



ENTREGA 7

PROYECTO SEMESTRAL 1-2016

El proyecto de este semestre consistirá en un catálogo de productos online, de 2 familias de productos y promociones.

El proyecto se desarrollará en 5 entregas parciales más la entrega final:

~~E3. Viernes 8 de abril – Maquetas en HTML + CSS~~

~~E4. Martes 3 de mayo – Autenticación y manejo de usuarios~~

~~E5. Martes 17 de mayo – Catálogos y administrador~~

~~E6. Lunes 30 de mayo – Mejoras varias JS~~

E7. Viernes 17 de junio – Web Service RESTful

EF. Viernes 24 de junio – Mejoras varias y detalles

REPOSITORIO Y ENTREGAS

Esta entrega se realizará en los mismos grupos registrados desde la entrega 3, al igual que el resto del proyecto semestral. El grupo no podrá cambiar durante el transcurso del proyecto.

ENTREGA: REPOSITORIO Y SERVIDOR

Esta entrega se realizará igual que las entregas anteriores, mediante el repositorio privado registrado. Recordar que las entregas son evaluadas de acuerdo a **la última copia presente en el repositorio del grupo al momento de la entrega**.

También deben actualizar su aplicación **en producción (Heroku o equivalente)**.

En esta entrega la publicación de su aplicación es fundamental y la no publicación inhabilita la entrega (por lo que será calificada con la nota mínima 1.0).

ABANDONO DE UN GRUPO

Entrega 7

En caso de que **un alumno no continúe con el proyecto**, será evaluado con la nota mínima en todas las entregas siguientes (incluyendo la entrega y presentación final, reprobando automáticamente el curso de acuerdo a las reglas publicadas en el programa).

Para **el integrante restante**, que deberá continuar sólo con el proyecto, se aplicarán reglas especiales en cada entrega (aunque probablemente esto no evitará que la carga individual sea mayor).

Estos casos excepcionales deben ser avisados a la brevedad al profesor.

ENTREGA 7

En esta entrega deberán implementar una API Rest para la manipulación de productos.

PRODUCTO

Para efectos de esta API, un producto estará compuesto por:

- **id**: (String, fully qualified URL) Identificador único del producto (siempre debe ser una **fully qualified url**, por ejemplo, `http://miserver.example.com/api/v1/products/8a2f9da64`).
- **name**: (String) Nombre del producto.
- **description**: (String) Descripción del producto.
- **image**: (String, fully qualified URL) URL con la imagen del producto.
- **Price**: (Integer) Precio del producto en pesos chilenos.

API

LISTADO

El listado de productos debe responder por **GET** y tener la siguiente URL fija:

`http://dominio.example.com[/raiz_aplicacion]/api/v1/products`

Por ejemplo:

- <http://alguna-cosa.herokuapp.com/api/v1/products>
- <http://otra-cosa.herokuapp.com/myapp/api/v1/products>

El resultado debe ser una respuesta **200** y el cuerpo debe ser un json que contenga un arreglo de identificadores de productos (urls).

CREACIÓN

La creación se efectuará mediante el envío de un **PUT** que en el cuerpo contendrá un producto en formato json. Notar que este PUT se efectuará siempre a una URL compuesta por la misma URL de listado de productos + un hash de 32 caracteres único, por ejemplo:

<http://alguna-cosa.herokuapp.com/api/v1/products/e13f2f5fca2b61a446d3354df14962d6>

Nótese que esta URL es el id del producto.

Entrega 7

La respuesta, tras la creación, debe ser un **201** (Created). Recordar que un PUT se supone "idempotente", por lo que sucesivos requests PUT con el mismo contenido a la misma URL no debieran hacer diferencias en el resultado; sin embargo, requests posteriores pueden considerarse como una actualización del recurso y deben tratarse así (tanto en efecto como en respuesta); ver la sección de actualización más adelante.

OBTENCIÓN

Un producto puede obtenerse al efectuar un request **GET** a la url del producto (id). La respuesta debe ser un **200** OK con el producto en formato json. Además, un producto se considera cacheable (que se puede almacenar en Caché), por lo que es importante considerar los encabezados que informan sobre la vigencia de la respuesta.

Además, un producto debe poder ser consultado mediante un request **HEAD**, que simplemente informará con los encabezados apropiados sobre la vigencia del recurso.

ACTUALIZACIÓN

La actualización se realizará mediante 2 posibles métodos:

Si se usa el método PUT, el producto se actualiza en su totalidad, entregando una respuesta acorde (obviamente no debe ser un 201 created).

Si se usa el método PATCH, el producto sólo debe cambiar la parte especificada, dando una respuesta acorde.

Para todos los efectos, el campo id debe considerarse como inmutable (es decir, cualquier intento de cambiarlo es un error).

ELIMINACIÓN

La eliminación se realizará mediante un request DELETE a la url del producto. Notar que el método DELETE también es idempotente.

