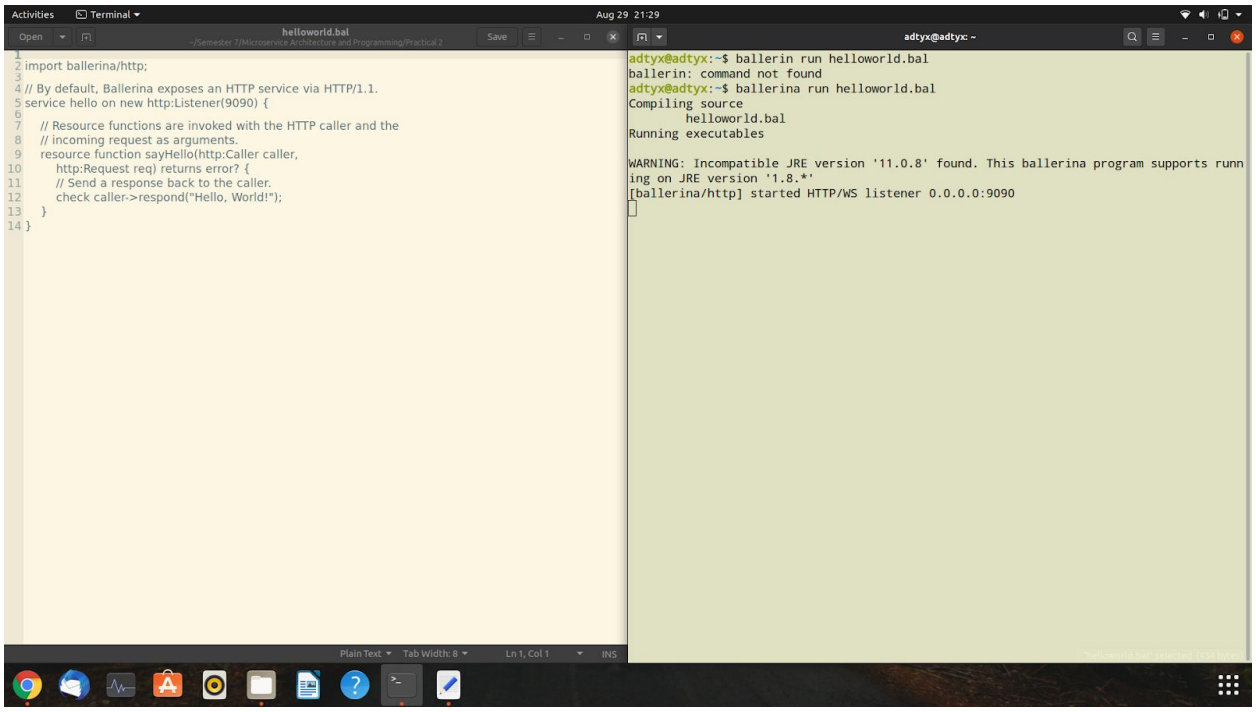
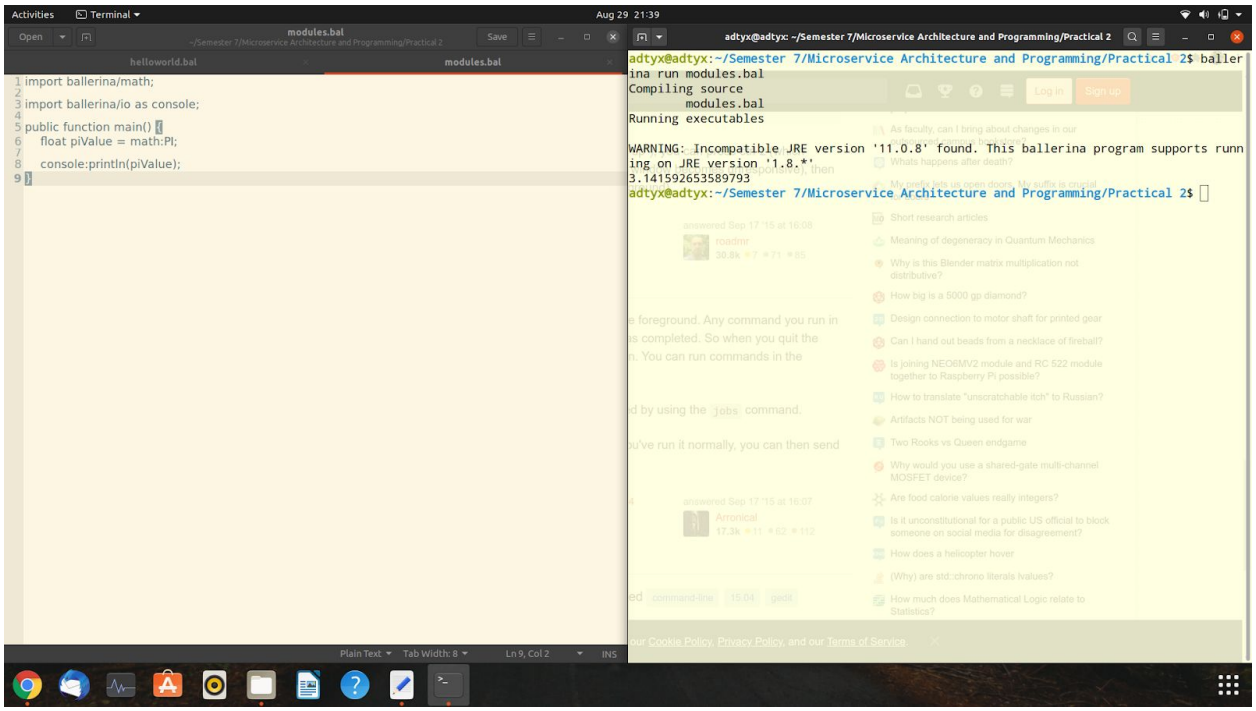
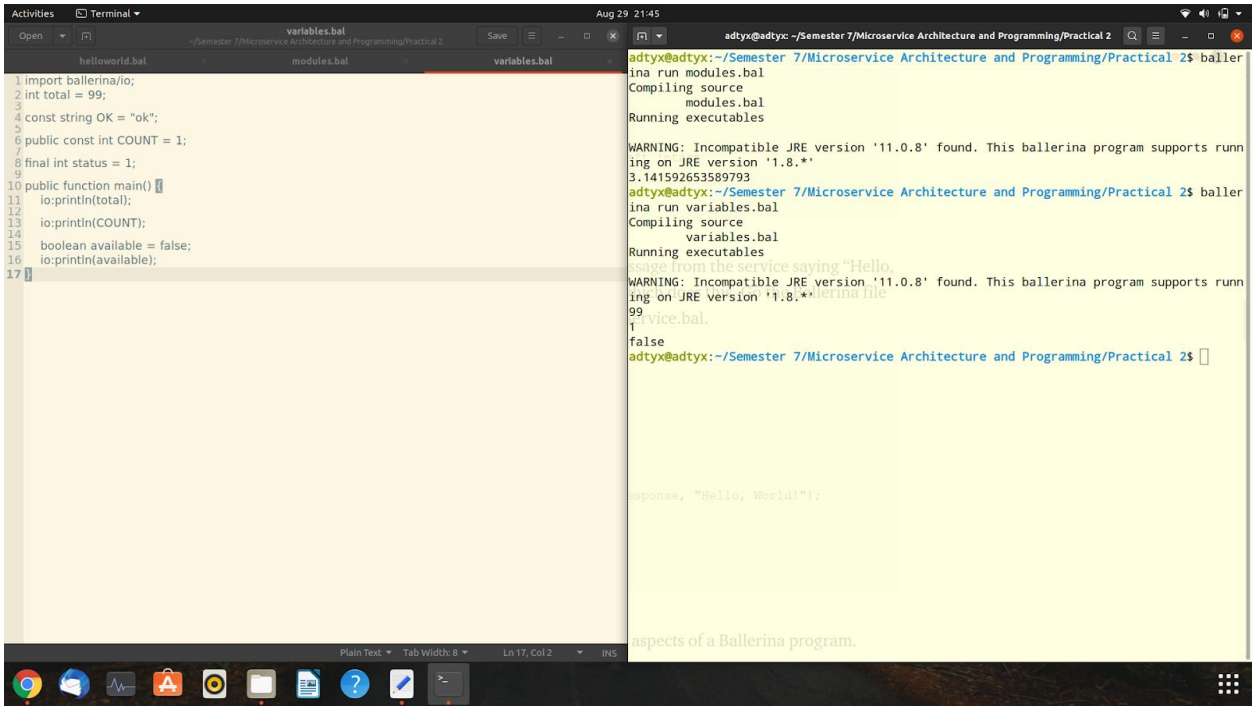
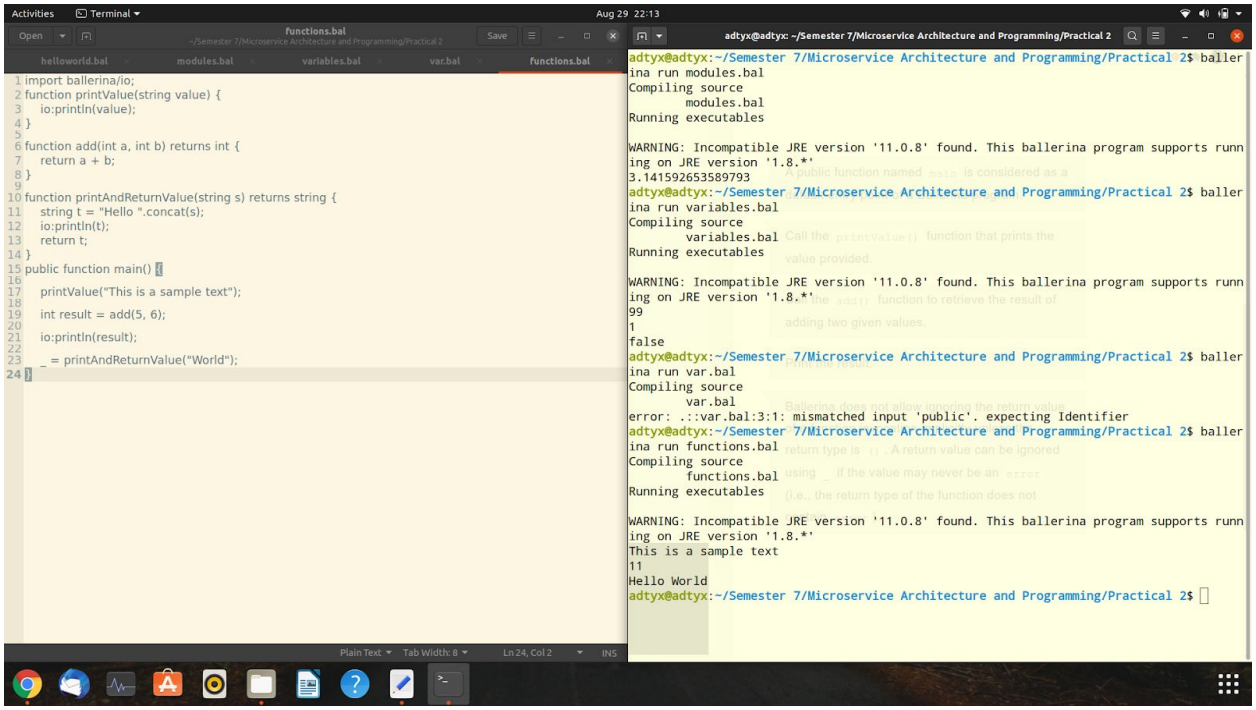
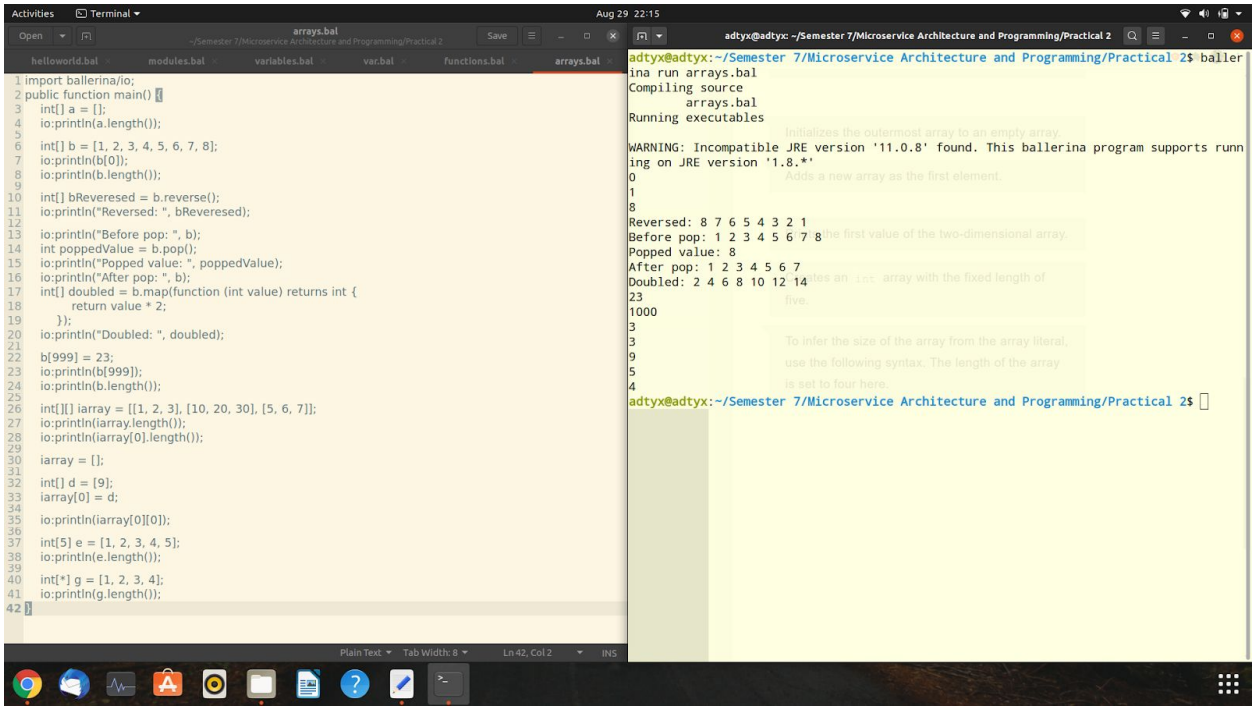
Code/screens showing basic concepts of Ballerina Language



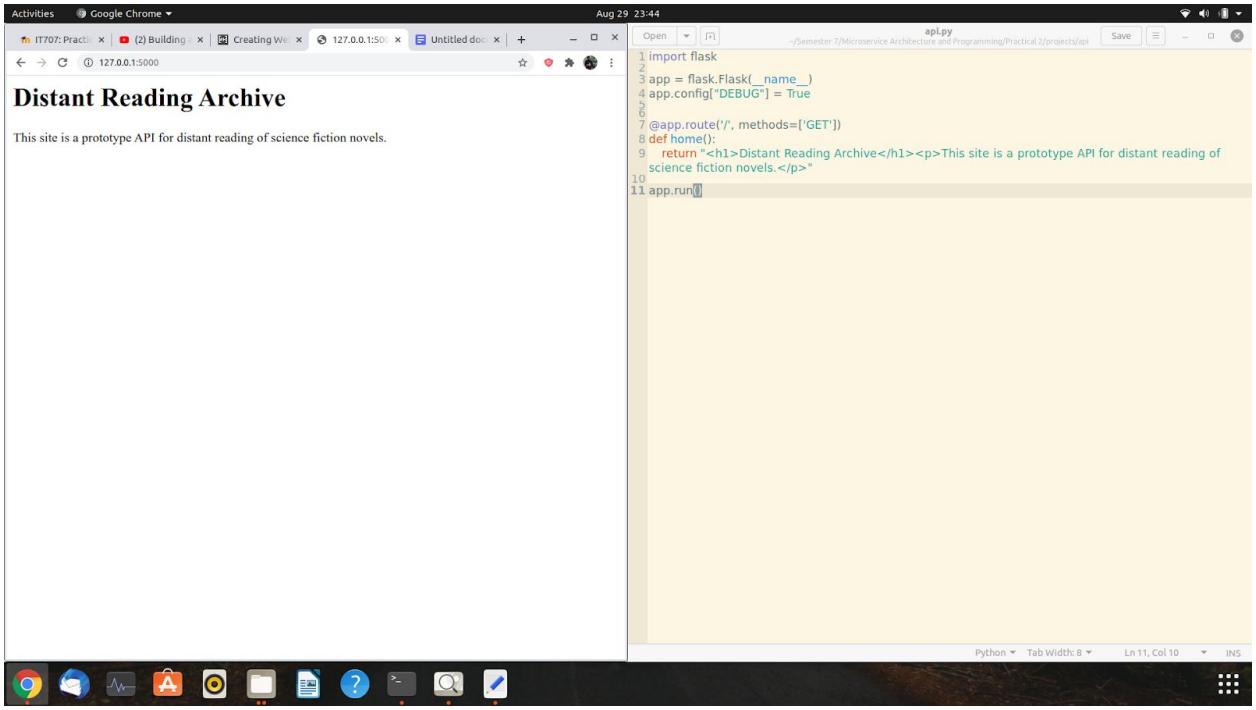


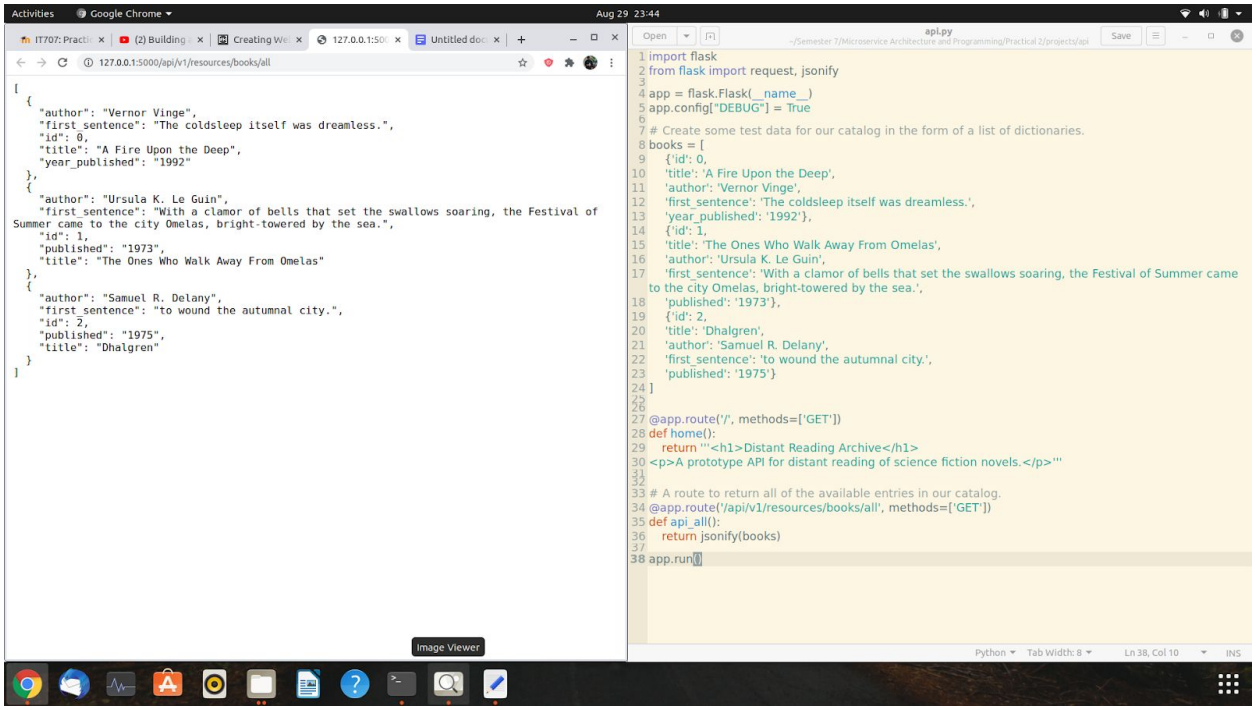

Activities
Terminal
Aug 29 21:45
variables.bal
~/Semester 7/Microservice Architecture and Programming/Practical 2
Open
Save
helloworld.bal
modules.bal
variables.bal
1 import ballerina/io;
2 int total = 99;
3
4 const string OK = "ok";
5
6 public const int COUNT = 1;
7
