KotlinConf

Isomorphic Kotlin

Troy Miles
@therockncoder

Troy Miles
@therockncoder

Troy Miles, aka the Rockncoder, began writing computer games in assembly language for early computers like the Apple II, Commodore C64, and the IBM PC over 38 years ago. Nowadays he writes web & mobile apps for a Southern California based automotive valuation and information company. Troy is fluent in JavaScript, C#, C ++, Kotlin, and Clojure. Check out my Kotlin video:

https://www.lynda.com/Java-tutorials/Kotlin-Java-Developers/562926-2.html

He can be reached at: rockncoder@gmail.com

https://github.com/Rockncoder/rk1

The source code

KotlinConf

FuelEconomy.gov

Source of the example data

the official government source
for fuel economy information

FuelEconomy.gov

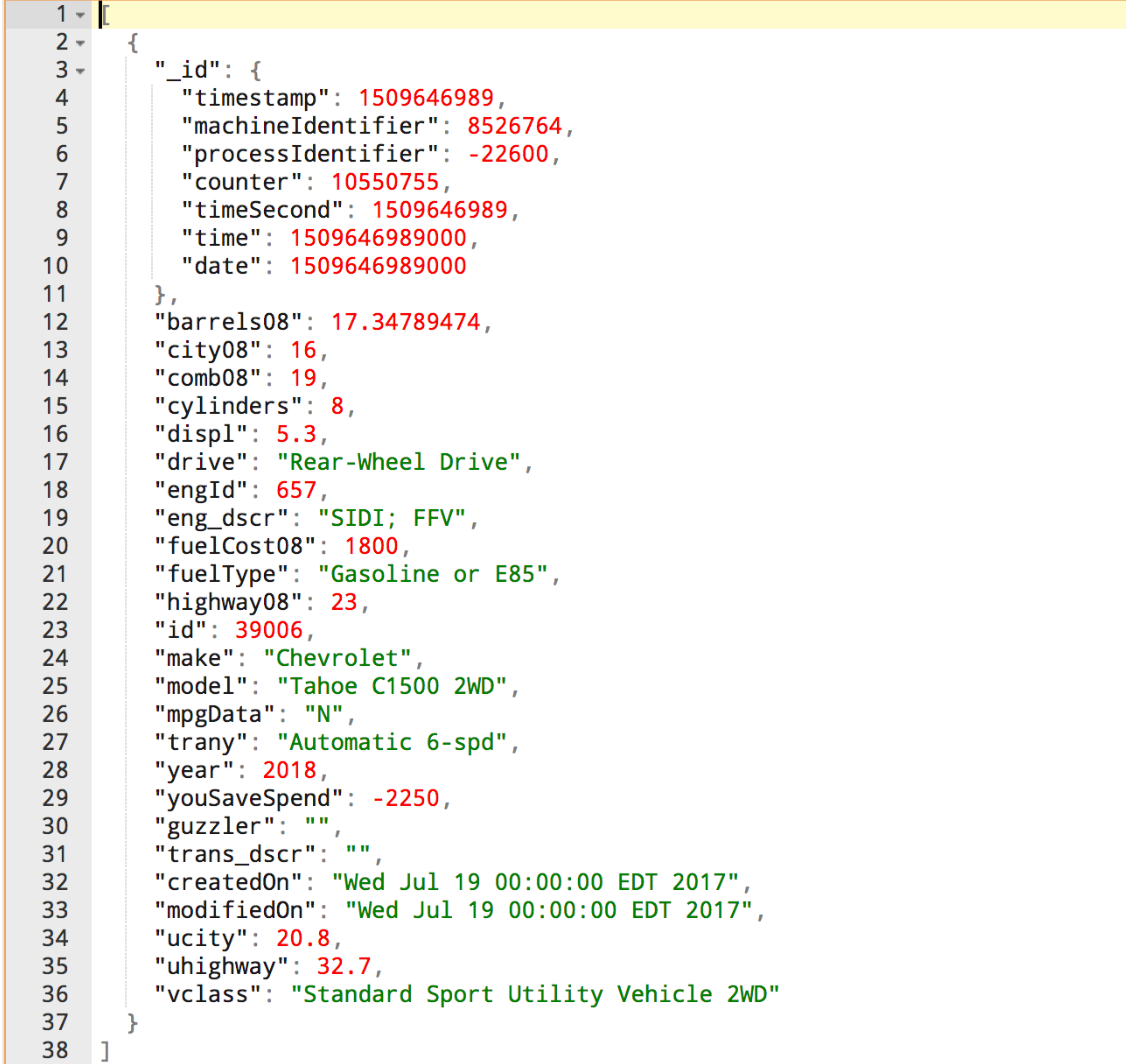
- Free vehicle information
- Source of MPG information
- Web service
- Download data in either XML or CSV

```
15.69571429,19,21,4,2,Rear-Wheel Drive,9011,(FFS),1650,Regular,25,1,Alfa Romeo,Spider Veloce 2000,Y,Manual 5-sp
29.96454545,9,11,12,4.9,Rear-Wheel Drive,22020,(GUZZLER),3150,Regular,14,10,Ferrari,Testarossa,N,Manual 5-spd,1
12.20777778,23,27,4,2.2,Front-Wheel Drive,2100,(FFS),1300,Regular,33,100,Dodge,Charger,Y,Manual 5-spd,29,47,Sub
29.96454545,10,11,8,5.2,Rear-Wheel Drive,2850,,3150,Regular,12,1000,Dodge,B150/B250 Wagon 2WD,N,Automatic 3-spd
17.34789474,17,19,4,2.2,4-Wheel or All-Wheel Drive,66031,"(FFS,TRBO)",2200,Premium,23,10000,Subaru,Legacy AWD T
14.98227273,21,22,4,1.8,Front-Wheel Drive,66020,(FFS),1550,Regular,24,10001,Subaru,Loyale,N,Automatic 3-spd,27,
13.1844,22,25,4,1.8,Front-Wheel Drive,66020,(FFS),1400,Regular,29,10002,Subaru,Loyale,Y,Manual 5-spd,28,41,Comp
13.73375,23,24,4,1.6,Front-Wheel Drive,57005,(FFS),1450,Regular,26,10003,Toyota,Corolla,Y,Automatic 3-spd,29,37
12.67730769,23,26,4,1.6,Front-Wheel Drive,57005,(FFS),1350,Regular,31,10004,Toyota,Corolla,Y,Manual 5-spd,30,43
13.1844,23,25,4,1.8,Front-Wheel Drive,57006,(FFS),1400,Regular,30,10005,Toyota,Corolla,Y,Automatic 4-spd,29,42
12.67730769,23,26,4,1.8,Front-Wheel Drive,57006,(FFS),1350,Regular,30,10006,Toyota,Corolla,Y,Manual 5-spd,30,42
