



EBook Gratis

APRENDIZAJE dropwizard

Free unaffiliated eBook created from
Stack Overflow contributors.

#dropwizard

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con dropwizard.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Capítulo 2: Creación de comandos.....	3
Examples.....	3
Subcomandos.....	3
Capítulo 3: Creando Recursos.....	5
Examples.....	5
Peticiones GET.....	5
Respuestas personalizadas.....	5
Creditos.....	7

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [dropwizard](#)

It is an unofficial and free dropwizard ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official dropwizard.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con dropwizard

Observaciones

Dropwizard reúne **bibliotecas estables y maduras** del ecosistema de Java en un paquete **simple y ligero** que le permite concentrarse en hacer las cosas.

Dropwizard tiene compatibilidad inmediata con **configuraciones** sofisticadas , **métricas de aplicaciones, registro, herramientas operativas** y mucho más, lo que le permite a usted y su equipo enviar un servicio web de calidad de producción en el menor tiempo posible.

Usa:

- [Embarcadero](#) para HTTP
- [Jersey](#) para descanso
- [Jackson](#) para JSON
- [Métricas](#) para métricas

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar Dropwizard.

Lea [Empezando con dropwizard en línea](#):

<https://riptutorial.com/es/dropwizard/topic/3884/empezando-con-dropwizard>

Capítulo 2: Creación de comandos

Examples

Subcomandos

Este ejemplo muestra cómo definir subcomandos para un comando dado y cómo asociar fácilmente un controlador de comandos. Este ejemplo define un comando `thing` con los subcomandos `get` y `set`.

```
package things;

import io.dropwizard.cli.Command;
import io.dropwizard.setup.Bootstrap;
import net.sourceforge.argparse4j.inf.FeatureControl;
import net.sourceforge.argparse4j.inf.Namespace;
import net.sourceforge.argparse4j.inf.Subparser;

public class ThingCommand extends Command {

    private static final String THING_COMMAND_NAME = "thing";

    private static final String THING_SUBCOMMAND_GET = "get";
    private static final String THING_SUBCOMMAND_SET = "set";

    public ThingCommand() {
        super(THING_COMMAND_NAME, "Thing management.");
    }

    @Override
    public void configure(Subparser subparser) {
        addSubCommand(subparser.addSubparsers().addParser(THING_SUBCOMMAND_GET)
            .help("Gets a thing."), new GetCommand());

        addSubCommand(subparser.addSubparsers().addParser(THING_SUBCOMMAND_SET)
            .help("Sets a thing."), new SetCommand());
    }

    private void addSubCommand(Subparser parser, Command cmd) {
        // associate the subcommand with a Command object
        parser.addArgument("--subcommand").help(FeatureControl.SUPPRESS).setDefault(cmd);
        parser.description(cmd.getDescription());
        cmd.configure(parser);
    }

    @Override
    public void run(Bootstrap<?> bootstrap, Namespace namespace) throws Exception {
        Command cmd = namespace.get("subcommand");
        assert cmd != null;
        cmd.run(bootstrap, namespace);
    }

    class GetCommand extends Command {

        public GetCommand() {
            super(THING_SUBCOMMAND_GET, "Gets a thing.");
        }
    }
}
```

```

    }

    @Override
    public void configure(Subparser cmd) {
        super.configure(cmd);
        cmd.addArgument("--name").type(String.class).required(true).help("Name of the
thing");
    }

    @Override
    public void run(Bootstrap<?> bootstrap, Namespace namespace) throws Exception {
        String thingName = namespace.getString("name");
        System.out.println("Getting the thing named: " + thingName);
    }
}

class SetCommand extends Command {
    // ...
}
}

```

Aquí hay un ejemplo de invocación del comando (asumiendo una aplicación llamada `myapp`):

```

$ bin/myapp thing get --name mything
Getting the thing named: mything

```

Lea Creación de comandos en línea: <https://riptutorial.com/es/dropwizard/topic/9323/creacion-de-comandos>

Capítulo 3: Creando Recursos

Examples

Peticiones GET

```
@Path("/hello")
public class HelloResource {

    /**
     * A request to /hello would get the response "Hello World"
     */
    @GET
    public String exampleGet() {
        return "Hello World";
    }

    /**
     * A request to /hello/bob would get the response "Hello bob"
     */
    @Path("/{name}")
    @GET
    public String exampleWithParameter(@PathParam("name") String name) {
        return "Hello " + name;
    }
}
```

Respuestas personalizadas

```
import javax.ws.rs.*;
import javax.ws.rs.core.Response;
import java.util.HashMap;
import java.util.Map;
import java.util.Objects;

@Path("/alphabet/{letter}")
public class AlphabetResource {

    private final Map<String, String> alphabet;

    public AlphabetResource() {
        this.alphabet = new HashMap<>();
        this.alphabet.put("A", "Apple");
    }

    @GET
    public Response get(@PathParam("letter") String letter) {
        if (alphabet.containsKey(letter)) {
            return Response.ok(alphabet.get(letter)).build();
        } else {
            return Response.status(Response.Status.NOT_FOUND).build();
        }
    }
}
```

```

@PUT
public Response put(@PathParam("letter") String letter, String value) {
    if (alphabet.containsKey(letter)) {
        return Response.status(Response.Status.CONFLICT).build();
    } else {
        alphabet.put(letter, value);
        return Response.noContent().build();
    }
}

@POST
public Response post(@PathParam("letter") String letter, String value) {
    if (alphabet.containsKey(letter) && Objects.equals(alphabet.get(letter), value)) {
        return Response.notModified().build();
    } else {
        alphabet.put(letter, value);
        return Response.noContent().build();
    }
}
}

```

Estos pueden ser llamados con los siguientes comandos curl

```

curl -v -X GET http://localhost:8080/alphabet/A
curl -v -X PUT http://localhost:8080/alphabet/A -d "Avacado"
curl -v -X PUT http://localhost:8080/alphabet/B -d "Banana"
curl -v -X POST http://localhost:8080/alphabet/A -d "Apple"
curl -v -X POST http://localhost:8080/alphabet/A -d "Avacado"

```

Lea Creando Recursos en línea: <https://riptutorial.com/es/dropwizard/topic/7633/creando-recursos>

Creditos

S. No	Capítulos	Contributors
1	Empezando con dropwizard	Community , Tom Reay
2	Creación de comandos	Eron Wright
3	Creando Recursos	Tom Reay