BUSTRACKER

APLICACIÓN PARA VER LOS TIEMPOS DEL TRANSPORTE PUBLICO

Roberto Blázquez Martín

TECNOLOGIAS

- Ktor
- React
- Mongo
- Docker
- Nginx

KTOR

Ktor es un marco de desarrollo de aplicaciones web en Kotlin. Proporciona un conjunto de herramientas y características que permiten crear rápidamente aplicaciones web y API.

¿POR QUÉ KTOR?

- Ktor permite desarrollar apis rápido y además tiene mucho mejor rendimiento que alternativas sus alternativas como Spring.

REACT

- React es una biblioteca de JavaScript utilizada para construir interfaces de usuario interactivas y reactivas.

¿POR QUÉ REACT?

- Eficiencia y rendimiento
- Componentes reutilizables
- Gran ecosistema y comunidad
- Multiplataforma

MONGO

- MongoDB es una base de datos NoSQL (No Relacional) que se caracteriza por su enfoque en la escalabilidad, la flexibilidad y el almacenamiento.

¿POR QUÉ MONGO?

- Flexibilidad en el esquema de datos
- Escalabilidad horizontal
- Consultas flexibles

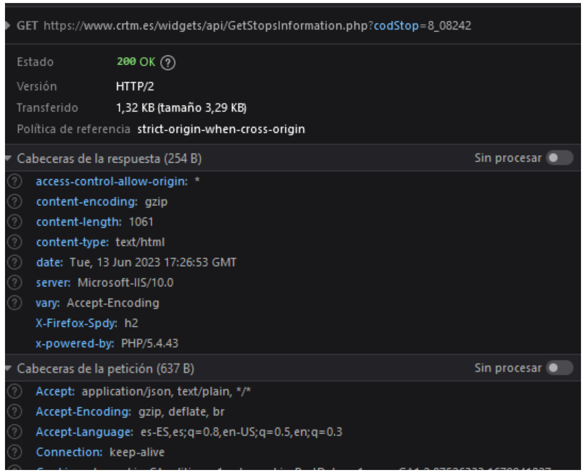
BACKEND

- Tiempos de las paradas de bus
- Localización de los autbuses
- Tiempos de las estaciones de metro
- Usuarios
- Autenticación
- Paradas Favoritas

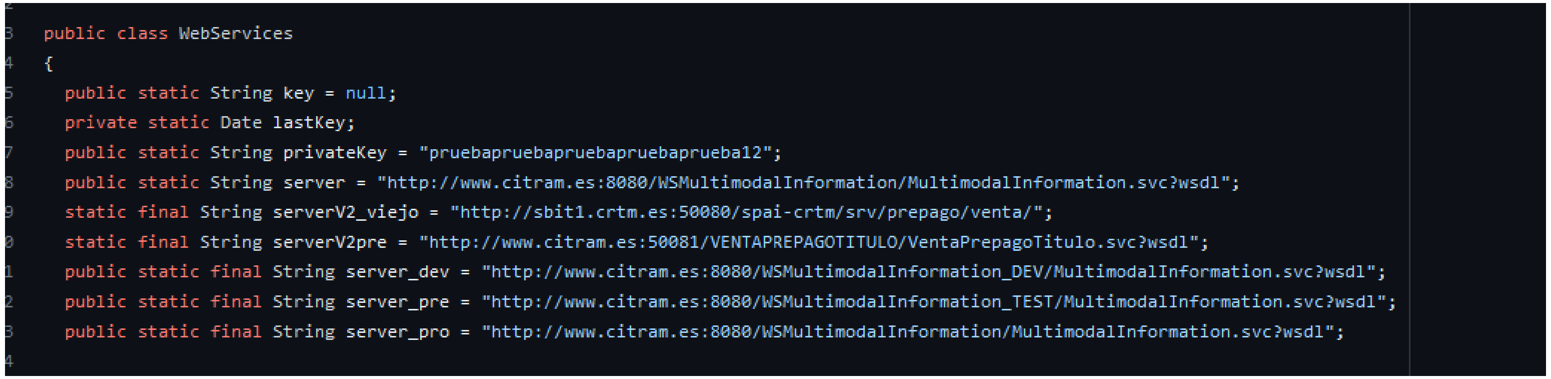
TIEMPOS BUS

- ¿Dónde obtengo los datos?
- Normalizar los datos

OBTENCIÓN DATOS

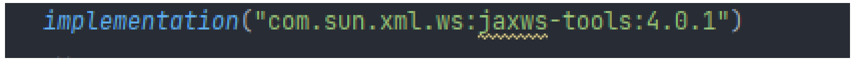


```
GET https://www.crtm.es/widgets/api/GetStopsInformation.php?codStop=8_08242

Estado                      200 OK
Versión                     HTTP/2
Transferido                 1,32 KB (tamaño 3,29 KB)
Política de referencia strict-origin-when-cross-origin

Cabeceras de la respuesta (254 B)                    Sin procesar
access-control-allow-origin: *
content-encoding: gzip
content-length: 1061
content-type: text/html
date: Tue, 13 Jun 2023 17:26:53 GMT
server: Microsoft-IIS/10.0
vary: Accept-Encoding
X-Firefox-Spdy: h2
x-powered-by: PHP/5.4.43

Cabeceras de la petición (637 B)                     Sin procesar
Accept: application/json, text/plain, */*
Accept-Encoding: gzip, deflate, br
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Connection: keep-alive
```

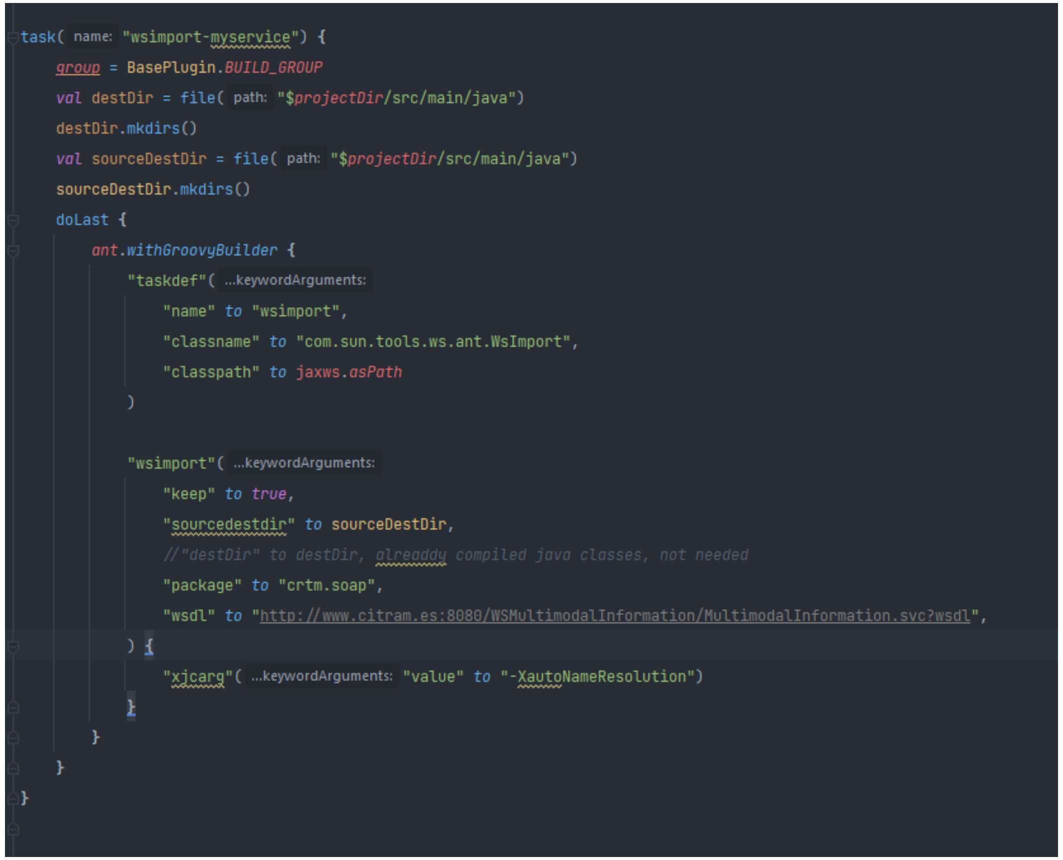


```java
public class WebServices
{
  public static String key = null;
  private static Date lastKey;
  public static String privateKey = "pruebapruebapruebapruebaprueba12";
  public static String server = "http://www.citram.es:8080/WSMultimodalInformation/MultimodalInformation.svc?wsdl";
  static final String serverV2_viejo = "http://sbit1.crtm.es:50080/spai-crtm/srv/prepago/venta/";
  static final String serverV2pre = "http://www.citram.es:50081/VENTAPREPAGOTITULO/VentaPrepagoTitulo.svc?wsdl";
  public static final String server_dev = "http://www.citram.es:8080/WSMultimodalInformation_DEV/MultimodalInformation.svc?wsdl";
  public static final String server_pre = "http://www.citram.es:8080/WSMultimodalInformation_TEST/MultimodalInformation.svc?wsdl";
  public static final String server_pro = "http://www.citram.es:8080/WSMultimodalInformation/MultimodalInformation.svc?wsdl";
```

GENERAR SOAP

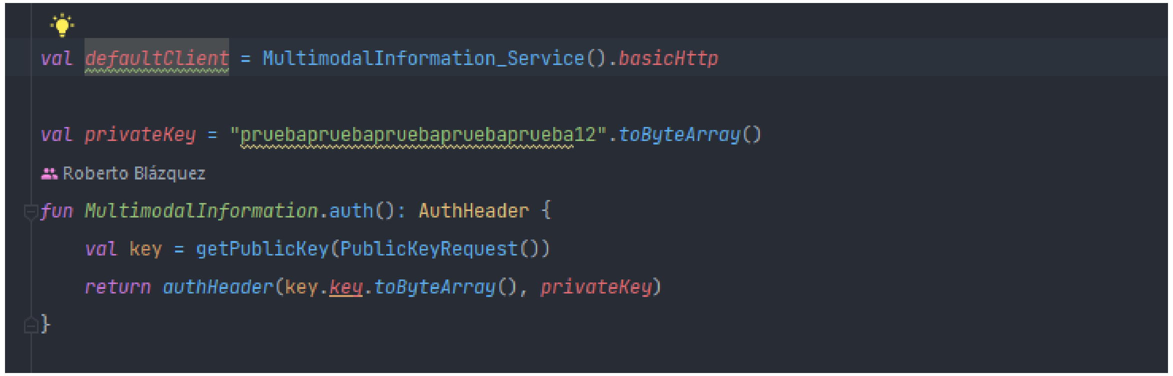


```
implementation("com.sun.xml.ws:jaxws-tools:4.0.1")
```