15.69571429,18,21,4,2,Front-Wheel Drive,59007,(FFS),1650,Regular,26,10007,Volkswagen,Golf III / GTI,N,Automatic
13.73375,21,24,4,2,Front-Wheel Drive,59007,(FFS),1450,Regular,29,10008,Volkswagen,Golf III / GTI,Y,Manual 5-spd
15.69571429,18,21,4,2,Front-Wheel Drive,59007,(FFS),1650,Regular,26,10009,Volkswagen,Jetta III,N,Automatic 4-sp
25.35461538,12,13,8,5.2,Rear-Wheel Drive,2850,,2650,Regular,15,1001,Dodge,B150/B250 Wagon 2WD,N,Automatic 3-spd
14.33086957,20,23,4,2,Front-Wheel Drive,59007,(FFS),1500,Regular,28,10010,Volkswagen,Jetta III,N,Manual 5-spd,2
16.4805,18,20,4,2.3,Rear-Wheel Drive,60030,(FFS),1700,Regular,23,10011,Volvo,240,Y,Automatic 4-spd,22,32.0513,C
15.69571429,19,21,4,2.3,Rear-Wheel Drive,60030,(FFS),1650,Regular,26,10012,Volvo,240,Y,Manual 5-spd,23.3333,36,
17.34789474,17,19,6,2.8,Front-Wheel Drive,64012,(FFS),2200,Premium,22,10013,Audi,100,Y,Automatic 4-spd,21,31,Mi
17.34789474,17,19,6,2.8,Front-Wheel Drive,64012,(FFS),2200,Premium,24,10014,Audi,100,N,Manual 5-spd,21,33.3333,
20.600625,14,16,8,4,Rear-Wheel Drive,12071,(GUZZLER)  (FFS),2600,Premium,20,10015,BMW,740i,N,Automatic 5-spd,17
20.600625,14,16,8,4,Rear-Wheel Drive,12071,(GUZZLER)  (FFS),2600,Premium,20,10016,BMW,740il,Y,Automatic 5-spd,1
25.35461538,11,13,12,5,Rear-Wheel Drive,12080,(GUZZLER)  (FFS),2650,Regular,17,10017,BMW,750il,N,Automatic 4-sp
14.33086957,21,23,4,2.2,Front-Wheel Drive,4112,(FFS),1500,Regular,28,10018,Buick,Century,N,Automatic 3-spd,26,3
17.34789474,17,19,6,3.3,Front-Wheel Drive,4410,(FFS),1800,Regular,24,10019,Buick,Century,Y,Automatic 3-spd,21.1
25.35461538,11,13,8,5.2,Rear-Wheel Drive,2850,,2650,Regular,17,1002,Dodge,B150/B250 Wagon 2WD,N,Manual 4-spd,14
15.69571429,18,21,6,3.3,Front-Wheel Drive,4410,(FFS),1650,Regular,26,10020,Buick,Century,N,Automatic 4-spd,22,3
16.4805,17,20,6,3.1,Front-Wheel Drive,4117,(FFS),1700,Regular,25,10021,Buick,Regal,N,Automatic 3-spd,21,35,Mids
16.4805,17,20,6,3.1,Front-Wheel Drive,4118,(FFS),1700,Regular,27,10022,Buick,Regal,N,Automatic 4-spd,21,38,Mids
16.4805,17,20,6,3.8,Front-Wheel Drive,4400,(FFS),1700,Regular,26,10023,Buick,Regal,Y,Automatic 4-spd,21,36,Mids
16.4805,17,20,6,3.8,Front-Wheel Drive,4400,(FFS),1700,Regular,25,10024,Buick,Riviera,N,Automatic 4-spd,21,35,M
```

