

{ }

MADA

Metodología Ágil de Desarrollo de APIs

Marco Antonio Sanz

ApiAddicts

¿Quienes somos?



Grupo de meetup

<http://www.meetup.com/API-Addicts/>



Meetups realizados

- ❑ MADA. Metodología ágil de definición de APIs
- ❑ Taller: Definición de APIs
- ❑ Taller: Desarrolla tu primera API
- ❑ Seguridad en las APIs
- ❑ Las APIs en el mundo Big Data
- ❑ Las APIs en el mundo Cloud
- ❑ Apis como modelo de negocio
- ❑ Define y desarrolla tu primera API

Marco Antonio Sanz: <http://es.linkedin.com/pub/marco-antonio-sanz-molina-prados/18/335/97/>

ApiAddicts

Patrocinadores



CloudAppi

Calle Velasco 13

Tlf: **658 89 75 75**

admin@cloudappi.net ·

www.cloudappi.net

¿qué nos ofrece?

- know - how de apis
- Experiencia en el gobierno de Apis
- Ejemplos de arquitecturas
- Experiencia en el mundo Cloud

Objetivo

{ }

¿Cual es el objetivo del meetup?



Intercambiar conocimientos sobre todos los aspectos de las Apis, desde tecnologías, seguridad, definición...

Api como producto

{ }

API



Al pensar en una API, hay que pensar en desarrollar productos. Es un traje para varios clientes, por lo que a todos no les puede quedar bien.

Backend

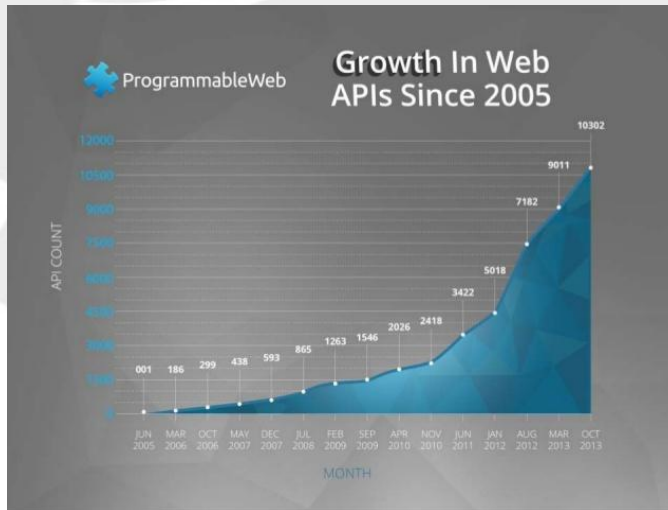


Un backend se desarrolla pensando en tu cliente, es un traje hecho a medida.

Conociendo las Apis



Crecimiento de las Apis



APIs más populares

[Google Maps](#)

[Twitter](#)

[YouTube](#)

[Flickr](#)

[Amazon Product](#)

[Advertising](#)

[Facebook](#)

Datos recogidos de programmable web

Desarrollo de Apis



Pasos:

- 1) Realizar un documento funcional
- 2) Realizar el diseño de la API
- 3) Realizar una implementación fake
- 4) Implementar la API
- 5) Validar la API
- 6) Generar documentación para developers
- 7) Generar casos de prueba (códigos de ejemplo)
- 8) Generar los SDKs

Documentación funcional

{ }

Descripción a alto nivel de la API

2.1.1 Crear un usuario

Se podrá crear un usuario para acceder a la plataforma.

