

Candidate: Quan Ha Le.

This is the final server

```
root@ip-172-31-95-82:~/mariadbapi
[root@ip-172-31-95-82 mariadbapi]#
[root@ip-172-31-95-82 mariadbapi]# php -S 0.0.0.0:8000 -t public
PHP 7.2.11 Development Server started at Tue Oct 30 20:21:59 2018
Listening on http://0.0.0.0:8000
Document root is /root/mariadbapi/public
Press Ctrl-C to quit.
[Tue Oct 30 20:22:09 2018] 76.69.93.134:58783 [200]: /product/all
[Tue Oct 30 20:22:09 2018] 76.69.93.134:58782 [404]: /favicon.ico - No such file
[Tue Oct 30 20:26:32 2018] 127.0.0.1:37240 [200]: /product/remove/6
[Tue Oct 30 20:26:39 2018] 127.0.0.1:37244 [200]: /product/remove/7
[Tue Oct 30 20:26:43 2018] 127.0.0.1:37248 [200]: /product/remove/8
[Tue Oct 30 20:26:46 2018] 127.0.0.1:37252 [200]: /product/remove/9
[Tue Oct 30 20:26:49 2018] 127.0.0.1:37256 [200]: /product/remove/10
[Tue Oct 30 20:26:52 2018] 127.0.0.1:37260 [200]: /product/remove/11
[Tue Oct 30 20:26:55 2018] 76.69.93.134:58818 [200]: /product/all
[Tue Oct 30 20:30:25 2018] 127.0.0.1:37266 [201]: /category/new
[Tue Oct 30 20:30:35 2018] 127.0.0.1:37270 [201]: /category/new
[Tue Oct 30 20:30:48 2018] 127.0.0.1:37274 [201]: /category/new
[Tue Oct 30 20:31:29 2018] 127.0.0.1:37278 [200]: /category/all
[Tue Oct 30 20:31:47 2018] 127.0.0.1:37282 [200]: /category/all
[Tue Oct 30 20:33:23 2018] 127.0.0.1:37286 [200]: /category/all
[Tue Oct 30 20:34:14 2018] 76.69.93.134:58869 [200]: /category/all
[Tue Oct 30 20:39:34 2018] 127.0.0.1:37292 [401]: /product/new
[Tue Oct 30 20:43:17 2018] 127.0.0.1:37294 [201]: /product/new
```

```
composer create-project symfony/skeleton mariadbapi
```

```
cd mariadbapi
```

```
composer require api
```

```
vi .env
```

```
-----
DATABASE_URL=mysql://root:admin2018@127.0.0.1:3306/eStore
-----
```

```
composer require symfony/orm-pack
```

```
composer require symfony/maker-bundle --dev
```

```
composer require symfony/form
```

```
composer require sensio/framework-extra-bundle
```

```
composer require symfony/monolog-bundle
```

```
composer require server
```

php bin/console make:entity Product

-----  
created: src/Entity/Product.php  
created: src/Repository/ProductRepository.php

Entity generated! Now let's add some fields!  
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):  
> name

Field type (enter ? to see all types) [string]:  
>

Field length [255]:  
> 30

Can this field be null in the database (nullable) (yes/no) [no]:  
>

updated: src/Entity/Product.php

Add another property? Enter the property name (or press <return> to stop adding fields):  
> category

Field type (enter ? to see all types) [string]:  
>

Field length [255]:  
> 20

Can this field be null in the database (nullable) (yes/no) [no]:  
>

updated: src/Entity/Product.php

Add another property? Enter the property name (or press <return> to stop adding fields):  
> sku

Field type (enter ? to see all types) [string]:  
>

Field length [255]:  
> 5

Can this field be null in the database (nullable) (yes/no) [no]:  
>

updated: src/Entity/Product.php

Add another property? Enter the property name (or press <return> to stop adding fields):

> price

Field type (enter ? to see all types) [string]:

> decimal

Precision (total number of digits stored: 100.00 would be 5) [10]:

> 8

Scale (number of decimals to store: 100.00 would be 2) [0]:

> 2

Can this field be null in the database (nullable) (yes/no) [no]:

>

updated: src/Entity/Product.php

Add another property? Enter the property name (or press <return> to stop adding fields):

> quantity

Field type (enter ? to see all types) [string]:

> integer

Can this field be null in the database (nullable) (yes/no) [no]:

>

updated: src/Entity/Product.php

Add another property? Enter the property name (or press <return> to stop adding fields):

> created\_at

Field type (enter ? to see all types) [datetime]:

>

Can this field be null in the database (nullable) (yes/no) [no]:

> yes

updated: src/Entity/Product.php

Add another property? Enter the property name (or press <return> to stop adding fields):

> modified\_at

Field type (enter ? to see all types) [datetime]:

>

Can this field be null in the database (nullable) (yes/no) [no]:

> yes

updated: src/Entity/Product.php

Add another property? Enter the property name (or press <return> to stop adding fields):

>

Success!

Next: When you're ready, create a migration with make:migration

-----

php bin/console make:entity Category

-----

created: src/Entity/Category.php

created: src/Repository/CategoryRepository.php

Entity generated! Now let's add some fields!

You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):

> name

Field type (enter ? to see all types) [string]:

>

Field length [255]:

> 20

Can this field be null in the database (nullable) (yes/no) [no]:

>

updated: src/Entity/Category.php

Add another property? Enter the property name (or press <return> to stop adding fields):

> created\_at

Field type (enter ? to see all types) [datetime]:

> yes

Can this field be null in the database (nullable) (yes/no) [no]:

>

updated: src/Entity/Category.php

Add another property? Enter the property name (or press <return> to stop adding fields):

> modified\_at

Field type (enter ? to see all types) [datetime]:

>

Can this field be null in the database (nullable) (yes/no) [no]:

> yes

updated: src/Entity/Category.php

Add another property? Enter the property name (or press <return> to stop adding fields):

>

Success!

Next: When you're ready, create a migration with make:migration

-----  
php bin/console make:entity User

created: src/Entity/User.php

created: src/Repository/UserRepository.php

Entity generated! Now let's add some fields!

You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):

> name

Field type (enter ? to see all types) [string]:

>

Field length [255]:

> 30

Can this field be null in the database (nullable) (yes/no) [no]:

>

updated: src/Entity/User.php

Add another property? Enter the property name (or press <return> to stop adding fields):  
> email

Field type (enter ? to see all types) [string]:  
>

Field length [255]:  
> 30

Can this field be null in the database (nullable) (yes/no) [no]:  
>

updated: src/Entity/User.php

Add another property? Enter the property name (or press <return> to stop adding fields):  
> created\_at

Field type (enter ? to see all types) [datetime]:  
> yes

Can this field be null in the database (nullable) (yes/no) [no]:  
> yes

updated: src/Entity/User.php

Add another property? Enter the property name (or press <return> to stop adding fields):  
> modified\_at

Field type (enter ? to see all types) [datetime]:  
>

Can this field be null in the database (nullable) (yes/no) [no]:  
> yes

updated: src/Entity/User.php

Add another property? Enter the property name (or press <return> to stop adding fields):  
>

Success!

