

Факультет программной̆ инженерии и компьютерной̆ техники

Сервис-ориентированная архитектура

Лабораторная работа №4

Вариант 1012

Преподаватель: Усков Иван Владимирович

Выполнил: Кульбако Артемий Юрьевич Р34112

# Задание

**Переработать сервисы из лабораторной работы #3 следующим образом:**

* Первый ("вызываемый") сервис переписать в соответствии с требованиями протокола SOAP.
* Развернуть переработанный сервис на сервере приложений по собственному выбору.
* Оставшийся сервис не модифицировать, не менять его API, протокол и используемый сервер приложений.
* Установить и сконфигурировать на сервере Helios программное обеспечение Mule ESB.
* Настроить интеграцию двух сервисов с использованием установленного программного обеспечения.
* Реализовать дополнительную REST-"прослойку", обеспечивающую возможность доступа к переработанному сервису клиентского приложения без необходимости его модификации. Никакой дополнительной логики, помимо вызовов SOAP-сервиса, разработанная REST-прослойка содержать не должна.

# API

## service1

<xs:schema *xmlns:xs*="http://www.w3.org/2001/XMLSchema"  
 *xmlns:tns*="http://testpassword/dtos" *xmlns:xsd*="http://www.w3.org/2001/XMLSchema"  
 *targetNamespace*="http://testpassword/dtos"  
 *elementFormDefault*="qualified">  
  
 <xs:element *name*="getPersonByIdRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="id" *type*="xs:int"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="getPersonByIdResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="person" *type*="tns:PersonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="deletePersonByIdRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="id" *type*="xs:int"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="deletePersonByIdResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="msg" *type*="xs:string"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
   
 <xs:element *name*="addPersonRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="person" *type*="tns:PersonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="addPersonResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="id" *type*="xs:int"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="modifyPersonByIdRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="person" *type*="tns:PersonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="modifyPersonByIdResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="msg" *type*="xs:string"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="getPersonsIncludedInNameRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="name" *type*="xs:string"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="getPersonsIncludedInNameResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *minOccurs*="0" *maxOccurs*="unbounded" *name*="dtos" *nillable*="true" *type*="tns:PersonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="getPersonsRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="limit" *type*="xs:int" *minOccurs*="0"/>  
 <xs:element *name*="offset" *type*="xs:int" *minOccurs*="0"/>  
 <xs:element *name*="sorters" *type*="xs:string" *minOccurs*="0"/>  
 <xs:element *name*="filters" *type*="xs:string" *minOccurs*="0"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="getPersonsResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *minOccurs*="0" *maxOccurs*="unbounded" *name*="dtos" *nillable*="true" *type*="tns:PersonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:complexType *name*="PersonDto">  
 <xs:sequence>  
 <xs:element *name*="id" *type*="xs:int" *minOccurs*="0"/>  
 <xs:element *name*="name" *type*="xs:string"/>  
 <xs:element *name*="height" *type*="xs:int"/>  
 <xs:element *name*="weight" *type*="xs:int"/>  
 <xs:element *name*="passportId" *type*="xs:string"/>  
 <xs:element *name*="hairColor" *type*="xs:string"/>  
 <xs:element *name*="teamId" *type*="xs:int" *minOccurs*="0"/>  
 </xs:sequence>  
 </xs:complexType>  
  
 <xs:element *name*="getDragonByIdRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="id" *type*="xs:int"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="getDragonByIdResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="dragon" *type*="tns:DragonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="deleteDragonByIdRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="id" *type*="xs:int"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="deleteDragonByIdResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="msg" *type*="xs:string"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="addDragonRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="dragon" *type*="tns:DragonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="addDragonResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="id" *type*="xs:int"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="modifyDragonByIdRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="dragon" *type*="tns:DragonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="modifyDragonByIdResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="msg" *type*="xs:string"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="getDragonsWithKillerWeakerThenRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="killerId" *type*="xs:int"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="getDragonsWithKillerWeakerThenResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *minOccurs*="0" *maxOccurs*="unbounded" *name*="dtos" *nillable*="true" *type*="tns:DragonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="getDragonsGroupedByTypeRequest">  
 <xs:complexType>  
 <xs:sequence>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:complexType *name*="DragonStat">  
 <xs:sequence>  
 <xs:element *name*="type" *type*="xs:string"/>  
 <xs:element *name*="count" *type*="xs:int"/>  
 </xs:sequence>  
 </xs:complexType>  
 <xs:element *name*="getDragonsGroupedByTypeResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *minOccurs*="0" *maxOccurs*="unbounded" *name*="stat" *type*="tns:DragonStat"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element *name*="getDragonsRequest">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *name*="limit" *type*="xs:int" *minOccurs*="0"/>  
 <xs:element *name*="offset" *type*="xs:int" *minOccurs*="0"/>  
 <xs:element *name*="sorters" *type*="xs:string" *minOccurs*="0"/>  
 <xs:element *name*="filters" *type*="xs:string" *minOccurs*="0"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element *name*="getDragonsResponse">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element *minOccurs*="0" *maxOccurs*="unbounded" *name*="dtos" *nillable*="true" *type*="tns:DragonDto"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:complexType *name*="CoordinateDto">  
 <xs:sequence>  
 <xs:element *name*="id" *type*="xs:int" *minOccurs*="0"/>  
 <xs:element *name*="x" *type*="xs:double"/>  
 <xs:element *name*="y" *type*="xs:double"/>  
 <xs:element *name*="z" *type*="xs:double"/>  
 </xs:sequence>  
 </xs:complexType>  
  
 <xs:complexType *name*="DragonDto">  
 <xs:sequence>  
 <xs:element *name*="id" *type*="xs:int" *minOccurs*="0"/>  
 <xs:element *name*="name" *type*="xs:string"/>  
 <xs:element *name*="coordinates" *type*="tns:CoordinateDto"/>  
 <xs:element *name*="creationDate" *type*="xs:long"/>  
 <xs:element *name*="age" *type*="xs:int"/>  
 <xs:element *name*="wingspan" *type*="xs:double"/>  
 <xs:element *name*="color" *type*="xs:string"/>  
 <xs:element *name*="type" *type*="xs:string"/>  
 <xs:element *name*="killerId" *type*="xs:int" *minOccurs*="0"/>  
 </xs:sequence>  
 </xs:complexType>  
</xs:schema>

## service1.5 (mule)

{  
 "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"  
 }  
 }  
 }  
 }  
 }  
 }  
}

# Код

<https://github.com/testpassword/Service-oriented-architecture/tree/master/lab4-12.02.22>

# Выводы

В процессе выполнения лабораторной работы я научился пользоваться технологией esb - шиной для обмена данными между сервиса, в том числе, позволяющей эти данные преобразовать в нужный формат. Концепция и идея мне понравились, она действительно полезна для организации взаимодействия сервисов, а вот сам Mule не очень – он тормозит, выкидывает ошибки, а однажды просто снёс мне весь проект 👍