

Descripción de los métodos para la Web Api

Historial

Creación	27/07/2014
Actualización	31/07/2014
Actualización	06/11/2015
Actualización	15/02/2015

URL BASE

[Base de la llamada al API.](#)

TOKEN

[Obtener un token para un actor](#)

CONFIGURACIÓN

[Obtener la configuración del grupo](#)

ACTOR

[Obtener el perfil de un actor](#)

[Devuelve el perfil de un actor](#)

[Devuelve el perfil del actor que está logueado](#)

[Actualizar el perfil de un actor](#)

[Elimina un actor](#)

CATEGORIAS

[Devolver las categorías de un actor](#)

[Crear una categoría](#)

[Eliminar una categoría](#)

[Actualizar una categoría](#)

GRUPOS

[Crear un grupo](#)

[Eliminar un grupo](#)

[Actualizar un grupo](#)

COMMENT

[Consultar los datos de un comentario](#)

[Consultar los comentarios de un post o evento](#)

[Eliminar un comentario](#)

[Actualizar un comentario](#)

[Crear un comentario](#)

EVENT

[Consultar los datos de un evento](#)

[Consultar los eventos de un actor](#)

[Eliminar un evento](#)

[Actualizar un evento](#)

[Crear un evento](#)

FEEDS

[Consultar los feeds de un actor](#)

[Consultar las notificaciones de un autor](#)

[Eliminar un Feed](#)

FILE

[Consultar los datos de un fichero](#)

[Consultar los ficheros de una carpeta](#)

[Actualizar un archivo](#)

[Eliminar un archivo](#)

[Copiar un archivo a otra carpeta](#)

[Mover un archivo a otra carpeta](#)

[Copiar un archivo a la nube del usuario activo](#)

[Crear un archivo](#)

FOLDER

[Consultar los datos de un actor y las carpetas de la raíz](#)

[Consultar los datos de una carpeta](#)

[Crear una carpeta](#)

[Actualizar una carpeta](#)

[Eliminar una carpeta](#)

LISTAS

[Devuelve los datos de una lista](#)

[Consultar los miembros de una lista](#)

[Consultar las listas de un actor](#)

[Eliminar una lista](#)

[Eliminar un miembro de una lista](#)

[Crea una lista](#)

[Adiciona un miembro a la lista](#)

POST

[Devolver los datos de un post](#)

[Devuelve los post de un actor/evento](#)

[Eliminar un post](#)

[Actualizar un post](#)

[Crear un post](#)

FOLLOWERS

[Devolver los seguidores de un actor/evento](#)

[Devolver los actores a los que el actor sigue](#)

[Seguir a un actor/evento](#)

[Dejar de seguir Actor/evento](#)

LIKES

[Devuelve los actores que han hecho like a un post/evento](#)

[Hacer like a post/evento](#)

[Hacer unlike a post/evento](#)

[Zones](#)

[Devuelve las zonas de un actor en un sitio](#)

URL BASE

Base de la llamada al API.

http://138.100.136.142:9091

TOKEN

Obtener un token para un actor

POST: /token

Hay que pasarle los siguientes parámetros.

Recibe:

HEADER: Content-Type: application/x-www-form-urlencoded

BODY: grant_type=password&username=pepe@chil.org&password=abc**

Devuelve:

access_token:

"8DHRnzejR-712HblAwSIQ1EjidobSacdPpJfHlae84GeWM_RZ2keDcrHd0J39DNE3jjkT4xL9ZVVJv8xN7CroC2gfJaqmcgQ3KLZ8qPIpanWkZfS73zXn2DWMKVCW-85z407kFHyREmo-8wp-RocOtcxxLwoIj_OASRI SdBRTNmAXM1VyHpoU6q1aZd91Xrf-lhWPK7Rd7bqRMEyD1ltkBpAPAxENKAe6D9PnP_hngJGiFcjHiDw0rlWEaZMJ1btGvh5bKJ6Zl3wOKQo1FBLLBE7ytM0k58iaaOYfZx_iYLX4eVh91w2M8ppP2yb"

token_type: "bearer"

expires_in: 31103999

refresh_token: "7F5FDE514E9674DD81668B32B44D231A133414307E1F3AA72A7105161E486936"

The screenshot shows the Advanced Rest Client interface. The request is a POST to `http://138.100.136.142:9091/token` with the following headers and body:

```
Headers: Content-Type: application/x-www-form-urlencoded
Body: grant_type=password&username=pepe@chil.org&password=abc**
```

The response is a 200 OK with the following headers:

```
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 466
Content-Type: application/json; charset=UTF-8
Expires: -1
Server: Microsoft-IIS/5.0
Access-Control-Allow-Origin: *
X-Powered-By: ASP.NET
Date: Fri, 06 Nov 2015 08:48:32 GMT
```