mongoimport

Imports data to MongoDB

```
mongoimport -h dsXXXXXX.mlab.com:XXXXX -d users -c vehicle -u admin --file vehicles-1997.csv
--type csv --columnsHaveTypes --fields
"barrels08.double(),city08.double(),comb08.double(),cylinders.int32(),displ.double(),drive.string(),e
ngId.int32(),eng_dscr.string(),fuelCost08.double(),fuelType.string(),highway08.double(),id.int32(),m
ake.string(),model.string(),mpgData.string(),trany.string(),UCity.double(),UHighway.double(),VClass
.string(),year.int32(),youSaveSpend.double(),guzzler.string(),trans_dscr.string(),createdOn.string(),
modifiedOn.string()" --parseGrace skipField -p
```



```
[
  {
    "_id": {
      "timestamp": 1509646989,
      "machineIdentifier": 8526764,
      "processIdentifier": -22600,
      "counter": 10550755,
      "timeSecond": 1509646989,
      "time": 1509646989000,
      "date": 1509646989000
    },
    "barrels08": 17.34789474,
    "city08": 16,
    "comb08": 19,
    "cylinders": 8,
    "displ": 5.3,
    "drive": "Rear-Wheel Drive",
    "engId": 657,
    "eng_dscr": "SIDI; FFV",
    "fuelCost08": 1800,
    "fuelType": "Gasoline or E85",
    "highway08": 23,
    "id": 39006,
    "make": "Chevrolet",
    "model": "Tahoe C1500 2WD",
    "mpgData": "N",
    "trany": "Automatic 6-spd",
    "year": 2018,
    "youSaveSpend": -2250,
    "guzzler": "",
    "trans_dscr": "",
    "createdOn": "Wed Jul 19 00:00:00 EDT 2017",
    "modifiedOn": "Wed Jul 19 00:00:00 EDT 2017",
    "ucity": 20.8,
    "uhighway": 32.7,
    "vclass": "Standard Sport Utility Vehicle 2WD"
  }
]
```

```kotlin
package com.tekadept.rk1.models

data class Vehicle(
        val _id: org.bson.types.ObjectId,
        val barrels08: Double,
        val city08: Double,
        val comb08: Double,
        val cylinders: Int?,
        val displ: Double?,
        val drive: String,
        val engId: Int,
        val eng_dscr: String,
        val fuelCost08: Double,
        val fuelType: String,
        val highway08: Double,
        val id: Int,
        val make: String,
        val model: String,
        val mpgData: String,
        val trany: String,
        val UCity: Double,
        val UHighway: Double,
        val VClass: String,
        val year: Int,
        val youSaveSpend: Double,
        val guzzler: String,
        val trans_dscr: String,
        val createdOn: String,
        val modifiedOn: String
)
```

KotlinConf

Microservice

A service with one and only one, very narrowly focused capability that a remote API exposes to the rest of the system.