Parámetros de entrada

User

Dato	Descripción	Tipo
external_id	Identificador del usuario en otros sistemas	String
password	Contraseña del usuario.	hash
first_name	Nombre del usuario	String
last_name	Primer apellido	String
last_name_ext	Segundo apellido	String
birth_date	Fecha de nacimiento	Date



Descripción técnica de la API

3.5.6 Consultar un grupo

Descripción

Servicio permite consultar un grupo

Información del servicio

URL	<code>https://XXXX/groups/{group_id}?[token=XXX]</code>
Requiere autenticación	Authorization: Basic
Formato petición	json, xml
Formato respuesta	json, xml
Métodos HTTP	GET
Objeto entrada	
Objeto devuelto	Lista de Group
Documentación web	
¿Público o privado?	Público

uri parameters

group_id	identificador del grupo
----------	-------------------------

Petición de ejemplo JSON

GET	<code>https://xxx/groups/group1?token=<token></code>
-----	--

Respuesta de ejemplo JSON

<pre>{ "header": { "code": "200", "message": "group updated" } }</pre>
--



Consideraciones generales

- Formato de la API (SOAP vs REST)
- Seguridad de la API, métodos de autenticación y autorización. Pj: Basic, oauth1, aouth2...
- API Manager (wso2, apigee, genoa) vs ESB (Oracle Service Bus..)

Documentación - fakes

{ }

Implementando un fake con las interfaces de entrada y salida

http://xxxxxx/users POST

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "birth_country": "ES",
3   "birth_date": "23/11/1981",
4   "email": "user@gmail.com",
5   "password": "prueba",
6   "external_id": "prueba",
7   "first_name": "Nombre",
8   "gender": "M",
9   "last_name": "Apellido1",
10  "second_last_name": "Apellido1",
11  "username": "prueba",
```

Send Save Preview Tests Add to collection

Body Cookies Headers (11) Tests STATUS 201 Created TIME 863 ms

Pretty Raw Preview

