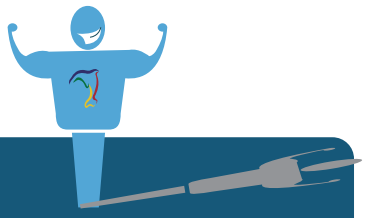
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

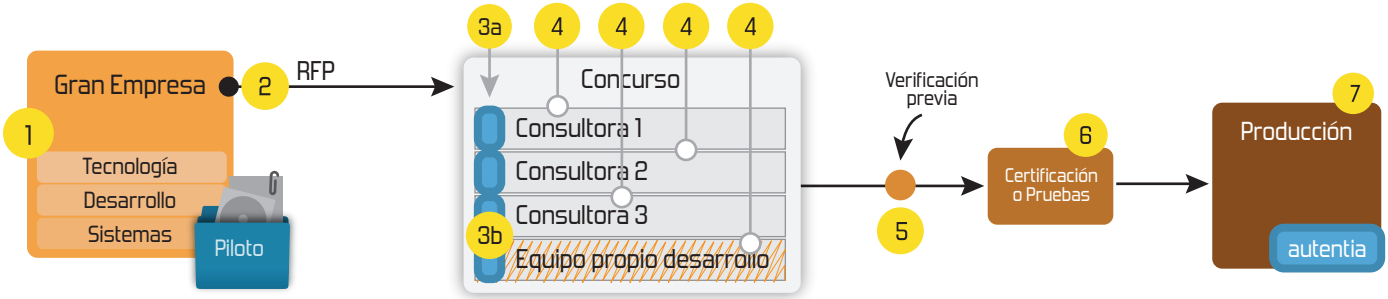
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.

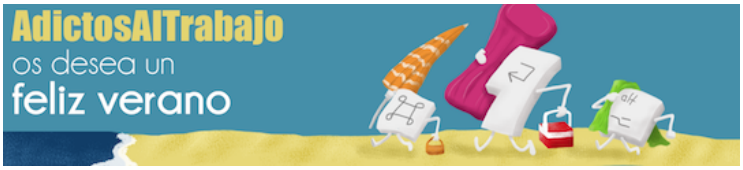


4. Cursos de formación (impartidos por desarrolladores en activo)

- Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery
- Gestor portales (Liferay)
- Gestor de contenidos (Alfresco)
- Aplicaciones híbridas
- Control de autenticación y acceso (Spring Security)
- UDDI
- Web Services
- Rest Services
- Social SSO
- SSO (Cas)
- JPA-Hibernate, MyBatis
- Motor de búsqueda empresarial (Solr)
- ETL (Talend)
- Dirección de Proyectos Informáticos.
- Metodologías ágiles
- Patrones de diseño
- TDD
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com



autentia
Soporte a desarrollo informático
Hosting patrocinado por
enredados

Entra en Adictos a través de
E-mail
Contraseña
Entrar
Registrarme
Olvidé mi contraseña

Inicio Quiénes somos Formación Comparador de salarios Nuestros libros Más

» Estás en: Inicio Tutoriales Introducción a WSO2 API Manager

Miguel Arlandy Rodríguez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/JEE

Ver todos los tutoriales del autor

Fecha de publicación del tutorial: 2014-07-03

Tutorial visitado 1 veces Descargar en PDF

Introducción a WSO2 API Manager.

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. ¿Qué es WSO2 API Manager?.
- 4. ¿Qué vamos a hacer?.
- 5. Nuestro servicio.
- 6. Exponiendo y securizando el servicio en WSO2 API Manager.
  - 6.1. Instalando WSO2 API Manager.
  - 6.2. Creando usuarios para publicar y consumir API´s.
  - 6.3. Publicando una API.
  - 6.4. Suscribiéndonos a un API.
- 7. Consumiendo el API a través de WSO2 API Manager.
- 8. Referencias.
- 9. Conclusiones.

1. Introducción

Cada vez son más las compañías y departamentos de IT los que se suben al carro de los **servicios como forma de exponer parte de su lógica de negocio** o simplemente datos. Los servicios, tanto SOAP como REST, nos permiten poder compartir esta lógica de una manera muy sencilla de forma que pueda ser **consumida desde diferentes canales**: móviles, aplicaciones web, procesos batch o lo que sea... Incluso, si diseñamos correctamente nuestros servicios podríamos llegar a tener una verdadera arquitectura SOA.

Todo esto suena muy bien pero todavía podríamos ir más lejos. Imaginemos que, además de exponer nuestra lógica a través de servicios, quisiéramos **securizarlos**, establecer políticas de **disponiblidad**, **monitorizar** el tráfico, controlar **quién consume** cada servicio, **monetizar** su uso, **balancear** la carga, tener un **catálogo** con las APIs de los servicios, poder **documentar** las APIs, etc, etc, etc... Pues bien, ahí es donde entra en juego **WSO2 API Manager**.

En este tutorial intentaremos explicar qué es WSO2 API Manager, cuáles son sus principales características y veremos un ejemplo exponiendo un servicio REST a través del API Manager.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

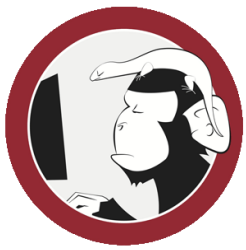
- Hardware: Portátil MacBook Pro 15' (2.2 Ghz Intel Core I7, 8GB DDR3).
- Sistema Operativo: Mac OS X Mavericks 10.9
- WSO2 API Manager 1.6.0

3. ¿Qué es WSO2 API Manager?.

WSO2 API Manager es una pieza de software 100% Open Source cuyo objetivo es crear, publicar y gestionar todos los aspectos relativos a APIs (interfaces de acceso a nuestros recursos o servicios) y su ciclo de vida.

Gracias a WSO2 API Manager podemos **monetizar** nuestras APIs, **securizarlas**, gestionar la **disponibilidad** de

Catálogo de servicios Autentia



Síguenos a través de:

Últimas Noticias

» Screencasts de programación narrados en Español

» Sorteo de entradas para APIdays Mediterranea

» Concurso del Día de la Madre:

» Aprende gratis ReactiveCocoa

» Checklist de Scrum de Autentia

Histórico de noticias

Últimos Tutoriales

» Introducción a Groovy y Grails con Maven: el patrón CRUD

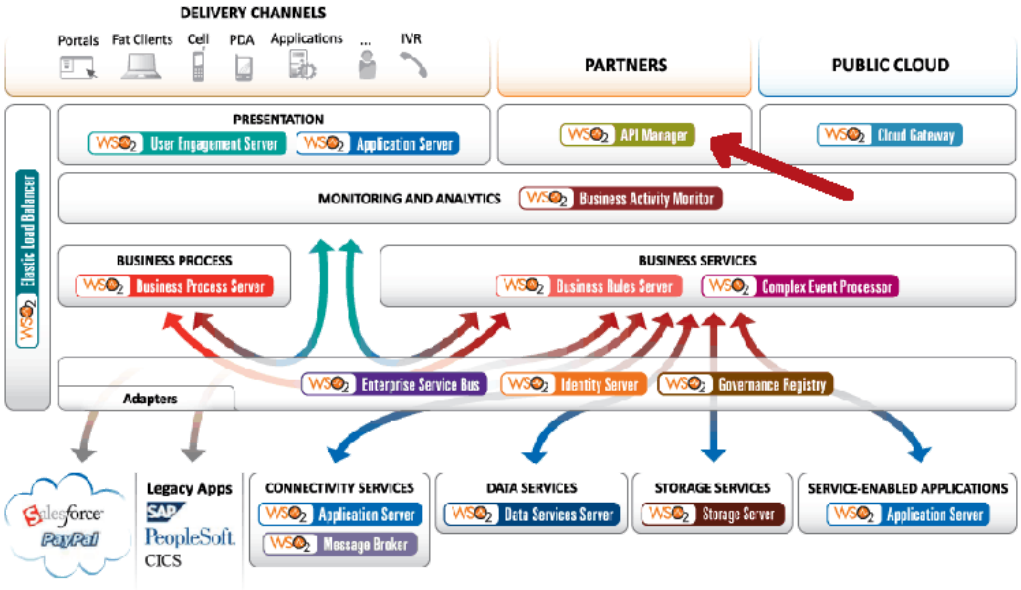
» Menu.bat Una forma cómoda de ejecutar comandos y aplicaciones, por ejemplo para Maven