8 final int status = 1;
9
10 public function main() {
11 io:println(total);
12
13 io:println(COUNT);
14
15 boolean available = false;
16 io:println(available);
17 }
Plain Text
Tab Width: 8
Ln 17, Col 2
INS
adtyx@adtyx: ~/Semester 7/Microservice Architecture and Programming/Practical 2
adtyx@adtyx:~/Semester 7/Microservice Architecture and Programming/Practical 2$ baller
ina run modules.bal
Compiling source
modules.bal
Running executables
WARNING: Incompatible JRE version '11.0.8' found. This ballerina program supports runn
ing on JRE version '1.8.*'
3.141592653589793
adtyx@adtyx:~/Semester 7/Microservice Architecture and Programming/Practical 2$ baller
ina run variables.bal
Compiling source
variables.bal
Running executables
WARNING: Incompatible JRE version '11.0.8' found. This ballerina program supports runn
ing on JRE version '1.8.*'
99
1
false
adtyx@adtyx:~/Semester 7/Microservice Architecture and Programming/Practical 2$

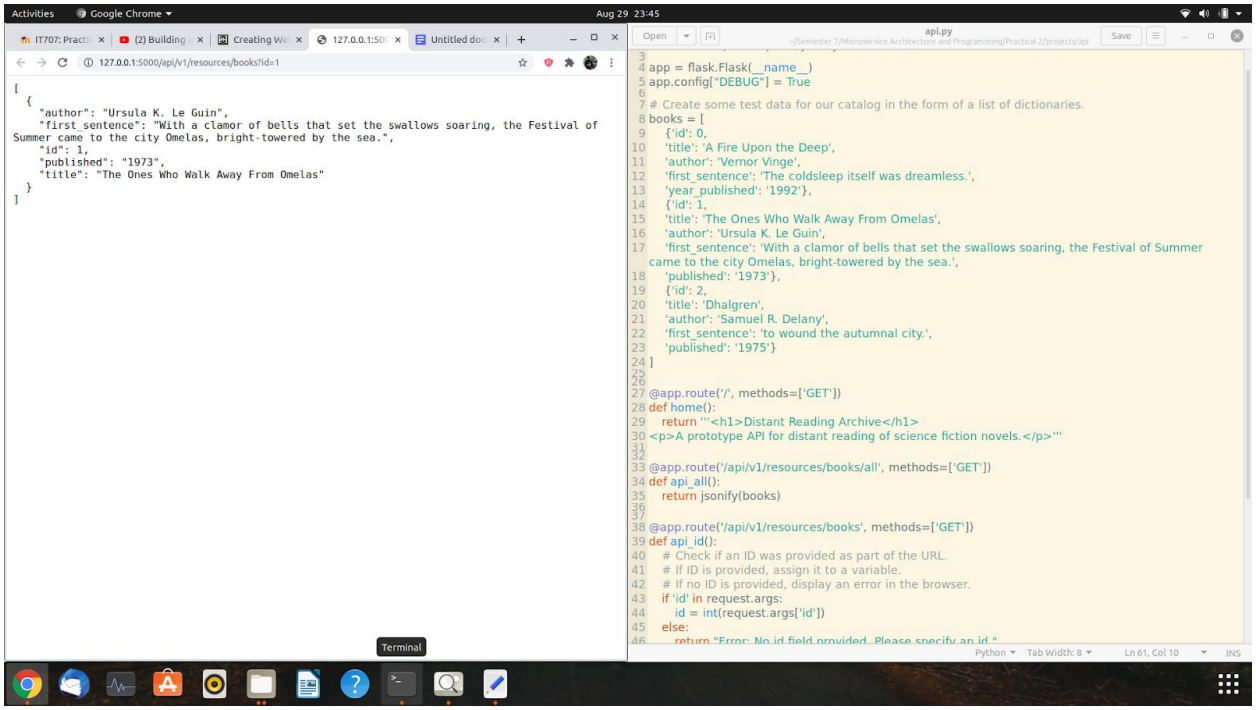

Activities
Terminal
Aug 29 22:13
functions.bal
~/Semester 7/Microservice Architecture and Programming/Practical 2
Open
Save
helloworld.bal
modules.bal
variables.bal
var.bal
functions.bal
1 import ballerina/io;
2 function printValue(string value) {
3 io:println(value);
4 }
5
