

Etapa 1 - Grupo 3

Integrantes: Minetto Fermín - Spinelli Sacha - Céspedes Brian

La primer etapa realizada incluye los servicios de stock de productos y empleados del supermercado, cuyas APIs permiten manipular los recursos a través de las operaciones CRUD. Para el desarrollo de ambos servicios se utilizó Django 2, un Framework Python elegido por conocimiento de la tecnología y por sus características de rapidez en el desarrollo, seguridad y escalabilidad.

Operaciones de acceso y manipulación de los recursos:

GET: para consultar información, POST para almacenar una nueva identidad, PUT y PATCH para actualizaciones de los recursos y DELETE para eliminarlos.

El parámetro ID de la URI es un valor numérico entero requerido el cual identifica un recurso único. Los parámetros para actualización y creación se envían en el cuerpo del requerimiento HTTP.

Employees:

GET : /api/employees/employees/
POST: /api/employees/employees/
GET: /api/employees/employees/{id}/
PUT: /api/employees/employees/{id}/
PATCH: /api/employees/employees/{id}/
DELETE: /api/employees/employees/{id}/

Employee types:

GET: /api/employees/employeetypes/
POST: /api/employees/employeetypes/
GET: /api/employees/employeetypes/{id}/
PUT: /api/employees/employeetypes/{id}/
PATCH: /api/employees/employeetypes/{id}/
DELETE: /api/employees/employeetypes/{id}/

Products:

GET: /api/products/products/
POST: /api/products/products/
GET: /api/products/products/{id}/
PUT: /api/products/products/{id}/
PATCH: /api/products/products/{id}/
DELETE: /api/products/products/{id}/

Product types:

GET: /api/products/producttypes/
POST: /api/products/producttypes/
GET: /api/products/producttypes/{id}/
PUT: /api/products/producttypes/{id}/
PATCH: /api/products/producttypes/{id}/
DELETE: /api/products/producttypes/{id}/