

Quarkus - Funqy HTTP (Standalone)

The guide walks through quickstart code to show you how you can deploy Funqy as a standalone service and invoke on Funqy functions using HTTP.



The Funqy HTTP binding is not a replacement for REST over HTTP. Because Funqy needs to be portable across a lot of different protocols and function providers its HTTP binding is very minimalistic and you will lose REST features like linking and the ability to leverage HTTP features like cache-control and conditional GETs. You may want to consider using Quarkus's JAX-RS, Spring MVC, or Vert.x Web Reactive Routes support instead, although Funqy will have less overhead than these alternatives (except Vert.x which is still super fast).

Prerequisites

To complete this guide, you need:

- less than 15 minutes
- Read about [Funqy Basics](#). This is a short read!
- an IDE
- JDK 1.8+ installed with `JAVA_HOME` configured appropriately
- Apache Maven 3.6.3

The Quickstart

Clone the Git repository: `git clone https://github.com/quarkusio/quarkus-quickstarts.git`, or download an [archive](#).

The solution is located in the `funqy-http-quickstart` directory.

The Code

If you look at the Java code, you'll see that there is no HTTP specific API. Its just simple Java methods annotated with `@Funq`. Simple, easy, straightforward.

Build Project

```
mvn clean quarkus:dev
```

This starts your functions in Quarkus dev mode.

Execute Funqy HTTP functions

Funqy HTTP input and output is always JSON. The Jackson JSON parser is used behind the scenes and you can use Jackson annotations to fine tune your JSON mappings for your input and output objects.

The HTTP POST method can be used to execute any Funqy function. HTTP GET can be used if your function does not have any input. No other HTTP method can be used besides POST and GET.

The URL path to execute a function is the function name. For example if your function name is `foo` then the URL path to execute the function would be `/foo`.

You can invoke the `hello` function defined in [PrimitiveFunctions.java](#) by pointing your browser to <http://localhost:8080/hello>

Invoking the other functions in the quickstart requires an HTTP POST. To execute the `greet` function defined in [GreetingFunction.java](#) invoke this curl script.

```
curl "http://localhost:8080/greet" \  
-X POST \  
-H "Content-Type: application/json" \  
-d '{"name":"Bill"}'
```

Primitive types can also be passed as input using the standard JSON mapping for them. To execute the `toLowerCase` function defined in [PrimitiveFunctions.java](#) invoke this curl script:

```
curl "http://localhost:8080/toLowerCase" \  
-X POST \  
-H "Content-Type: application/json" \  
-d '"HELLO WORLD"'
```

To execute the `double` function defined in [PrimitiveFunctions.java](#) invoke this curl script:

```
curl "http://localhost:8080/double" \  
-X POST \  
-H "Content-Type: application/json" \  
-d '2'
```

Maven Dependencies

To write Funqy HTTP functions, simply include the `quarkus-funqy-http` dependency into your Quarkus `pom.xml` file:

```
<dependency>  
  <groupId>io.quarkus</groupId>  
  <artifactId>quarkus-funqy-http</artifactId>  
</dependency>
```