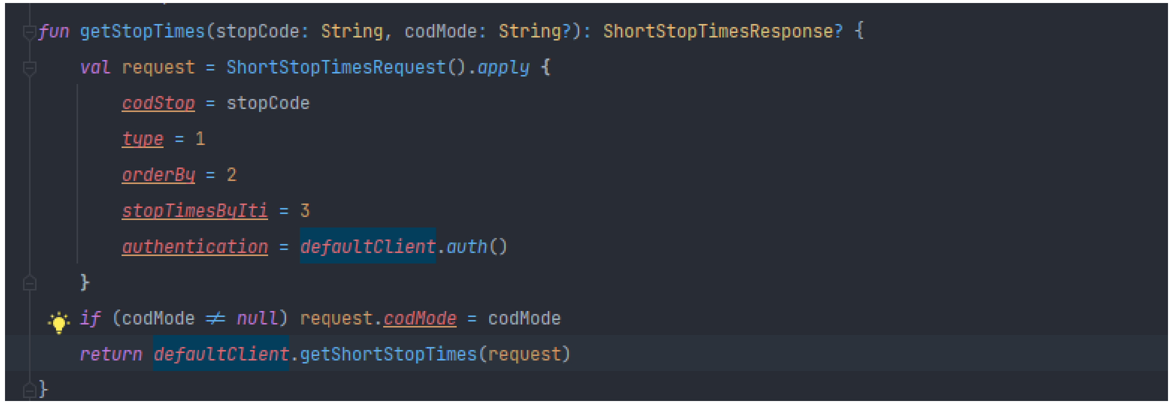


```
task( name: "wsimport-myservice") {
    group = BasePlugin.BUILD_GROUP
    val destDir = file( path: "$projectDir/src/main/java")
    destDir.mkdirs()
    val sourceDestDir = file( path: "$projectDir/src/main/java")
    sourceDestDir.mkdirs()
    doLast {
        ant.withGroovyBuilder {
            "taskdef"( ...keywordArguments:
                "name" to "wsimport",
                "classname" to "com.sun.tools.ws.ant.WsImport",
                "classpath" to jaxws.asPath
            )

            "wsimport"( ...keywordArguments:
                "keep" to true,
                "sourcedestdir" to sourceDestDir,
                //"destDir" to destDir, alreaddy compiled java classes, not needed
                "package" to "crtm.soap",
                "wsdl" to "http://www.citram.es:8080/WSMultimodalInformation/MultimodalInformation.svc?wsdl",
            ) {
                "xjcarg"( ...keywordArguments: "value" to "-XautoNameResolution")
            }
        }
    }
}
```

CONSUMIR SOAP

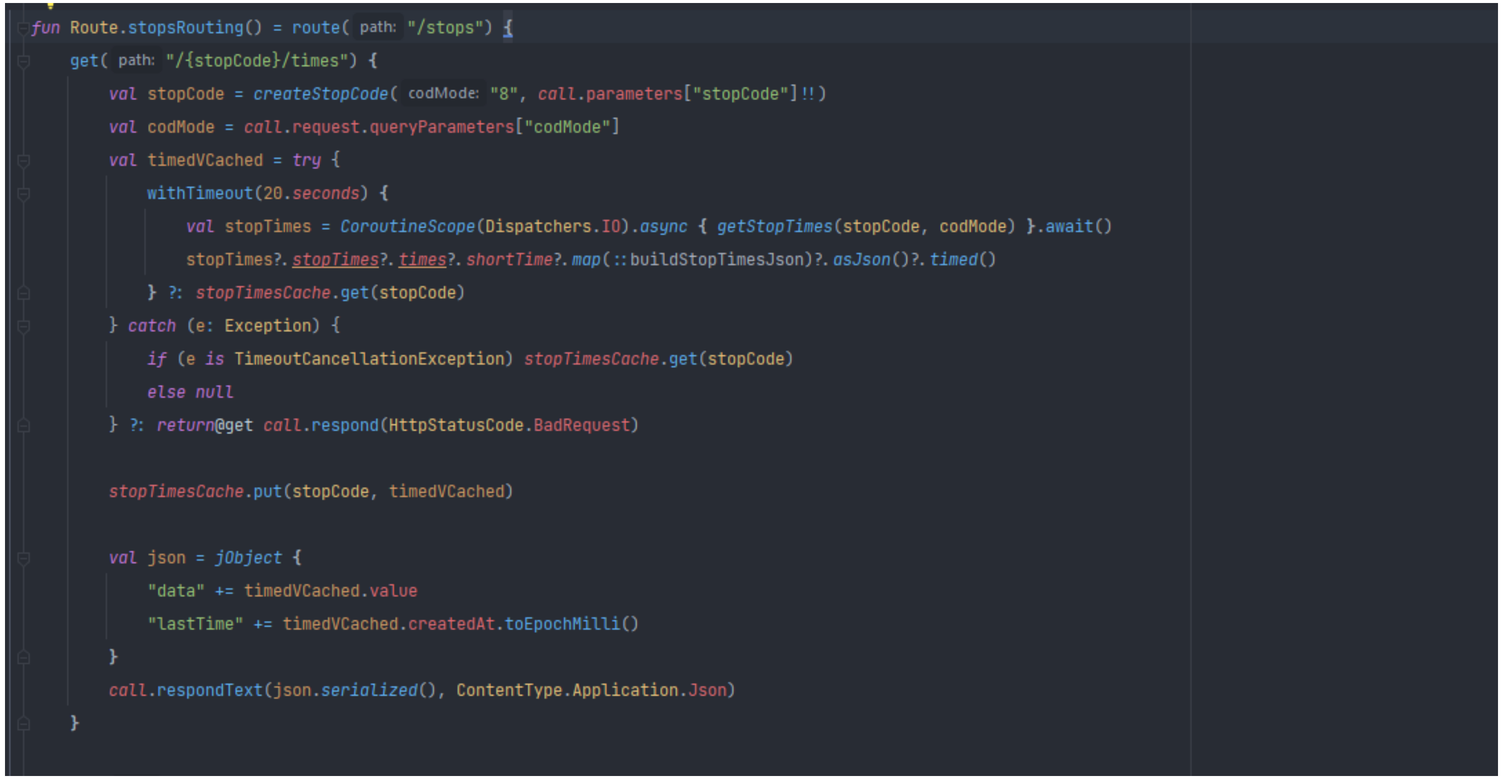


```
val defaultClient = MultimodalInformation_Service().basicHttp

val privateKey = "pruebapruebapruebapruebaprueba12".toByteArray()
Roberto Blázquez
fun MultimodalInformation.auth(): AuthHeader {
    val key = getPublicKey(PublicKeyRequest())
    return authHeader(key.key.toByteArray(), privateKey)
}
```



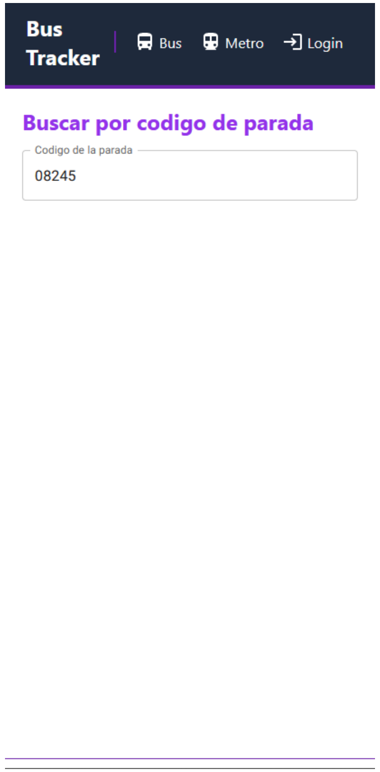
```
fun getStopTimes(stopCode: String, codMode: String?): ShortStopTimesResponse? {
    val request = ShortStopTimesRequest().apply {
        codStop = stopCode
        type = 1
        orderBy = 2
        stopTimesByIti = 3
        authentication = defaultClient.auth()
    }
    if (codMode ≠ null) request.codMode = codMode
    return defaultClient.getShortStopTimes(request)
}
```

EXPONER LOS ENDPOINTS



```
fun Route.stopsRouting() = route( path: "/stops") {
    get( path: "/{stopCode}/times") {
        val stopCode = createStopCode( codMode: "8", call.parameters["stopCode"]!!)
        val codMode = call.request.queryParameters["codMode"]
        val timedVCached = try {
            withTimeout(20.seconds) {
                val stopTimes = CoroutineScope(Dispatchers.IO).async { getStopTimes(stopCode, codMode) }.await()
                stopTimes?.stopTimes?.times?.shortTime?.map(::buildStopTimesJson)?.asJson()?.timed()
            } ?: stopTimesCache.get(stopCode)
        } catch (e: Exception) {
            if (e is TimeoutCancellationException) stopTimesCache.get(stopCode)
            else null
        } ?: return@get call.respond(HttpStatusCode.BadRequest)

        stopTimesCache.put(stopCode, timedVCached)

        val json = jObject {
            "data" += timedVCached.value
            "lastTime" += timedVCached.createdAt.toEpochMilli()
        }
        call.respondText(json.serialized(), ContentType.Application.Json)
    }
```