Next: When you're ready, create a migration with make:migration

-----

```
vi src/Entity/Category.php
```

```
-----  
<?php
```

```
namespace App\Entity;
```

```
use ApiPlatform\Core\Annotation\ApiResource;  
use Doctrine\ORM\Mapping as ORM;
```

```
/**  
 * @ORM\Entity(repositoryClass="App\Repository\CategoryRepository")  
 * @ApiResource  
 */  
-----
```

```
vi src/Entity/Product.php
```

```
-----  
<?php
```

```
namespace App\Entity;
```

```
use ApiPlatform\Core\Annotation\ApiResource;  
use Doctrine\ORM\Mapping as ORM;
```

```
/**  
 * @ORM\Entity(repositoryClass="App\Repository\ProductRepository")  
 * @ApiResource  
 */  
-----
```

```
./bin/console doctrine:database:create
```

```
-----  
Created database `eStore` for connection named default  
-----
```

```
./bin/console doctrine:schema:update
```

```
./bin/console doctrine:schema:create
```

```
[root@ip-172-31-95-82 mariadbapi]# mysql --user=root --password=admin2018 eStore;  
MariaDB [eStore]> ALTER TABLE category MODIFY `created_at` timestamp NOT NULL DEFAULT  
current_timestamp();  
MariaDB [eStore]> ALTER TABLE product MODIFY `created_at` timestamp NOT NULL DEFAULT  
current_timestamp();  
MariaDB [eStore]> ALTER TABLE user MODIFY `created_at` timestamp NOT NULL DEFAULT  
current_timestamp();
```

```
MariaDB [eStore]> insert into user(name, email) values ("Bobby Fischer", "bobby@foo.com"), ("Betty Rubble", "betty@foo.com");
MariaDB [eStore]> commit;
```

```
[root@ip-172-31-95-82 mariadbapi]# php bin/console make:controller CategoryController
```

```
-----
created: src/Controller/CategoryController.php
created: templates/category/index.html.twig
```

```
Success!
-----
```

```
[root@ip-172-31-95-82 mariadbapi]# php bin/console make:controller ProductController
```

```
-----
created: src/Controller/ProductController.php
created: templates/product/index.html.twig
```

```
Success!
-----
```

```
[root@ip-172-31-95-82 mariadbapi]vi src/Controller/ApiController.php
```

```
-----
<?php
namespace App\Controller;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\HttpFoundation\Request;
class ApiController extends AbstractController
{
    /**
     * @var integer HTTP status code - 200 (OK) by default
     */
    protected $statusCode = 200;
    /**
     * Gets the value of statusCode.
     *
     * @return integer
     */
    public function getStatusCode()
    {
        return $this->statusCode;
    }
    /**
     * Sets the value of statusCode.
     */
}
```



```

* @param integer $statusCode the status code
*
* @return self
*/
protected function setStatusCode($statusCode)
{
    $this->statusCode = $statusCode;
    return $this;
}
/**
* Returns a JSON response
*
* @param array $data
* @param array $headers
*
* @return Symfony\Component\HttpFoundation\JsonResponse
*/
public function respond($data, $headers = [])
{
    return new JsonResponse($data, $this->getStatusCode(), $headers);
}
/**
* Sets an error message and returns a JSON response
*
* @param string $errors
*
* @return Symfony\Component\HttpFoundation\JsonResponse
*/
public function respondWithErrors($errors, $headers = [])
{
    $data = [
        'errors' => $errors,
    ];
    return new JsonResponse($data, $this->getStatusCode(), $headers);
}
/**
* Returns a 401 Unauthorized http response
*
* @param string $message
*
* @return Symfony\Component\HttpFoundation\JsonResponse
*/
public function respondUnauthorized($message = 'Not authorized!')
{
    return $this->setStatusCode(401)->respondWithErrors($message);
}
/**
* Returns a 201 Created

```

```

*
* @param array $data
*
* @return Symfony\Component\HttpFoundation\JsonResponse
*/
public function respondCreated($data = [])
{
    return $this->setStatusCode(201)->respond($data);
}
/**
* Returns a 404 Not Found
*
* @param string $message
*
* @return Symfony\Component\HttpFoundation\JsonResponse
*/
public function respondNotFound($message = 'Not found!')
{
    return $this->setStatusCode(404)->respondWithErrors($message);
}
protected function transformJsonBody(Request $request)
{
    $data = json_decode($request->getContent(), true);
    if (json_last_error() !== JSON_ERROR_NONE) {
        return null;
    }

    if ($data === null) {
        return $request;
    }
    $request->request->replace($data);
    return $request;
}
}