Microservice
Key ideas

- Runs in its own process
- Owns its data store
- Can be deployed on its own
- Can be written in different languages

Java Web Frameworks

What do these all have in common?

- Spring MVC
- Struts 2
- JavaServer Faces (JSF)
- Play!

Java Microservice Frameworks

- Spark aka. SparkJava
- Ratpack

KotlinConf

Spark

aka SparkJava

A micro framework for creating web applications in Kotlin and Java 8 with minimal effort

Spark aka SparkJava

- Supports Java and Kotlin
- First released Feb 7, 2013
- Latest May 13, 2017, version 2.6.0
- Lots of docs and tutorials

KotlinConf

Ratpack

Ratpack
Lean & powerful HTTP apps

- Supports Groovy, Java and Kotlin
- First released Jul 21, 2012
- Latest release Sept. 3, 2017, version 1.5.0
- Lots of docs but not a lot of examples

RESTful API

- Defined by RFC 2616 protocol
- Preferred over SOAP since it uses less bandwidth
- Breaks down a transaction into a series of HTTP methods
- Stateless by design

GET Method

GET /resource	
Request has body	No
Successful response has body	Yes
Safe	Yes
Idempotent	Yes
Cacheable	Yes

HEAD Method

HEAD /resource (HEAD *)	
Request has body	No
Successful response has body	No
Safe	Yes
Idempotent	Yes
Cacheable	Yes

POST Method

POST /resource	
Request has body	Yes
Successful response has body	Yes
Safe	No
Idempotent	No
Cacheable	No*

PUT Method

PUT /new-resource	
Request has body	Yes
Successful response has body	No
Safe	No
Idempotent	Yes
Cacheable	No

PATCH Method

PATCH /resource	
Request has body	Yes
Successful response has body	No
Safe	No
Idempotent	No
Cacheable	No

DELETE Method

DELETE /resource	
Request has body	No
Successful response has body	No
Safe	No
Idempotent	Yes
Cacheable	No