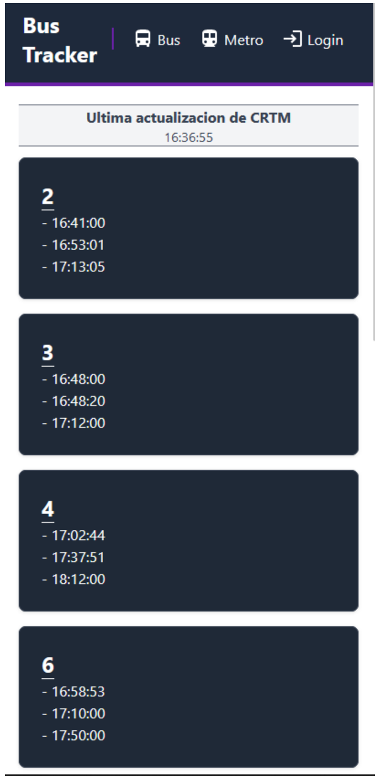
FRONT



Bus Tracker | Bus | Metro | Login

Buscar por codigo de parada

Codigo de la parada: 08245



Bus Tracker | Bus | Metro | Login

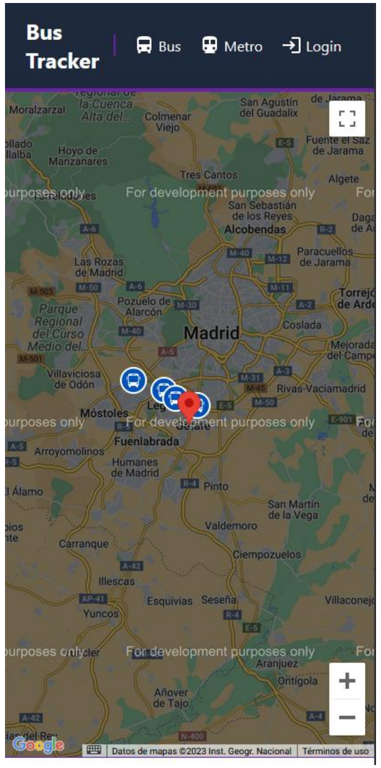
Ultima actualizacion de CRTM
16:36:55

2
- 16:41:00
- 16:53:01
- 17:13:05

3
- 16:48:00
- 16:48:20
- 17:12:00

4
- 17:02:44
- 17:37:51
- 18:12:00

6
- 16:58:53
- 17:10:00
- 17:50:00

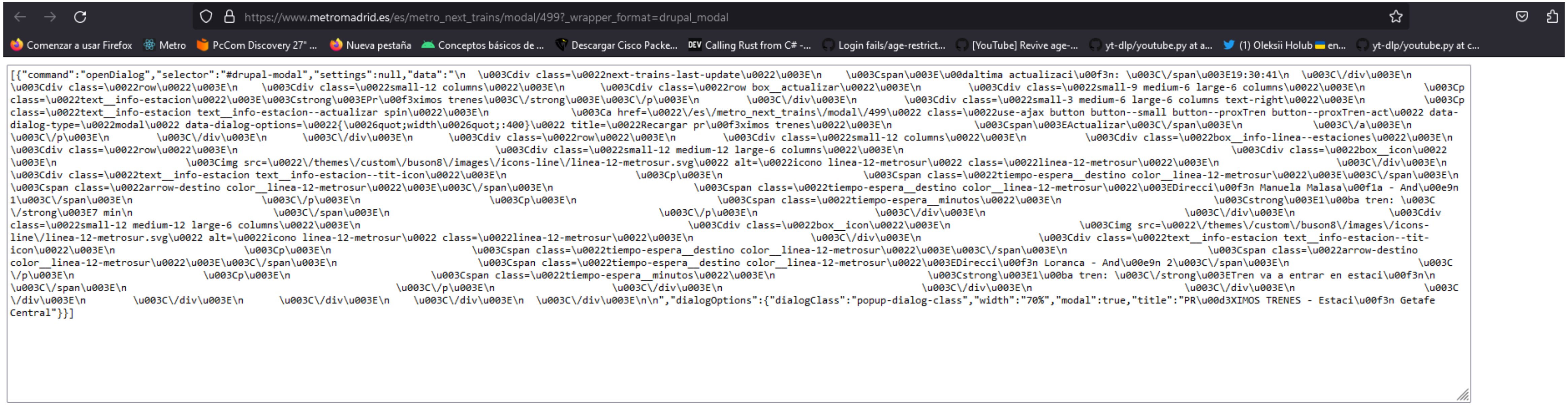


Bus Tracker | Bus | Metro | Login

TIEMPO METRO

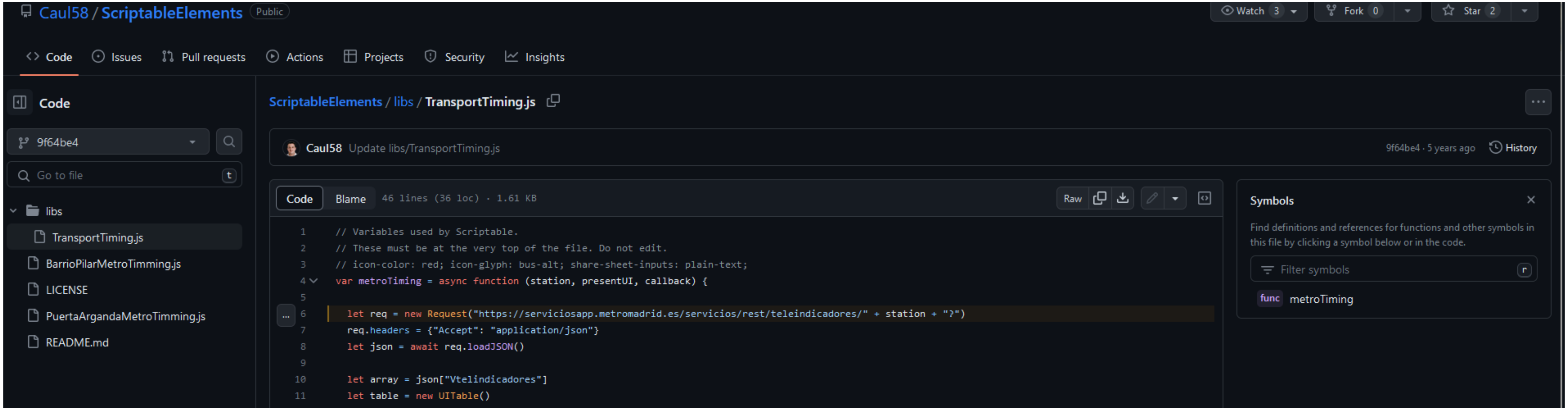
- ¿Dónde obtengo los datos?
- Normalizar los datos

OBTENCIÓN DATOS


https://www.metromadrid.es/es/metro_next_trains/modal/499?_wrapper_format=drupal_modal
Comenzar a usar Firefox
Metro
PcCom Discovery 27" ...
Nueva pestaña
Conceptos básicos de ...
Descargar Cisco Packe...
Calling Rust from C# -...
Login fails/age-restrict...
[YouTube] Revive age-...
yt-dlp/youtube.py at a...
(1) Oleksii Holub 🇺🇦 en...
yt-dlp/youtube.py at c...

```
[{"command":"openDialog","selector":"#drupal-modal","settings":null,"data":"\n  \u003Cdiv class=\u0022next-trains-last-update\u0022\u003E\n    \u003Cspan\u003E\u00daltima actualizaci\u00f3n: \u003C\/span\u003E19:30:41\n  \u003C\/div\u003E\n
\u003Cdiv class=\u0022row\u0022\u003E\n    \u003Cdiv class=\u0022small-12 columns\u0022\u003E\n      \u003Cdiv class=\u0022row box__actualizar\u0022\u003E\n        \u003Cdiv class=\u0022small-9 medium-6 large-6 columns\u0022\u003E\n          \u003Cp
class=\u0022text__info-estacion\u0022\u003E\u003Cstrong\u003EPr\u00f3ximos trenes\u003C\/strong\u003E\u003C\/p\u003E\n        \u003C\/div\u003E\n        \u003Cdiv class=\u0022small-3 medium-6 large-6 columns text-right\u0022\u003E\n          \u003Cp
class=\u0022text__info-estacion text__info-estacion--actualizar spin\u0022\u003E\n            \u003Ca href=\u0022\/es\/metro_next_trains\/modal\/499\u0022 class=\u0022use-ajax button button--small button--proxTren button--proxTren-act\u0022 data-
dialog-type=\u0022modal\u0022 data-dialog-options=\u0022{\u0026quot;width\u0026quot;:400}\u0022 title=\u0022Recargar pr\u00f3ximos trenes\u0022\u003E\n              \u003Cspan\u003EActualizar\u003C\/span\u003E\n            \u003C\/a\u003E\n
\u003C\/p\u003E\n        \u003C\/div\u003E\n      \u003C\/div\u003E\n      \u003Cdiv class=\u0022row\u0022\u003E\n        \u003Cdiv class=\u0022small-12 columns\u0022\u003E\n          \u003Cdiv class=\u0022box__info-linea--estaciones\u0022\u003E\n
\u003Cdiv class=\u0022row\u0022\u003E\n                              \u003Cdiv class=\u0022small-12 medium-12 large-6 columns\u0022\u003E\n                              \u003Cdiv class=\u0022box__icon\u0022
\u003E\n                    \u003Cimg src=\u0022\/themes\/custom\/buson8\/images\/icons-line\/linea-12-metrosur.svg\u0022 alt=\u0022icono linea-12-metrosur\u0022 class=\u0022linea-12-metrosur\u0022\u003E\n                  \u003C\/div\u003E\n
\u003Cdiv class=\u0022text__info-estacion text__info-estacion--tit-icon\u0022\u003E\n                      \u003Cp\u003E\n                          \u003Cspan class=\u0022tiempo-espera__destino color__linea-12-metrosur\u0022\u003E\u003C\/span\u003E\n
\u003Cspan class=\u0022arrow-destino color__linea-12-metrosur\u0022\u003E\u003C\/span\u003E\n                          \u003Cspan class=\u0022tiempo-espera__destino color__linea-12-metrosur\u0022\u003EDirecci\u00f3n Manuela Malasa\u00f1a - And\u00e9n
1\u003C\/span\u003E\n                      \u003C\/p\u003E\n                      \u003Cp\u003E\n                          \u003Cspan class=\u0022tiempo-espera__minutos\u0022\u003E\n                            \u003Cstrong\u003E1\u00ba tren: \u003C
\/strong\u003E7 min\n                          \u003C\/span\u003E\n                          \u003C\/p\u003E\n                      \u003C\/div\u003E\n                    \u003C\/div\u003E\n                    \u003Cdiv
class=\u0022small-12 medium-12 large-6 columns\u0022\u003E\n                              \u003Cdiv class=\u0022box__icon\u0022\u003E\n                    \u003Cimg src=\u0022\/themes\/custom\/buson8\/images\/icons-
line\/linea-12-metrosur.svg\u0022 alt=\u0022icono linea-12-metrosur\u0022 class=\u0022linea-12-metrosur\u0022\u003E\n                  \u003C\/div\u003E\n                  \u003Cdiv class=\u0022text__info-estacion text__info-estacion--tit-
icon\u0022\u003E\n                      \u003Cp\u003E\n                          \u003Cspan class=\u0022tiempo-espera__destino color__linea-12-metrosur\u0022\u003E\u003C\/span\u003E\n                          \u003Cspan class=\u0022arrow-destino
color__linea-12-metrosur\u0022\u003E\u003C\/span\u003E\n                          \u003Cspan class=\u0022tiempo-espera__destino color__linea-12-metrosur\u0022\u003EDirecci\u00f3n Loranca - And\u00e9n 2\u003C\/span\u003E\n                      \u003C
\/p\u003E\n                      \u003Cp\u003E\n                          \u003Cspan class=\u0022tiempo-espera__minutos\u0022\u003E\n                            \u003Cstrong\u003E1\u00ba tren: \u003C\/strong\u003ETren va a entrar en estaci\u00f3n\n
\u003C\/span\u003E\n                          \u003C\/p\u003E\n                      \u003C\/div\u003E\n                    \u003C\/div\u003E\n                    \u003C\/div\u003E\n                      \u003C
\/div\u003E\n        \u003C\/div\u003E\n      \u003C\/div\u003E\n    \u003C\/div\u003E\n  \u003C\/div\u003E\n\n","dialogOptions":{"dialogClass":"popup-dialog-class","width":"70%","modal":true,"title":"PR\u00d3XIMOS TRENES - Estaci\u00f3n Getafe
Central"}}]
```