» Testing de Hadoop con MRUnit

» Dart, el lenguaje de programación web del futuro creado por Google

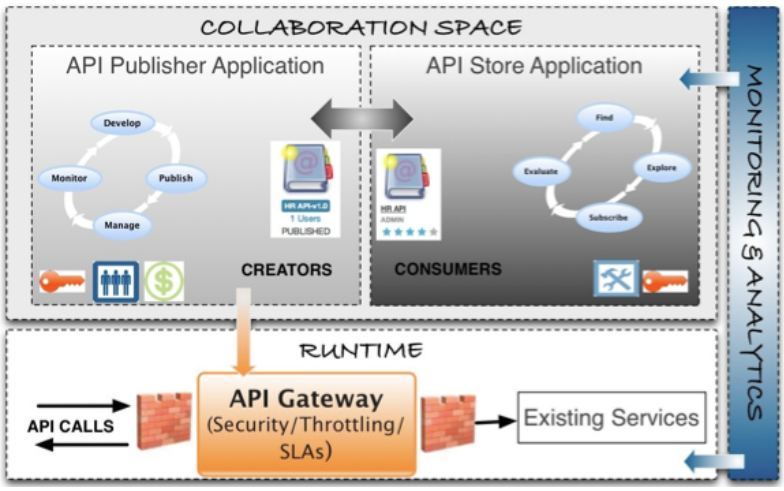
» Cómo integrar en Gradle un servidor Jetty o Tomcat

nuestros servicios de backend, **versionado**, **monitorizar** el tráfico, conocer **quién está consumiendo** nuestros servicios y mucho mucho más... Todo esto es posible debido a que WSO2 API Manager hace uso de otros componentes de la plataforma WSO2 tales como WSO2 Identity Server, WSO2 ESB o WSO2 Gobernance Registry.



La arquitectura de WSO2 API Manager está basada en 5 componentes:

- **API Publisher**: Aplicación web utilizada para publicar APIs, gestionarlas y documentarlas.
- **API Store**: Aplicación web utilizada como catálogo de APIs que permite la suscripción a las mismas.
- **API Gateway**: Esta pieza está basada en el ESB de WSO2. Actúa como un proxy que intercepta las peticiones entrantes y aplica políticas de seguridad y disponibilidad.
- **API Handlers**: Componentes utilizados por API Gateway para añadir determinados comportamientos ante los mensajes entrantes como pueden ser: monitorización, autenticación, publicación en Google Analytics, etc...
- **API Key Manager**: Componente que gestiona la seguridad y tokens de acceso. Está basado en el protocolo OAuth 2.0.



4. ¿Qué vamos a hacer?.

Supongamos que tenemos una instancia de un servicio REST desplegado en un servidor de aplicaciones. Además, nuestro servicio recibe peticiones HTTP sin ningún tipo de credenciales de autenticación. Partiendo de este escenario, lo que haremos será utilizar WSO2 API Manager para:

- **Balancear** la carga entre **dos instancias** de dicho servicio REST.
- **Securizar** las peticiones mediante HTTPS + credenciales de acceso.
- Controlar el flujo de peticiones entrantes por **consumidor**.

Partimos de algo como esto:

Últimos Tutoriales del Autor

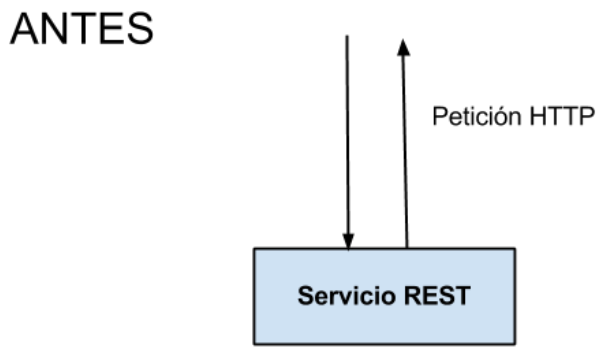
» SOA vs. SOAP y REST

» REST, el principio HATEOAS y Spring HATEOAS

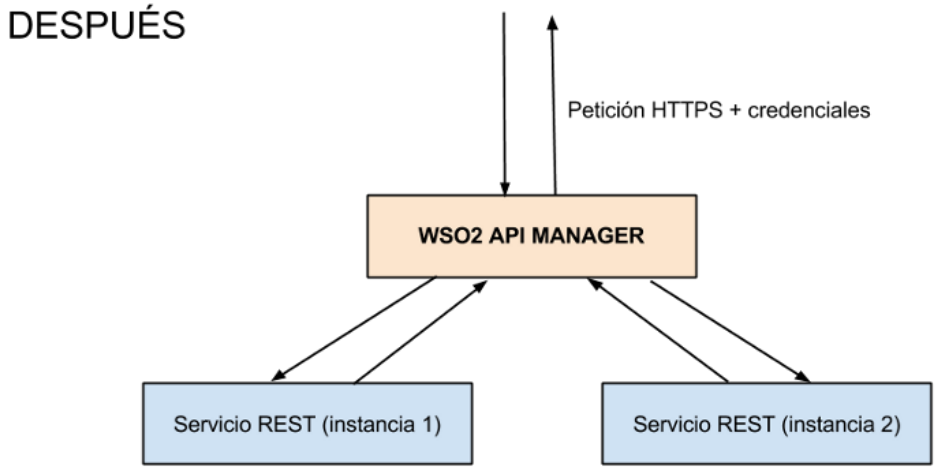
» SOA y los tipos de servicios

» Introducción a Spring Batch

» Iconos increíbles para nuestra web con Font Awesome Icons



Y lo que intentaremos obtener será algo como esto.



5. Nuestro servicio.

El API del servicio REST que publicaremos a través de WSO2 API Manager tendrá como objetivo el fútbol :). Estará compuesta de tres recursos distintos: equipos, estadios y jugadores. Para más información sobre los detalles del API consultar este otro tutorial.

Nuestro servicio está preparado para recibir y devolver datos tanto en representación XML como JSON. Está compuesto por tres recursos y sería algo como:

	Resource	Descripción
GET	/teams	Devuelve el listado de equipos
POST	/teams	Da de alta un equipo. Es necesario enviar en el cuerpo de la petición los siguientes campos: *name*, *foundationYear* y *rankingPosition*.
GET	/teams/:id	Devuelve el equipo cuyo identificador coincida con *:id*
GET	/teams/:id/stadium	Devuelve el estadio del equipo cuyo identificador coincida con *:id*
POST	/teams/:id/stadium	Da de alta el estadio del equipo cuyo identificador coincida con *:id*. Es necesario enviar en el cuerpo de la petición los siguientes campos: *capacity*, *name* y *city*.
GET	/teams/:id/players	Devuelve los jugadores del equipo cuyo identificador coincida con *:id*
POST	/teams/:id/players	Da de alta un jugador perteneciente al equipo cuyo identificador coincida con *:id*. Es necesario enviar en el cuerpo de la petición los siguientes campos: *name*, *goals*, *country* y *age*.
GET	/teams/:id/players/:id_player	Devuelve los datos del jugador que juegue en un equipo cuyo identificador coincida con *:id* y con un identificador de jugador que coincida con *:id_player*.