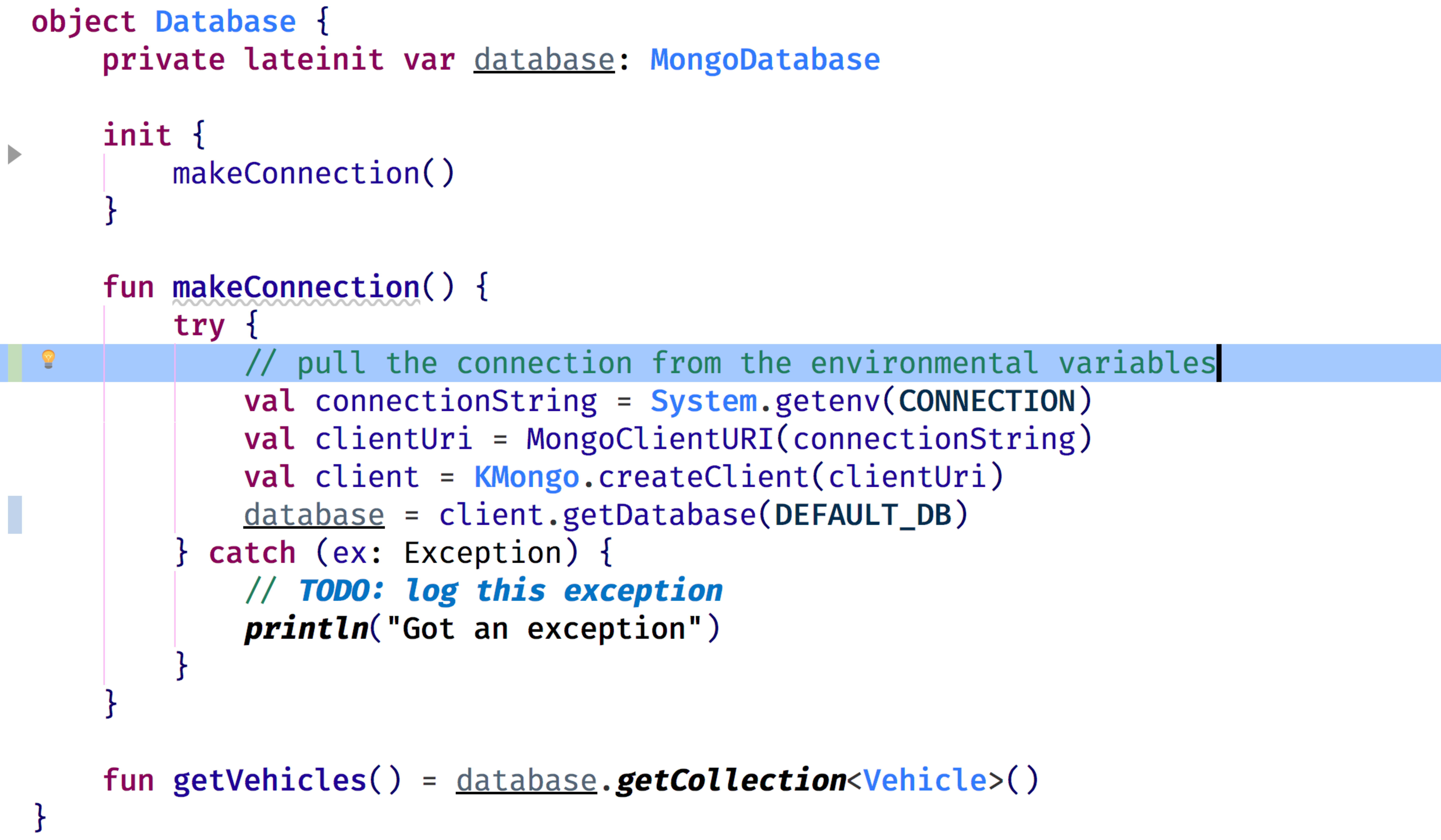
OPTIONS Method

OPTIONS /resource	
Request has body	No
Successful response has body	Yes
Safe	Yes
Idempotent	No
Cacheable	No

```
group 'com.tekadept.rk1'
version '1.0-SNAPSHOT'

buildscript {
    ext.kotlin_version = '1.1.60-eap-43'

    repositories {
        maven { url 'http://dl.bintray.com/kotlin/kotlin-eap-1.1' }
        mavenCentral()
        jcenter()
    }
    dependencies {
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        classpath 'io.ratpack:ratpack-gradle:1.3.3'
    }
}

apply plugin: 'java'
apply plugin: 'kotlin'
apply plugin: 'io.ratpack.ratpack-java'

sourceCompatibility = 1.8

repositories {
    maven { url 'http://dl.bintray.com/kotlin/kotlin-eap-1.1' }
    mavenCentral()
    jcenter()
}
```