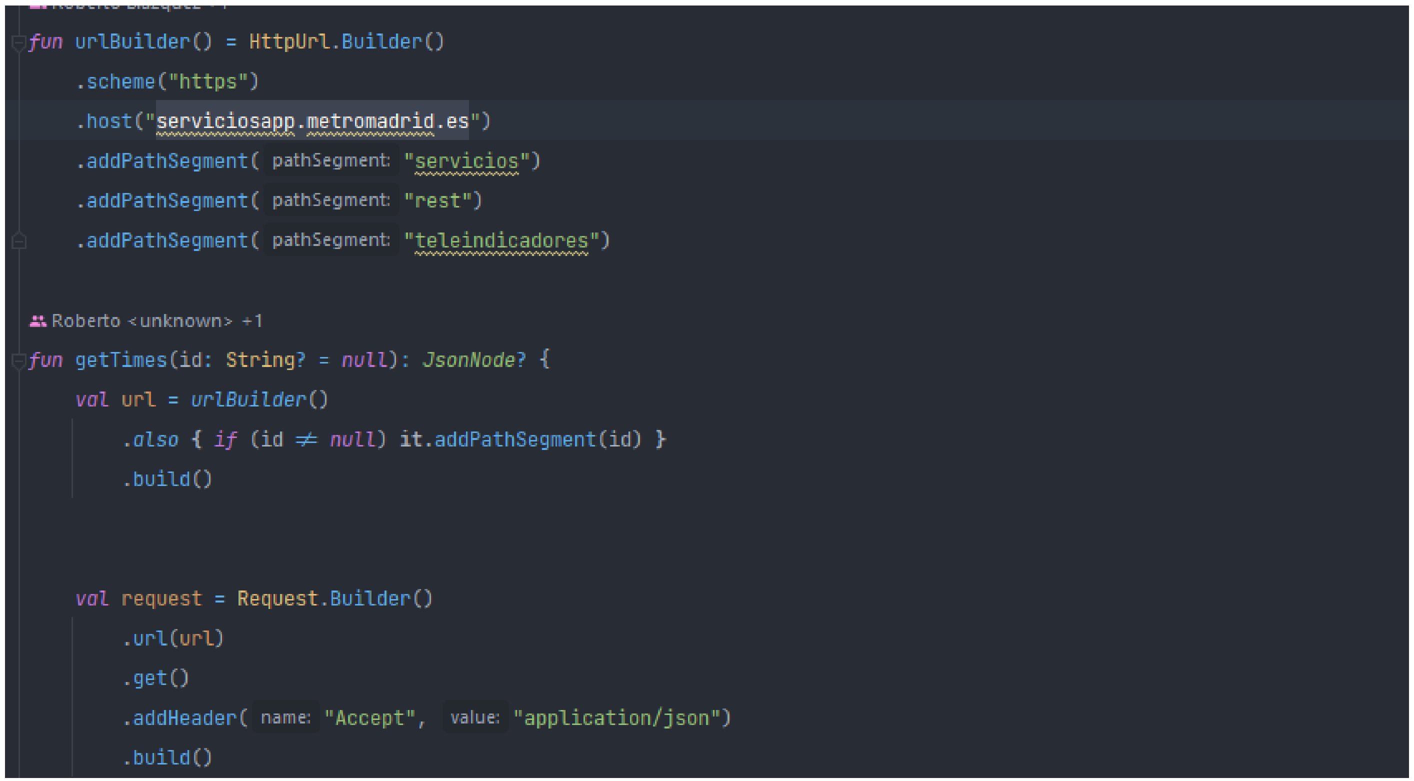
OBTENCIÓN DATOS



Find definitions and references for functions and other symbols in this file by clicking a symbol below or in the code.

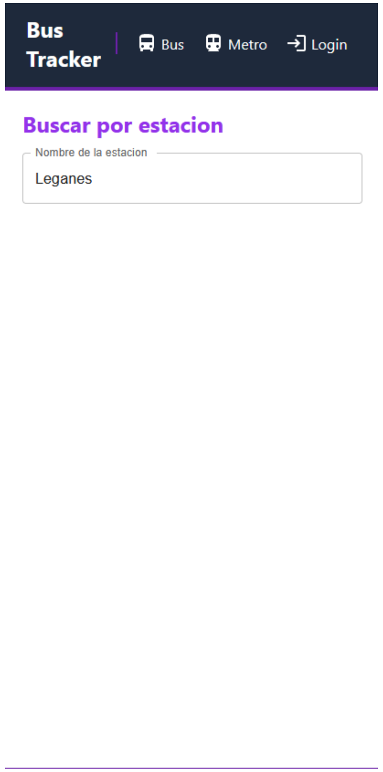
```
// Variables used by Scriptable.
// These must be at the very top of the file. Do not edit.
// icon-color: red; icon-glyph: bus-alt; share-sheet-inputs: plain-text;
var metroTiming = async function (station, presentUI, callback) {

  let req = new Request("https://serviciosapp.metromadrid.es/servicios/rest/teleindicadores/" + station + "?")
  req.headers = {"Accept": "application/json"}
  let json = await req.loadJSON()

  let array = json["Vtelindicadores"]
  let table = new UITable()
```

CONSUMIR API



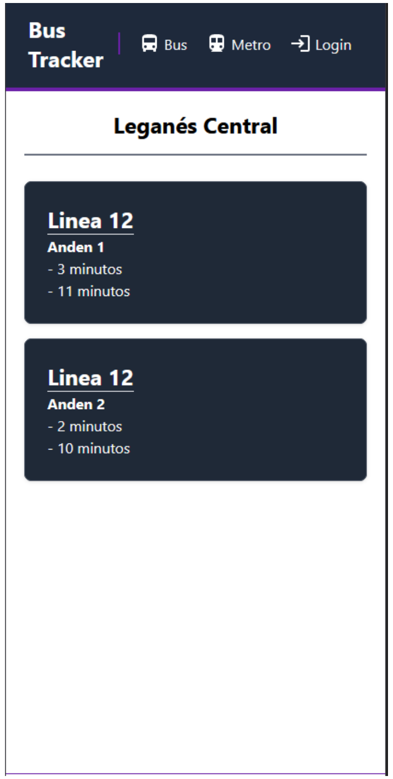
```
fun urlBuilder() = HttpUrl.Builder()
    .scheme("https")
    .host("serviciosapp.metromadrid.es")
    .addPathSegment( pathSegment: "servicios")
    .addPathSegment( pathSegment: "rest")
    .addPathSegment( pathSegment: "teleindicadores")

Roberto <unknown> +1
fun getTimes(id: String? = null): JsonNode? {
    val url = urlBuilder()
        .also { if (id ≠ null) it.addPathSegment(id) }
        .build()


    val request = Request.Builder()
        .url(url)
        .get()
        .addHeader( name: "Accept",  value: "application/json")
        .build()
```

FRONT



Buscar por estacion

Nombre de la estacion: Leganes



Leganés Central

Linea 12

Anden 1

- 3 minutos
- 11 minutos

Linea 12

Anden 2

- 2 minutos
- 10 minutos

USUARIOS

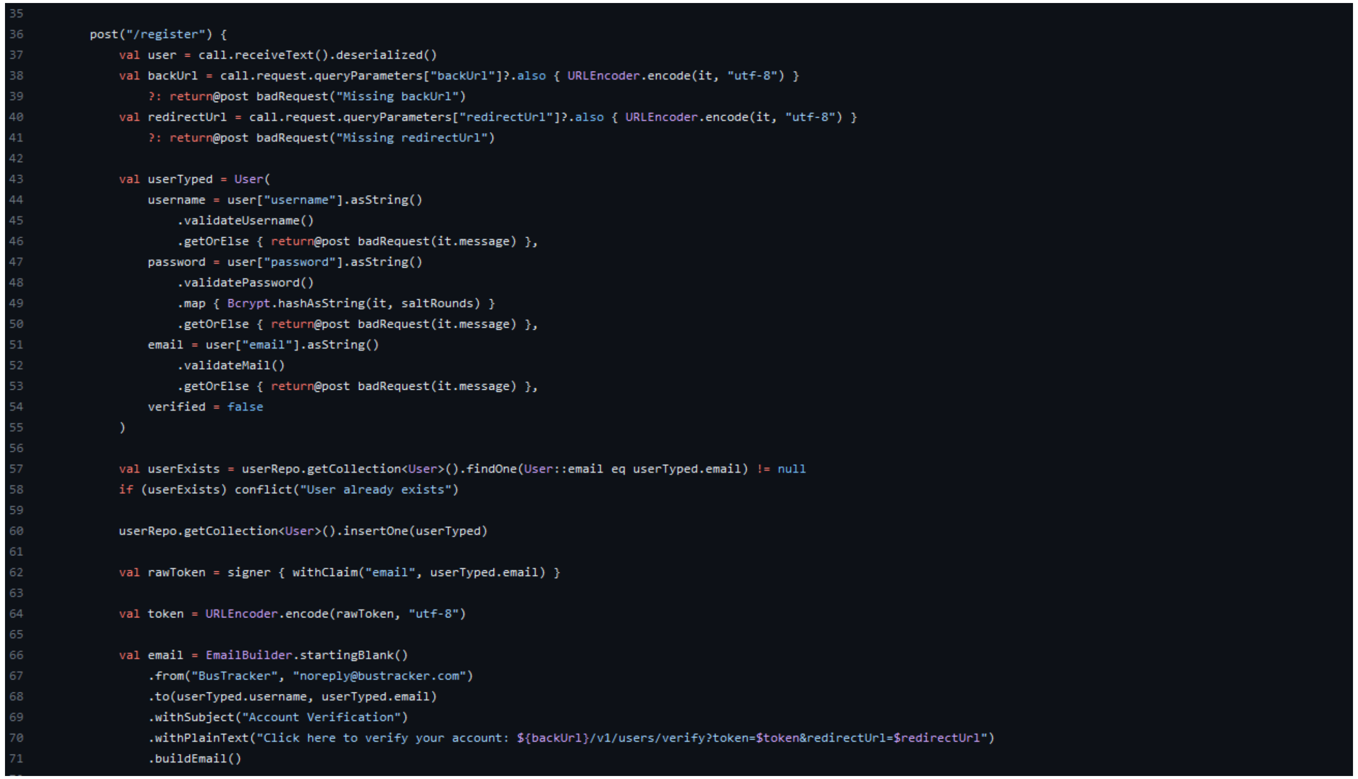
- Registro
- Inicio de sesión
- Verificar
- Restaurar contraseña

REGISTRO

- Introduce los datos
- Se validan y se envía el correo
- Usuario verifica su cuenta

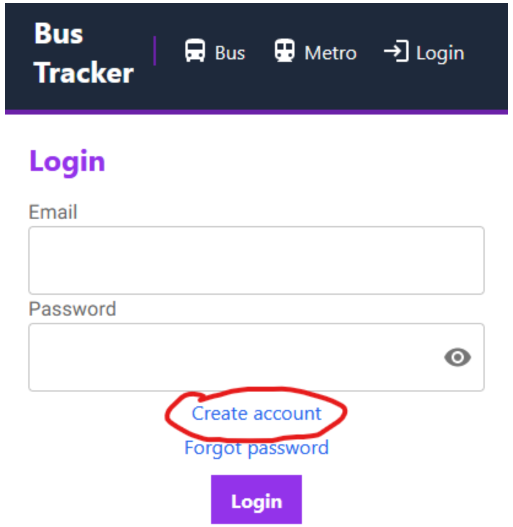
REGISTRO

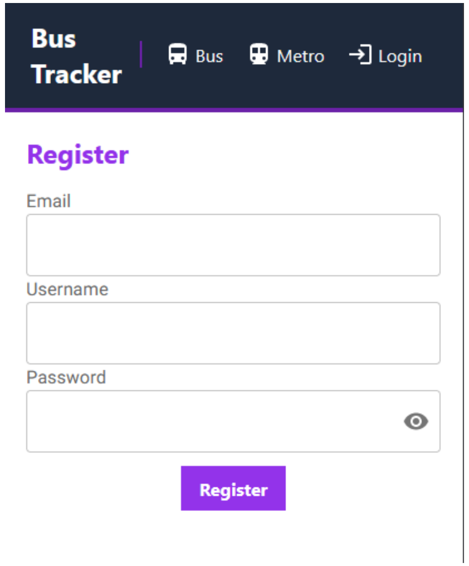
- Bcrypt para la contraseña
- Enviar correo