6 function add(int a, int b) returns int {
7 return a + b;
8 }
9
10 function printAndReturnValue(string s) returns string {
11 string t = "Hello ".concat(s);
12 io:println(t);
13 return t;
14 }
15 public function main() {
16
17 printValue("This is a sample text");
18
19 int result = add(5, 6);
20
21 io:println(result);
22
23 _ = printAndReturnValue("World");
24 }
Plain Text
Tab Width: 8
Ln 24, Col 2
INS
adtyx@adtyx: ~/Semester 7/Microservice Architecture and Programming/Practical 2
adtyx@adtyx:~/Semester 7/Microservice Architecture and Programming/Practical 2$ baller
ina run modules.bal
Compiling source
modules.bal
Running executables
WARNING: Incompatible JRE version '11.0.8' found. This ballerina program supports runn
ing on JRE version '1.8.*'
3.141592653589793
adtyx@adtyx:~/Semester 7/Microservice Architecture and Programming/Practical 2$ baller
ina run variables.bal
Compiling source
variables.bal
Running executables
WARNING: Incompatible JRE version '11.0.8' found. This ballerina program supports runn
ing on JRE version '1.8.*'
99
1
false
adtyx@adtyx:~/Semester 7/Microservice Architecture and Programming/Practical 2$ baller
ina run var.bal
Compiling source
var.bal
error: .::var.bal:3:1: mismatched input 'public'. expecting Identifier
