RESTful Web Service

Descripción general

La aplicación contiene la funcionalidad para poder insertar, actualizar, eliminar y consultar los siguientes componentes:

Objeto	Descripción
Person	Contiene los datos de identidad (nombre, apellido, correo, fecha de ingreso, registroactivo). Este es el registro principal, sin el identificador PersonId no se permitiran registrar los componentes siguientes.
Address	Contiene los datos de ubicación/dirección de la persona registrada, **depende del registro principal (person) para poder relacionar persona-dirección con el identificador PersonId**.
Product	Contiene los datos para ingresar un producto relacionado con el registro persona, de igual forma, **depende del registro principal (person) para poder relacionar persona-producto con el identificador PersonId**.

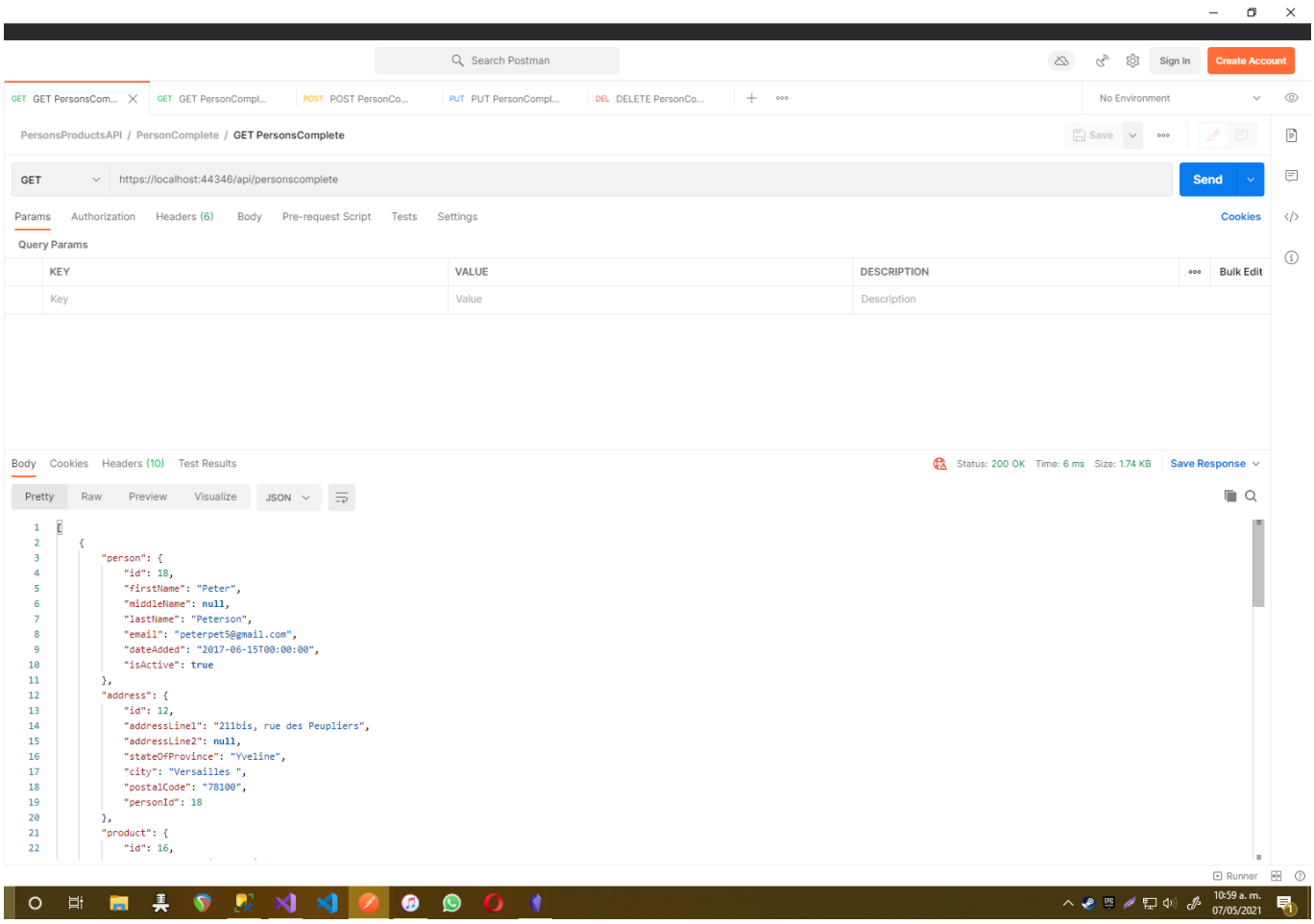
Funcionamiento

El registro completo de estos datos se realiza utilizando el controlador PersonCompleteController en una sola operación, el cual tiene los distintos métodos para consultar uno o todos los registros de tipo person-address-product.

Para esto, utiliza un DataTransferObject (DTO) el cual contiene los 3 objetos a recibir, esto para poder manejar cada una de las operaciónes (get, post, put y delete).