The response body is a JSON object:

```
{
  "access_token": "8DHRnzejR-712HblAwSIQ1EjidobSacdPpJfHlae84GeWM_RZ2keDcrHd0J39DNE3jjkT4xL9ZVVJv8xN7CroC2gfJaqmcgQ3KLZ8qPIpanWkZfS73zXn2DWMKVCW-85z407kFHyREmo-8wp-RocOtcxxLwoIj_OASRI SdBRTNmAXM1VyHpoU6q1aZd91Xrf-lhWPK7Rd7bqRMEyD1ltkBpAPAxENKAe6D9PnP_hngJGiFcjHiDw0rlWEaZMJ1btGvh5bKJ6Zl3wOKQo1FBLLBE7ytM0k58iaaOYfZx_iYLX4eVh91w2M8ppP2yb",
  "token_type": "bearer",
  "expires_in": 31103999,
  "refresh_token": "7F5FDE514E9674DD81668B32B44D231A133414307E1F3AA72A7105161E486936"
}
```

Una vez conseguido el token debe usarse para todas las llamadas que requieren autenticación, tiene una duración de 6 meses, por tanto no es necesario solicitar uno nuevo cada vez. Para hacer las llamadas con el token se hace de la siguiente manera:

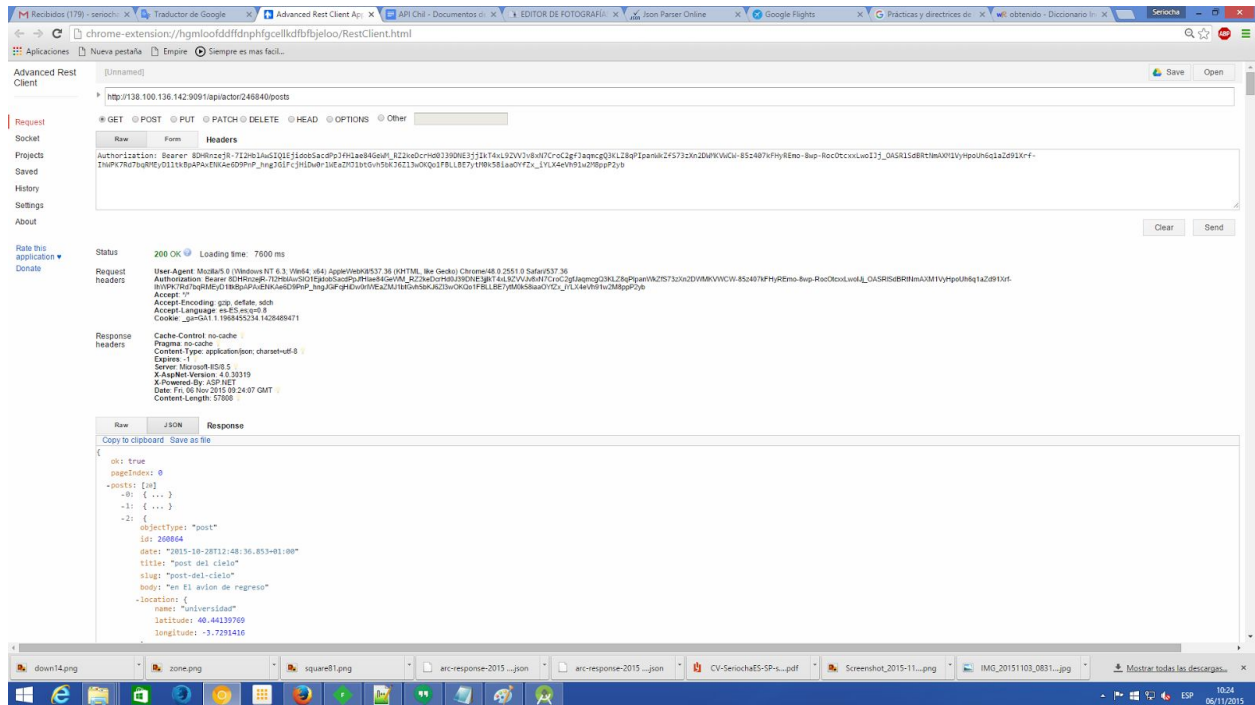
HEADER: Authorization: Bearer + access_token

GET: URL COMPLETA Ej: <http://138.100.136.142:9091/api/actor/{idActor}/posts>

Devuelve:

Objeto JSON de respuesta esta compuesto por:

```
ok: true
pageIndex: 0
posts: [20] //arreglo de objetos post
0: {...} ...
```



CONFIGURACIÓN

Obtener la configuración del grupo

GET: /api/actor/{id}/mobile-config: Devuelve toda la configuración del sitio. Nombre del sitio, pantalla de inicio, opciones de menu para toda la aplicacion.

Recibe => long id

Devuelve => **ok**= true,

permissions : Listado de permisos

config : JSON

Ejemplo:

```
{
  "appName" : "Aprogip",
  "splashScreen" :
"http://138.100.136.142:8080/v2/media/64270f38-157c-430e-b0b3-ad31952ea4e0.jpg",

  "menu" : {"mainWall" : 55607,
            "typeMenu" : "group",
  "header" : {"imagePath" : "@vinotinto2",
              "text" : "AGROGIP"},
  "items":[{"formatType" : "RegularItem1",
            "UID" : "55607",
            "action" : "News",
            "type" : "news",
            "scope" : "Group",
            "text" : "Noticias",
            "publishStatus" : "Public",
            "state" : "Active",
            "subItems" : [{}]}
            ],
  {"formatType" : "RegularItem1",
   "UID" : "55607",
   "action" : "Docs",
   "type" : "document",
   "scope" : "Group",
   "text" : "Documentos",
   "publishStatus" : "Public",
   "state" : "Active",
   "subItems" : [{}]}
}
```

