

Факультет программной̆ инженерии и компьютерной̆ техники

Сервис-ориентированная архитектура

Лабораторная работа №3

Вариант 1012.90.9

Преподаватель: Усков Иван Владимирович

Выполнил: Кульбако Артемий Юрьевич Р34112

# Задание

Переработать веб-сервисы из лабораторной работы #2 таким образом, чтобы они реализовывали основные концепции микросервисной архитектуры. Для этого внести в оба сервиса -- "вызываемый" (из лабораторной работы #1) и "вызывающий" (добавленный в лабораторной работе #2) перечисленные ниже изменения.

**Изменения в "вызываемом" сервисе:**

* Сконфигурировать окружение для работы сервиса на платформе Spring Boot.
* Запустить второй экземпляр сервиса на другом порту. Реализовать балансировку нагрузки между экземплярами с помощью Haproxy.
* Реализовать механизм Service Discovery. Для этого установить Consul и интегрировать свой сервис с ним, автоматически регистрируя в момент запуска.

**Изменения в "вызывающем" сервисе:**

* Разделить приложение на два модуля -- веб-приложение с веб-сервисом и EJB-jar с бизнес-компонентами.
* Переместить всю логику из класса сервиса в Stateless EJB. В классе сервиса оставить только обращение к методам бизнес-интерфейса. EJB-компонент должен быть доступен удалённо (иметь Remote-интерфейс).
* Сформировать на уровне сервера приложений пул компонентов EJB настраиваемой мощности, динамически расширяемый при увеличении нагрузки.
* Настроить второй экземпляр сервера приложений на другом порту, "поднять" на нём вторую копию веб-сервиса и пула EJB.
* Настроить балансировку нагрузки на оба запущенных узла через Haproxy.

**Оба веб-сервиса и клиентское приложение должны сохранить полную совместимость с API, реализованными в рамках предыдущих лабораторных работ.**