```

?>

[root@ip-172-31-95-82 mariadbapi]# vi src/Repository/CategoryRepository.php

<?php

namespace App\Repository;

use App\Entity\Category;

use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;

use Symfony\Bridge\Doctrine\RegistryInterface;

...

```
public function transform(Category $category)
{
    return [
        'id' => (int) $category->getId(),
        'name' => (string) $category->getName(),
        'created_at' => (string) ($category->getCreatedAt() ? $category->getCreatedAt()->format('Y-m-d
H:i:s') : NULL),
        'modified_at' => (string) ($category->getModifiedAt() ? $category->getModifiedAt()->format('Y-m-
d H:i:s') : NULL)
    ];
}
public function transformAll()
{
    $categories = $this->findAll();
    $categoriesArray = [];
    foreach ($categories as $category) {
        $categoriesArray[] = $this->transform($category);
    }
    return $categoriesArray;
}
```

-----  
[root@ip-172-31-95-82 mariadbapi]# vi src/Controller/CategoryController.php

```
-----
<?php
namespace App\Controller;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\HttpFoundation\Exception\HttpException;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
use Doctrine\ORM\EntityManagerInterface;
use App\Entity\Category;
use App\Repository\CategoryRepository;
use App\Form\CategoryType;
class CategoryController extends ApiController
{
    /**
     * @Route("/category/all")
     * @Method("GET")
     */
    public function index(CategoryRepository $categoryRepository)
    {
```

```

        $categories = $categoryRepository->transformAll();
        return $this->respond($categories);
    }
    /**
     * @Route("/category/new")
     * @Method("POST")
     */
    public function postCategoryAction(Request $request, CategoryRepository $categoryRepository,
    EntityManagerInterface $em)
    {
        $request = $this->transformJsonBody($request);
        if (!$request) {
            return $this->respondValidationError('Please provide a valid request!');
        }
        if (!$request->get('name')) {
            return $this->respondValidationError('Please provide a name!');
        }
        // persist the new category
        $category = new Category;
        $category->setName($request->get('name'));
        $em->persist($category);
        $em->flush();
        return $this->respondCreated($categoryRepository->transform($category));
    }
}
?>

```

[root@ip-172-31-95-82 mariadbapi]# vi src/Repository/ProductRepository.php

```

<?php

```

```

namespace App\Repository;

```

```

use App\Entity\Product;
use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;
use Symfony\Bridge\Doctrine\RegistryInterface;

```

```

...

```

```

public function transform(Product $product)
{
    return [
        'id' => (int) $product->getId(),
        'name' => (string) $product->getName(),
        'category' => (string) $product->getCategory(),
    ];
}

```

```

        'sku' => (string) $product->getSku(),
        'price' => (float) $product->getPrice(),
        'quantity' => (int) $product->getQuantity(),
        'created_at' => (string) ($product->getCreatedAt() ? $product->getCreatedAt()->format('Y-m-d
H:i:s') : NULL),
        'modified_at' => (string) ($product->getModifiedAt() ? $product->getModifiedAt()->format('Y-
m-d H:i:s') : NULL)
    ];
}
public function transformAll()
{
    $products = $this->findAll();
    $productsArray = [];
    foreach ($products as $product) {
        $productsArray[] = $this->transform($product);
    }
    return $productsArray;
}

```

-----

[root@ip-172-31-95-82 mariadbapi]# vi src/Controller/ProductController.php

```

<?php
namespace App\Controller;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\HttpFoundation\Exception\HttpException;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
use Doctrine\ORM\EntityManagerInterface;
use Doctrine\Common\Collections\ArrayCollection;
use Symfony\Component\DependencyInjection\Parameter;
use App\Entity\Product;
use App\Repository\ProductRepository;
use App\Form\ProductType;
use App\Entity\User;
use App\Repository\UserRepository;
class ProductController extends ApiController
{
    private function userAuthorization(Request $request, EntityManagerInterface $em)
    {
        if (!$request->get('username')) {
            return false;
        }
        if (!$request->get('password')) { // the password is here simply as the email