ACTOR

Obtener el perfil de un actor

GET: /api/actor/{id}: Devuelve el perfil de un actor

Recibe => long id

Devuelve JSON =>

```
{ "lastName": "Alfaro", "gender": "female", "birthdate": "1979-06-22T00:00:00", "interests": [], "skills": [], "id": 13, "identifier": "yisel.alfaro", "description": "Software engineer", "headerPictureUrl": "http://138.100.136.142:8080/30095.jpg?w=960&h=140&mode=crop", "email": "yisel.alfaro@gmail.com", "name": "Yisel", "displayName": "Yisel Alfaro", "pictureUrl": "http://138.100.136.142:8080/140742.jpg?w=180&h=180&mode=crop&anchor=topcenter", "type": "user", "thumbUrl": "http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter", "headerType": "image_only", "socialProfiles": {} }
```

Devuelve el perfil de un actor

GET: /api/actor/{identifier}: Devuelve el perfil de un actor. Si el identifier tiene un "punto", hay que

incluir un "/" al final de la url. Ejemplo: http://localhost:59520//api/actor/yisel.alfaro/

Recibe => string identifier

Devuelve JSON => { "lastName": "Alfaro", "gender": "female", "birthdate": "1979-06-22T00:00:00", "interests": [], "skills": [], "id": 13, "identifier": "yisel.alfaro", "description": "Software engineer", "headerPictureUrl": "http://138.100.136.142:8080/30095.jpg?w=960&h=140&mode=crop", "email": "yisel.alfaro@gmail.com", "name": "Yisel", "displayName": "Yisel Alfaro", "pictureUrl": "http://138.100.136.142:8080/140742.jpg?w=180&h=180&mode=crop&anchor=topcenter", "type": "user", "thumbUrl": "http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter", "headerType": "image_only", "socialProfiles": {} }

Devuelve el perfil del actor que está logueado

GET: /api/actor/me: Devuelve el perfil del actor que está logueado

Devuelve JSON=>

```
{ "lastName": "Alfaro", "gender": "female", "birthdate": "1979-06-22T00:00:00", "interests": [], "skills": [], "id": 13, "identifier": "yisel.alfaro", "description": "Software engineer", "headerPictureUrl": "http://138.100.136.142:8080/30095.jpg?w=960&h=140&mode=crop", "email": "yisel.alfaro@gmail.com", "name": "Yisel", "displayName": "Yisel Alfaro", "pictureUrl": "http://138.100.136.142:8080/140742.jpg?w=180&h=180&mode=crop&anchor=topcenter", "type": "user", "thumbUrl": "http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter", "headerType": "image_only", "socialProfiles": {} }
```

w=180&h=180&mode=crop&anchor=topcenter", "type": "user", "thumbUrl": "http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter", "headerType": "image_only", "socialProfiles": {}}

Actualizar el perfil de un actor

POST: /api/actor/{id}/profile-update: Actualiza todo el perfil del actor, se debe pasar todos los datos del actor, nuevos y los que no han cambiado.

Recibe => long id

Devuelve => OK (código de estado HTTP 200)

Devuelve => OK = true

Ejemplo de uso:

- Deben subirse primero las imágenes
- Atrapar el Id y pasarlo en los atributos de header y picture respectivamente.
- Si se quiere una parte específica de la imagen de pasarse en el headerCrop o pictureCrop.

name=Andrés&Birthdate=9/20/1985&phone=55566555&description=Ing.
Agrónomo&lastName=Pérez&Location.Name=Puerta del
Sol&Location.Latitude=40.4168778402635&Location.Longitude=-3.70354056358337&skills=inf
ormatica&interests=TICs&is=ingeniero&interests=project management&skills=project
analisis&picture=106680&header=106942&headerCrop=0,40,600,160,600,375

Elimina un actor

POST: /api/actor/{id}/remove: Elimina un actor

Recibe => long id

Devuelve => OK (código de estado HTTP 200)

POST: /api/actor/add: Crea un actorCATEGORY

GET: /api/actor/{id}: Devuelve los datos de una categoría.

Recibe => long id

Devuelve JSON => {"id":132831,"name":"Prueba","slug":"prueba"}

CATEGORIAS

Devolver las categorías de un actor

GET: /api/actor/{id}/categories: Devuelve las categorías de un actor

Recibe => long id

Devuelve JSON =>

```
[{"id":132831,"name":"Prueba","slug":"prueba"}, {"id":153304,"name":"otraaa categoria web api","slug":"otraaa-categoria-web-api","group":{"id":153305,"name":"grupo para testear web api","slug":"grupo-para-testear-web-api"}}]
```

Crear una categoría

POST: /api/actor/{hostId}/categories/add-category: Crea una categoría de tipo "Category"

Recibe => long hostId, long Id, string Name, string Slug, long? GroupId

Devuelve => long Id

Eliminar una categoría

POST: /api/category/{id}/remove-category: Elimina una categoría de tipo "Category"

Recibe => long id

Devuelve => OK (código de estado HTTP 200)

Actualizar una categoría

POST: /api/category/update-category: Actualiza una categoría de tipo "Category"