adtyx@adtyx:~/Semester 7/Microservice Architecture and Programming/Practical 2$ baller
ina run functions.bal
Compiling source
functions.bal
Running executables
WARNING: Incompatible JRE version '11.0.8' found. This ballerina program supports runn
ing on JRE version '1.8.*'
This is a sample text
11
Hello World
adtyx@adtyx:~/Semester 7/Microservice Architecture and Programming/Practical 2$



RESTful API implementation code/demo screens




Activities
Google Chrome
Aug 29 23:44
127.0.0.1:5000/api/v1/resources/books/all
[
{
"author": "Vernor Vinge",
"first_sentence": "The coldsleep itself was dreamless.",
"id": 0,
"title": "A Fire Upon the Deep",
"year_published": "1992"
},
{
"author": "Ursula K. Le Guin",
"first_sentence": "With a clamor of bells that set the swallows soaring, the Festival of Summer came to the city Omelas, bright-towered by the sea.",
"id": 1,
"published": "1973",
"title": "The Ones Who Walk Away From Omelas"
},
{
"author": "Samuel R. Delany",
"first_sentence": "to wound the autumnal city.",
"id": 2,
"published": "1975",
"title": "Dhalgren"
}
]
Image Viewer
api.py
~/Semester 7/Microservice Architecture and Programming/Practical 2/projects/api
Open
Save
import flask
from flask import request, jsonify
app = flask.Flask(__name__)