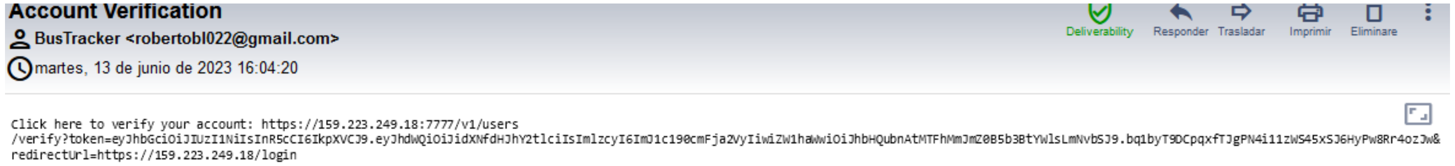


```
        post("/register") {
            val user = call.receiveText().deserialized()
            val backUrl = call.request.queryParameters["backUrl"]?.also { URLEncoder.encode(it, "utf-8") }
                ?: return@post badRequest("Missing backUrl")
            val redirectUrl = call.request.queryParameters["redirectUrl"]?.also { URLEncoder.encode(it, "utf-8") }
                ?: return@post badRequest("Missing redirectUrl")

            val userTyped = User(
                username = user["username"].asString()
                    .validateUsername()
                    .getOrElse { return@post badRequest(it.message) },
                password = user["password"].asString()
                    .validatePassword()
                    .map { Bcrypt.hashAsString(it, saltRounds) }
                    .getOrElse { return@post badRequest(it.message) },
                email = user["email"].asString()
                    .validateMail()
                    .getOrElse { return@post badRequest(it.message) },
                verified = false
            )

            val userExists = userRepo.getCollection<User>().findOne(User::email eq userTyped.email) != null
            if (userExists) conflict("User already exists")

            userRepo.getCollection<User>().insertOne(userTyped)

            val rawToken = signer { withClaim("email", userTyped.email) }

            val token = URLEncoder.encode(rawToken, "utf-8")

            val email = EmailBuilder.startingBlank()
                .from("BusTracker", "noreply@bustracker.com")
                .to(userTyped.username, userTyped.email)
                .withSubject("Account Verification")
                .withPlainText("Click here to verify your account: ${backUrl}/v1/users/verify?token=$token&redirectUrl=$redirectUrl")
                .buildEmail()
```

FRONT







Account Verification

BusTracker <robertobl022@gmail.com>

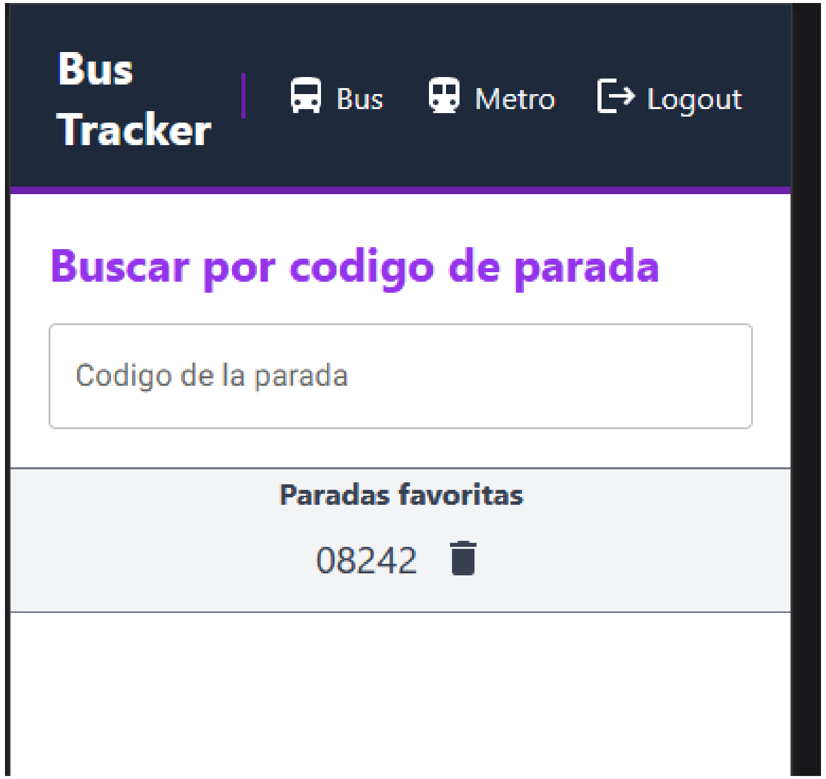
martes, 13 de junio de 2023 16:04:20

```
Click here to verify your account: https://159.223.249.18:7777/v1/users
/verify?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJidXNfdHJhY2tlciIsImlzcyI6ImJ1c190cmFja2VyIiwiZW1haWwiOiJhbHQubnAtMTFhMmJmZ0B5b3BtYWlsLmNvbSJ9.bq1byT9DCpqxfTJgPN4i11zWS45xSJ6HyPw8Rr4ozJw&
redirectUrl=https://159.223.249.18/login
```

LOGIN

- Introducen los datos
- Devuelve jwt con la información del usuario

FRONT



Buscar por codigo de parada

Codigo de la parada

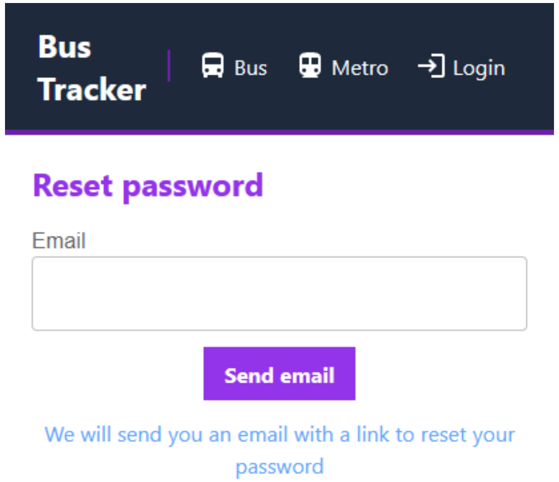
Paradas favoritas

08242

RESETEAR CONTRASEÑA

- Introducen los datos
- Se envía un correo con el link para resetear la contraseña
- Se resetea la contraseña

FRONT

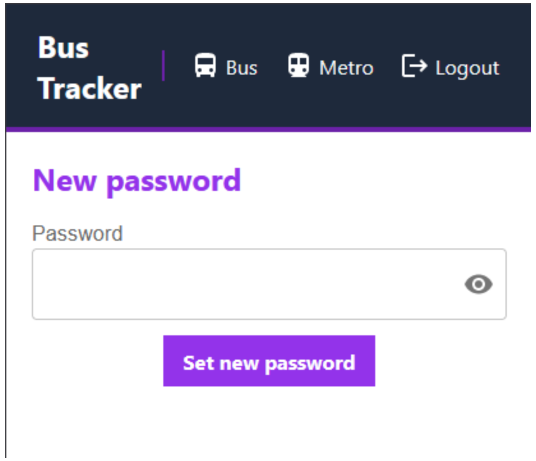


Reset password

Email

Send email

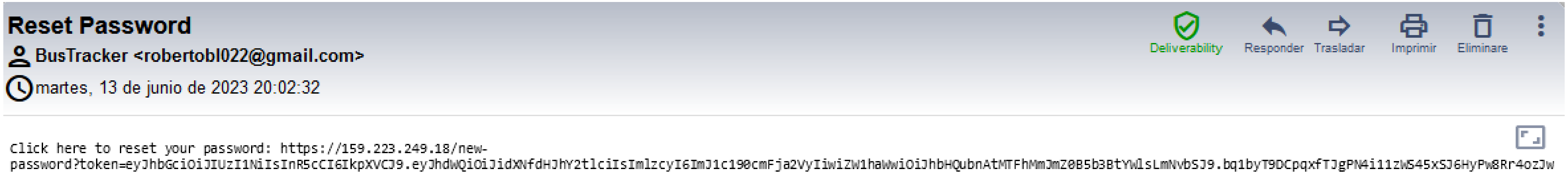
We will send you an email with a link to reset your password



New password

Password

Set new password



Reset Password

BusTracker <robertobl022@gmail.com>

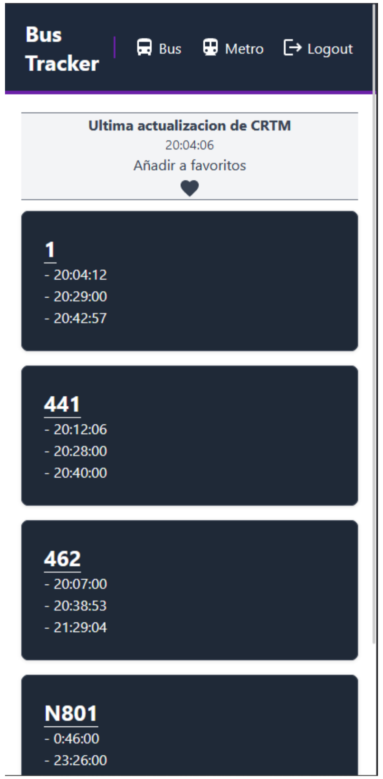
martes, 13 de junio de 2023 20:02:32

```
Click here to reset your password: https://159.223.249.18/new-
password?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJidXNfdHJhY2tlciIsImlzcyI6ImJ1c190cmFja2VyIiwiZW1haWwiOiJhbHQubnAtMTFhMmJmZ0B5b3BtYWlsLmNvbSJ9.bq1byT9DCpqxfTJgPN4i11zWS45xSJ6HyPw8Rr4ozJw
```