```

```

        return false;
    }
    $user = $request->get('username');
    $password = $request->get('password');
    $qb = $em->createQueryBuilder()
        ->select('u')
        ->from(User::class, 'u')
        ->where('u.name = :user_name AND u.email = :user_email')
        ->setParameters(array( 'user_name' => $user, 'user_email' => $password ));
    $qResult = $qb->getQuery()->execute();
    return ( count($qResult) > 0 );
}
/**
 * @Route("/product/all")
 * @Method("GET")
 */
public function index(ProductRepository $productRepository)
{
    $products = $productRepository->transformAll();
    return $this->respond($products);
}
/**
 * @Route("/product/entry/{id}")
 * @Method("GET")
 */
public function getProductAction(int $id, ProductRepository $productRepository): ?object
{
    if (!$id) {
        throw new HttpException(400, "Invalid id");
    }
    $product = $productRepository->find($id);
    if (!$product) {
        return $this->respondNotFound();
    }
    $product = $productRepository->transform($product);
    return $this->respond($product);
}
/**
 * @Route("/product/new")
 * @Method("POST")
 */
public function postProductAction(Request $request, ProductRepository $productRepository,
EntityManagerInterface $em)
{
    $request = $this->transformJsonBody($request);
    if (!$request) {
        return $this->respondValidationError('Please provide a valid request!');
    }
}

```

```

        if (! $this->userAuthorization($request, $sem) ) {
            return $this->respondUnauthorized();
        }
        if (! $request->get('name')) {
            return $this->respondValidationError('Please provide a name!');
        }
        if (! $request->get('category')) {
            return $this->respondValidationError('Please provide a category!');
        }
        if (! $request->get('sku')) {
            return $this->respondValidationError('Please provide a sku!');
        }
        if (! $request->get('price')) {
            return $this->respondValidationError('Please provide a price!');
        }
        if (! $request->get('quantity')) {
            return $this->respondValidationError('Please provide a quantity!');
        }
        // persist the new product
        $product = new Product;
        $product->setName($request->get('name'));
        $product->setCategory($request->get('category'));
        $product->setSku($request->get('sku'));
        $product->setPrice($request->get('price'));
        $product->setQuantity($request->get('quantity'));
        $sem->persist($product);
        $sem->flush();
        return $this->respondCreated($productRepository->transform($product));
    }
    /**
     * @Route("/product/update")
     * @Method("PUT")
     */
    public function updateProductAction(Request $request, ProductRepository $productRepository,
    EntityManagerInterface $sem)
    {
        $request = $this->transformJsonBody($request);
        if (! $request) {
            return $this->respondValidationError('Please provide a valid request!');
        }
        if (! $this->userAuthorization($request, $sem) ) {
            return $this->respondUnauthorized();
        }
        if (! $request->get('id')) {
            return $this->respondValidationError('Please provide an id!');
        }
        $id = $request->get('id');
        $product = $productRepository->find($id);
    }

```

```

        if (! $product) {
            return $this->respondNotFound();
        }

        if ($request->get('name')) {
            $product->setName($request->get('name'));
        }
        if ($request->get('category')) {
            $product->setCategory($request->get('category'));
        }
        if ($request->get('sku')) {
            $product->setSku($request->get('sku'));
        }
        if ($request->get('price')) {
            $product->setPrice($request->get('price'));
        }
        if ($request->get('quantity')) {
            $product->setQuantity($request->get('quantity'));
        }
        $product->setModifiedAt(new \DateTime());
        $sem->persist($product);
        $sem->flush();
        return $this->respond($product);
    }
    /**
     * @Route("/product/remove/{id}")
     * @Method("DELETE")
     */
    public function deleteProductAction(int $id, Request $request, ProductRepository
$productRepository, EntityManagerInterface $sem): ?object
    {
        $request = $this->transformJsonBody($request);
        if (! $request) {
            return $this->respondValidationError('Please provide a valid request!');
        }
        if (! $this->userAuthorization($request, $sem) ) {
            return $this->respondUnauthorized();
        }
        if (! $id) {
            throw new HttpException(400, "Invalid id");
        }
        $product = $productRepository->find($id);
        if (! $product) {
            return $this->respondNotFound();
        }
        $sem->remove($product);
        $sem->flush();
        return $this->respond([

