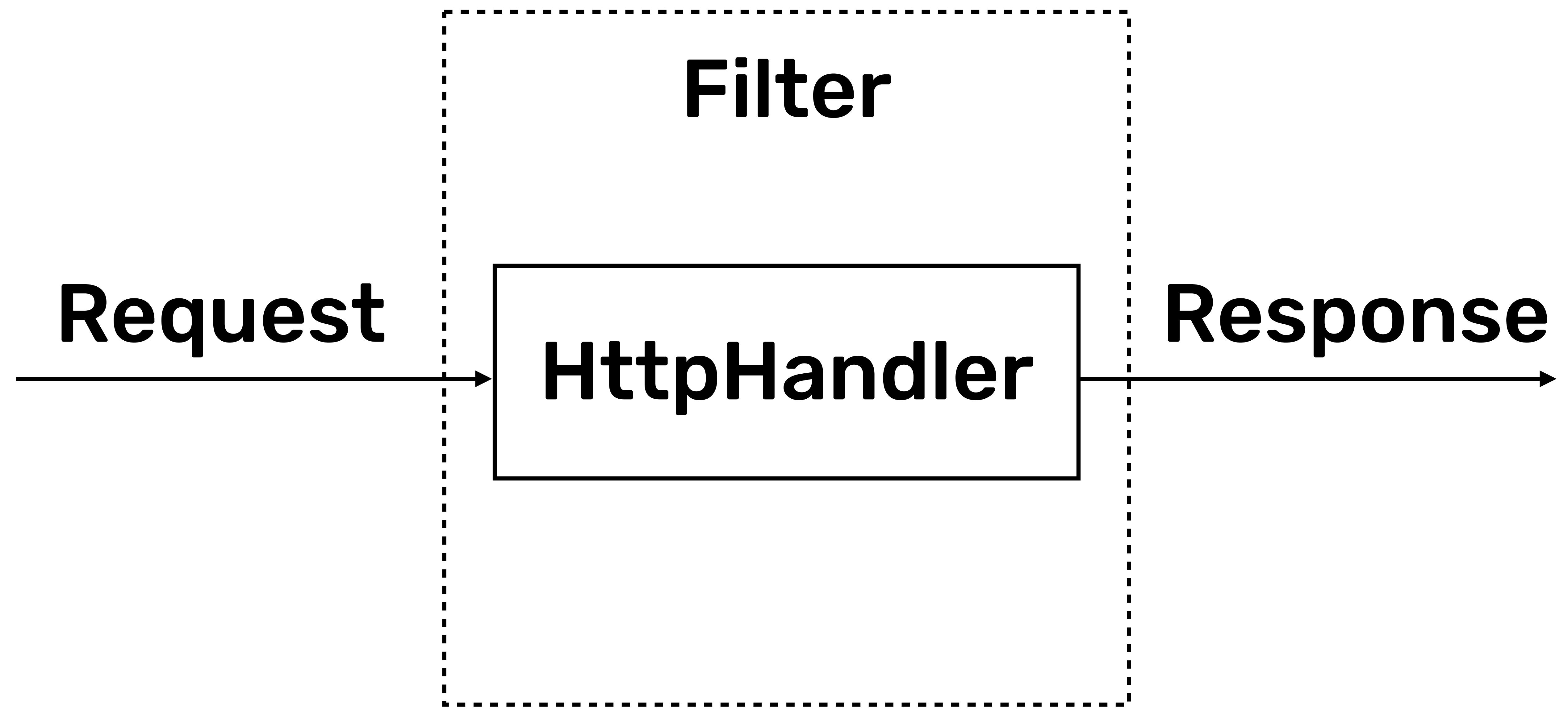
**Client and server
have the same
type signature!!**

HttpHandler

Request







- keep it simple

- keep it simple
- test everything

- keep it simple
- test everything
- libraries over frameworks

Next steps



Home



Trending



Subscriptions



Library



KotlinConf 2018 - Server as a Function in Kotlin by Ivan Sanchez & David Denton

JetBrainsTV • 2.1K views • 1 year ago

Recording brought to you by American Express <https://americanexpress.io/kotlin-jobs> In this talk, you'll learn about how we ...



KotlinConf 2019: Asynchronous Data Streams with Kotlin Flow by Roman Elizarov

JetBrainsTV • 15K views • 2 months ago

Recording brought to you by American Express. <https://americanexpress.io/kotlin-jobs> Kotlin Flow is a declarative mechanism ...



KotlinConf 2019: What's New in Java 19: The end of Kotlin? by Jake Wharton

JetBrainsTV • 111K views • 2 months ago

Recording brought to you by American Express. <https://americanexpress.io/kotlin-jobs> Kotlin's introduction was a breath of ...



