

MTIS	Metodologías y Tecnologías de Integración de Sistemas
	Práctica 2
	Diseño Api Rest: Raml

Preámbulo

Una de las tareas más importantes a la hora de crear una SOA es definir su modelo de servicios: o sea, qué servicios hay y qué tareas en concreto hace cada uno. En una SOA basada en servicios web tipo Rest debemos de poder diseñar nuestros servicios al igual que hacíamos con los servicios web tipo SOAP mediante los contratos WSDL, para poder definir en un documento todas las funcionalidades que presenta nuestra API Rest.

RESTful API Modeling Language (RAML) hace que sea fácil administrar todo el ciclo de vida de la API, desde el diseño hasta el uso compartido. Es conciso, solo se escribe lo que se necesita definir, y es reutilizable. Es un diseño API legible por máquina y amigable para los humanos.

El Objetivo de esta práctica es documentar un servicio mediante el RAML, sin necesidad de crear un servicio web. Posteriormente crearemos un servicio web basándonos en el documento RAML y por último generaremos un cliente.

Pasos a realizar

Para poder realizar el documento RAML, contamos con varias herramientas:

- Api Designer, podemos encontrarla en esta URL:

<http://rawgit.com/mulesoft/api-designer/master/dist/index.html>

Esta herramienta es proporcionada por muleESB y también podemos encontrarla en la consola de desarrolladores de muleESB.

La herramienta API Designer, nos permitirá verificar el documento RAML, incluso realizar alguna prueba.

- Atom editor, incluyendo el paquete de Api Designer.
<https://atom.io/>
<https://blog.atom.io/2015/11/03/mulesoft-releases-api-workbench-built-on-atom.html>

Las especificaciones de RAML las podemos encontrar en la siguiente URL:

<https://github.com/raml-org/raml-spec/blob/master/versions/raml-10/raml-10.md/>

Tutoriales básicos:

<https://raml.org/developers/raml-100-tutorial>
<https://raml.org/developers/raml-200-tutorial>

Ejemplo 1:

```
##RAML 1.0
title: Hello world # required title

/HelloWorld: # optional resource
  get: # HTTP method declaration
    queryParameters:
      entrada:
        example: "Hello world"
    responses: # declare a response
      200: # HTTP status code
        body: # declare content of response
          application/json:
            type: | # structural definition of a response (schema or type)
              {
                "title": "Hello world Response",
                "type": "object",
                "properties": {
                  "message": {
                    "type": "string"
                  }
                }
              }
            example: | # example how a response looks like
              {
                "message": "Hello world"
              }
```

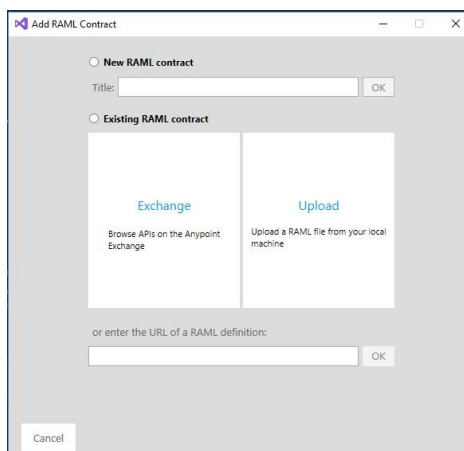
Creación del servicio web

Una vez realizado el documento RAML, el siguiente paso consistirá en crear un servicio web tipo Rest, mediante Visual Studio 2017/2019. Para ello haremos uso de la herramienta RAML Tools for NET:

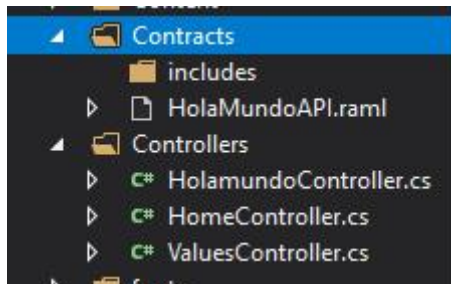
<https://marketplace.visualstudio.com/items?itemName=MuleSoftInc.RAMLTtoolsforNET>

Dentro de Visual Studio 2017/2019, para generar el esqueleto del servicio web Rest, crearemos un proyecto **Aplicación web ASP.NET** (.NET Framework), eligiendo como plantilla Web API.