```
{
  - header: {
    code: "201",
    message: "User created"
  },
  - data: {
    __v: 0,
    created: "2014-07-02T23:16:27.218Z",
    birth_country: "ES",
    birth_date: "23/11/1981",
```

Implementación

{ }

Consideraciones generales:

- Lenguaje de programación y frameworks a utilizar (java/springmvc, php/zend framework, node/express,.net/.net asp Web API)
- Base de datos SQL vs noSQL
- Instalación en Cloud vs in-house
- Utilización de PaaS, IaaS. ¿Se deben utilizar servicios propios de los Clouds?
- Pruebas de estrés, carga, rendimiento y volumen.

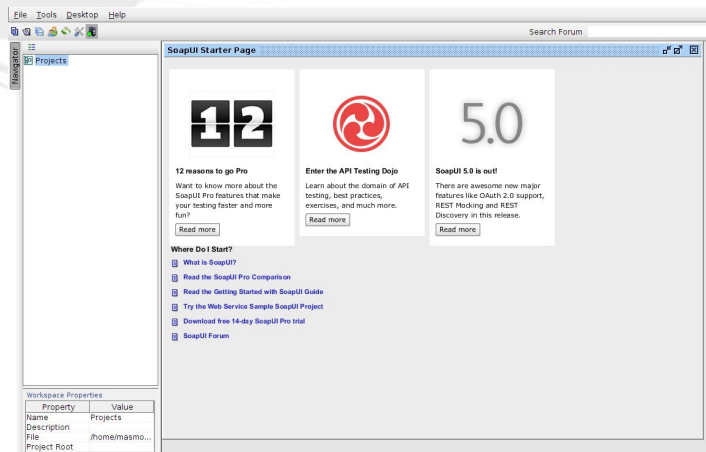
Testing



Validación de la API

Una vez implementada la API, hay que validar que la implementación cumple con las especificaciones.

- Validación manual: Postman
- Validación automática (SOAPUI, jMETER)

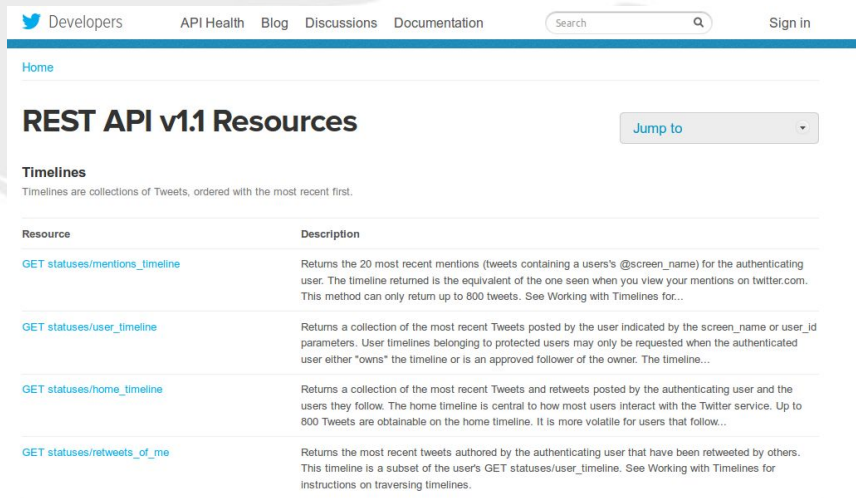


Documentación



Generar la documentación para el developer

Se debe generar documentación clara y comprensible para los developers.



The screenshot shows the Twitter Developers page for REST API v1.1 Resources. The page has a navigation bar with links for Developers, API Health, Blog, Discussions, and Documentation. A search bar and a 'Sign in' button are also present. The main heading is 'REST API v1.1 Resources' with a 'Jump to' dropdown menu. Below this, there is a section for 'Timelines' with a brief description. A table follows, listing four resources with their descriptions.

Resource	Description
GET statuses/mentions_timeline	Returns the 20 most recent mentions (tweets containing a users's @screen_name) for the authenticating user. The timeline returned is the equivalent of the one seen when you view your mentions on twitter.com. This method can only return up to 800 tweets. See Working with Timelines for...
GET statuses/user_timeline	Returns a collection of the most recent Tweets posted by the user indicated by the screen_name or user_id parameters. User timelines belonging to protected users may only be requested when the authenticated user either "owns" the timeline or is an approved follower of the owner. The timeline...
GET statuses/home_timeline	Returns a collection of the most recent Tweets and retweets posted by the authenticating user and the users they follow. The home timeline is central to how most users interact with the Twitter service. Up to 800 Tweets are obtainable on the home timeline. It is more volatile for users that follow...
GET statuses/retweets_of_me	Returns the most recent tweets authored by the authenticating user that have been retweeted by others. This timeline is a subset of the user's GET statuses/user_timeline. See Working with Timelines for instructions on traversing timelines.

Documentación



Casos de prueba

Una de las cosas más importantes es generar casos de prueba para que los developers puedan guiarse en la implementación.

Example Request

GET <https://api.twitter.com/1.1/statuses/retweets/21947795900469248.json>

