Chapter 1: Introduction to Microservices
Chapter 2: Developing a Microservice with the Hyper Crate

Rust Microservice

Rust Microservice - Mozilla Firefox
Mozilla Firefox

localhost:8080/funny_cats  ×

404
GET    funny...  localhost:8080/funny...  0 ms

One request | 0 B / 82 B transferred | Finish: 0 ms | DOMContentLoaded: 115 ms | load: 177 ms

Mozilla Firefox

localhost:8080/users/  ×

0, 2
Chapter 3: Logging and Configuring Microservice

Mozilla Firefox

File Edit View History Bookmarks Tools Help

localhost:8080/

RUST_LOG=debug cargo run

INFO random_service_with_logging > Rand Microservice - v0.1.0
DEBUG random_service_with_logging > Trying to bind server to address: 127.0.0.1:8080
INFO random_service_with_logging > Used address: 127.0.0.1:8080
DEBUG random_service_with_logging > Run!
DEBUG hyper::proto::h1::io > read 747 bytes
DEBUG hyper::proto::h1::io > parsed 9 headers
DEBUG hyper::proto::h1::conn > incoming body is empty
DEBUG random_service_with_logging > Generated value is: 106
DEBUG hyper::proto::h1::io > flushed 78 bytes
DEBUG hyper::proto::h1::io > read 747 bytes
DEBUG hyper::proto::h1::io > parsed 9 headers
DEBUG hyper::proto::h1::io > incoming body is empty
DEBUG random_service_with_logging > Generated value is: 142
DEBUG hyper::proto::h1::io > flushed 78 bytes

Rust Microservice

Finished dev [unoptimized + debuginfo] target(s) in 0.10s
Running "target/debug/random-service-with-config"
INFO 2018-07-26T16:34:32Z: random_service_with_config: Rand Microservice - v0.1.0
DEBUG 2018-07-26T16:34:32Z: random_service_with_config: Trying to bind server to address: 0.0.0.0:9876
INFO 2018-07-26T16:34:32Z: random_service_with_config: Used address: 0.0.0.0:9876
DEBUG 2018-07-26T16:34:32Z: random_service_with_config: Run!
DEBUG 2018-07-26T16:34:32Z: tokio_reactor::background: starting background reactor
Chapter 4: Data Serialization and Deserialization with the Serde Crate

No images
Chapter 5: Understanding Asynchronous Operations with Futures Crate
Chapter 6: Reactive Microservices - Increasing Capacity and Performance

No images
Chapter 7: Reliable Integration with Databases

No images
Chapter 8: Interaction to Database with Object-Relational Mapping
Chapter 9: Simple REST Definition and Request Routing with Frameworks
Chapter 10: Background Tasks and Thread Pools in Microservices

No images
Chapter 11: Involving Concurrency with Actors and the Actix Crate

No images
## Chapter 12: Scalable Microservices Architecture

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Timestamp</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>89a6fd6129d047039402f073daa33148</td>
<td>2019-01-18 19:18:02.308320080 UTC</td>
<td>in progress</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Timestamp</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>89a6fd6129d047039402f073daa33148</td>
<td>2019-01-18 19:18:02.308320080 UTC</td>
<td>done: <a href="https://www.rust-lang.org/">https://www.rust-lang.org/</a></td>
</tr>
</tbody>
</table>
Chapter 13: Testing and Debugging Rust Microservices

http://localhost:7000/api/signin

POST http://localhost:7000/api/signin

Params
Authorization
Headers (1)
Body
Pre-request Script
Tests
Cookies
Code
Comments (0)

Postman

Params
Authorization
Headers (1)
Body
Pre-request Script
Tests
Cookies
Code
Comments (0)