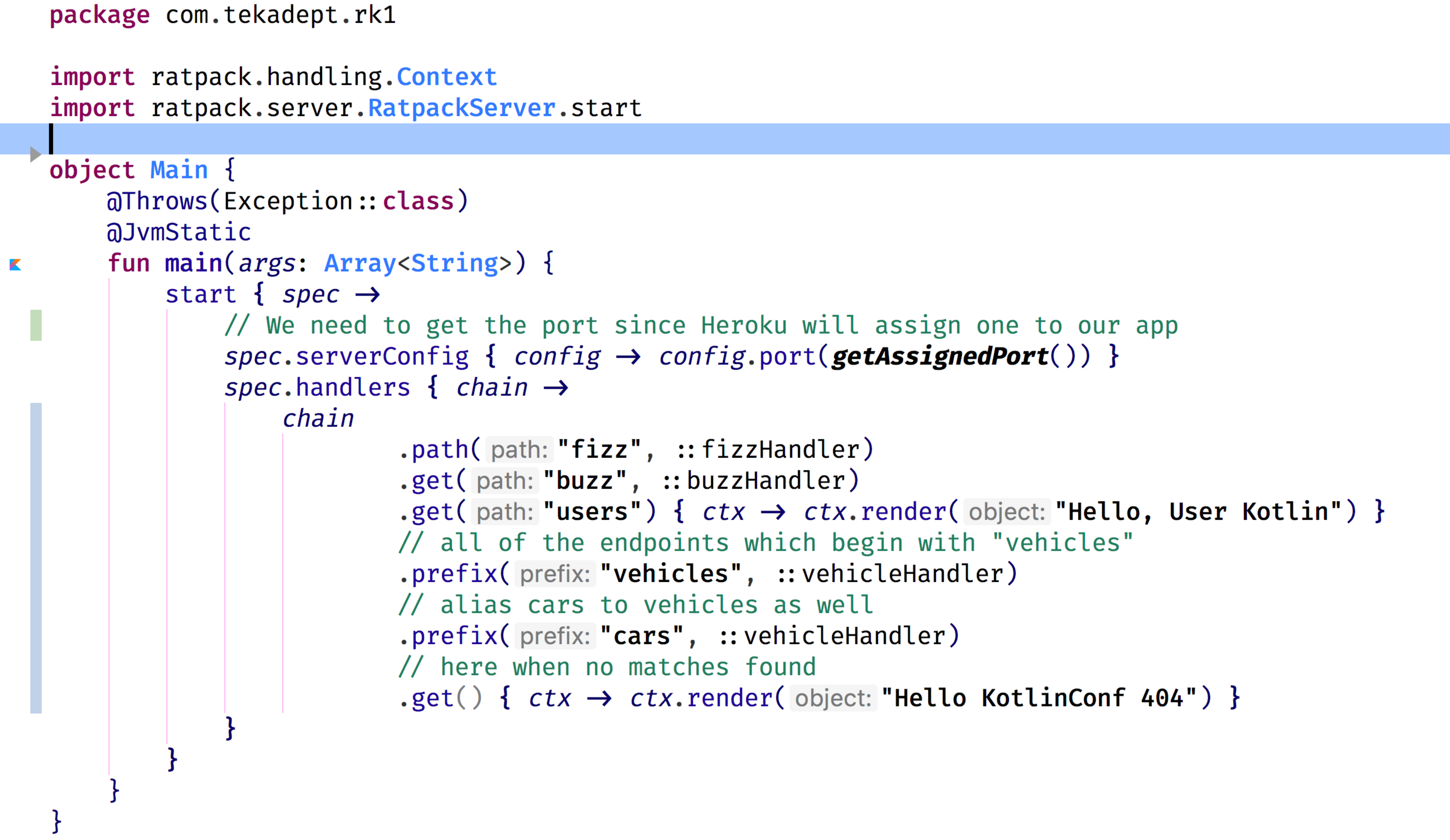
```
dependencies {
    compile "org.jetbrains.kotlin:kotlin-stdlib-jre8:$kotlin_version"
    compile 'org.mongodb:mongodb-driver:3.4.3'
    compile 'org.litote.kmongo:kmongo-native:3.5.0'
    testCompile group: 'junit', name: 'junit', version: '4.12'
}

compileKotlin {
    kotlinOptions.jvmTarget = "1.8"
}
compileTestKotlin {
    kotlinOptions.jvmTarget = "1.8"
}

mainClassName = 'com.tekadept.rk1.Main'

task stage {
    dependsOn installDist
}
```

```kotlin
@file:JvmName("Constants")
package com.tekadept.rk1

const val DEFAULT_PORT = 5050
const val PAGE_COUNT = 20
const val SORT_YMM = "{year: -1, make: 1, model: 1}"
const val CONNECTION = "MONGODB_CONNECTION"
const val DEFAULT_DB = "users"
```