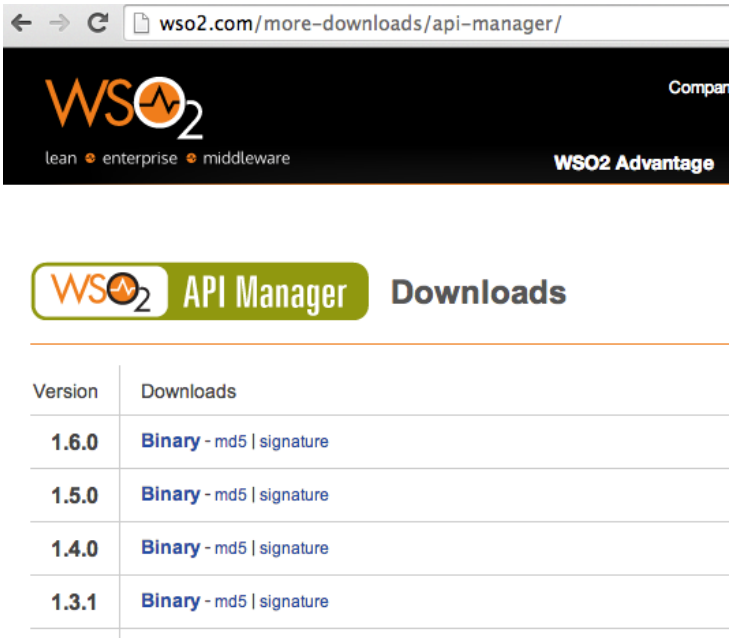
6. Exponiendo y securizando el servicio en WSO2 API Manager.

Una vez tenemos claro el escenario, procederemos con los pasos necesarios para poder publicar, securizar y consumir nuestro servicio en WSO2 API Manager.

6.1 Instalando WSO2 API Manager.

La versión que hemos elegido de WSO2 API Manager es la 1.6.0. Nótese que a fecha de publicación de este tutorial ya está disponible la 1.7.0, pero la hemos descartado por la gran cantidad de bug´s que contiene (algunos realmente preocupantes) y lo inestable de dicha versión. Una pena :(...

Para instalar WSO2 API Manager lo primero que haremos será descargarnos el producto directamente desde la página. Requiere un registro previo.



Una vez descargado, tendremos un fichero del tipo **wso2am-X.X.X.zip**, donde X.X.X será la versión del producto, en nuestro caso 1.6.0. Descomprimimos el empaquetado en el directorio deseado y listo.

Lo siguiente que haremos será asegurarnos de que tenemos la **variable de entorno JAVA_HOME** correctamente configurada apuntando a nuestra JDK. Luego nos dirigimos a nuestro directorio **${WSO2_API_MANAGER_HOME}/bin/** y ejecutamos el comando (UNIX):

```
./wso2server.sh start
```

Los ficheros de log los encontramos en el directorio **${WSO2_API_MANAGER_HOME}/repository/logs/**. Si todo salió correctamente, en el fichero ${WSO2_API_MANAGER_HOME}/repository/logs/wso2carbon.log deberíamos ver algo como:

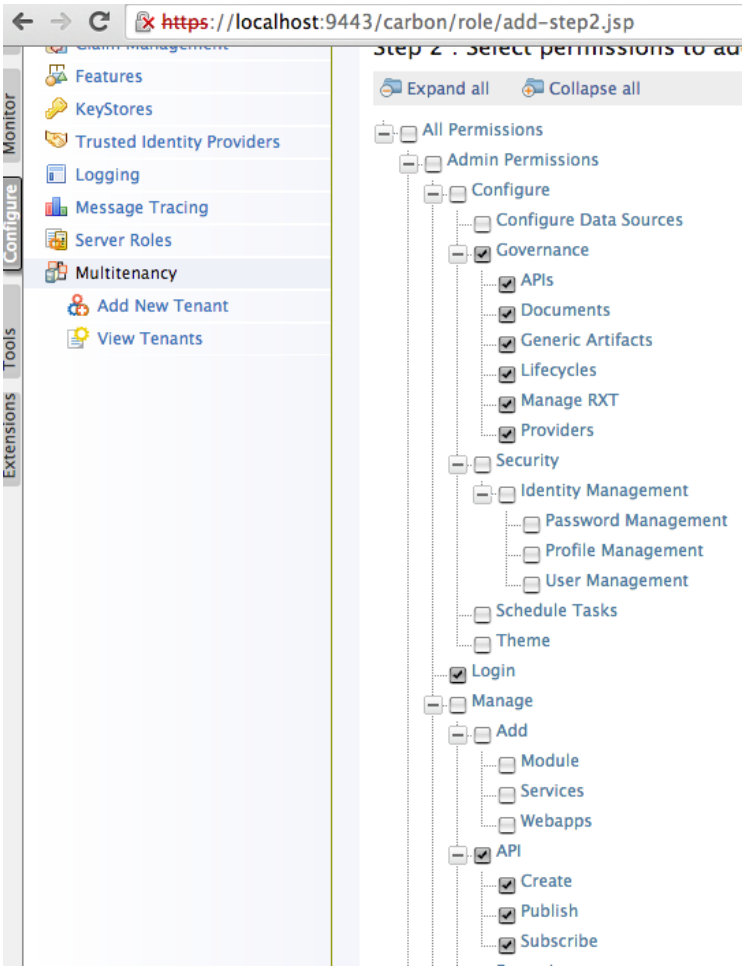
```
TID: [0] [AM] [2014-05-24 18:19:43,575]  INFO {org.wso2.carbon.ui.internal.CarbonUIServiceCom
ponent} -  Mgt Console URL  : https://192.168.1.38:9443/carbon/ {org.wso2.carbon.ui.internal.
CarbonUIServiceComponent}
TID: [0] [AM] [2014-05-24 18:19:43,575]  INFO {org.wso2.carbon.ui.internal.CarbonUIServiceCom
ponent} -  API Publisher Default Context : http://192.168.1.38:9763/publisher {org.wso2.carbo
n.ui.internal.CarbonUIServiceComponent}
TID: [0] [AM] [2014-05-24 18:19:43,575]  INFO {org.wso2.carbon.ui.internal.CarbonUIServiceCom
ponent} -  API Store Default Context : http://192.168.1.38:9763/store {org.wso2.carbon.ui.int
ernal.CarbonUIServiceComponent}
TID: [0] [AM] [2014-05-24 18:19:43,637]  INFO {org.wso2.carbon.apimgt.keymgt.internal.APIKeyM
gtServiceComponent} -  Setting default carbon host for thrift key management service: 192.168
.1.38 {org.wso2.carbon.apimgt.keymgt.internal.APIKeyMgtServiceComponent}
TID: [0] [AM] [2014-05-24 18:19:43,648]  INFO {org.wso2.carbon.apimgt.keymgt.internal.APIKeyM
gtServiceComponent} -  Started thrift key mgt service at port:10397 {org.wso2.carbon.apimgt.k
eymgt.internal.APIKeyMgtServiceComponent}
```

6.2 Creando usuarios para publicar y consumir API´s.

Una vez tenemos instalado y arrancado nuestro API Manager, lo primero que haremos será crear dos usuarios: uno para crear y publicar API´s y otro para consumirlas.