Body
Cookies (1)
Headers (8)
Test Results

Name
Value
Domain
Path
Expires
HttpOnly
Secure

auth-example
mysqlWRRH3NnDs/sy/Deo5Wm4LM
w0zK0QoWkUn
o1nwKvqg5UqofE
DN9We3RN3Z7N3
e6ly7aLfe5cpoP
2fM9g---

localhost
/
true
false
<table>
<thead>
<tr>
<th>Path</th>
<th>Method</th>
<th>Status</th>
<th>Size</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://localhost:7000/">http://localhost:7000/</a></td>
<td>GET</td>
<td>302</td>
<td>0</td>
<td>34ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/index.html">http://localhost:7000/index.html</a></td>
<td>GET</td>
<td>200</td>
<td>271b</td>
<td>41ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/style.css">http://localhost:7000/style.css</a></td>
<td>GET</td>
<td>404</td>
<td>0</td>
<td>53ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/script.js">http://localhost:7000/script.js</a></td>
<td>GET</td>
<td>200</td>
<td>303b</td>
<td>52ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/api/members">http://localhost:7000/api/members</a></td>
<td>GET</td>
<td>200</td>
<td>103b</td>
<td>38ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/style.css">http://localhost:7000/style.css</a></td>
<td>GET</td>
<td>404</td>
<td>0</td>
<td>24ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/api/comments">http://localhost:7000/api/comments</a></td>
<td>GET</td>
<td>200</td>
<td>103b</td>
<td>33ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/api/signup">http://localhost:7000/api/signup</a></td>
<td>POST</td>
<td>302</td>
<td>45b</td>
<td>917ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/login.html">http://localhost:7000/login.html</a></td>
<td>GET</td>
<td>200</td>
<td>274b</td>
<td>30ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/style.css">http://localhost:7000/style.css</a></td>
<td>GET</td>
<td>404</td>
<td>0</td>
<td>24ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/api/comments">http://localhost:7000/api/comments</a></td>
<td>GET</td>
<td>200</td>
<td>103b</td>
<td>33ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/api/signin">http://localhost:7000/api/signin</a></td>
<td>POST</td>
<td>302</td>
<td>45b</td>
<td>928ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/comments.html">http://localhost:7000/comments.html</a></td>
<td>GET</td>
<td>200</td>
<td>264b</td>
<td>35ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/style.css">http://localhost:7000/style.css</a></td>
<td>GET</td>
<td>404</td>
<td>0</td>
<td>26ms</td>
</tr>
<tr>
<td><a href="http://localhost:7000/api/comments">http://localhost:7000/api/comments</a></td>
<td>GET</td>
<td>200</td>
<td>103b</td>
<td>37ms</td>
</tr>
</tbody>
</table>

**Request**

POST http://localhost:7000/api/signup HTTP/1.1

**Host**

localhost

**User-Agent**

Mozilla/5.0 (X11; Fedora; Linux x86_64; rv:64.0) Gecko/20100101 Firefox/64.0

**Accept**

text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

**Accept-Language**
en-US,en;q=0.5

**Accept-Encoding**
gzip, deflate

**Referer**

http://localhost:7700/login.html

**Content-Type**

application/x-www-form-urlencoded

**Content-Length**

45

**Connection**

keep-alive

**Cookie**

auth-example:tgBqtwImFzcxBzziPIapsow/cpm=xyhJU60at1yh0FN6TfYy3TFIbt08+31vzi23PCnQ0KrvTSw49bo2TRIp2w==

**Upgrade-Insecure-Requests**

1

**email:** userproxy@example.com

**password:** secret

**View:** auto • ▲ ▼ URLEncoded form
fn comments(req: HttpRequest<State>) -> FutureResponse<HttpResponse> {
    debug("/api/comments called");
    let url = format!("{}\{:id\}", req.state().content());
    let Tut = get_req(&url)
        .map(|data| {
            let mut response = HttpResponse::ok().body(data);
            Box::new(tut)
        });
    fn healthcheck(req: HttpRequest<State>) -> &'static str {
        "Router Microservice"
    }
    //[derive(Debug, Deserialize)]
    struct Config {
        address: Option<String>,
        users: Option<String>
    }
    PROBLEMS OUTPUT DEBUG CONSOLE ...
    microservices/router/target/debug/router-microservice "
    (lldb) command script import
    
    Listening on port 34519
Chapter 14: Optimization of Microservices

<table>
<thead>
<tr>
<th>generating response</th>
<th>database query</th>
<th>incoming request</th>
</tr>
</thead>
</table>

Chapter 15: Packing Servers to Containers

No images
Chapter 16: DevOps of Rust Microservices - Continuous Integration and Delivery
Install Steps For First-time Run

If you're running Gogs inside Docker, please read Guidelines carefully before you change anything in this page!