# API

## service1

{  
 "openapi": "3.0.1",  
 "info": {  
 "title": "service1\_api",  
 "version": "1"  
 },  
 "servers": [  
 {  
 "url": "https://localhost:26557/",  
 "description": "localhost and helios with port forwarding"  
 }  
 ],  
 "paths": {  
 "/api/persons/{id}": {  
 "get": {  
 "tags": [  
 "persons-controller"  
 ],  
 "operationId": "getById",  
 "parameters": [  
 {  
 "name": "id",  
 "in": "path",  
 "required": *true*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "$ref": "#/components/schemas/Person"  
 }  
 }  
 }  
 }  
 }  
 },  
 "put": {  
 "tags": [  
 "persons-controller"  
 ],  
 "operationId": "modify",  
 "parameters": [  
 {  
 "name": "id",  
 "in": "path",  
 "required": *true*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 ],  
 "requestBody": {  
 "content": {  
 "application/json": {  
 "schema": {  
 "$ref": "#/components/schemas/Person"  
 }  
 }  
 },  
 "required": *true* },  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "object",  
 "additionalProperties": {  
 "type": "string"  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "delete": {  
 "tags": [  
 "persons-controller"  
 ],  
 "operationId": "delete",  
 "parameters": [  
 {  
 "name": "id",  
 "in": "path",  
 "required": *true*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "object",  
 "additionalProperties": {  
 "type": "string"  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "/api/persons/": {  
 "get": {  
 "tags": [  
 "persons-controller"  
 ],  
 "operationId": "get",  
 "parameters": [  
 {  
 "name": "limit",  
 "in": "query",  
 "required": *false*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 },  
 {  
 "name": "offset",  
 "in": "query",  
 "required": *false*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 },  
 {  
 "name": "sorters",  
 "in": "query",  
 "required": *false*,  
 "schema": {  
 "type": "string",  
 "default": ""  
 }  
 },  
 {  
 "name": "filters",  
 "in": "query",  
 "required": *false*,  
 "schema": {  
 "type": "string",  
 "default": ""  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "array",  
 "items": {  
 "$ref": "#/components/schemas/Person"  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "post": {  
 "tags": [  
 "persons-controller"  
 ],  
 "operationId": "add",  
 "requestBody": {  
 "content": {  
 "application/json": {  
 "schema": {  
 "$ref": "#/components/schemas/Person"  
 }  
 }  
 },  
 "required": *true* },  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "/api/dragons/": {  
 "get": {  
 "tags": [  
 "dragons-controller"  
 ],  
 "operationId": "get\_1",  
 "parameters": [  
 {  
 "name": "limit",  
 "in": "query",  
 "required": *false*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 },  
 {  
 "name": "offset",  
 "in": "query",  
 "required": *false*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 },  
 {  
 "name": "sorters",  
 "in": "query",  
 "required": *false*,  
 "schema": {  
 "type": "string",  
 "default": ""  
 }  
 },  
 {  
 "name": "filters",  
 "in": "query",  
 "required": *false*,  
 "schema": {  
 "type": "string",  
 "default": ""  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "array",  
 "items": {  
 "$ref": "#/components/schemas/Dragon"  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "post": {  
 "tags": [  
 "dragons-controller"  
 ],  
 "operationId": "add\_1",  
 "requestBody": {  
 "content": {  
 "application/json": {  
 "schema": {  
 "$ref": "#/components/schemas/Dragon"  
 }  
 }  
 },  
 "required": *true* },  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "/api/dragons/{id}": {  
 "get": {  
 "tags": [  
 "dragons-controller"  
 ],  
 "operationId": "getById\_1",  
 "parameters": [  
 {  
 "name": "id",  
 "in": "path",  
 "required": *true*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "$ref": "#/components/schemas/Dragon"  
 }  
 }  
 }  
 }  
 }  
 },  
 "delete": {  
 "tags": [  
 "dragons-controller"  
 ],  
 "operationId": "delete\_1",  
 "parameters": [  
 {  
 "name": "id",  
 "in": "path",  
 "required": *true*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "object",  
 "additionalProperties": {  
 "type": "string"  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "patch": {  
 "tags": [  
 "dragons-controller"  
 ],  
 "operationId": "modify\_1",  
 "parameters": [  
 {  
 "name": "id",  
 "in": "path",  
 "required": *true*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 ],  
 "requestBody": {  
 "content": {  
 "application/json": {  
 "schema": {  
 "$ref": "#/components/schemas/Dragon"  
 }  
 }  
 },  
 "required": *true* },  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "object",  
 "additionalProperties": {  
 "type": "string"  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "/api/persons/find\_person\_included\_in\_name": {  
 "get": {  
 "tags": [  
 "persons-controller"  
 ],  
 "operationId": "getIncludedInName",  
 "parameters": [  
 {  
 "name": "name",  
 "in": "query",  
 "required": *true*,  
 "schema": {  
 "type": "string"  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "array",  
 "items": {  
 "$ref": "#/components/schemas/Person"  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "/api/dragons/grouped\_by\_type": {  
 "get": {  
 "tags": [  
 "dragons-controller"  
 ],  
 "operationId": "getGroupedByType",  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "object",  
 "additionalProperties": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "/api/dragons/find\_with\_killer\_weaker\_then": {  
 "get": {  
 "tags": [  
 "dragons-controller"  
 ],  
 "operationId": "getWithKillerWeakerThen",  
 "parameters": [  
 {  
 "name": "killer\_id",  
 "in": "query",  
 "required": *true*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "array",  
 "items": {  
 "$ref": "#/components/schemas/Dragon"  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "components": {  
 "schemas": {  
 "Person": {  
 "type": "object",  
 "properties": {  
 "id": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "name": {  
 "type": "string"  
 },  
 "height": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "weight": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "passportId": {  
 "type": "string"  
 },  
 "hairColor": {  
 "type": "string",  
 "enum": [  
 "GREEN",  
 "YELLOW",  
 "BROWN"  
 ]  
 },  
 "team": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 },  
 "Coordinates": {  
 "type": "object",  
 "properties": {  
 "id": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "x": {  
 "type": "number",  
 "format": "double"  
 },  
 "y": {  
 "type": "number",  
 "format": "double"  
 },  
 "z": {  
 "type": "number",  
 "format": "double"  
 }  
 }  
 },  
 "Dragon": {  
 "type": "object",  
 "properties": {  
 "id": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "name": {  
 "type": "string"  
 },  
 "coordinates": {  
 "$ref": "#/components/schemas/Coordinates"  
 },  
 "creationDate": {  
 "type": "string",  
 "format": "date-time"  
 },  
 "age": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "wingspan": {  
 "type": "number",  
 "format": "double"  
 },  
 "color": {  
 "type": "string",  
 "enum": [  
 "GREEN",  
 "YELLOW",  
 "BROWN"  
 ]  
 },  
 "type": {  
 "type": "string",  
 "enum": [  
 "WATER",  
 "AIR",  
 "FIRE",  
 "UNKNOWN"  
 ]  
 },  
 "killer": {  
 "$ref": "#/components/schemas/Person"  
 }  
 }  
 }  
 }  
 }  
}

