Poné un poco de swag a tu API

Una breve introducción a Swagger y Bravado

Swagger 2.0 (AKA OpenAPI Specification)

- Especificación formal para representar APIs ReST
- Agnóstico al lenguaje
- Legible por humanos y computadoras
- Ecosistema de herramientas

Ecosistema de herramientas para Swagger

- Swagger UI (Documentación)
- Swagger editor
  - Tricky con los errores!
  - local (http://swagger.io/swagger-editor/)
  - **Deshabilitar live rendering**
- Swagger Codegen: Clientes / servidores basados en spec (Bravado)

La especificación (AKA ‘spec’)

- Versión 2.0
- https://github.com/OAI/OpenAPI-Specification
- Usos
  - autodocumentación a partir del código → actualizada
    - Django Rest Swagger, Pyramid Swagger
  - Creación de la interfaz de la API antes de tener el código (contrato)

¡Vamos a la spec!

Parámetros

- Tipos: body, path, query, form
- Sólo UN BODY por acción → usamos un objeto

Responses

- Obligatoria al menos una
- Conviene especificar schema de respuesta (para Bravado)

Definitions

- Tipos de dato que consume/devuelve API
- No hace falta mapear todas las propiedades
- x-nullable
- Modelo de objetos para Bravado

Problemas

- Nullable
- No hay endpoints polimórficos
- No soporta envío de múltiples files en un form
- Archivo de spec muy grande
- Parámetros en body poco intuitivos
- Errores poco intuitivos

Bravado

- Biblioteca generadora de cliente Python para Swagger 2.0
- Cliente generado dinámicamente
- Validación automática de spec
- Validación automática de requests y responses contra spec
- Modelos Swagger como tipos de dato (mejor que interactuar con dicts)
- Permite spec local (archivo) o remoto (url)

¡Vamos al ejemplo!

¿Preguntas?

Acerca de mí

@cynpy

- Programadora
- Co-fundadora de LinuxChix Argentina
- Entusiasta de eventos técnicos