```
1. [
2.   {
3.     "coordinates": null,
4.     "created_at": "Mon Mar 05 18:31:56 +0000 2012",
5.     "favorited": false,
6.     "truncated": false,
7.     "retweeted_status": {
8.       "coordinates": null,
9.       "created_at": "Mon Jan 03 15:15:36 +0000 2011",
10.      "favorited": false,
11.      "truncated": false,
12.      "id_str": "21947795900469248",
13.      "in_reply_to_user_id_str": null,
14.      "entities": {
15.        "urls": [
16.          ],
17.        "media": [
18.          ],
19.        "hashtags": [
20.          ],
21.        "user_mentions": [
22.          ],
23.        ],
24.      ],
25.    },
26.  ],
27. ],
28. "text": "Tale 2011 I work at Twitter. And so I'm a bird now, too."
```

Please [Sign in](#) with your Twitter account in order to use the OAuth tool.

Tags

- Retweets (43)

POSTMAN

History

Collections

Notify - User



Normal

Basic Auth

Digest Auth

OAuth 1.0



No environment ▼

Enter request URL here

Send

Preview

Add to collection

SDKs



Facilitan la integración con las Apis

Google Cloud SDK

Search

Products > Google Cloud Platform > Google Cloud SDK

- Google Cloud Platform
 - Pricing
 - Pricing Calculator
- Developer Tools
 - Cloud SDK
 - Overview
 - Command-Line Tool Guide (gcloud)
 - Cloud Platform Tools
 - Client Libraries for Cloud Platform APIs
 - Usage Statistics
 - Android Studio
 - Android Studio Backend Templates
 - Google Plugin for Eclipse
 - Cloud Repositories

Google Cloud SDK contains tools and libraries that enable you to easily create and manage resources on Google Cloud Platform, including [App Engine](#), [Compute Engine](#), [Cloud Storage](#), [BigQuery](#), [Cloud SQL](#), and [Cloud DNS](#).

System Requirements: Google Cloud SDK runs on Windows, Mac OS X and Linux, and requires Python 2.7.x. Some of the individual tools bundled with Cloud SDK have more stringent requirements: using App Engine tools for Java development requires Java 1.7+.

Installation and Quick Start

Linux/Mac OS X Windows Alternative methods

Sign Up My Account / Console English

AWS Products & Solutions

Entire Site

Developers Support

RESOURCES

AWS SDK for Java

RELATED LINKS

Documentation

Tools

AWS SDK for Java

Get started quickly using AWS with the AWS SDK for Java. The SDK helps take the complexity out of coding by providing Java APIs for many AWS services including Amazon S3, Amazon EC2, DynamoDB, and more. The single, downloadable package includes the AWS Java library, code samples, and documentation.

1

Getting Started »

Developer Guide »

API Documentation »

Developer Blog »

Downloads

AWS SDK for Java »
[Get the source on GitHub »](#)

AWS Toolkit for Eclipse »
[Get the source on GitHub »](#)

Follow @awsforjava on Twitter

¿Donde aporta Valor?

- 1) Realizar un documento funcional **NO**
- 2) Realizar el diseño de la API **SI**
- 3) Realizar una implementación fake **SI**
- 4) Implementar la API **SI**
- 5) Validar la API **SI**
- 6) Generar documentación **SI**
- 7) Generar SDKS **NO** (por el momento)



Objetivo

Todo los pasos en el desarrollo de una API deben partir de un único documento, el de definición de la API.

Existen varios lenguaje de definición de APIs que permiten obtener nuestra meta, de los cuales los tres más importantes son RAML, SWAGGER y BLUEPRINT



Documento de la API

La API se define en RAML, un lenguaje de definición de APIs.

Parámetros generales de la API

```
#%RAML 0.8

title: World Music API
baseUri: http://example.api.com/{version}
version: v1
traits:
  - paged:
      queryParameters:
        pages:
          description: The number of pages to return
          type: number
  - secured: !include
http://raml-example.com/secured.yml
```

Permite:

- *Describir la API*
- *Incluir ficheros externos*
- *Utilizar propiedades*
- *Incluir schemas*
- *Definir la versión*
- *Definir el tipo de mediaAType (pj:application/json)*
- *Protocolos (HTTP,HTTPS)*
- *Definir la URL base (URL en la que estará desplegada)*
- *Definir documentación en formato Markdown [MARKDOWN].*

Documento de la API

Definiendo métodos GET y DELETE

```
/songs:
  is: [ paged, secured ]
  get:
    queryParameters:
      genre:
        description: filter the songs by genre
    delete:
      description: |
        This method will *delete* an **individual
      responses:
        200:
          body:
            application/json:
              example: !include
              examples/instagram-v1-media-popular-example.json
```

Permite:

- *Describir los parámetros de entrada, tanto query parameters como uriParameters, indicando tipo, descripción, valores por defecto, ejemplos de valores...*
- *Definir los parámetros de salida (definirlo tanto como json schema como por xml). Por ejemplo:*

Documento de la API

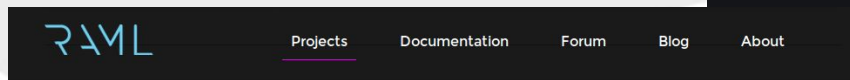
Definiendo métodos POST