Database Settings

Gogs requires MySQL, PostgreSQL, SQLite3, MSSQL or TiDB.

Database Type*  SQLite3

Path*  data/gogs.db

The file path of SQLite3 database.
Please use absolute path when you start as service.

Application General Settings

Application Name*  Gogs

Put your organization name here huge and loud!

Repository Root Path*  /data/git/gogs-repositories

All Git remote repositories will be saved to this directory.
Sign Up

Username: developer
Email: developer@example.com
Password: ********
Re-Type: ********
Captcha: 533413

Create New Account

Already have an account? Sign in now!
No Description

1 Commits
1 Branches
0 Releases
TeamCity First Start

Please review the settings below before proceeding with the first TeamCity start.

TeamCity server stores server configuration settings, project definitions, build results and caches on disk in a Data Directory.

Location of the Data Directory: /data/teamcity_server/datadir

If you already worked with TeamCity and want to use existing directory or you want to use another location for creating fresh setup, check the documentation to change the directory location.

Proceed

TeamCity 2018.2.1 (build 61078)
Database connection setup

TeamCity server stores builds history and users-related data in an SQL database.

**Select the database type**: Internal (HSQLDB)

The internal database suits evaluation purposes only and is not intended for production. We strongly recommend using an external database in a production environment.

You can start with the internal database and then migrate the data to an external one after successful evaluation.

Proceed

TeamCity 2018.2.1 (build 61078)
Create Administrator Account

Username
admin

Password
●●●●●●

Confirm password
●●●●●●

Create Account

Login as Super user

Version 2018.2.1 (build 61078)
Agents

Install Build Agents

There are 1 currently unauthorized agent. Agent licenses left: 3.

Authorize

172.20.0.4

Connected

Disconnected

Unauthorized

Pools

Parameters Report

Matrix

Statistics

Agent Push

Last communication a few seconds ago

Create Project

Create Project

From a repository URL

Manually

Parent project:

<Root project>

Repository URL:

http://git-server:3000/developer/microservice.git

A VCS repository URL. Supported formats: http(s), svn, gitURL, etc. as well as URLs in Maven format.

Username:

developer

Provide username if access to repository requires authentication.

Password:

******

Provide password if access to repository requires authentication.
Create Project From URL

✓ The connection to the VCS repository has been verified

**Project name:**  Microservice

**Build configuration name:**  Build

**VCS root:**  (Git) http://git-server:3000/developer/microservice.git

**Proceed**  **Cancel**
New Build Step

Runnertype: Command Line
Simple command execution

Stepname: Format Check
Optional, specify to distinguish this build step from other steps.

Run: Custom script

Custom script:
```
cargo fmt -- --check
```

A platform-specific script, which will be executed as a .cmd file on Windows or as a shell script in Unix-like environments.
New Build Step

Runnertype: Command Line
Simple command execution

Stepname: Build
Optional, specify to distinguish this build step from other steps.

Run: Custom script

Custom script:
```
cargo build
```

A platform-specific script, which will be executed as a .cmd file on Windows or as a shell script in Unix-like environments.
**Build Configuration**

**Build Steps**

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool.

<table>
<thead>
<tr>
<th>Build Step</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Format Check</td>
<td>Command Line: <code>cargo fmt --check</code></td>
<td>Execute if all previous steps finished successfully</td>
</tr>
<tr>
<td>2. Build</td>
<td>Command Line: <code>cargo build</code></td>
<td>Execute if all previous steps finished successfully</td>
</tr>
</tbody>
</table>