FAVORITOS

- Guardar
- Leer
- Borrar

FRONT



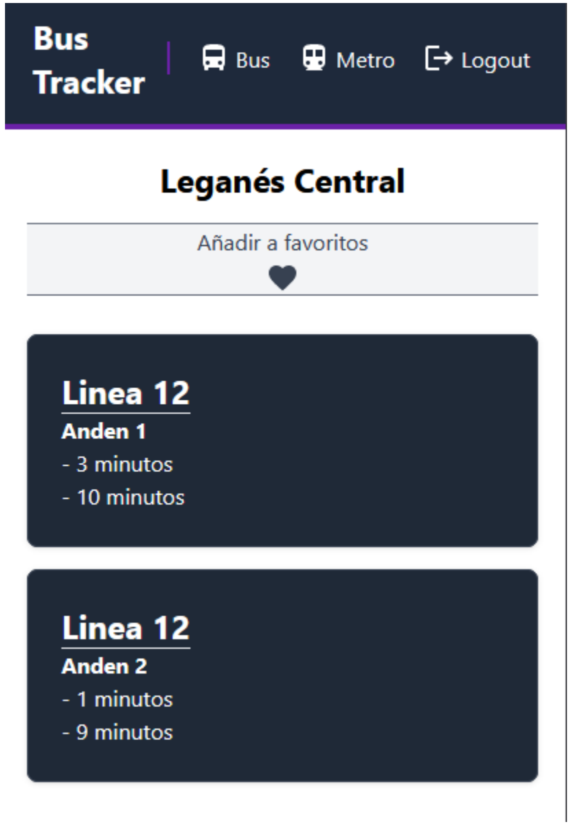
Ultima actualizacion de CRTM
20:04:06

1
- 20:04:12
- 20:29:00
- 20:42:57

441
- 20:12:06
- 20:28:00
- 20:40:00

462
- 20:07:00
- 20:38:53
- 21:29:04

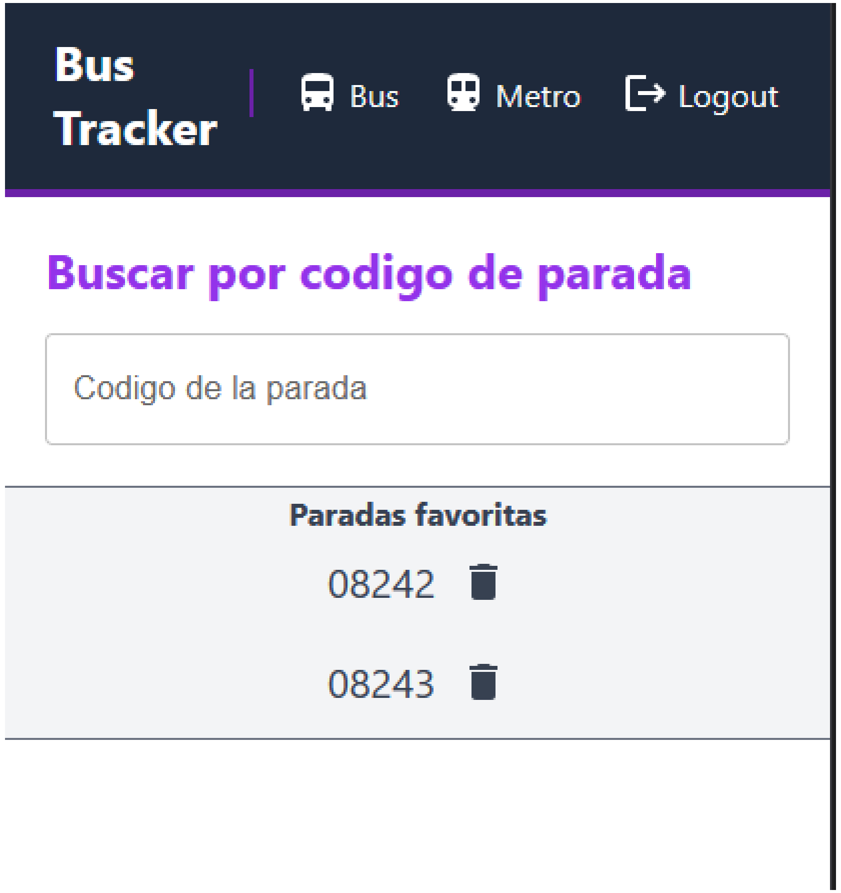
N801
- 0:46:00
- 23:26:00



Leganés Central

Linea 12
Anden 1
- 3 minutos
- 10 minutos

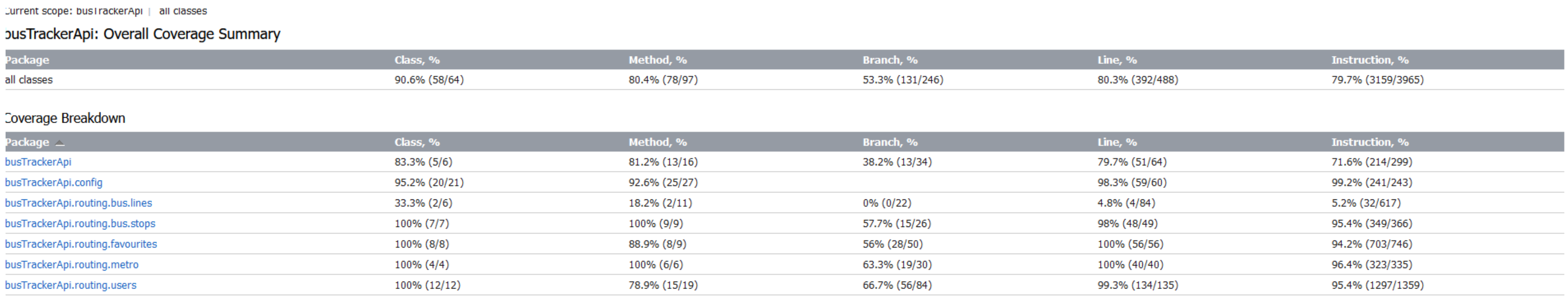
Linea 12
Anden 2
- 1 minutos
- 9 minutos



Buscar por codigo de parada

Paradas favoritas
08242
08243

TESTS



Current scope: busTrackerApi | all classes

busTrackerApi: Overall Coverage Summary

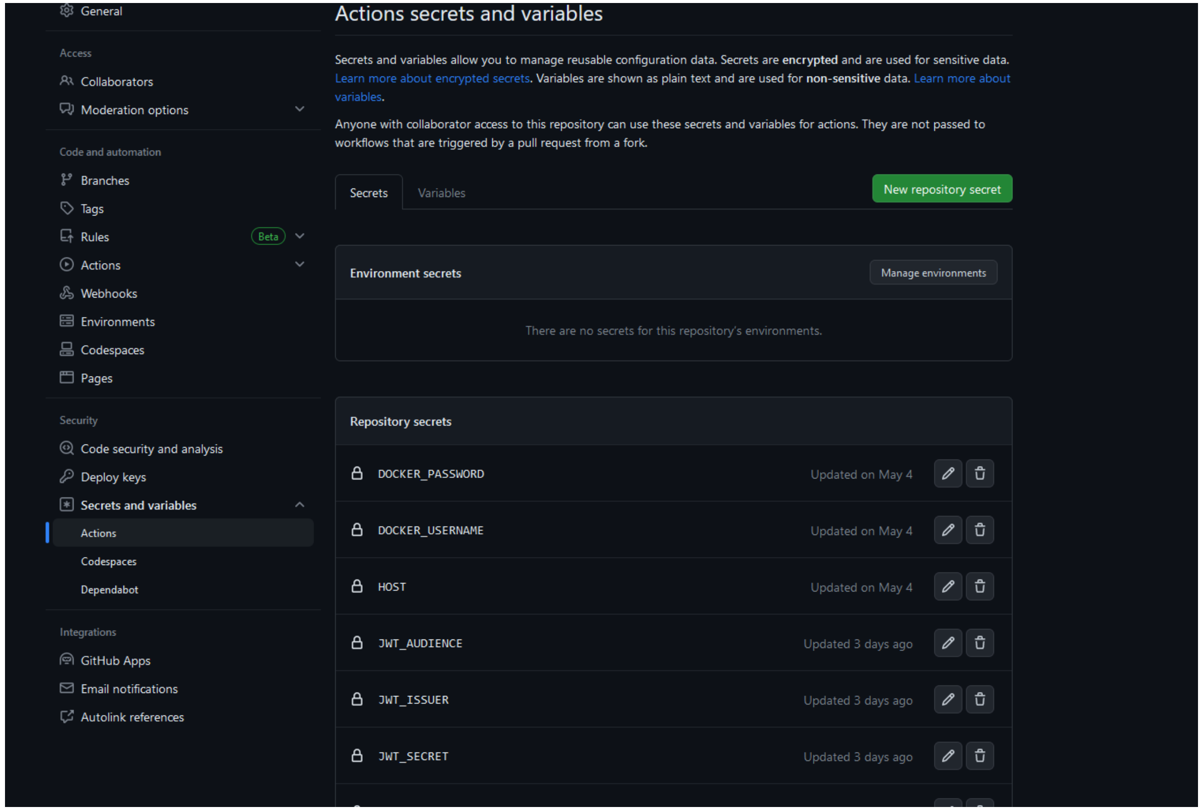
Package	Class, %	Method, %	Branch, %	Line, %	Instruction, %
all classes	90.6% (58/64)	80.4% (78/97)	53.3% (131/246)	80.3% (392/488)	79.7% (3159/3965)

Coverage Breakdown

Package	Class, %	Method, %	Branch, %	Line, %	Instruction, %
busTrackerApi	83.3% (5/6)	81.2% (13/16)	38.2% (13/34)	79.7% (51/64)	71.6% (214/299)
busTrackerApi.config	95.2% (20/21)	92.6% (25/27)		98.3% (59/60)	99.2% (241/243)
busTrackerApi.routing.bus.lines	33.3% (2/6)	18.2% (2/11)	0% (0/22)	4.8% (4/84)	5.2% (32/617)
busTrackerApi.routing.bus.stops	100% (7/7)	100% (9/9)	57.7% (15/26)	98% (48/49)	95.4% (349/366)
busTrackerApi.routing.favourites	100% (8/8)	88.9% (8/9)	56% (28/50)	100% (56/56)	94.2% (703/746)
busTrackerApi.routing.metro	100% (4/4)	100% (6/6)	63.3% (19/30)	100% (40/40)	96.4% (323/335)
busTrackerApi.routing.users	100% (12/12)	78.9% (15/19)	66.7% (56/84)	99.3% (134/135)	95.4% (1297/1359)

CI

Configurar variables de entorno



Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

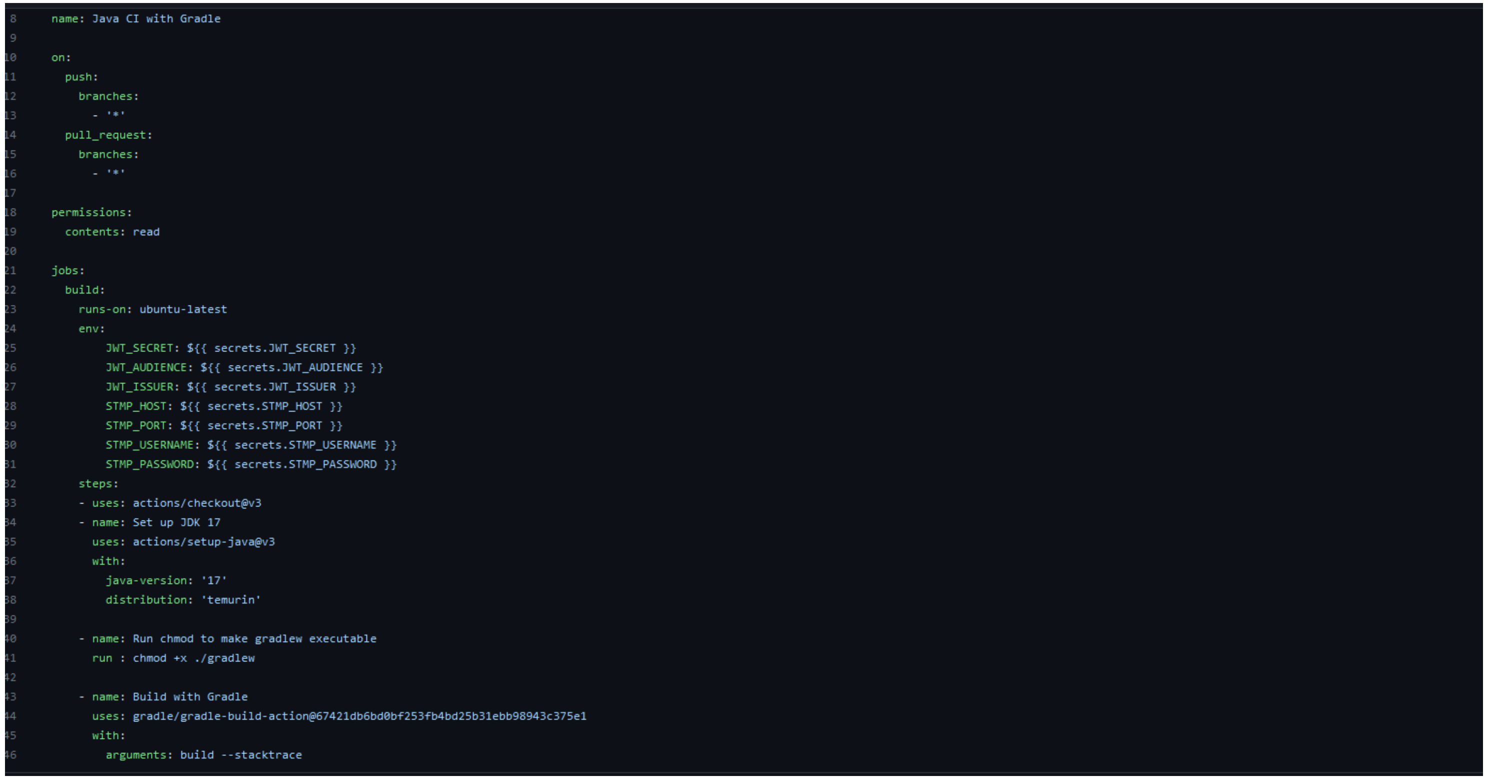
Environment secrets

There are no secrets for this repository's environments.