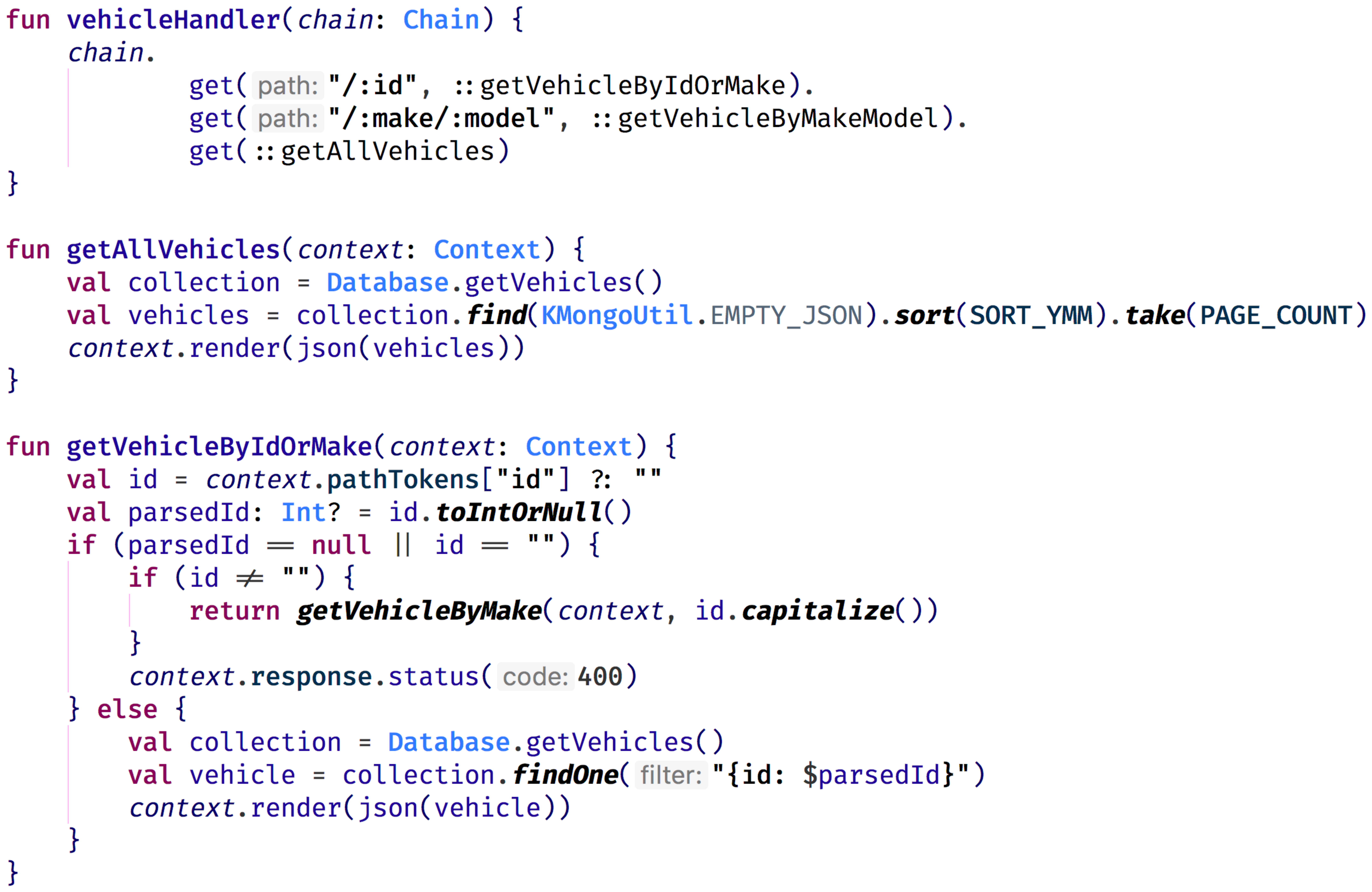
```kotlin
object Database {
    private lateinit var database: MongoDatabase

    init {
        makeConnection()
    }

    fun makeConnection() {
        try {
            // pull the connection from the environmental variables
            val connectionString = System.getenv(CONNECTION)
            val clientUri = MongoClientURI(connectionString)
            val client = KMongo.createClient(clientUri)
            database = client.getDatabase(DEFAULT_DB)
        } catch (ex: Exception) {
            // TODO: log this exception
            println("Got an exception")
        }
    }

    fun getVehicles() = database.getCollection<Vehicle>()
}
```

```
package com.tekadept.rk1

import ratpack.server.RatpackServer.start


object MainX {
    @Throws(Exception::class)
    @JvmStatic
    fun main(args: Array<String>) {
        start { server →
            server.handlers { chain →
                chain
                        .get { ctx → ctx.render( object: "Hello World!") }
                        .get( path: ":name") { ctx → ctx.render( object: "Hello " + ctx.pathTokens["name"] + "!") }
            }
        }
    }
}
```



```kotlin
package com.tekadept.rk1

import ratpack.handling.Context
import ratpack.server.RatpackServer.start

object Main {
    @Throws(Exception::class)
    @JvmStatic
    fun main(args: Array<String>) {
        start { spec →
            // We need to get the port since Heroku will assign one to our app
            spec.serverConfig { config → config.port(getAssignedPort()) }
            spec.handlers { chain →
                chain
                        .path( path: "fizz", ::fizzHandler)
                        .get( path: "buzz", ::buzzHandler)
                        .get( path: "users") { ctx → ctx.render( object: "Hello, User Kotlin") }
                        // all of the endpoints which begin with "vehicles"
                        .prefix( prefix: "vehicles", ::vehicleHandler)
                        // alias cars to vehicles as well
                        .prefix( prefix: "cars", ::vehicleHandler)
                        // here when no matches found
                        .get() { ctx → ctx.render( object: "Hello KotlinConf 404") }
            }
        }
    }
}
```

```kotlin
// all of the methods for the "fizz" path are handled here
fun fizzHandler(context: Context) {
    context.byMethod { t ->
        t
                .get { context.render( object: "GET FIZZ") }
                .post { context.render( object: "POST FIZZ") }
                .patch { context.render( object: "PATCH FIZZ") }
                .put { context.render( object: "PATCH FIZZ") }
                .options { context.render( object: "OPTIONS FIZZ") }
                .delete { context.render( object: "DELETE FIZZ") }
    }
}

fun buzzHandler(context: Context) {
    context.render( object: "from the baz handler")
}

fun getAssignedPort() = ProcessBuilder().environment()["PORT"]?.toInt() ?: DEFAULT_PORT
```



```
fun vehicleHandler(chain: Chain) {
    chain.
            get( path: "/:id", ::getVehicleByIdOrMake).
            get( path: "/:make/:model", ::getVehicleByMakeModel).
            get(::getAllVehicles)
}

fun getAllVehicles(context: Context) {
    val collection = Database.getVehicles()
    val vehicles = collection.find(KMongoUtil.EMPTY_JSON).sort(SORT_YMM).take(PAGE_COUNT)
    context.render(json(vehicles))
}

fun getVehicleByIdOrMake(context: Context) {
    val id = context.pathTokens["id"] ?: ""
    val parsedId: Int? = id.toIntOrNull()
    if (parsedId == null || id == "") {
        if (id != "") {
            return getVehicleByMake(context, id.capitalize())
        }
        context.response.status( code: 400)
    } else {
        val collection = Database.getVehicles()
        val vehicle = collection.findOne( filter: "{id: $parsedId}")
        context.render(json(vehicle))
    }
}
```

```
fun getVehicleByMake(context: Context, make: String) {
    val collection = Database.getVehicles()
    val vehicles = collection.find( filter: "{make: '$make'}")
            .sort(SORT_YMM)
            .take(PAGE_COUNT)
    context.render(json(vehicles))
}

fun getVehicleByMakeModel(context: Context) {
    val make = context.pathTokens["make"]?.capitalize()
    val model = context.pathTokens["model"]?.capitalize()
    if (make == null || model == null) {
        context.response.status( code: 400)
    } else {
        val collection = Database.getVehicles()
        val vehicles = collection.find( filter: "{make: '$make', model: '$model'}")
                .sort(SORT_YMM)
                .take(PAGE_COUNT)
        context.render(json(vehicles))
    }
}
```