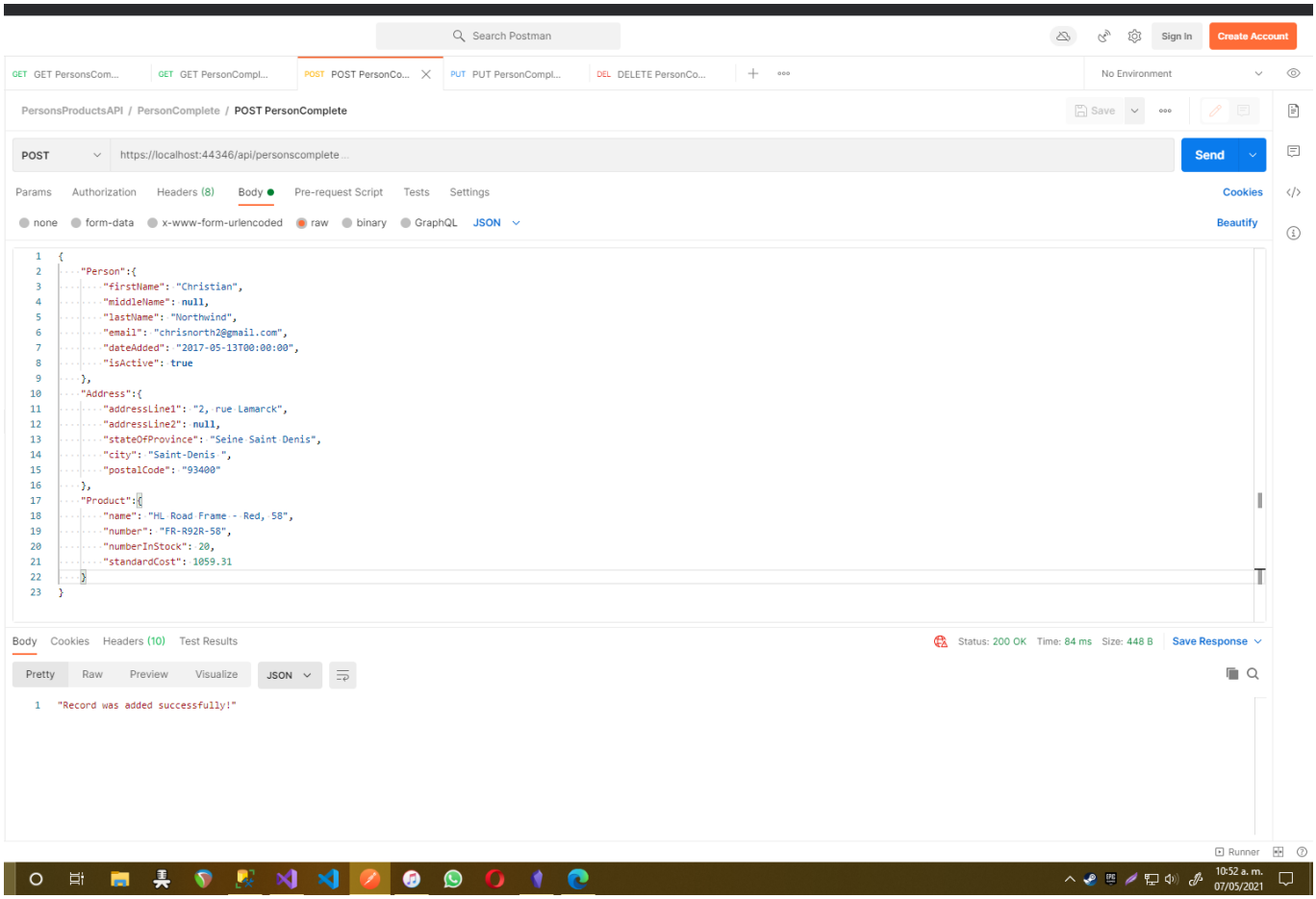
GET

Al consultar la ruta **/api/personscomplete** obtendremos una respuesta compuesta de la siguiente forma:



POST

Para hacer la carga de un registro completo, se requiere recibir un objeto JSON de esta forma:

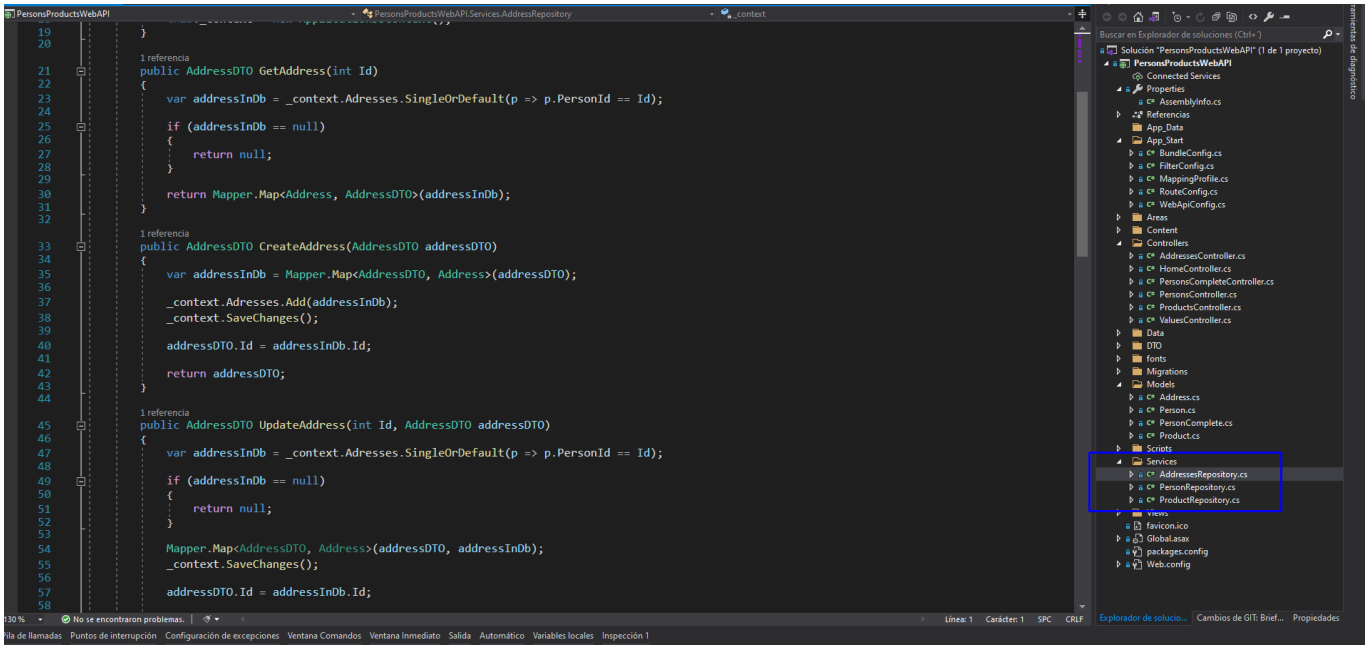


Objeto JSON:

{
"Person":{
"firstName": "Christian",
"middleName": null,
"lastName": "Northwind",
"email": "chrisnorth2@gmail.com",
"dateAdded": "2017-05-13T00:00:00",
"isActive": true
},
"Address":{
"addressLine1": "2, rue Lamarck",
"addressLine2": null,
"stateOfProvince": "Seine Saint Denis",
"city": "Saint-Denis ",
"postalCode": "93400"
},

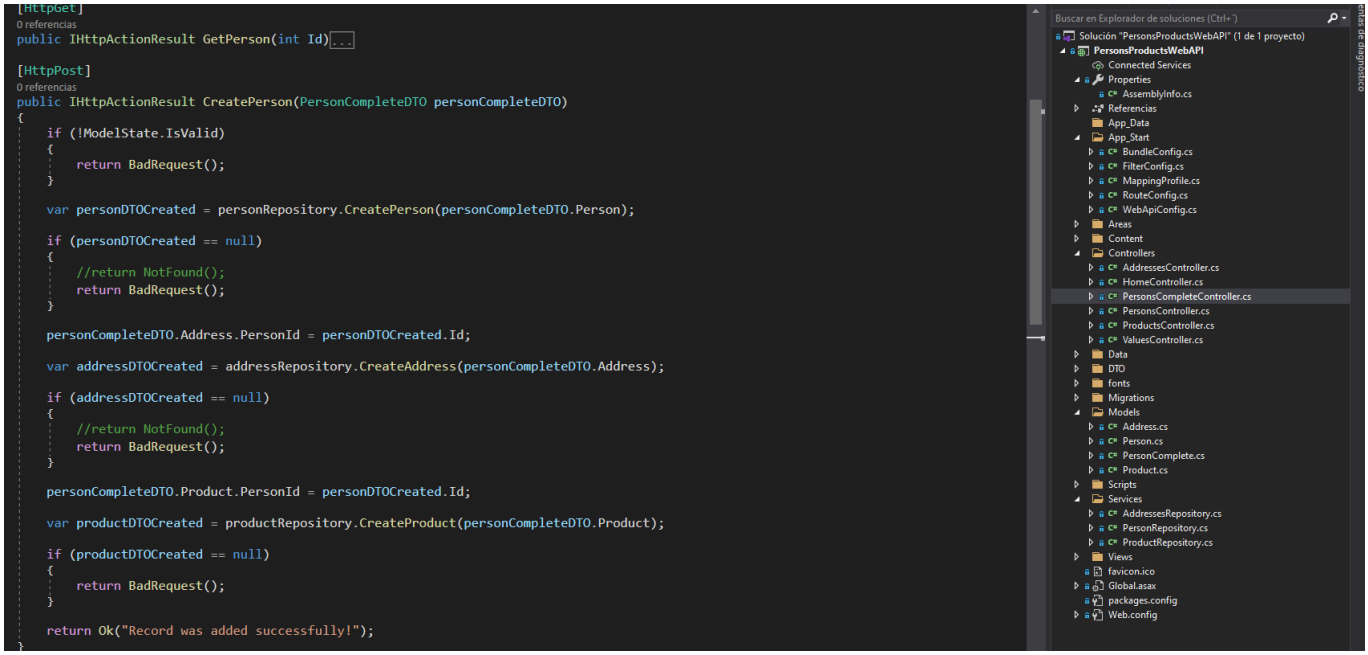
"Product":{
"name": "HL Road Frame - Red, 58",
"number": "FR-R92R-58",
"numberInStock": 20,
"standardCost": 1059.31
}
}

Podemos notar que es un objeto compuesto de las tres entradas distintas **(person-address-product)**, una vez recibiendo la entrada, **llamará a cada uno de los servicios encontrados como PersonRepository, AddressRepository y ProductRepository**, los cuales **contienen las distintas operaciones para crear, actualizar y eliminar** cada uno de los elementos.



En la instrucción POST, **insertará los datos de identidad de primera instancia, ya que sin este registro, no se podrán realizar las inserciones de los objetos tipo address ni product**.

Dentro del método POST se consumen cada uno de los servicios mencionados para poder insertar los componentes:



Debido a esto, es posible operar con las distintas funciones de REST: GET, POST, PUT y DELETE.