

# Microservice Architecture and Programming

Practical 2

Harsh Chaplot - 17BIT026

---

## Hello World Main

Let's print "Hello, World!" using the main function in Ballerina. A public function named main is considered as an entry point to a Ballerina program.

**Code:**

```
import ballerina/io;

public function main() {

    io:println("Hello, World!");

}
```

**Output:** Hello, World!

## Variables

Ballerina provides you with two ways to declare variables. You can either use the var keyword followed by the variable name and an initializer expression or use a type name followed by the variable name in which you can optionally specify the initializer expression.

**Code:**

```
import ballerina/io;

int total = 99;
```

---

```
const string OK = "ok";

public const int COUNT = 1;

final int status = 1;

public function main() {

    io.println(total);

    io.println(COUNT);

    boolean available = false;

    io.println(available);

}
```

**Output: 99**

1

false

---

# Functions

Functions operate in the same way as in any other language. It is a mechanism to create a reusable unit of functionality within a program.

## Code:

```
import ballerina/io;

function printValue(string value) {
    io:println(value);
}

function add(int a, int b) returns int {
    return a + b;
}

function printAndReturnValue(string s) returns string {
    string t = "Hello ".concat(s);
    io:println(t);
    return t;
}

public function main() {

    printValue("This is a sample text");

    int result = add(5, 6);

    io:println(result);

    _ = printAndReturnValue("World");
}
```

**Output:** This is a sample text

11

Hello World

---

# Hello World Service

In Ballerina, services represent collections of network-accessible entry points. Resources represent one such entry point. How a resource is exposed over a network protocol depends on the listener to which a service is attached.

## Code:

```
import ballerina/http;

service hello on new http:Listener(9090) {

    resource function sayHello(http:Caller caller,
        http:Request req) returns error? {

        check caller->respond("Hello, World!");
    }
}
```

**Output:** Hello, World!

# RESTful API

## Code:

```
import ballerina/http;
import ballerina/log;

listener http:Listener httpListener = new(9090);

map<json> ordersMap = {};

@http:ServiceConfig { basePath: "/ordermgt" }
service orderMgt on httpListener {

    @http:ResourceConfig {
        methods: ["GET"],
        path: "/order/{orderId}"
    }
}
```

```

    }

    resource function findOrder(http:Caller caller, http:Request req,
string orderId) {
        json? payload = ordersMap[orderId];
        http:Response response = new;
        if (payload == null) {
            payload = "Order : " + orderId + " cannot be found.";
        }

        response.setJsonPayload(<@untainted> payload);

        var result = caller->respond(response);
        if (result is error) {
            log:printError("Error sending response", err = result);
        }
    }

    @http:ResourceConfig {
        methods: ["POST"],
        path: "/order"
    }

    resource function addOrder(http:Caller caller, http:Request req) {
        http:Response response = new;
        var orderReq = req.getJsonPayload();
        if (orderReq is json) {
            string orderId = orderReq.Order.ID.toString();
            ordersMap[orderId] = <@untainted> orderReq;

            json payload = { status: "Order Created.", orderId: orderId };
            response.setJsonPayload(<@untainted> payload);

            response.statusCode = 201;

            response.setHeader("Location",
                "http://localhost:9090/ordermgt/order/" + orderId);
        }
    }

```

```

        var result = caller->respond(response);
        if (result is error) {
            log:printError("Error sending response", err = result);
        }
    } else {
        response.statusCode = 400;
        response.setPayload("Invalid payload received");
        var result = caller->respond(response);
        if (result is error) {
            log:printError("Error sending response", err = result);
        }
    }
}

@http:ResourceConfig {
    methods: ["PUT"],
    path: "/order/{orderId}"
}

resource function updateOrder(http:Caller caller, http:Request req,
string orderId) {
    var updatedOrder = req.getJsonPayload();
    http:Response response = new;
    if (updatedOrder is json) {
        json existingOrder = ordersMap[orderId];

        if (existingOrder != null) {

            ordersMap[orderId] = <@untainted> updatedOrder;
        } else {
            existingOrder = "Order : " + orderId + " cannot be
found.";
        }
        response.setJsonPayload(<@untainted> existingOrder);
        var result = caller->respond(response);
        if (result is error) {
            log:printError("Error sending response", err = result);

```

---

```

    }
  } else {
    response.statusCode = 400;
    response.setPayload("Invalid payload received");
    var result = caller->respond(response);
    if (result is error) {
      log:printError("Error sending response", err = result);
    }
  }
}

@http:ResourceConfig {
  methods: ["DELETE"],
  path: "/order/{orderId}"
}

resource function cancelOrder(http:Caller caller, http:Request req,
string orderId) {
  http:Response response = new;
  _ = ordersMap.remove(orderId);

  json payload = "Order : " + orderId + " removed.";
  response.setJsonPayload(<@untainted> payload);

  var result = caller->respond(response);
  if (result is error) {
    log:printError("Error sending response", err = result);
  }
}
}

```

**Output:**

Activities Postman Wed 11:54 AM

Postman

File Edit View Help

New Import Runner My Workspace Invite

Filter

History Collections APIs

+ New Collection Trash

Smart hisab  
19 requests

POST localhost:9090/ordermgt/order GET localhost:9090/ordermgt/order...

No Environment

Untitled Request

POST localhost:9090/ordermgt/order

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "Order": {
3     "ID": "xyz",
4     "Name": "harsh",
5     "Description": "harsh"
6   }
7 }
```

Body Cookies Headers (6) Test Results

Status: 201 Created Time: 860 ms Size: 277 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "Order Created.",
3   "orderId": "xyz"
4 }
```

Find and Replace Console

Bootcamp Build Browse

Activities Postman Wed 11:54 AM

Postman

File Edit View Help

New Import Runner My Workspace Invite

Filter

History Collections APIs

+ New Collection Trash

Smart hisab  
19 requests

POST localhost:9090/ordermgt/order GET localhost:9090/ordermgt/order...

No Environment

Untitled Request

GET localhost:9090/ordermgt/order/xyz

Send Save

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 17 ms Size: 229 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "Order": {
3     "ID": "xyz",
4     "Name": "harsh",
5     "Description": "harsh"
6   }
7 }
```

Find and Replace Console

Bootcamp Build Browse