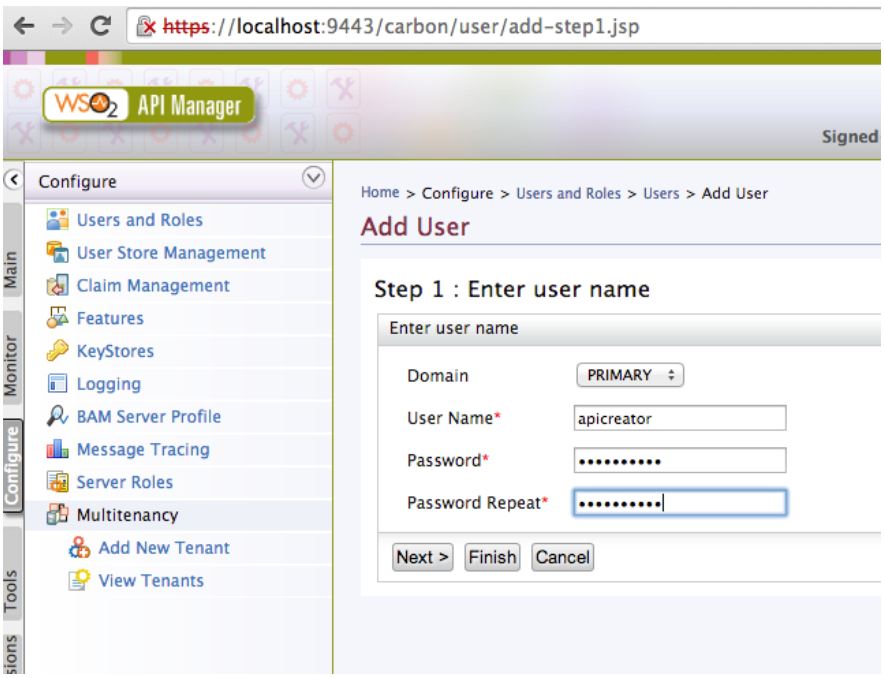
Para crear los usuarios accedemos a la consola de administración del API Manager en **https://<NOMBRE_MAQUINA>:9443/carbon** con usuario y clave admin (usuario administrador que ya viene creado por defecto).

Para crear el usuario con capacidad para crear y publicar API´s lo haremos en: **Configure > Users and Roles > Roles > Add New Role** y creamos un nuevo rol de usuario con nombre **creator**, pulsamos en el botón *Next* y seleccionamos los siguientes permisos: Governance (y todas sus opciones), Login y API (y todas sus opciones) tal y como se puede apreciar en la siguiente imagen (una vez hecho esto pulsamos Finish):



Ahora haremos lo mismo creando un rol de usuario **subscriber** seleccionando los permisos Login y Subscribe (dentro del grupo API). Pulsamos el botón Finish.

Una vez tenemos los roles para crear y suscribirse a las API´s, el siguiente paso es crear un usuario de cada tipo. Para ello, en la consola de aministración nos vamos a: **Configure > Users and Roles > Users > Add New User** y creamos dos usuarios: **apicreator** y **apisubscriber**. Pulsando sobre **Next** asociamos a cada usuario su rol correspondiente.



Y con esto ya tenemos todo lo necesario para empezar a publicar y suscribise a API´s.

6.3 Publicando una API.

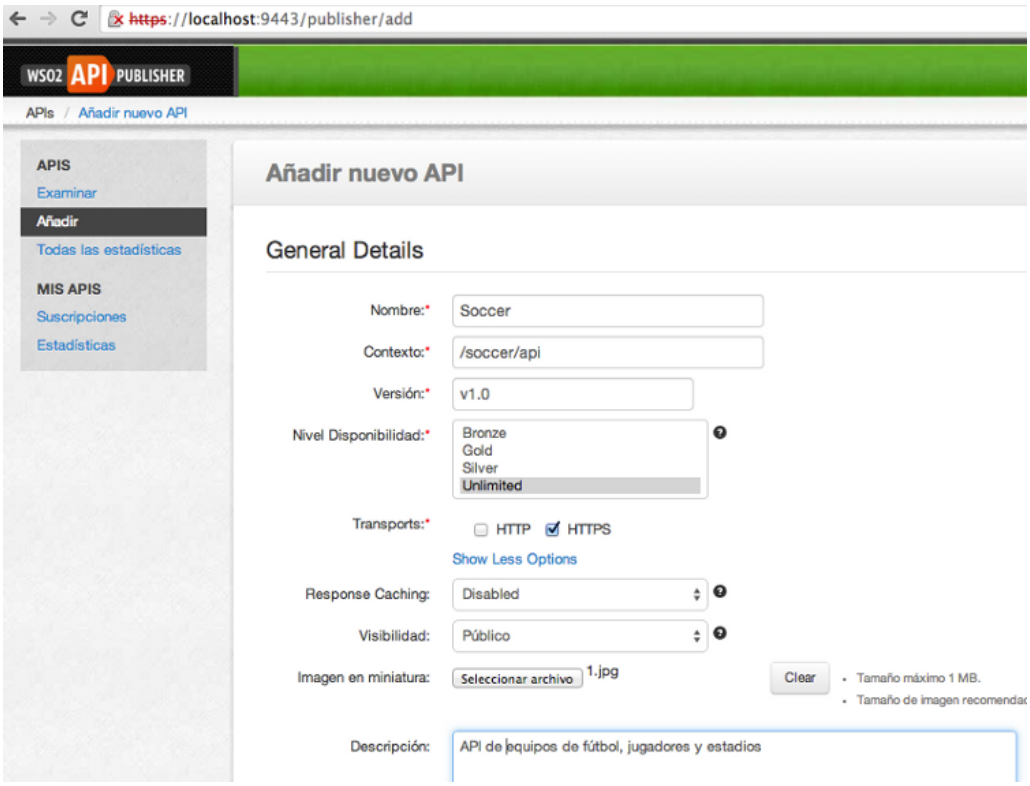
Lo siguiente que haremos será publicar nuestra API en WSO2 API Manager. Para ello accedemos a la herramienta **API Publisher** en **https://<nombre_maquina>:9443/publisher/** y nos autenticamos con el usuario que creamos anteriormente para tal fin (apicreator).

Una vez hemos accedido, pulsamos sobre "Añadir" y comenzará el proceso de registro de una API que estará compuesto de un formulario que debemos rellenar.