**Build step settings updated.**
The build is removed from the queue to be prepared for the start.
Starting the build on the agent rustci_agent_1.
Clearing temporary directory: /opt/buildagent/temp/buildTemp
Publishing internal artifacts (1s)
Will perform clean checkout, Reason: Checkout directory is empty or doesn't exist
Checkout directory: /opt/buildagent/work/6bb39fb1710c6cb
Updating sources: auto checkout (on server) (8s)
Step 1/2: Format Check (Command Line) (1s)
Step 2/2: Build (Command Line) (running for 33s)
[Step 2/2] Starting: /opt/buildagent/temp/agentTemp/custom_script7816632993483962962
[Step 2/2] in directory: /opt/buildagent/work/6bb39fb1710c6cb
[Step 2/2] Updating crates.io index
Running...
<table>
<thead>
<tr>
<th>Build</th>
<th>Running</th>
<th>deniskolodin (1)</th>
<th>14m:52s left</th>
<th>20 minutes age (14m:39s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Microservice**

- No hidden
Chapter 17: Bounded Microservices with AWS Lambda

Lambda Management Console - Mozilla Firefox

Author from scratch

Name
minimal-lambda

Runtime
You can select a supported AWS Lambda runtime or provide your own runtime as part of the function deployment package or Lambda layer after creating the function.
Use custom runtime in function code or layer

Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. Learn more about Lambda execution roles.

Create a new role from one or more templates.

Role name
Enter a name for your new role.
minimal-lambda-role

This new role will be scoped to the current function. To use it with other functions, you can modify it in the IAM console.

Policy templates
Choose one or more policy templates. A role will be generated for you before your function is created. Learn more about the permissions that each policy template will add to your role.
### Function code

<table>
<thead>
<tr>
<th>Code entry type</th>
<th>Runtime</th>
<th>Handler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload a zip file</td>
<td>Use custom runtime</td>
<td>hello.handler</td>
</tr>
</tbody>
</table>

**Function package**

- **Upload minimal-lambda.zip (9.4 MB)**

For files larger than 10 MB, consider uploading using Amazon S3.
Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

- Create new test event
- Edit saved test events

Event template
Hello World

Event name
uniform

```
1
2    "distribution": "uniform",
3    "parameters": {
4        "start": 0,
5        "end": 100
6    }
7 }
```
Execution result: succeeded (logs)

The area below shows the result returned by your custom runtime function execution. Learn more about returning results from your function.

```json
{
    "value": 56
}
```

Summary

<table>
<thead>
<tr>
<th>Code SHA-256</th>
<th>Request ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>JivXyU7v27XkP0C0W/cWh3IQD56QxrDZ6WKhIO+</td>
<td>905e3963-6cb5-40ba-b00a-a0244cf48954</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Duration</th>
<th>Billed duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.22 ms</td>
<td>100 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resources configured</th>
<th>Max memory used</th>
</tr>
</thead>
<tbody>
<tr>
<td>128 MB</td>
<td>21 MB</td>
</tr>
</tbody>
</table>

Log output

The section below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. Click here to view the CloudWatch log group.

```
START RequestId: 905e3963-6cb5-40ba-b00a-a0244cf48954 Version: $LATEST
2019-01-25 16:10:35 INFO [lambda_runtime_core::runtime] Received new event with AWS request id: 905e3963-6cb5-40ba-b00a-a0244cf48954
```
Welcome! Click the map to set your pickup location.

You are authenticated. Click to see your auth token.

Shadowfax, your White unicorn, is on his way.

Shadowfax has arrived. Giddy up!
CloudWatch Management Console - Mozilla Firefox

CloudWatch > Log Groups > /aws/lambda/ust-sls-dev-lambda_1 > 2019/01/25/[$LATEST]01a201b5d1c458fda1fd854faa09ef

Expand all  ▪ Row  ▪ Text  ▪ ▪ ▪

Filter events

<table>
<thead>
<tr>
<th>Time (UTC +00:00)</th>
<th>Message</th>
</tr>
</thead>
</table>
| 22:30:06         | 2019-01-25 22:30:06 INFO [lambda_runtime:runtime] Received new event with AWS request ID: 1d9abf1d-
| 22:30:06         | 2019-01-25 22:30:06 DEBUG [lambda_1:request] Full request: |
| 22:30:06         | method: POST |
| 22:30:06         | final_url: https://dynamodb.us-east-1.amazonaws.com/ |
| 22:30:06         | payload: {"Item":{"User":{"B":"null","S":"fe7e941e-beaf-4046-a980-5d8997287ae4"},"RId":"B""null","S":456",...

Feedback  ▪ English (US)