```
post:
  /{songId}:
    get:
      responses:
        200:
          body:
            application/json:
              schema: |
                { "$schema": "http://json-schema.org/schema",
                  "type": "object",
                  "description": "A canonical song",
                  "properties": {
                    "title": { "type": "string" },
                    "artist": { "type": "string" }
                  },
                  "required": [ "title", "artist" ]
                }
            application/xml:
```

Permite:

- *Describir los valores de entrada mediante schemas (ya sean json o xsd)*

Api Designer



TOOLS

API Designer

Developed by MuleSoft

License: CPAL-1.0

STABLE

[View on Github](#)

[Try it out](#)

API Designer is a web-based API development tool that allows API providers to design their API quickly, efficiently, and consistently, and socialize the design. It consists of a RAML editor side-by-side with an embedded RAML console (the API Console).

API Console

Developed by MuleSoft

License: CPAL-1.0

STABLE

[View on Github](#)

[Try it out](#)

API Console is a graphical user interface for a RAML-defined API that visually exposes the API's structure and important patterns and serves as interactive API documentation.

API Notebook

Developed by MuleSoft

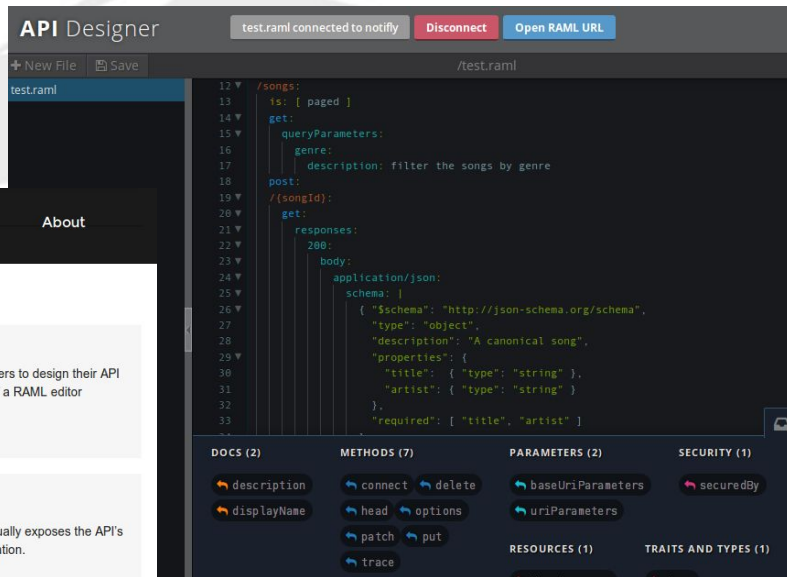
License: CPAL-1.0

STABLE

[View on Github](#)

[Try it out](#)

API Notebook is a web-based, persistent, JavaScript scripting workspace that enables live testing and exploring of APIs, and saving API use cases as markdown gists, so they are versioned, forkable and shareable. It's an example of [iterate programming](#).

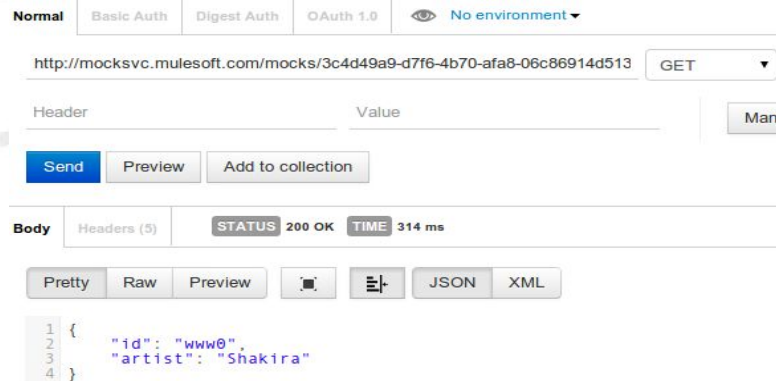




Implementando el servicio fake

Desarrollo de una implementación con las interfaces de entrada y salida.