Publicando una API: Los datos generales

La primera sección del formulario de alta de una nueva API contiene campos relativos a la información general de la misma. Dichos campos son:

- **Nombre**: con el que identificaremos nuestra API en las diferentes herramientas como API store o la consola de administración.
- **Contexto**: de la URI donde será expuesta nuestra API. Ej: https://<MAQUINA_AM>:<PUERTO_AM>**/miapi/**
- **Versión**: que daremos a nuestra API. Este es un campo crucial para la gestión del ciclo de vida. Además, el contexto base de la URL de nuestra API estará compuesto por **contexto / versión**, algo del tipo https://maquina:puerto/contexto/version/loquesea. Nótese que añadir la versión como parte del contexto de la URL de un API está considerado muy buena práctica a la hora de gestionarlas con el fin de evitar conflictos en los consumidores y poder administrar correctamente su ciclo de vida.
- **Nivel de disponibilidad**: Este es un campo muy interesante. Su propósito es limitar el número de peticiones por consumidor que recibe nuestra API en un periodo de tiempo. De esta forma podemos evitar ataques, monetizar nuesta API en función de los diferentes niveles de disponibilidad (suscripción normal, premium, etc...) o regular el tráfico en función de nuestra infraestructura. Por defecto WSO2 API Manager viene con 4 niveles de disponibilidad predeterminados (aunque se pueden añadir más):
  - **Bronze:** Permite 1 petición por minuto.
  - **Silver:** Permite 5 peticiones por minuto.
  - **Gold:** Permite 20 peticiones por minuto.
  - **Unlimited:** No hay límite de peticiones por periodo de tiempo.
- **Transports**: Capa de transporte sobre la que publicaremos nuestra API. Puede ser HTTP, HTTPS o ambas.
- **Response Caching**: WSO2 API Manager viene con un sistema de caché de forma que es capaz de devolver la misma respuesta para la misma petición sin necesidad de enviar la petición al servicio de backend.
- **Visibilidad**: El propósito de este campo es poder decidir si nuestra API es visible desde nuestro **API store** a: cualquier usuario (incluyendo usuarios anónimos) o sólo a usuarios de un rol (o roles) concreto.
- **Imagen**: que daremos a nuestra API. Algo así como los iconos de las aplicaciones en Google Play o Apple Store.
- **Descripción**: texto descriptivo de nuestra API.
- **Etiquetas**: para categorizar nuestra API.
- **Sequences**: uno de los componentes de WSO2 API Manager es Apache Synapse. Mediante este campo podemos personalizar el flujo del mensaje entrante en el API Manager mediante una secuencia de Synapse.

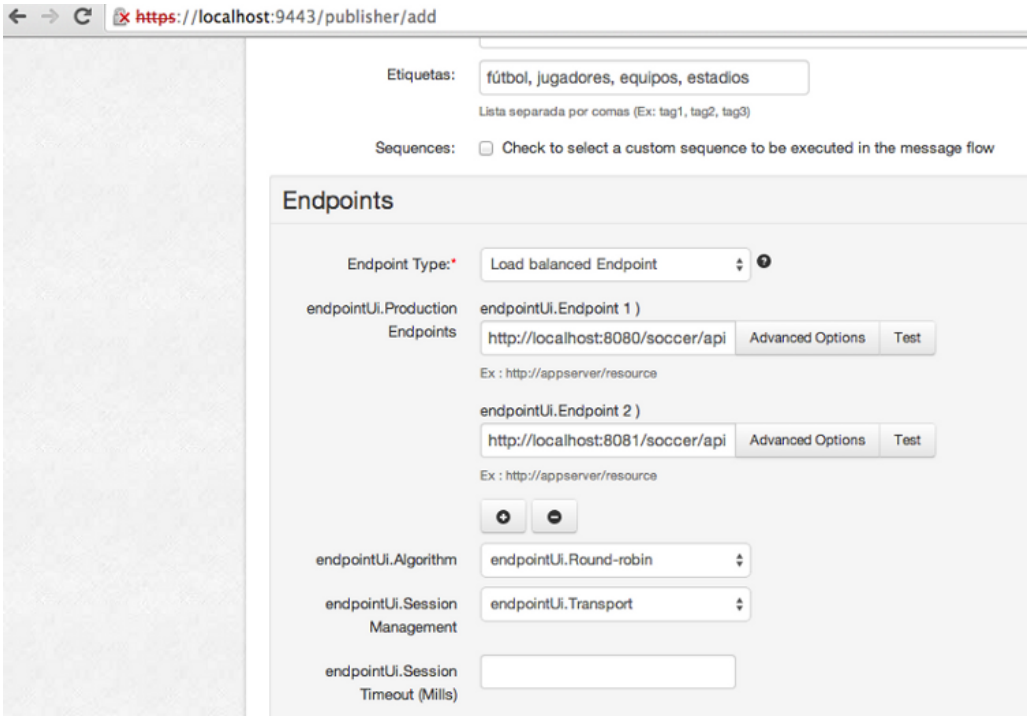


Publicando un API: Enpoints

La siguiente sección del formulario hace referencia al enpoint (uno o varios) al que "atacará" WSO2 API Manager. Será el destino del mensaje entrante. En nuestro caso será nuestra API REST de fútbol que tendremos desplegada en 2 instancias de Apache Tomcat (escuchando en el puerto 8080 y 8081 respectivamente).

- **HTTP Enpoint**: Tipo de enpoint utilizado cuando nuestro destino es una URI con protocolo HTTP.
- **URL Enpoint**: Muy similar al anterior aunque algo más genérico ya que el protocolo de nuestra URI no tiene por qué ser HTTP.
- **WSDL Enpoint**: Cuando la definición de nuestro endpoint está basada en un documento WSDL. Especificamos el servicio, el puerto y nuestro WSDL.
- **Failover Enpoint**: Ideal para definir una lista de enpoints y asegurar la alta disponibilidad. Cuando falle el envío de un mensaje a al primer enpoint, automáticamente lo intentará con el segundo, luego con el tercero, etc...
- **Load balanced Enpoint**: Ideal para distribuir la carga y asegurar la alta disponibilidad. Definimos una lista de enpoints y las peticiones que entren por nuestro API Manager se irá distribuyendo entre cada uno de ellos. Además, si alguno de los servicios que responden en cualquiera de los enpoints deja de dar servicio, la carga se distribuirá entre el resto de los enpoints activos. Este será el **tipo de enpoint que elegiremos en nuestro ejemplo** (tenemos el nuestro soccer API en dos instancias de Apache Tomcat: puertos 8080 y 8081).

Para cualquier tipo de enpoint de los descritos anteriormente podemos definir tanto el destino de entorno pruebas como el de producción (aunque probablemente no sea lo más recomendable para no ver comprometido el rendimiento de nuestro API Manager).