Next On Track

Ivan Sanchez & David Denton

[HTTP4K](#)

Writing Test Driven Apps with http4k



The Functional toolkit for Kotlin HTTP applications

http4k provides a simple and uniform way to serve, consume, and test HTTP services.

[Quick Start](#)

HTTP as a Function

```
val app = { request: Request ->
    Response(OK)
        .body("Hello, ${request.query("name")}!")
}
```



```
app.asServer(Jetty(9000)).start()
```

```
val client = ApacheClient()

val request = Request(GET, "http://localhost:9000")
    .query("name", "John Doe")

val response: Response = client(request)
```



Search or jump to...

Pull requests Issues Marketplace Explore

[http4k / http4k](#)[Sponsor](#)[Watch](#) 29[Unstar](#) 1.5k[Fork](#) 136[Code](#)[Issues 21](#)[Pull requests 6](#)[Actions](#)[Projects](#)[Security](#)[Insights](#)[master](#)

9 branches

13 tags

[Go to file](#)[Add file](#)[Code](#)

About

The Functional toolkit for Kotlin HTTP applications. http4k provides a simple and uniform way to serve, consume, and test HTTP services.

[http4k.org](#)

kotlin http-client typesafe
testability tdd http-server
http http4k immutability

[Readme](#)

[Apache-2.0 License](#)

Releases 13

3.276.0 Latest
2 hours ago

+ 12 releases

 daviddenton [skip travis]	checkout main repo!	✓ 1f4e517 16 minutes ago	4,900 commits
 .github	[skip travis] checkout main repo!	16 minutes ago	
 buildSrc	upgrade versions	2 hours ago	
 gradle/wrapper	upgrade libs and gradle	21 days ago	
 http4k-aws	tidying kotlin after analysis	10 days ago	
 http4k-bom	added BOM module	now	
 http4k-client	tidying kotlin after analysis	10 days ago	
 http4k-cloudnative	tidying kotlin after analysis	10 days ago	
 http4k-contract	Split out core modules for format and templating modules....	2 days ago	
 http4k-core	Split out core modules for format and templating modules....	2 days ago	
 http4k-format	add core to test deps	23 hours ago	
 http4k-graphql	Graphql module added	4 hours ago	
 http4k-incubator	Rename Matched to MatchedWithoutHandler	5 days ago	
 http4k-jsonrpc	Split out core modules for format and templating modules....	2 days ago	



Jump to... < >

coroutines

datascience

🔒 duncans_place

eap

feed

fosdem

functional

general

getting-started

gradle

hiring

http4k

intelliJ

intelliJ-plugins

javascript

kontributors

kotlin-native

kotlinconf

kotlinlondon

kotlinx-files

ktor

language-proposals

london

↓ More unreads

#http4k

☆ | 213 | 2 | <http://www.http4k.org>



Search



Thursday, February 13th



Jamie Hodkinson 09:49

Hi! We are responding with an input stream from a temporary file on disk.

```
Response(OK).body(tempFile.inputStream())
```

Is there any way to detect when the response has been fully read so that we can then delete the temp file on disk?



7 replies

Last reply 12 days ago

Friday, February 14th



Mehdi Mohammadi 23:47

Could anybody help that what is wrong with this code?

```
val app = { request: Request -> Response(OK).body("Hello, ${request.query("name")!}") }

val jettyServer = app.asServer(Jetty(9006)).start()
val client = WebSocketClient.blocking(Uri.of("wss://echo.websocket.org"), headers = listOf(Pair("a
header", "value")))

client.send(WsMessage("hello"))

// read all of the messages from the socket until it is closed (by the server).
// we expect to get one message back before the stream is closed.
client.received().toList().forEach(::println)
```

I am able to connect but I am not receiving any message. I want to develop a websocket client always send and receive messages to/from the server (edited)



14 replies

Last reply 1 day ago

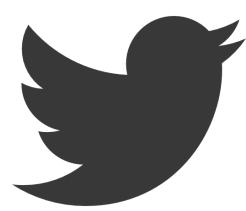


Message #http4k

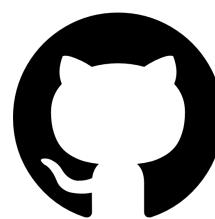
Aa @ 😊

[https://github.com/dkandalov/
tictactoe4k](https://github.com/dkandalov/tictactoe4k)

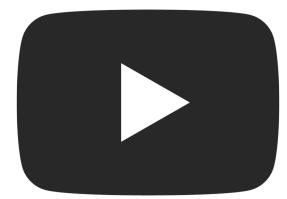
The End



@dmitrykandalov



github.com/dkandalov



youtube.com/dkandalov