app.config["DEBUG"] = True
# Create some test data for our catalog in the form of a list of dictionaries.
books = [
{'id': 0,
'title': 'A Fire Upon the Deep',
'author': 'Vernor Vinge',
'first_sentence': 'The coldsleep itself was dreamless.',
'year_published': '1992'},
{'id': 1,
'title': 'The Ones Who Walk Away From Omelas',
'author': 'Ursula K. Le Guin',
'first_sentence': 'With a clamor of bells that set the swallows soaring, the Festival of Summer came to the city Omelas, bright-towered by the sea.',
'published': '1973'},
{'id': 2,
'title': 'Dhalgren',
'author': 'Samuel R. Delany',
'first_sentence': 'to wound the autumnal city.',
'published': '1975'}
]
@app.route('/', methods=['GET'])
def home():
return '''<h1>Distant Reading Archive</h1>
A prototype API for distant reading of science fiction novels. '''
A route to return all of the available entries in our catalog.
@app.route('/api/v1/resources/books/all', methods=['GET'])
def api_all():
return jsonify(books)
app.run()
Python
Tab Width: 8
Ln 38, Col 10
INS


Activities
Google Chrome
Aug 29 23:45
127.0.0.1:5000/api/v1/resources/books?id=1
[
{
"author": "Ursula K. Le Guin",
"first_sentence": "With a clamor of bells that set the swallows soaring, the Festival of Summer came to the city Omelas, bright-towered by the sea.",
"id": 1,
"published": "1973",
"title": "The Ones Who Walk Away From Omelas"
}
]
Terminal
api.py
~/Semester 7/Microservice Architecture and Programming/Practical 2/projects/api
Open
Save
app = flask.Flask(__name__)
app.config["DEBUG"] = True
# Create some test data for our catalog in the form of a list of dictionaries.
books = [
{'id': 0,
'title': 'A Fire Upon the Deep',
'author': 'Vernor Vinge',
'first_sentence': 'The coldsleep itself was dreamless.',
'year_published': '1992'},
{'id': 1,
'title': 'The Ones Who Walk Away From Omelas',
'author': 'Ursula K. Le Guin',
'first_sentence': 'With a clamor of bells that set the swallows soaring, the Festival of Summer came to the city Omelas, bright-towered by the sea.',
'published': '1973'},
{'id': 2,
'title': 'Dhalgren',
'author': 'Samuel R. Delany',
'first_sentence': 'to wound the autumnal city.',
'published': '1975'}
]
@app.route('/', methods=['GET'])
def home():
return '''<h1>Distant Reading Archive</h1>
A prototype API for distant reading of science fiction novels. '''
@app.route('/api/v1/resources/books/all', methods=['GET'])
def api_all():
return jsonify(books)
@app.route('/api/v1/resources/books', methods=['GET'])
def api_id():
# Check if an ID was provided as part of the URL.
# If ID is provided, assign it to a variable.
# If no ID is provided, display an error in the browser.
if 'id' in request.args:
id = int(request.args['id'])
else:
Python
Tab Width: 8
Ln 61, Col 10
INS