```
/test.raml
12 /songs:
13   is: [ paged ]
14   get:
15     queryParameters:
16       genre:
17         description: filter the songs by genre
18     responses:
19       200:
20         body:
21           application/json:
22             example: |
23               {
24                 "id": "www0",
25                 "artist": "Shakira"
26             }
27   post:
28     /{songId}:
29       get:
30         responses:
31           200:
32             body:
33               application/json:
34                 schema: |
35                   {
36                     "$schema": "http://json-schema.org/schema"
```





Implementación

Existen Herramientas para generación de parte del código automáticamente

JAX-RS Codegen

Developed by MuleSoft

License: Apache 2

EARLY RELEASE

[View on Github](#)

JAX-RS Codegen is a set of Java tools that support an API-first approach to implementing RAML-described RESTful services in Java with JAX-RS. Given a RAML specification of an API, it generates corresponding JAX-RS-annotated interfaces and supporting classes, which can be used in your project to implement those spec-compliant APIs. *This project is in early alpha.*

Osprey

Developed by MuleSoft

License: Apache 2

EARLY RELEASE

[View on Github](#)

Osprey is a JavaScript framework for rapidly building applications that expose RAML APIs. It's based on Node and Express.

Permite:

- **Generar código en Java o Node**
- **Coexistir implementación fake con**

Implementación

Creando un esquema de aplicación con Osprey

```
var express = require('express');
var path = require('path');
var osprey = require('osprey');

var app = module.exports = express();

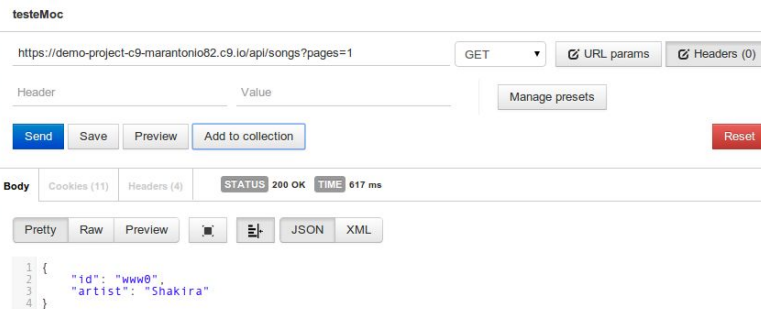
app.use(express.bodyParser());
app.use(express.methodOverride());
app.use(express.compress());
app.use(express.logger('dev'));

app.set('port', process.env.PORT || 3000);

api = osprey.create('/api', app, {
  ramlFile: path.join(__dirname, '/assets/raml/api.raml'),
  logLevel: 'debug' // logLevel: off->No logs | info->Show Osprey modules initializations | debug->Show all
});

// Adding business logic to a valid RAML Resource
// api.get('/examples/exampleId', function(req, res) {
//   res.send({ name: 'example' });
// });