Recibe => long hostId, long Id, string Name, string Slug, long? GroupId

Devuelve => OK (código de estado HTTP 200)

GRUPOS

Crear un grupo

POST: /api/actor/{hostId}/categories/add-group: Crea una categoría de tipo "Group"

Recibe => long hostId, long Id, string Name, string Slug

Devuelve => long Id

Eliminar un grupo

POST: /api/category/{id}/remove-group: Elimina una categoría de tipo "Group"

Recibe => long id

Devuelve => OK (código de estado HTTP 200)

Actualizar un grupo

POST: /api/category/update-group: Actualiza una categoría de tipo "Group"

Recibe => long hostId, long Id, string Name, string Slug

Devuelve=> OK (código de estado HTTP 200)

COMMENT

Consultar los datos de un comentario

GET: `/api/comment/{id}`: Devuelve los datos de un comentario

Recibe => long id

Devuelve JSON => `{"id":246,"date":"2012-11-`

`29T08:30:36.883","creator":{"id":13,"identifier":"yisel.alfaro","displayName":"Yisel`

`Alfaro","thumbUrl":"http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter"},"message":"Testing UI...
 Very nice"}`

Consultar los comentarios de un post o evento

GET: `/api/{objectType:regex(post|event)}/{id}/comments`

Recibe => string objectType, long id

Devuelve JSON =>

`[{"id":246,"date":"2012-11-29T08:30:36.883","creator":{"id":13,"identifier":"yisel.alfaro","displayName":`

`ame":"Yisel`
`Alfaro","thumbUrl":"http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter"},"message":"Testing UI...
 Very`

`nice"}, {"id":244,"date":"2012-11-29T08:29:34.327","creator":{"id":10,"identifier":"hectorluis.collado",`

`displayName":"Héctor L`
`Collado","thumbUrl":"http://graph.facebook.com/100000132657631/picture?type=square"},"message":"I'm playing`

`too"}, {"id":242,"date":"2012-11-29T08:24:28.527","creator":{"id":10,"identifier":"hectorluis.collado",`

`displayName":"Héctor L`
`Collado","thumbUrl":"http://graph.facebook.com/100000132657631/picture?type=square"},"message":"I'm playing..."}]`

Eliminar un comentario

POST: `/api/comment/{id}/remove`

Recibe => long id

Devuelve => OK (código de estado HTTP 200)

Actualizar un comentario

POST: `/api/comment/{id}/update`: Actualiza un comentario

Recibe => long id, long objectId: Id del Post, string Message

Devuelve => OK (código de estado HTTP 200)

Crear un comentario

POST: `/api/comment/add`: Crea un comentario

Recibe => long ObjectId, string Message, ObjectTypeId=44 =>siempre

Devuelve => long Id

EVENT

Consultar los datos de un evento

GET: `/api/event/{id}`

Recibe => long id

Devuelve JSON =>

```
{
  "id": 122593,
  "name": "prueba desde el web api",
  "slug": "prueba-desde-el-web-api",
  "description": "texto desde web api",
  "startDate": "2014-07-23T00:00:00",
  "endDate": "2014-07-29T00:00:00",
  "tags": [],
  "categories": [],
  "creator": {
    "id": 13,
    "identifier": "yisel.alfaro",
    "displayName": "Yisel Alfaro",
    "thumbUrl": "http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter"
  },
  "body": "<p>blablablala\n</p>",
  "host": {
    "id": 13,
    "name": "Yisel Alfaro"
  },
  "attachments": []
}
```

Consultar los eventos de un actor

GET: `/api/actor/{id}/events`

Recibe => long id

Devuelve JSON =>

```
[
  {
    "id": 60858,
    "name": "Tomar cerveza en Barcelona",
    "slug": "tomar-cerveza-en-barcelona",
    "description": "A tomar cerveza en Barcelona junto al sol. Que maravilla",
    "startDate": "2013-07-12T04:00:00",
    "creator": {
      "id": 13,
      "identifier": "yisel.alfaro",
      "displayName": "Yisel Alfaro",
      "type": "user",
      "thumbUrl": "http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter"
    },
    "host": {
      "id": 13,
      "name": "Yisel Alfaro"
    },
    "location": {
      "name": "Barcelona, España",
      "latitude": 41.3850639,
      "longitude": 2.17340349999995
    },
    "attachments": [
      {
        "id": 880,
        "name": "Sagrada Familia",
        "slug": "sagrada-familia",
        "type": "image",
        "format": "jpg",
        "previewUrl": "http://138.100.136.142:8080/30094.jpg",
        "contentUrl": "",
        "url": "http://www.confortelhoteles.com/uploads/galeriaprovincias/hoteles-sagrada-familia-barcelona.jpg",
        "metadata": {
          "width": 800,
          "height": 600
        }
      },
      {
        "id": 882,
        "name": "Mobile WC",
        "slug": "mobile-wc",
        "description": "testing",
        "type": "image",
        "format": "jpg",
        "previewUrl": "http://138.100.136.142:8080/30095.jpg",
        "contentUrl": "",
        "url": "http://www.essentialstyleformen.com/wp-content/uploads/2009/09/barcelona.jpg",
        "metadata": {
          "width": 2105,
          "height": 1427
        }
      },
      {
        "id": 884,
        "name": "Montjuí",
        "slug": "montjuí",
        "type": "image",
        "format": "jpg",
        "previewUrl": "http://138.100.136.142:8080/30096.jpg",
        "contentUrl": "",
        "url": "http://static-c.localnomad.com/panoramic/barcelona-loft-gracia-1-69391.jpg",
        "metadata": {
          "width": 560,
          "height": 214
        }
      }
    ]
  },
  {
    "id": 122593,
    "name": "prueba desde el web api",
    "slug": "prueba-desde-el-web-api",
    "description": "texto desde web api",
    "startDate": "2014-07-23T00:00:00",
    "endDate": "2014-07-29T00:00:00",
    "creator": {
      "id": 13,
      "identifier": "yisel.alfaro",
      "displayName": "Yisel Alfaro",
      "type": "user",
      "thumbUrl": "http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter"
    },
    "body": "<p>blablablala\n</p>",
    "host": {
      "id": 13,
      "name": "Yisel Alfaro"
    },
    "permissions": [
      "update",
      "remove"
    ]
  },
  {
    "id": 153299,
    "name": "prueba"
  }
]
```