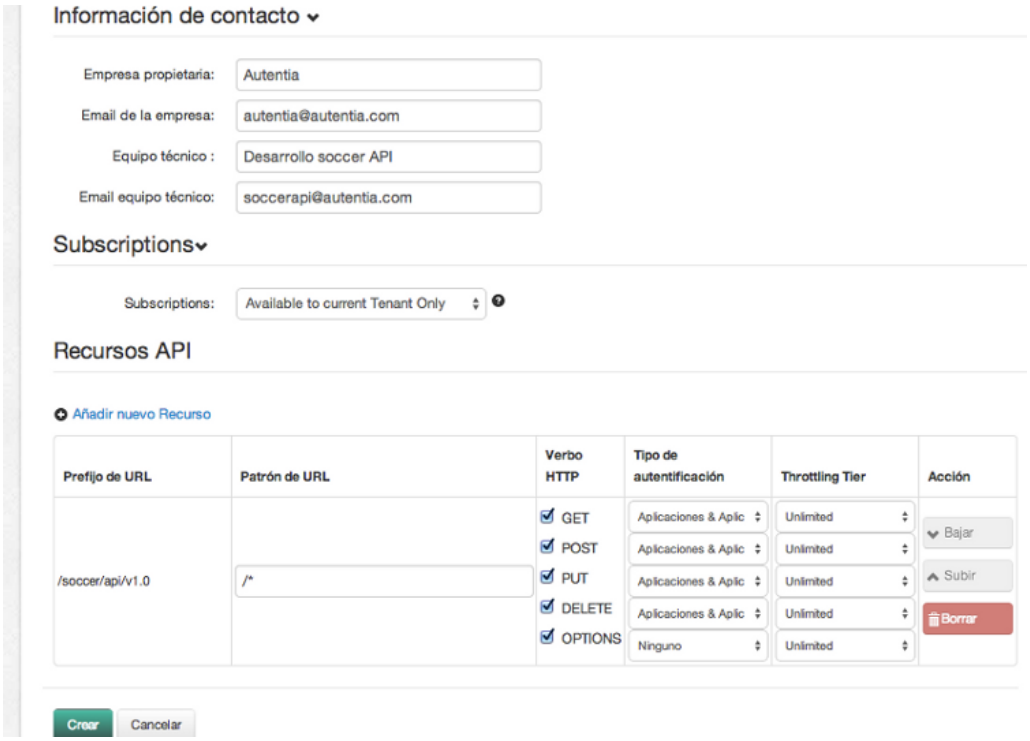


Publicando una API: Contacto, Subscripciones y Recursos

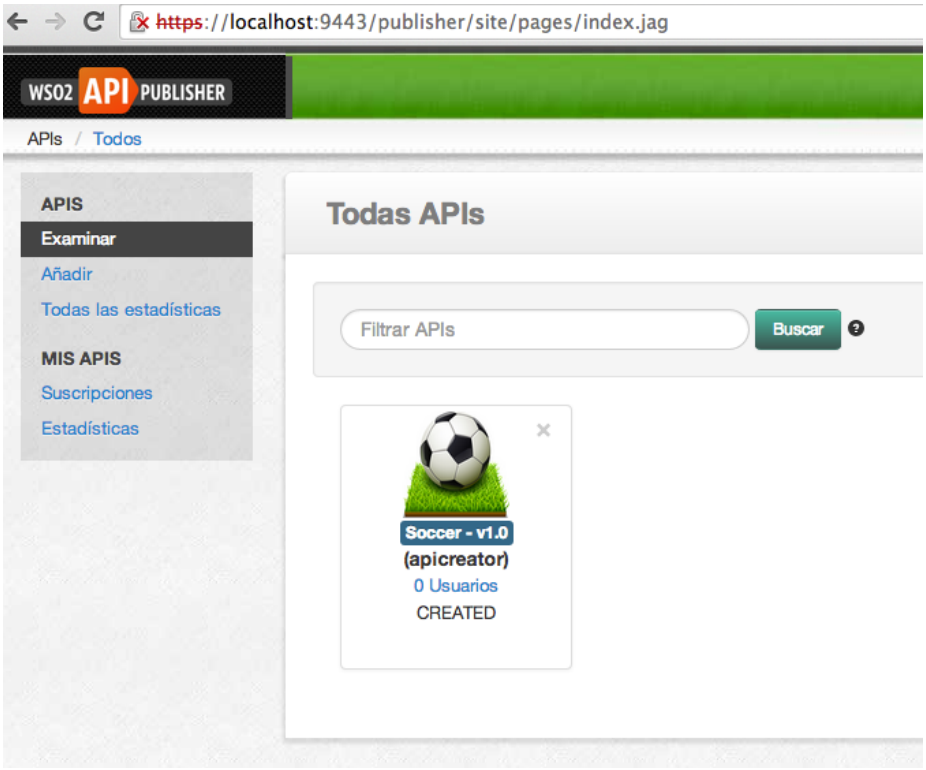
La sección de **contacto** contiene información relativa al propietario o responsable del servicio que publicamos en nuestro API Manager. La plataforma WSO2 cuenta con otro producto llamado **WSO2 Gobernance Registry** que permite completar muy bien toda esta información relativa al propietario del producto.

La sección de **suscripciones** nos permite configurar qué tipos de usuarios pueden suscribirse (y por tanto consumir) nuestro API. No debemos confundir este campo con el de visibilidad que vimos anteriormente. Una cosa es poder ver la información de nuestro API y otra es consumirla a través de una suscripción. Lo veremos mejor cuando veamos la herramienta **API store**.

Por último, la sección **Recursos** nos permite completar la información de nuestro API definiendo los diferentes recursos que la componen.



Cuando ya hemos rellenado los datos necesarios relativos a nuestra API, pulsamos sobre el botón "Crear" para crear nuestro API. OJO!!! con esto solo la creamos, no la publicamos (hasta que no se publique no se podrá consumir).



Publicando la API

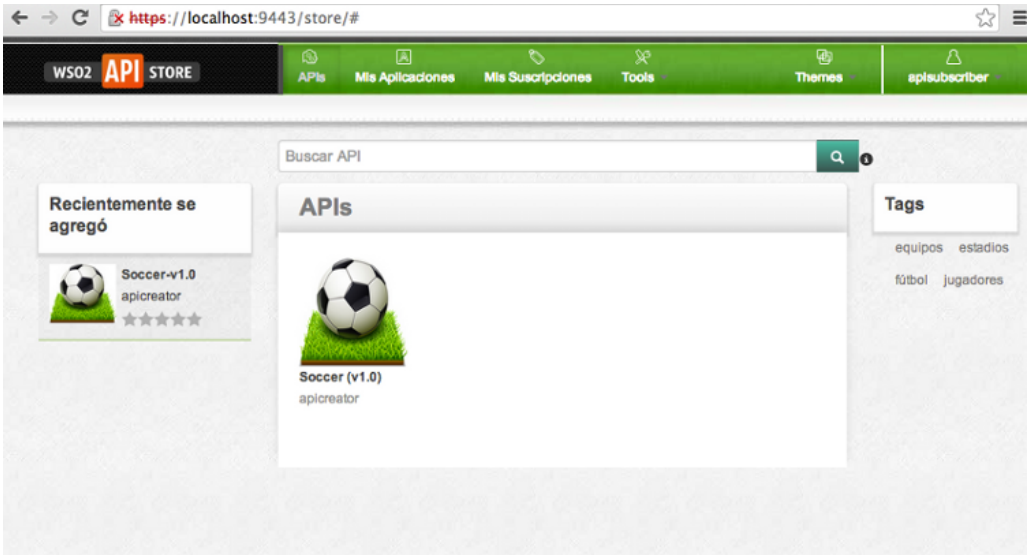
Para poder consumir nuestra API antes debemos publicarla. El proceso es muy sencillo. Pulsamos sobre nuestra API (que ya podremos ver con el icono que añadimos) y pulsamos sobre la pestaña **Ciclo de vida** de la parte superior. Seleccionamos la opción **PUBLICADO** y pulsamos sobre actualizar.

Y listo, ya tenemos nuestra API creada y publicada. Ahora veremos cómo nos podemos suscribir y consumirla a través de WSO2 API Manager y la aplicación "API store".