Repository secrets

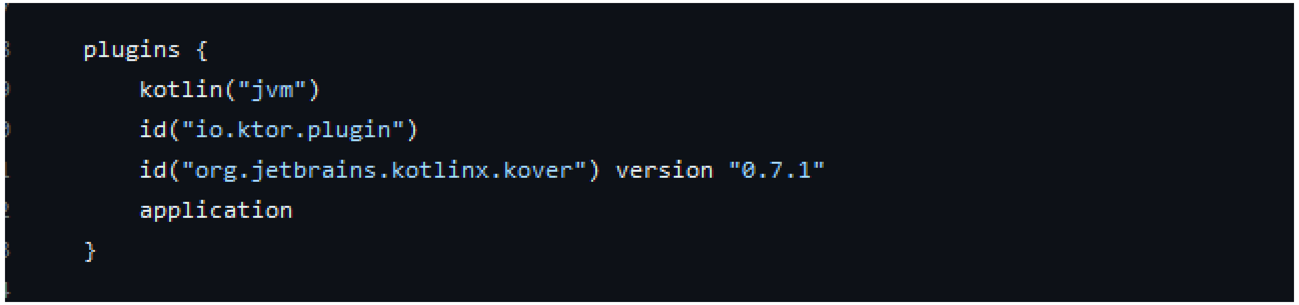
DOCKER_PASSWORD	Updated on May 4
DOCKER_USERNAME	Updated on May 4
HOST	Updated on May 4
JWT_AUDIENCE	Updated 3 days ago
JWT_ISSUER	Updated 3 days ago
JWT_SECRET	Updated 3 days ago

CI

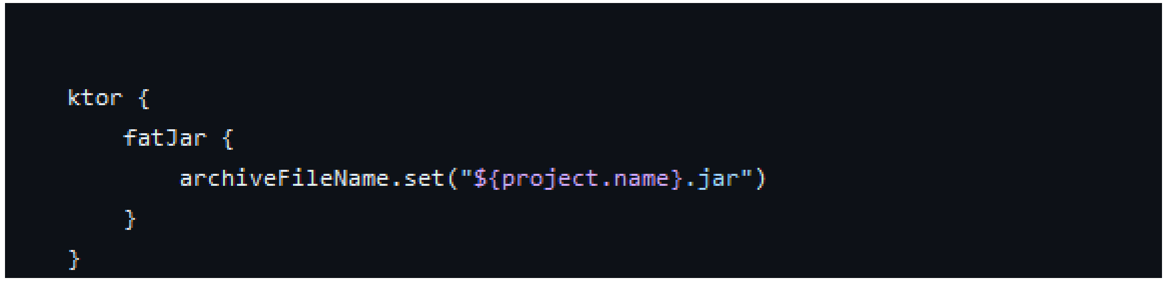


```yaml
name: Java CI with Gradle

on:
  push:
    branches:
      - '*'
  pull_request:
    branches:
      - '*'

permissions:
  contents: read

jobs:
  build:
    runs-on: ubuntu-latest
    env:
        JWT_SECRET: ${{ secrets.JWT_SECRET }}
        JWT_AUDIENCE: ${{ secrets.JWT_AUDIENCE }}
        JWT_ISSUER: ${{ secrets.JWT_ISSUER }}
        STMP_HOST: ${{ secrets.STMP_HOST }}
        STMP_PORT: ${{ secrets.STMP_PORT }}
        STMP_USERNAME: ${{ secrets.STMP_USERNAME }}
        STMP_PASSWORD: ${{ secrets.STMP_PASSWORD }}
    steps:
    - uses: actions/checkout@v3
    - name: Set up JDK 17
      uses: actions/setup-java@v3
      with:
        java-version: '17'
        distribution: 'temurin'

    - name: Run chmod to make gradlew executable
      run : chmod +x ./gradlew

    - name: Build with Gradle
      uses: gradle/gradle-build-action@67421db6bd0bf253fb4bd25b31ebb98943c375e1
      with:
        arguments: build --stacktrace
```

JAR

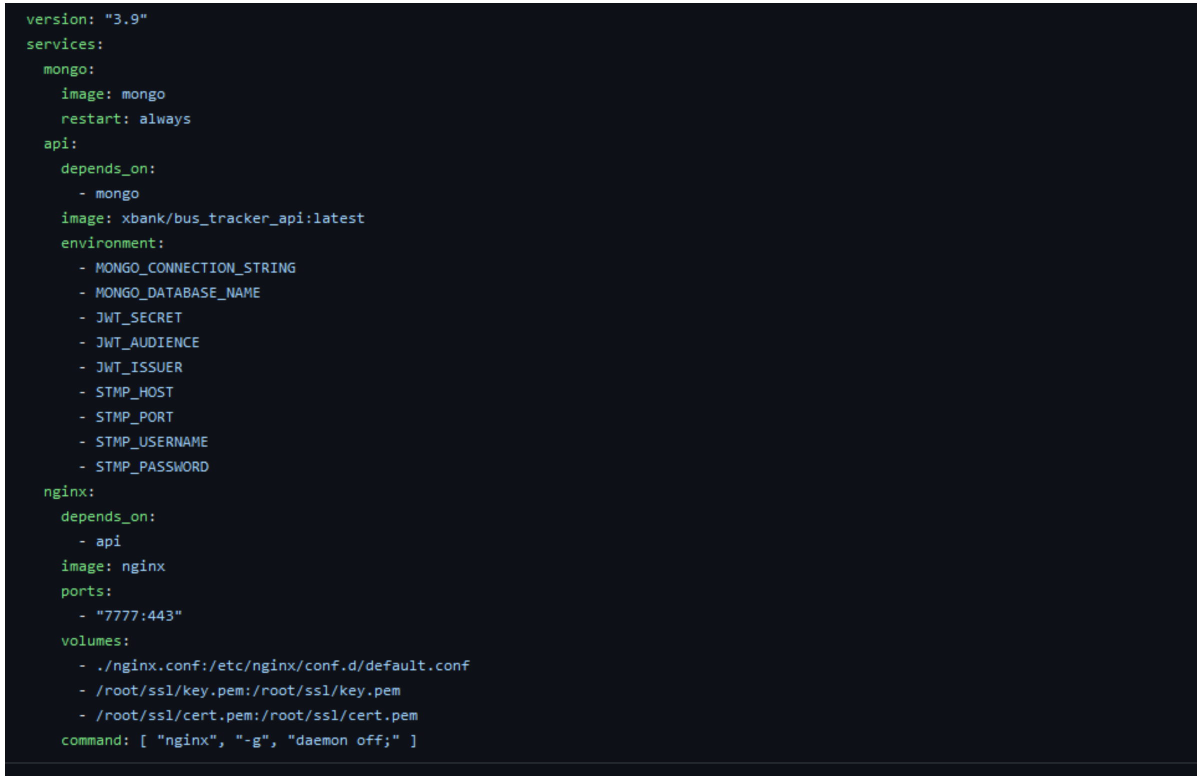


```
plugins {
    kotlin("jvm")
    id("io.ktor.plugin")
    id("org.jetbrains.kotlinx.kover") version "0.7.1"
    application
}
```



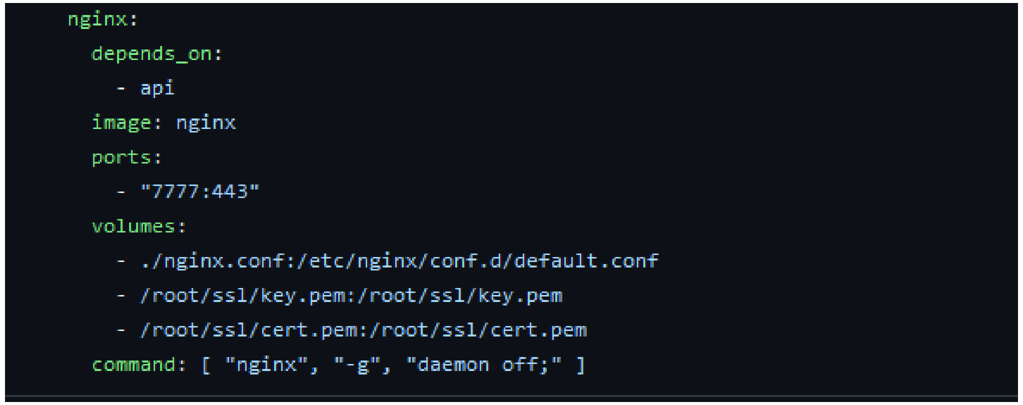
```
ktor {
    fatJar {
        archiveFileName.set("${project.name}.jar")
    }
}
```

DOCKER

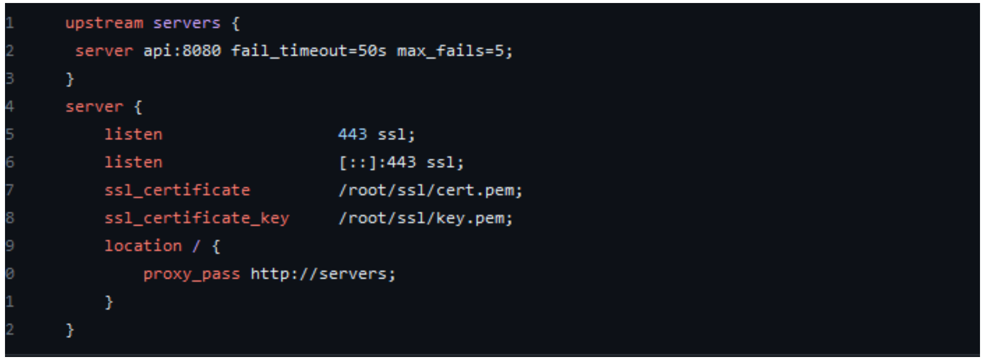


```yaml
version: "3.9"
services:
  mongo:
    image: mongo
    restart: always
  api:
    depends_on:
      - mongo
    image: xbank/bus_tracker_api:latest
    environment:
      - MONGO_CONNECTION_STRING
      - MONGO_DATABASE_NAME
      - JWT_SECRET
      - JWT_AUDIENCE
      - JWT_ISSUER
      - STMP_HOST
      - STMP_PORT
      - STMP_USERNAME
      - STMP_PASSWORD
  nginx:
    depends_on:
      - api
    image: nginx
    ports:
      - "7777:443"
    volumes:
      - ./nginx.conf:/etc/nginx/conf.d/default.conf
      - /root/ssl/key.pem:/root/ssl/key.pem
      - /root/ssl/cert.pem:/root/ssl/cert.pem
    command: [ "nginx", "-g", "daemon off;" ]
```