desde el web api", "slug": "prueba-desde-el-web-api", "description": "texto desde web api", "startDate": "2014-07-23T00:00:00", "endDate": "2014-07-29T00:00:00", "creator": {"id": 13, "identifier": "yisel.alfaro", "displayName": "Yisel Alfaro", "type": "user", "thumbUrl": "http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter"}, "body": "<p>blablabalabala\n</p>", "host": {"id": 13, "name": "Yisel Alfaro"}, "permissions": ["update", "remove"]}]

Eliminar un evento

POST: */api/event/{id}/remove*

Recibe => long id

Devuelve => OK (código de estado HTTP 200)

Actualizar un evento

POST: */api/event/{id}/update*

Recibe => long id, string Name, string Description, DateTime StartDate, DateTime? EndDate, long? PictureId, long HostId, LocationDto Location, IEnumerable<long> Attachments, string Tags, string Body, IEnumerable<long> Categories

Devuelve => OK (código de estado HTTP 200)

Crear un evento

POST: */api/event/add*

Recibe => string Name, string Description, DateTime StartDate, DateTime? EndDate, long? PictureId, long HostId, LocationDto Location, IEnumerable<long> Attachments, string Tags, string Body, IEnumerable<long> Categories

Devuelve => long Id

FEEDS

Consultar los feeds de un actor

GET: */api/actor/{id}/feeds*

Recibe => long id

Devuelve JSON => [{"id":162907,"date":"2014-07-24T10:44:57.087","actor":{"id":60748,"identifier":"saisoto","displayName":"Isa Soto","type":"user","thumbUrl":"http://138.100.136.142:8080/140755.jpg?w=50&h=50&mode=crop&anchor=topcenter"},"action":"follow","isRead":false},{id":162894,"date":"2014-07-22T11:17:53.723","actor":{"id":13,"identifier":"yisel.alfaro","displayName":"Yisel Alfaro","type":"user","thumbUrl":"http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter"},"action":"add_event","isRead":true,"object":{"id":153299,"name":"prueba desde el web api","slug":"prueba-desde-el-web-api","description":"texto desde web api","startDate":"2014-07-23T00:00:00","endDate":"2014-07-29T00:00:00","body":"<p>blablabla bala\n</p>","host":{"id":13,"name":"Yisel Alfaro"}}}]

Consultar las notificaciones de un autor

GET: */api/actor/{id}/notifications*

Recibe => long id

Devuelve JSON =>

[{"id":142882,"date":"2014-06-25T15:07:52.347","actor":{"id":10,"identifier":"hectorluis.collado","displayName":"Héctor L Collado","type":"user","thumbUrl":"http://graph.facebook.com/100000132657631/picture?type=square"},"action":"follow","isRead":true},{id":142878,"date":"2014-06-24T10:37:32.797","actor":{"id":10,"identifier":"hectorluis.collado","displayName":"Héctor L Collado","type":"user","thumbUrl":"http://graph.facebook.com/100000132657631/picture?type=square"},"action":"add_manager","isRead":true}]

Eliminar un Feed

POST: */api/feed/{id}/remove*

Recibe => long id

Devuelve => OK (código de estado HTTP 200)

FILE

Consultar los datos de un fichero

GET: `/api/file/{id}`

Recibe => long id

Devuelve JSON =>

```
{"id":153279,"name":"mas y mas prueba","slug":"mas-y-mas-prueba","description":"mas pruebas del doby","type":"image","format":"jpg","previewUrl":"http://138.100.136.142:8080/170883.jpg","contentUrl":"/download-doc/170883","metadata":{"width":1347,"height":1917,"size":482226}}
```