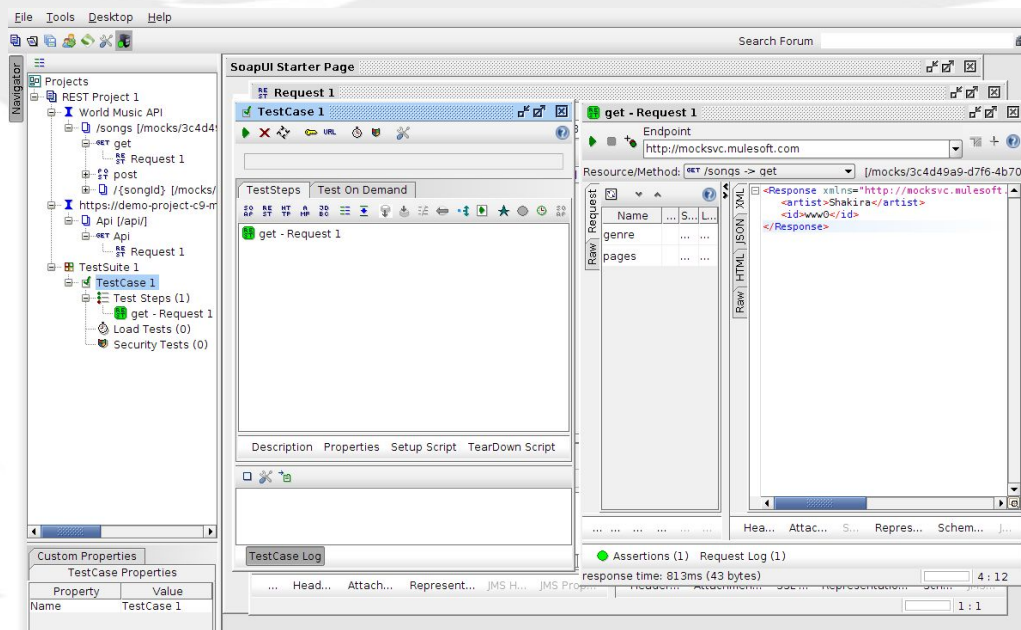
if (!module.parent) {
  var port = app.get('port');
  app.listen(port);
  console.log('listening on port ' + port);
}
```





Validando la API

Importando un raml desde SoapUI



Permite:

- Generar casos de prueba
- Validar los parámetros de entrada como de salida

Documentación

Generando la documentación con RAML

World Music API API documentation version v1

/songs

<http://mocksvc.mulesoft.com/mocks/3c4d49a9-d7f6-4b70-afa8-06c86914d513>

/songs

/songs

GET

POST

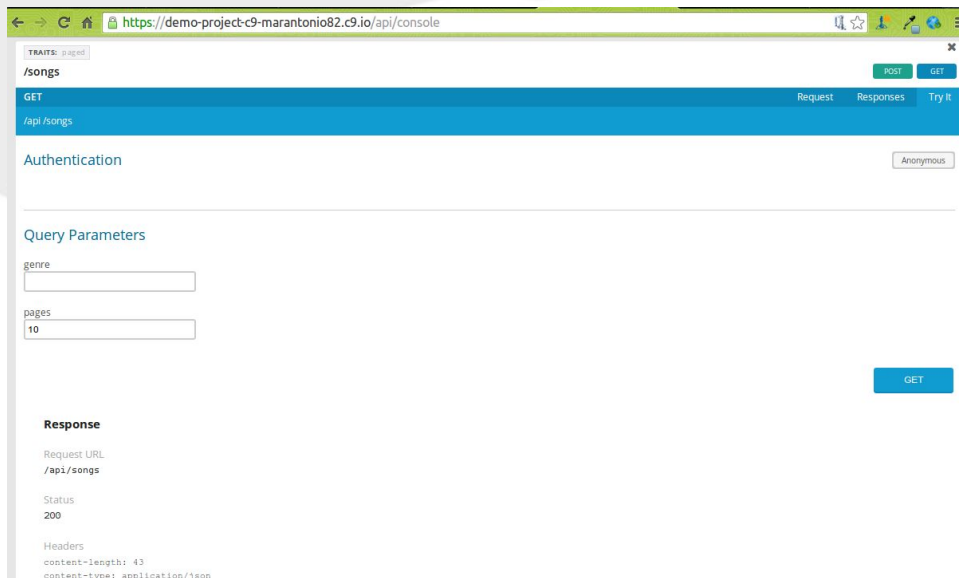
/songs/{songId}

GET

DELETE

Casos de prueba

Desde la consola podemos generar los casos de prueba a partir del RAML



Futuro

RAML RoadMAP

Generación de código de ejemplo en Java, PHP, .net.

{ }

Ruegos y preguntas



{ }



Contacta



Contacta en:

Email: admin@apiaddicts.org

Web:

<http://www.meetup.com/APIAddicts>

Síguenos en:

- LinkedIn: [ApiAddicts](#)
- Twitter: [@apiaddicts](#)
- Facebook: [APIAddicts](#)
- Meetup: [APIAddicts](#)