RESPUESTAS

Para la elaboración de las respuestas en cada caso considerar siempre:

- El código de respuesta (http status code).
- Encabezados de la respuesta.
- El cuerpo de la respuesta, si corresponde.

Por ejemplo, un GET no autenticado debiera responder distinto a un GET autenticado a la url de un producto inexistente (404, Not found) y distinto a un GET autenticado a una URL correcta.

AUTENTICACIÓN

La autenticación debe efectuarse mediante una implementación de HTTP Basic Access Authentication. Referencia: https://en.wikipedia.org/wiki/Basic_access_authentication.

Para efectos de evaluación, incluir la información de 2 usuarios y sus contraseñas en un archivo README en el home del proyecto (o directamente en la página inicial del repositorio).

Entrega 7

Nótese que la mayoría de servidores permiten configurar este acceso, pero para el caso de su aplicación, este acceso debe ser implementado en la aplicación contra usuarios en la base de datos (pueden ser los mismos usuarios regulares o usuarios específicamente creados para web services).

RESTRICCIONES

A partir de esta entrega pueden usar cualquier gema o librería.

PLAZO Y FORMA DE ENTREGA

El plazo máximo de entrega es el **viernes 17 de junio, a las 23:59 hrs.** La forma de entrega consistirá en:

- Tener la aplicación corriendo, disponible para pruebas automatizadas (no tenerla se evaluará con la nota mínima, 1.0).
- Entrega normal en el repositorio.

AYUDANTÍA DE AYUDA

El día **jueves 16 de junio** habrá una ayudantía para la resolución de dudas y apoyo de trabajo, en el módulo 3 (11:30 a 12:50) en la sala BC21 (horario y sala de clases).

EVALUACIÓN

En esta oportunidad la evaluación se efectuará mediante las siguientes pruebas:

PRUEBA 1: OBTENCIÓN

- Listado de productos.
- Obtención de un producto al azar del listado.
- Obtención del encabezado (HEAD).
- Obtención de un producto al azar (distinto).
- Obtención del encabezado (HEAD).

PRUEBA 2: CREACIÓN

- Listado de productos.
- Creación de un nuevo producto.
- Listado de productos (ahora debe aparecer el nuevo producto).
- Obtención del producto.
- Obtención del encabezado (HEAD).
- Actualización (PUT) sin cambios (es decir, mismo request de creación).
- Obtención del encabezado (HEAD) – se espera que no haya cambiado.

PRUEBA 3: ACTUALIZACIÓN 1

- Listado de productos.
- Obtención de un producto al azar del listado.
- Obtención del encabezado (HEAD).

Entrega 7

- Actualización (PATCH) de un campo del producto.
- Obtención del encabezado (HEAD) – se espera que haya cambiado.
- Obtención del producto y comprobación del cambio efectuado.

PRUEBA 4: ACTUALIZACIÓN 2

- Listado de productos.
- Obtención de un producto al azar del listado.
- Obtención del encabezado (HEAD).
- Actualización (PUT) de uno o más campos del producto.
- Obtención del encabezado (HEAD) – se espera que haya cambiado.
- Obtención del producto y comprobación del cambio efectuado.

PRUEBA 5: ELIMINACIÓN

- Listado de productos.
- Obtención de un producto al azar del listado.
- Eliminación del producto.
- Listado de productos (ya no debe aparecer el producto, pero si los demás K productos).
- Eliminación del mismo producto (chequeo de idempotencia).
- Listado de productos (aún no aparece el producto, pero si los mismos otros K productos).

PRUEBA 6: ERRORES

- Listado de productos, GET de cada producto y selección de uno al azar.
- Listado, creación, obtención (GET), obtención (HEAD), actualización (PATCH), eliminación (DELETE) sin autenticación – todos debieran fallar por autenticación, con respuesta y encabezado correspondiente.
- GET del producto usado para la prueba anterior – chequeo que nada haya cambiado.
- Listado, creación, ..., eliminación (DELETE) con autenticación, pero errónea.
- Obtención (GET), actualización (PATCH) y eliminación (DELETE) a urls inexistentes sin autenticación – debiera primar el error de autenticación.
- Intento de creación con producto incompleto.
- Intento de creación donde el id del producto no corresponde con la URL del PUT.
- Intento de actualización (PATCH) tratando de cambiar el id.
- Intento de actualización (PUT) con datos parciales.
- Intento de obtención de recurso inexistente.
- Intento de actualización de recurso inexistente.
- Intento de eliminación de recurso inexistente.
- Listado de producto y GET de cada producto – comprobación que nada ha cambiado desde el inicio.

Cada una de estas pruebas vale 10 puntos, con la siguiente escala:

| RESULTADO | PUNTAJE |
|---------------------------|-----------|
| Prueba exitosa | 10 puntos |
| Prueba falla en 1 paso | 8 puntos |
| Prueba falla en 2 pasos | 4 puntos |
| Prueba fallida (3+ pasos) | 0 puntos |