Consultar los ficheros de una carpeta

GET: `/api/folder/{id}/files`: Devuelve los archivos de una carpeta

Recibe => long id

Devuelve JSON =>

```
[{"id":133070,"name":"4.2_create_group","slug":"42_create_group","type":"image","format":"png","previewUrl":"http://138.100.136.142:8080/150821.jpg","contentUrl":"/download-file/0-133070","metadata":{"width":315,"height":167,"size":23507},"{"id":133075,"name":"La Revista del Consumidor","slug":"la-revista-del-consumidor","description":"Como ya habrás notado, un nuevo fenómeno está arrasando en Europa y Norteamérica: las páginas web de subastas a céntimo. Por todas partes en la web, en blogs, se pueden leer historias de felices ganadores que han conseguido increíbles productos, generalmente bastante caros, por una ridícula suma de dinero. Como no estábamos 100% convencidos, hemos decidido poner a prueba este tipo de páginas web – para asegurarnos de que estas gangas son verdaderamente reales. ¿Y qué mejor manera que probarlo con unomismo?","type":"web_page","previewUrl":"http://138.100.136.142:8080/150826.jpg","contentUrl":"","url":"http://megabargains24.com/ES/09/?KW=DSPLIT&CAMP=ES_A66"}]
```

Actualizar un archivo

POST: `/api/file/{id}/update`: Actualiza un archivo

Recibe => long id, string Name, string Description, string Url, long? HostId

Devuelve => OK (código de estado HTTP 200)

Eliminar un archivo

POST: `/api/file/{id}/remove/{folderId}`: Elimina un archivo

Recibe => long id, long folderId

Devuelve => OK (código de estado HTTP 200)

Copiar un archivo a otra carpeta

POST: `/api/file/{id}/copy/{sourceFolderId}/to/{destinationFolderId}`: Hacer copia de un archivo, de una carpeta hacia otra.

Recibe => long id, long sourceFolderId, long destinationFolderId
Devuelve => OK (código de estado HTTP 200)
Devuelve => OK = true

Mover un archivo a otra carpeta

POST: /api/file/{id}/move/{sourceFolderId}/to/{destinationFolderId}: Mover un archivo, de una carpeta hacia otra.
Recibe => long id, long sourceFolderId, long destinationFolderId
Devuelve => OK (código de estado HTTP 200)
Devuelve => OK = true

Copiar un archivo a la nube del usuario activo

POST: /api/file/{id}/clone
Recibe => long id
Devuelve => OK (código de estado HTTP 200)
Devuelve => OK = true
Nota: El id es del fichero no de Attachment, si es un attach se puede utilizar el campo FileId

Crear un archivo

POST: /api/file/add: Crea un archivo
Recibe => file, Name, Description, Url, HostId, FolderId
Devuelve => long Id
Description, Url: no son obligatorios.
Name = "file" siempre; filename: nombre del fichero. Esta información debe llegar al server en el "Content-Disposition"
En el header: "Content-Type: multipart/form-data"

ej programación móvil:
Content-Disposition: form-data; name="file"; filename=variable_filename

ej completo para Advanced REST Client:

- subir fichero en Tab Files de REST Client, con el filename = 'file' (único textbox que aparece)
- adicionar cada parámetro en Data Form

Significado por campo:

name=contenidos => nombre que tendrá el fichero visualmente en la nube.
filename=contenidos.jpg => nombre del contenido del fichero.
Description="texto para describir el fichero"
HostId=62834 => portal al que pertenece.
FolderId=119903 => carpeta en la que se subirá.

Queda así:

name=contenidos&filename=contenidos.jpg&HostId=62834&FolderId=119903

FOLDER

Consusltar los datos de un actor y las carpetas de la raiz

GET: /api/actor/{id}/folders: Devuelve los datos de un actor y las carpetas que están en su raíz.

Recibe => long id

Devuelve JSON =>

```
{"id":132941,"name": "_root", "subFolders":[{"id":112203,"name": "Cuba", "description": "La bella!!!"}, {"id":122550,"name": "EUA", "description": ""}, {"id":133069,"name": "Brasil", "description": ""}, {"id":153266,"name": "viaje a cuba", "description": "muy pronto"}]}
```

Consultar los datos de una carpeta

GET: /api/folder/{id}: Devuelve los datos de una carpeta

Recibe => long id

Devuelve JSON =>

```
{"id":112203,"name": "Cuba", "description": "La bella!!!", "subFolders":[{"id":112537,"name": "La Habana", "description": ""}]}
```

Crear una carpeta

POST: /api/folder/add

Recibe => string Name, string Description, long? ParentFolderId, long? HostId, bool IsPublic

Devuelve => long Id

Actualizar una carpeta

POST: /api/folder/{id}/update: Actualiza una carpeta

Recibe => long id, string Name, string Description, long? ParentFolderId, long? HostId, bool IsPublic

Devuelve => OK (código de estado HTTP 200)

Eliminar una carpeta

POST: /api/folder/{id}/remove

Recibe => long id

Devuelve => OK (código de estado HTTP 200)

LISTAS

Devuelve los datos de una lista

GET: /api/list/{id}

Recibe => long id Devuelve JSON => {"id":153301,"name":"nueva lista","description":"desde web api","isPublic":false}

Consultar los miembros de una lista

GET: /api/list/{id}/members

Recibe => long id

Devuelve JSON =>

{ OK: true,

members:

[{"id":696,"identifier":"alejandro.collado","displayName":"Alejandro Collado","thumbUrl":"/actor/default-thumb/696"}]

Consutar las listas de un actor

GET: /api/actor/{id}/lists: Devuelve las listas de un actor

Recibe => long id

Devuelve JSON =>

[{"id":14,"name":"Coworkers","description":"Your coworkers","isPublic":false}, {"id":15,"name":"Friends","description":"Your friends","isPublic":false}, {"id":133246,"name":"testList","isPublic":true}, {"id":153301,"name":"nueva lista","description":"desde web api","isPublic":false}]