6.4 Suscribiéndonos a un API.

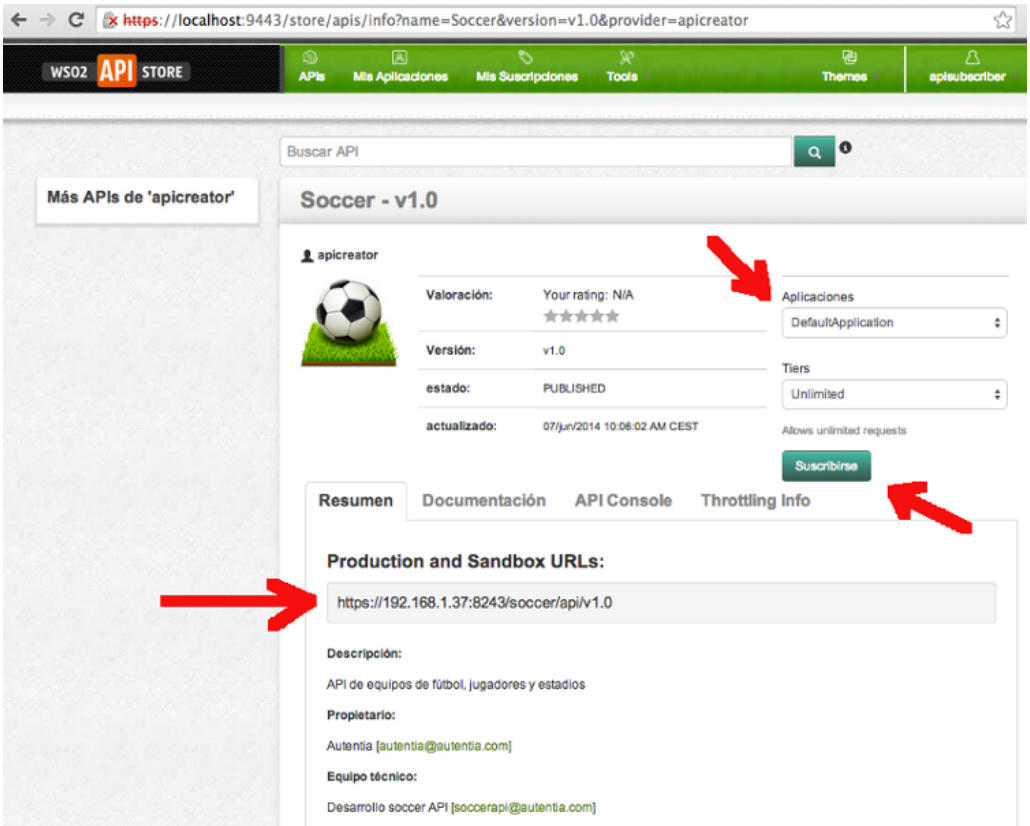
Lo siguiente que veremos será cómo podemos suscribirnos a una API. Para ello accedemos al "API store" en **https://<NOMBRE_MAQUINA>:9443/store** y nos autenticamos con el usuario **apisubscriber** que creamos anteriormente.

Una vez hecho esto, veremos una lista de APIs publicadas en el API Manager. En nuestro caso solo tendremos una. Pulsamos sobre el icono.

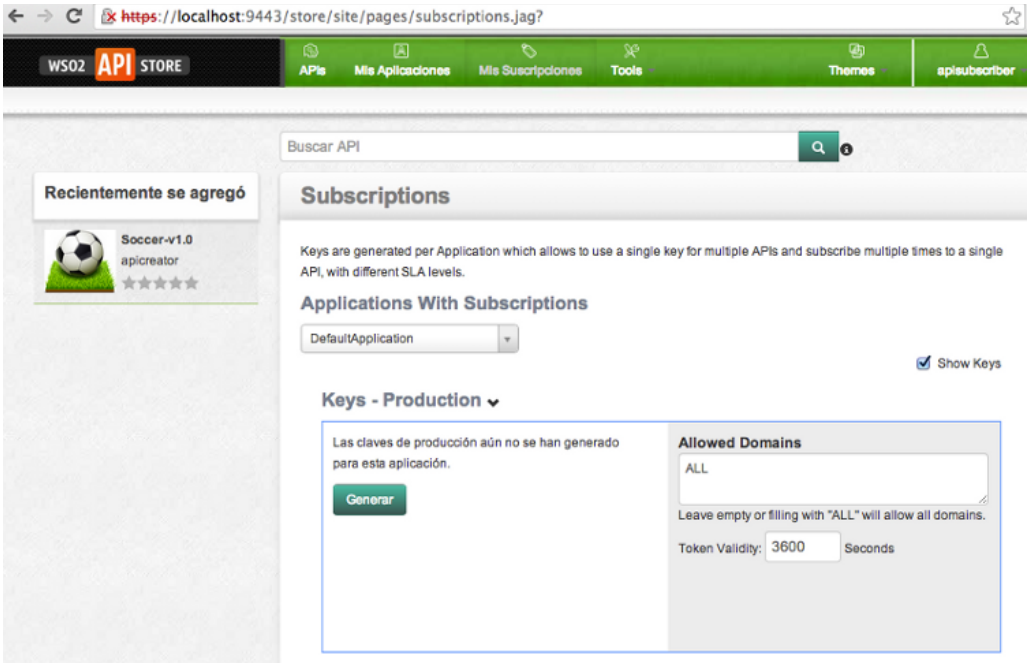


LLegados a este punto debemos comentar el **concepto de aplicación** en WSO2 API Manager. Un usuario puede tener una o varias aplicaciones. Por defecto tendremos una llamada **"DefaultApplication"**. Las **suscripciones a las APIs se realizan por usuario y aplicación**. De esta forma, a través de las credenciales propias del usuario/aplicación, podremos saber qué aplicaciones consumen nuestra API. Lo vemos a continuación.

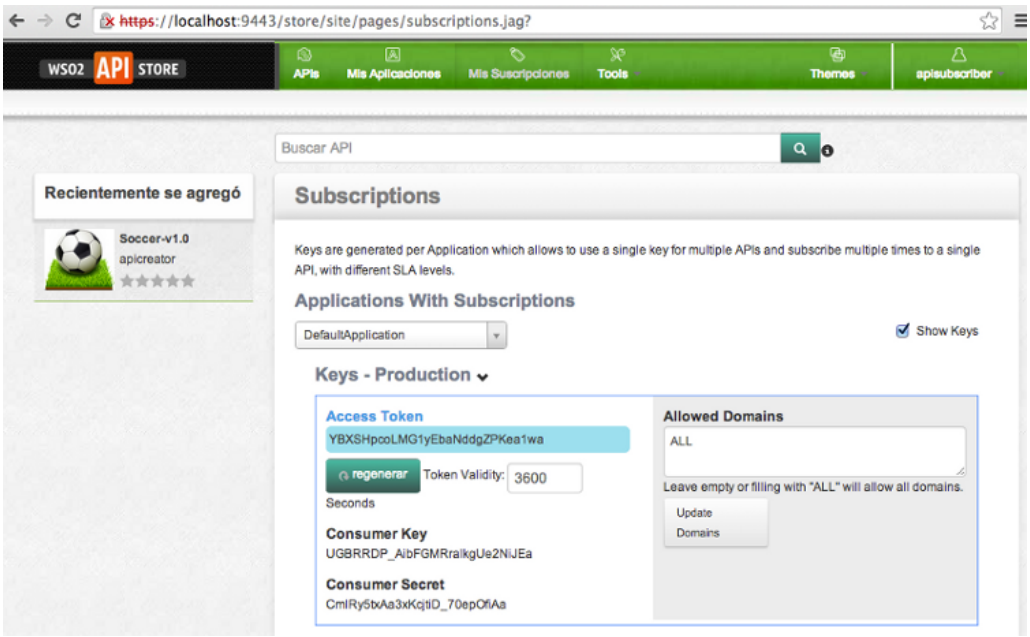
Apareceremos en la página del API a la que queremos suscribirnos donde podremos ver parte de la información de la misma como, por ejemplo, la URL de acceso. Para suscribirnos lo primero que debemos hacer es elegir una de las aplicaciones asociadas a nuestro usuario. Como no hemos creado ninguna aplicación adicional para nuestro usuario elegiremos "DefaultApplication". Por último pulsaremos sobre el botón **"Suscribirse"** y listo.