## service2

{  
 "openapi": "3.0.3",  
 "info": {  
 "title": "service2\_api",  
 "version": "1"  
 },  
 "servers": [  
 {  
 "url": "https://localhost:26558/killer",  
 "description": "localhost and helios with port forwarding"  
 }  
 ],  
 "paths": {  
 "/api/dragons/find\_by\_cave\_depth": {  
 "get": {  
 "parameters": [  
 {  
 "name": "type",  
 "in": "path",  
 "required": *true*,  
 "schema": {  
 "type": "string",  
 "enum": [  
 "MIN", "MAX"  
 ]  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "array",  
 "items": {  
 "$ref": "#/components/schemas/Dragon"  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "/api/teams": {  
 "post": {  
 "requestBody": {  
 "content": {  
 "application/json": {  
 "schema": {  
 "$ref": "#/components/schemas/Team"  
 }  
 }  
 },  
 "required": *true* },  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 }  
 }  
 }  
 },  
 "get": {  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "type": "array",  
 "items": {  
 "$ref": "#/components/schemas/Team"  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "/api/teams/{id}": {  
 "get": {  
 "parameters": [  
 {  
 "name": "id",  
 "in": "path",  
 "required": *true*,  
 "schema": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 ],  
 "responses": {  
 "200": {  
 "description": "OK",  
 "content": {  
 "\*/\*": {  
 "schema": {  
 "$ref": "#/components/schemas/Team"  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 },  
 "components": {  
 "schemas": {  
 "Coordinates": {  
 "type": "object",  
 "properties": {  
 "id": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "x": {  
 "type": "number",  
 "format": "double"  
 },  
 "y": {  
 "type": "number",  
 "format": "double"  
 },  
 "z": {  
 "type": "number",  
 "format": "double"  
 }  
 }  
 },  
 "Dragon": {  
 "type": "object",  
 "properties": {  
 "id": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "name": {  
 "type": "string"  
 },  
 "coordinates": {  
 "$ref": "#/components/schemas/Coordinates"  
 },  
 "creationDate": {  
 "type": "string",  
 "format": "date-time"  
 },  
 "age": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "wingspan": {  
 "type": "number",  
 "format": "double"  
 },  
 "color": {  
 "type": "string",  
 "enum": [  
 "GREEN",  
 "YELLOW",  
 "BROWN"  
 ]  
 },  
 "type": {  
 "type": "string",  
 "enum": [  
 "WATER",  
 "AIR",  
 "FIRE",  
 "UNKNOWN"  
 ]  
 },  
 "killer": {  
 "$ref": "#/components/schemas/Person"  
 }  
 }  
 },  
 "Team": {  
 "type": "object",  
 "properties": {  
 "id": {  
 "type": "integer",  
 "format": "int32"  
 },  
 "name": {  
 "type": "string"  
 },  
 "membersIds": {  
 "type": "array",  
 "items": {  
 "type": "integer",  
 "format": "int32"  
 }  
 }  
 }  
 }  
 }  
 }  
}

# Haproxy

global

defaults

log global

mode http

option dontlognull

timeout connect 5000

timeout client 50000

timeout server 50000

frontend service1-f

bind 0.0.0.0:26554

mode http

default\_backend service1

frontend service2-f

bind 0.0.0.0:26555

mode http

default\_backend service2

backend service1

balance roundrobin

server server-1 127.0.0.1:26556

server server-2 127.0.0.1:26557

http-request set-path %[path]

backend service2

balance roundrobin

server server-3 127.0.0.1:26558

server server-4 127.0.0.1:26559

http-request set-path %[path]

# Код

<https://github.com/testpassword/Service-oriented-architecture/tree/master/lab3-03.01.22>

# Выводы

Данная лабораторная даёт навыки работы с инфраструктурным ПО для управления сервисами – service discovery и load balancer – крайне полезными технологиями для администраторов и DevOps-инженеров. С обоими работать понравилось, сложностей не было: для простых целей порог вхождения минимальный.