Eliminar una lista

POST: /api/list/{id}/remove: Elimina una lista

Recibe => long id

Devuelve => OK (código de estado HTTP 200)

Eliminar un miembro de una lista

POST: /api/list/{id}/remove-member: Elimina un miembro de una lista

Recibe => long id, long actorId

Devuelve => OK (código de estado HTTP 200)

POST: /api/list/{id}/remove-member: Elimina un miembro de una lista

Recibe => long id, string email

Devuelve => OK (código de estado HTTP 200)

Crea una lista

POST: /api/list/add: Crea una lista

Recibe => long actorId, string Name, string Description, bool IsPublic

Devuelve => long Id

Adiciona un miembro a la lista

POST: /api/list/{id}/add-member: Adiciona un miembro a una lista

Recibe => long id, long actorId

Devuelve => OK (código de estado HTTP 200)

POST: /api/list/{id}/add-member: Adiciona un miembro a una lista

Recibe => long id, string Name, string Email

Devuelve => OK (código de estado HTTP 200)

POST

Devolver los datos de un post

GET: /api/post/{id}: Devuelve los datos de un post

Recibe => long id

Devuelve JSON =>

```
{ OK: true,
post: {"id":153281,"date":"2014-07-21T12:22:33.89","title":"esto es un nuevo post","slug":"esto-
es-un-nuevo-post","body":"<p>aaaaaaaaaaaaa bbbbbbbbbbb\n</
p>","description":"descripcion","attachments":[],"type":"post","host":{"id":13,"name":"Yisel
Alfaro"},"url":"https://
www.google.es/","sourceName":"aaasss","tags":["agua","prueba"],"categories":[],"comme
ntsCount":0,"likesCount":0,"creator":{"id":13,"identifier":"yisel.alfaro","displayName":"Yisel
Alfaro","type":"user","thumbUrl":"http://138.100.136.142:8080/140742.jpg?
w=50&h=50&mode=crop&anchor=topcenter"},"status":1} }
```

Devuelve los post de un actor/evento

GET: /api/{objectType:regex(actor|zone)}/{hostId}/posts

Ej: http://138.100.136.142:9091/api/actor/260874/posts

Recibe => string objectType, long hostId

Devuelve JSON =>

```
{ OK : true,
pageIndex: 0,
posts: [{"id":342,"date":"2013-01-17T12:47:51.327","body":"Probando la
UI","attachments":[{"id":113,"name":"yisel","slug":"yisel","type":"image","format":"jpg","previewUrl
":"http://138.100.136.142:8080/12.jpg","contentUrl":"/download-doc/12","metadata":{"width":"197
","height":"250"},"size":16384}],type":"post","host":{"id":13,"name":"Yisel
Alfaro"},"commentsCount":0,"likesCount":1,"creator":{"id":13,"identifier":"yisel.alfaro","displayNa
me":"Yisel Alfaro","type":"user","thumbUrl":"http://138.100.136.142:8080/140742.jpg?
w=50&h=50&mode=crop&anchor=topcenter"},"status":1,"permissions":["update","remove"]},{"id"
:892,"date":"2013-02-27T13:43:24.647","body":"Barcelona - Like a jewel in the
sun","location":{"name":"Barcelona,
España","latitude":41.3850639,"longitude":2.17340349999995},"attachments":[{"id":880,"name":
"Sagrada
Familia","slug":"sagrada-familia","type":"image","format":"jpg","previewUrl":"http://138.100.136.142:8080/30094.jpg","contentUrl":"","url":"http://www.confortelhoteles.com/uploads/galeriaprovincias/hoteles-sagrada-familia-barcelona.jpg","metadata":{"width":"800","height":"600"}}] }
```

También se pueden pasar algunos parámetros con el objetivo de limitar la respuesta:

1. Determinados tipos de post (post, news, question, document, event, link)
2. Categorías previamente seleccionadas.(filtrado de contenidos por id categorías)
3. Paginado.

Ej: <http://138.100.136.142:9091/api/actor/1010/posts?types=question.post&categories=25373,14582>

Eliminar un post

POST: `/api/post/{id}/remove`

Recibe => long id

Devuelve => OK : true // (código de estado HTTP 200)

Actualizar un post

POST: `/api/post/{id}/update`

Recibe => long id, string Title, string Body, string Description, LocationDto Location, IEnumerable<long> Attachments, string Type, long HostId, long? CoverImg, string Url, string SourceName, long? FormatId, string Tags, IEnumerable<long> Categories, bool includeFirstImageAttachmentAsCover

Devuelve => OK : true (código de estado HTTP 200)

Nota: El paso de parámetros es idéntico que cuando adionas un post nuevo.

- En el caso de la imagen del post: el CoverImg se pasa con otro valor.
- Si se quiere eliminar la imagen actual no se pasa.