Una vez creado el proyecto, añadiremos al proyecto con el botón derecho el contrato Raml:



Esto nos creará un esqueleto del contrato en el proyecto ASP:



Ahora deberemos de implementar la lógica de negocio del servicio web mediante ASP.NET.

En el siguiente tutorial, podemos ver todos los pasos necesarios para crear una API Rest:

<https://www.genbetadev.com/formacion/haciendo-una-api-rest-en-1-hora-con-asp-net-web-api-i>

Con los pasos anteriores tendremos construido: el documento RAML, el cual nos detalla las operaciones realizadas y el servicio web, ahora nos queda consumirlo, para ello crearemos un cliente.

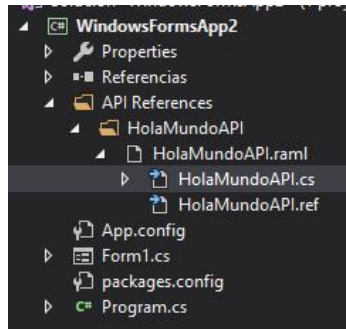
Creación de la aplicación cliente

Para la creación del cliente, emplearemos una aplicación Windows Forms, WPF, ASP, Consola, etc ..., mediante Visual Studio 2017/2019 y con la ayuda de la herramienta RAML Tools for NET :

Añadimos el esqueleto necesario para conectar con la API Rest creada anteriormente.

Pasos:

1. Creamos el proyecto de cliente en Visual Studio 2017/2019.
2. Botón derecho encima de *Referencias* y *Add RAML Reference*.
3. Esto nos creará el esqueleto para poder atacar a la API Rest:



4. El siguiente paso será la creación del cliente, instanciando la clase creada sobre el contrato Raml.

En la siguiente URL, podemos encontrar toda la documentación y ejemplos sobre RAML Tools for NET.

<https://github.com/mulesoft/raml-dotnet-tools>

Enunciado

1. Se deberán crear tres documentos RAML que contengan las mismas funcionalidades que los tres contratos WSDL de la práctica 1.
2. A partir de los documentos RAML creados generar los servicios web, mediante visual studio 2017/2019.

Para la comprobación del correcto funcionamiento de los servicios web, emplearemos las herramientas: SoapUI.

3. Se deberá crear una aplicación cliente, mediante cualquier lenguaje de programación .NET (Windows Forms, WPF, ASP), para la comprobación de dichos servicios.
4. A la hora de valorar la práctica para la máxima nota, se tendrán en cuenta las siguientes consideraciones:

Todos los servicios contendrán un parámetro adicional de entrada, a los especificados a continuación, el cual será un *string*, llamado **RestKey**, este contendrá una clave, la cual deberemos validar que existe en nuestra Base de Datos, para realizar o no, el método solicitado. (Deberemos de crear una tabla en BD llamada restkey).

Adicionalmente a la salida, todos los métodos contarán con un parámetro extra (string), para notificar posibles errores, siendo valorable el empleo de un tipo de datos de error que contenga un código de error y un mensaje.

Entrega

Se debe de entregar una pequeña memoria detallando la puesta en marcha y pasos seguidos para la elaboración de la práctica, así como los proyectos de servicios web y proyecto cliente, por el control creado en el CV.

La fecha límite será el 3/3/2020. En caso de entregar la práctica en una fecha posterior, únicamente se podrá optar a un 5.

El profesor corregirá la práctica durante la clase de prácticas del 4/3/2020.