NGINX



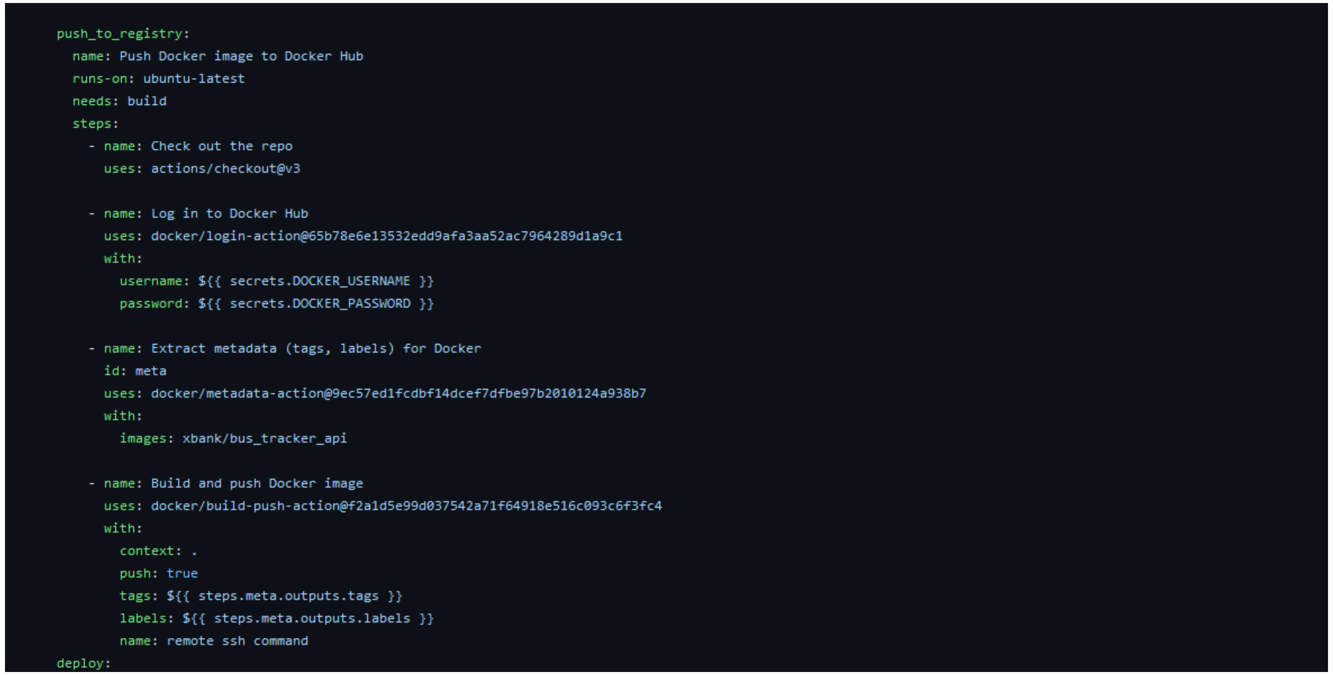
```yaml
nginx:
  depends_on:
    - api
  image: nginx
  ports:
    - "7777:443"
  volumes:
    - ./nginx.conf:/etc/nginx/conf.d/default.conf
    - /root/ssl/key.pem:/root/ssl/key.pem
    - /root/ssl/cert.pem:/root/ssl/cert.pem
  command: [ "nginx", "-g", "daemon off;" ]
```



```nginx
upstream servers {
 server api:8080 fail_timeout=50s max_fails=5;
}
server {
    listen                  443 ssl;
    listen                  [::]:443 ssl;
    ssl_certificate         /root/ssl/cert.pem;
    ssl_certificate_key     /root/ssl/key.pem;
    location / {
        proxy_pass http://servers;
    }
}
```

CD

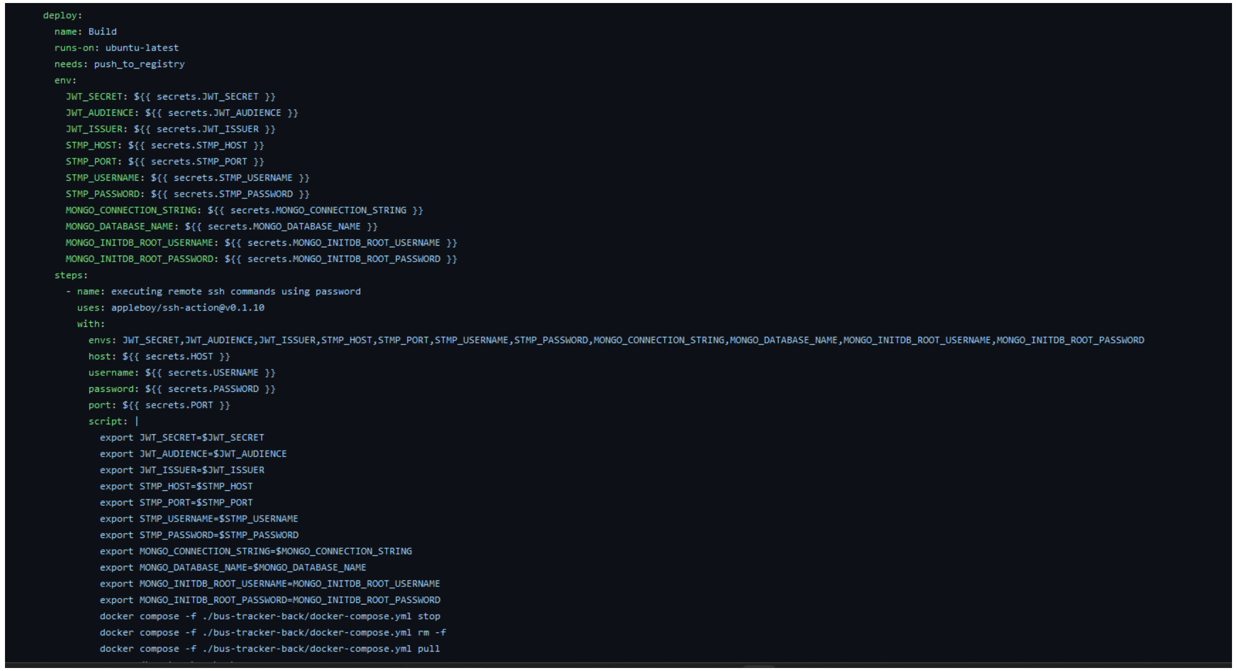
Primero se compila y se sube la imagen a Docker hub



```
push_to_registry:
  name: Push Docker image to Docker Hub
  runs-on: ubuntu-latest
  needs: build
  steps:
    - name: Check out the repo
      uses: actions/checkout@v3

    - name: Log in to Docker Hub
      uses: docker/login-action@65b78e6e13532edd9afa3aa52ac7964289d1a9c1
      with:
        username: ${{ secrets.DOCKER_USERNAME }}
        password: ${{ secrets.DOCKER_PASSWORD }}

    - name: Extract metadata (tags, labels) for Docker
      id: meta
      uses: docker/metadata-action@9ec57ed1fcdbf14dcef7dfbe97b2010124a938b7
      with:
        images: xbank/bus_tracker_api

    - name: Build and push Docker image
      uses: docker/build-push-action@f2a1d5e99d037542a71f64918e516c093c6f3fc4
      with:
        context: .
        push: true
        tags: ${{ steps.meta.outputs.tags }}
        labels: ${{ steps.meta.outputs.labels }}
        name: remote ssh command
deploy:
```

CD

Después se despliega en la maquina ejecutando el Docker compose



```
deploy:
  name: Build
  runs-on: ubuntu-latest
  needs: push_to_registry
  env:
    JWT_SECRET: ${{ secrets.JWT_SECRET }}
    JWT_AUDIENCE: ${{ secrets.JWT_AUDIENCE }}
    JWT_ISSUER: ${{ secrets.JWT_ISSUER }}
    STMP_HOST: ${{ secrets.STMP_HOST }}
    STMP_PORT: ${{ secrets.STMP_PORT }}
    STMP_USERNAME: ${{ secrets.STMP_USERNAME }}
    STMP_PASSWORD: ${{ secrets.STMP_PASSWORD }}
    MONGO_CONNECTION_STRING: ${{ secrets.MONGO_CONNECTION_STRING }}
    MONGO_DATABASE_NAME: ${{ secrets.MONGO_DATABASE_NAME }}
    MONGO_INITDB_ROOT_USERNAME: ${{ secrets.MONGO_INITDB_ROOT_USERNAME }}
    MONGO_INITDB_ROOT_PASSWORD: ${{ secrets.MONGO_INITDB_ROOT_PASSWORD }}
  steps:
    - name: executing remote ssh commands using password
      uses: appleboy/ssh-action@v0.1.10
      with:
        envs: JWT_SECRET,JWT_AUDIENCE,JWT_ISSUER,STMP_HOST,STMP_PORT,STMP_USERNAME,STMP_PASSWORD,MONGO_CONNECTION_STRING,MONGO_DATABASE_NAME,MONGO_INITDB_ROOT_USERNAME,MONGO_INITDB_ROOT_PASSWORD
        host: ${{ secrets.HOST }}
        username: ${{ secrets.USERNAME }}
        password: ${{ secrets.PASSWORD }}
        port: ${{ secrets.PORT }}
        script: |
          export JWT_SECRET=$JWT_SECRET
          export JWT_AUDIENCE=$JWT_AUDIENCE
          export JWT_ISSUER=$JWT_ISSUER
          export STMP_HOST=$STMP_HOST
          export STMP_PORT=$STMP_PORT
          export STMP_USERNAME=$STMP_USERNAME
          export STMP_PASSWORD=$STMP_PASSWORD
          export MONGO_CONNECTION_STRING=$MONGO_CONNECTION_STRING
          export MONGO_DATABASE_NAME=$MONGO_DATABASE_NAME
          export MONGO_INITDB_ROOT_USERNAME=MONGO_INITDB_ROOT_USERNAME
          export MONGO_INITDB_ROOT_PASSWORD=MONGO_INITDB_ROOT_PASSWORD
          docker compose -f ./bus-tracker-back/docker-compose.yml stop
          docker compose -f ./bus-tracker-back/docker-compose.yml rm -f
          docker compose -f ./bus-tracker-back/docker-compose.yml pull
```

GRACIAS