Crear un post

POST: `/api/post/add`

Recibe => string Title, string Body, string Description, LocationDto Location, IEnumerable<long> Attachments, string Type, long (HostId o Zoneld), long? CoverImg, string Url, string SourceName, long? FormatId, string Tags, IEnumerable<long> Categories, bool IncludeFirstImageAttachmentAsCover

Devuelve =>

{ OK : true,
post: long Id}

Nota : Los parámetros “Attachments” , “Tags” y “Categories” deben pasarse como se muestra en el siguiente ejemplo: Title=esto es un titulo de prueba&Body=esto es un texto de prueba&Url=https://www.google.es/

&Description=esto es la descripcion&SourceName=esto es la fuente&Attachments[0].FileId=1&Attachments[1].FileId=2&Attachments[2].FileId=3&Tags=agua, prueba&Categories=58777&Categories=56778&HostId=13&Type=post

- El parámetro *Location* es un objeto y hay que pasarlo todo por separado ejemplo:

Location.Name=Puerta del

Sol&Location.Latitude=40.4168778402635&Location.Longitude=-3.70354056358337

- Los parámetros HostId y Zoneld son excluyentes (solo uno de los dos).
- Si no eres administrador del sitio no hará caso al parámetro Status.
- Para la imagen del post se usa el CoverImg

FOLLOWERS

Devolver los seguidores de un actor/evento

GET: */api/{objectType:regex(actor|event)}/{hostId}/followers*

Recibe => string objectType, long hostId

Devuelve JSON =>

```
[{"id":10,"identifier":"hectorluis.collado","displayName":"Héctor L Collado","description":"Guajiro = War Hero","thumbUrl":"http://graph.facebook.com/100000132657631/picture?type=square"}, {"id":60748,"identifier":"saisoto","displayName":"Isa Soto","description":"Hola","thumbUrl":"http://138.100.136.142:8080/170884.jpg?w=50&h=50&mode=crop&anchor=topcenter"}, {"id":386,"identifier":"johndoe","displayName":"John Doe","description":"Usuario de pruebas de Chil","thumbUrl":"http://138.100.136.142:8080/70132.jpg?w=50&h=50&mode=crop&anchor=topcenter"}, {"id":34,"identifier":"Isoto.upm","displayName":"Luis Soto","thumbUrl":"http://graph.facebook.com/100003082699040/picture?type=square"}]
```

Devolver los actores a los que el actor sigue

GET: */api/actor/{id}/following*: Devuelve a los que el actor está siguiendo

Recibe => long id

Devuelve JSON =>

```
[{"id":696,"identifier":"alejandro.collado","displayName":"Alejandro Collado","description":"Pintor y diseñador escenográfico","thumbUrl":"/actor/default-thumb/696"}, {"id":20970,"identifier":"chil-dev","displayName":"Chil Development Team","description":"Noticias sobre el desarrollo de Chil","thumbUrl":"http://138.100.136.142:8080/50141.jpg?w=50&h=50&mode=crop&anchor=topcenter"}, {"id":10,"identifier":"hectorluis.collado","displayName":"Héctor L Collado","description":"Guajiro = War Hero","thumbUrl":"http://graph.facebook.com/100000132657631/picture?type=square"}]
```

Seguir a un actor/evento

POST: */api/{objectType:regex(actor|event)}/{hostId}/follow*

Recibe => string objectType, long hostId

Devuelve =>OK (código de estado HTTP 200)

Dejar de seguir Actor/evento

POST: */api/{objectType:regex(actor|event)}/{hostId}/unfollow*

Recibe => string objectType, long hostId

Devuelve =>OK (código de estado HTTP 200)

LIKES

Devuelve los actores que han hecho like a un post/evento

GET: */api/{objectType:regex(post|event)}/{hostId}/likes*

Recibe => string objectType, long hostId

Devuelve JSON => [{"id":10,"identifier":"hectorluis.collado","displayName":"Héctor L Collado","description":"Guajiro = War

Hero","thumbUrl":"http://graph.facebook.com/100000132657631/picture?type=square","isCurrentUserFollower":true}, {"id":13,"identifier":"yisel.alfaro","displayName":"Yisel

Alfaro","description":"Software engineer","thumbUrl":"http://138.100.136.142:8080/140742.jpg?w=50&h=50&mode=crop&anchor=topcenter"}]

Hacer like a post/evento

POST: */api/{objectType:regex(post|event)}/{hostId}/like*

Recibe => string objectType, long hostId

Devuelve =>OK (código de estado HTTP 200)

Hacer unlike a post/evento

POST: */api/{objectType:regex(post|event)}/{hostId}/unlike*

Recibe => string objectType, long hostId

Devuelve =>OK (código de estado HTTP 200)

Zones

Devuelve las zonas de un actor en un sitio

GET: `/api/actor/{HostId}/zones`

Nota: Identifica al usuario activo por el token que le pasa.

Devuelve =>

```
{ OK : true,  
  zones: listados de zonas}
```

Ejemplo:

```
"ok": true  
"zones": [1]  
  0: {  
    "objectType": "zone"  
    "id": 134  
    "name": "Prueba"  
    "slug": "prueba"  
    "description": "Prueba"  
    "permissions": [3]  
      0: "add_post"  
      1: "visibility"  
      2: "comment"  
    }  
}
```

Nota: El listado de zonas visible se corresponde con el nivel de acceso del usuario a zonas privadas dentro del grupo.

Importante: Si la zona es pública, cosa que no debería pasar, aparecerá el listado de permisos con los accesos públicos.