Por defecto, toda API publicada en **WSO2 API Manager require unas credenciales** para poder consumirla. Gracias a esas credenciales nuestra API estará securizada y WSO2 API Manager podrá controlar quién la está consumiendo. Para poder obtener dichas credenciales relativas a la aplicación "DefaultApplication" (con la que nos suscribimos), pulsamos sobre Mis Suscripciones (parte superior de la aplicación) y pulsamos sobre el botón **"Generar"**.

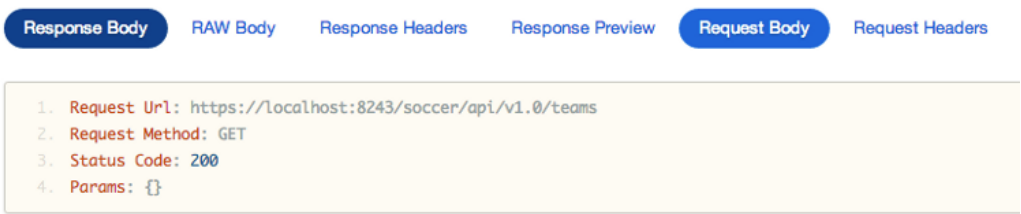


Una vez generadas, obtendremos una **clave de consumidor y una clave secreta** (Consumer Key y Consumer Secret: especificación Oauth2) además de un **token de acceso** al API. Dicho token de acceso tendrá un periodo de validez hasta que expire (podemos configurar el periodo de expiración) y podremos renovarlo (nuevo token de acceso) haciendo uso de nuestra clave de consumidor y secreta.



7. Consumiendo el API a través de WSO2 API Manager.

LLegados a este punto, ya estamos listos para poder consumir nuestra API a través de WSO2 API Manager. Para ello haremos una petición al **recurso teams**, que deberá devolvernos el listado de equipos. La URL de acceso podemos verla en la página propia del API dentro de la aplicación "API store". Hacemos una petición como la que se puede apreciar en la siguiente figura (nos ayudamos del complemento REST Console de Google Chrome):

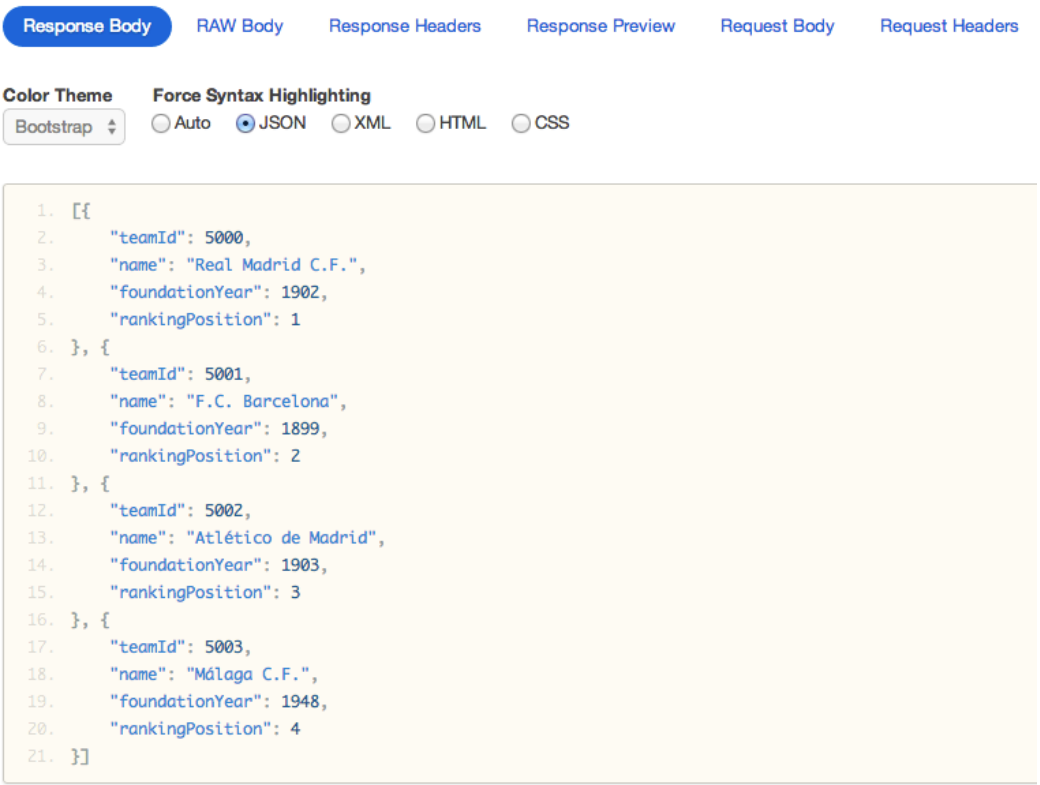


Sin embargo el resultado no es el esperado.

```
<ams:fault xmlns:ams="http://wso2.org/apimanager/security">
    <ams:code>900902</ams:code>
    <ams:message>Missing Credentials</ams:message>
    <ams:description>Required OAuth credentials not provided</ams:description>
</ams:fault>
```

¿Qué ha pasado?. Pues muy sencillo, que no hemos enviado las **credenciales de acceso al API** en nuestra petición (el token de acceso que vimos en el punto anterior). Incluimos el token de acceso en la cabecera de nuestra petición HTTP tal y como se puede apreciar en la siguiente figura:

Y ahora si, obtenemos el listado de equipos :). Además, debemos recordar que teníamos el nuestro "API backend" desplegado en dos instancias de Apache Tomcat que registramos en el API Manager de forma que los **mensajes entrantes se distribuyesen** entre las dos instancias. Si nos fijásemos en el log de cualquiera de las dos instancias comprobaríamos que el comportamiento es el esperado :D.



8. Referencias.

- WSO2 API Manager

9. Conclusiones.

En este tutorial hemos visto una pequeña parte de la funcionalidad que nos ofrece WSO2 API Manager, una herramienta muy interesante, cuyo "habitat natural" podría ser una DMZ de forma que pueda ser utilizado como canal de entrada a nuestos servicios backend con el objetivo de que éstos puedan estar securizados (si es que no lo estaban) y donde requiramos tener controlado el flujo de peticiones a los mismos.

Únicamente con esto podríamos estar hablando ya de un **API Gateway**, pero WSO2 API Manager ofrece muchísima más funcionalidad lo que lo convierte en un verdadero **API Manager** gracias a su catálogo de APIs, herramientas para creación y suscripción a las mismas, control del ciclo de vida, documentación, "enfoque social", etc, etc, etc...

Espero que este tutorial os haya sido de ayuda. Un saludo.

Miguel Arlandy

marlandy@autentia.com

Twitter: @m_arlandy

A continuación puedes evaluarlo:

Regístrate para evaluarlo

Por favor, vota +1 o compártelo si te pareció interesante

Share | +1 0

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

» Registrate y accede a esta y otras ventajas **«**

SOME RIGHTS RESERVED Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

IMPULSA Impulsores Comunidad ¿Ayuda?

---
sin clicks

0 personas han traído clicks a esta página

+ + + + + + + +

powered by karmacracy

Copyright 2003-2014 © All Rights Reserved | Texto legal y condiciones de uso | Banners | Powered by Autentia | Contacto

XHTML 1.0 CSS XML RSS XML ATOM