KotlinConf

MongoDB

MongoDB

- Document Database
- High Performance
- High Availability
- Easy Scalability
- Geospatial Data

Top DB Engines

October 2017

1. Oracle
2. MySQL
3. MS SQL Server
4. PostgreSQL
5. MongoDB

https://db-engines.com/en/ranking

SQL to MongoDB

SQL	MongoDB
column	field
row	document
table	collection
database	database
joins	none
transactions	none

CRUD Operations

- Create: insert()
- Read: find()
- Update: update()
- Delete: remove(<query>)

Query Modifiers

- db.<collection name>.find(<query>)
- skip()
- take()
- sort()
- pretty()

KMongo

A Kotlin toolkit for Mongo

- Wraps the MongoDB Java driver
- Converts objects from JSON to KOJO

KotlinConf

mLab

mLab

the leading Database-as-a-Service for MongoDB

- Database as a Service (DaaS) provider
- Supports AWS, Azure, App Engine
- Used by Fox, New York Times, Lyft, Toyota, SAP

KotlinConf

Heroku

Heroku
Platform-as-a-Service (PaaS)

- Supports PHP, Python, Node.js, Java, Scala, and Go
- Runs on top of AWS
- Bought by Salesforce in 2010
- Runs instances which require very little configuration
- heroic + haiku = heroku

Setup

free developer account

- Have Git installed
- Create a free Heroku account
- Install the Heroku CLI
- brew install heroku/brew/heroku

Deployment

- heroku login
- heroku create <app-name> (must be unique)
- git push heroku master
- heroku open

```
web: build/install/rk1/bin/rk1
```

KotlinConf

Live Demo

Cross your fingers…

Thank you!

Troy Miles
@therockncoder

#kotlinconf17