```



```
        'deleted' => $product->getId()
    });
}
}
?>
```

---

```
[root@ip-172-31-95-82 mariadbapi]# php bin/console make:form Product
```

---

The name of Entity or fully qualified model class name that the new form will be bound to (empty for none):

> Product

created: src/Form/ProductType.php

Success!

Next: Add fields to your form and start using it.

Find the documentation at <https://symfony.com/doc/current/forms.html>

---

```
[root@ip-172-31-95-82 mariadbapi]# php bin/console make:form Category
```

In order to start server

```
[root@ip-172-31-95-82 mariadbapi]# ./bin/console server:start
```

```
[OK] Server listening on http://127.0.0.1:8000
```

or you can start all the allowed host to HTTP GET from Chrome browser,

```
php -S 0.0.0.0:8000 -t public
```

In order to stop the local server

```
./bin/console server:stop
```

Now you can try to insert into Category table by the API

```
curl -H 'content-type: application/json' -v -X POST -d '{"name": "Games"}'
```

```
http://127.0.0.1:8000/category/new
```

```
curl -H 'content-type: application/json' -v -X POST -d '{"name": "Computers"}'
```

```
http://127.0.0.1:8000/category/new
```

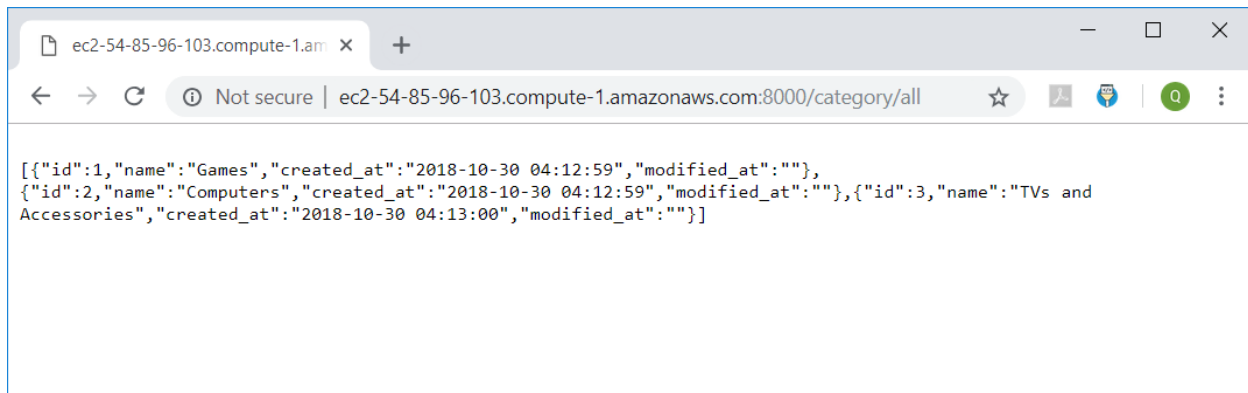
```
curl -H 'content-type: application/json' -v -X POST -d '{"name": "TVs and Accessories"}'  
http://127.0.0.1:8000/category/new
```

Now you can try on the ec2 instance to list all the category

```
curl -X GET --header 'Accept: application/ld+json' http://127.0.0.1:8000/category/all  
[root@ip-172-31-95-82 mariadbapi]# curl -X GET --header 'Accept: application/ld+json'  
http://127.0.0.1:8000/category/all
```

```
[{"id":1,"name":"Games","created_at":"2018-10-30  
04:12:59","modified_at":""}, {"id":2,"name":"Computers","created_at":"2018-10-30  
04:12:59","modified_at":""}, {"id":3,"name":"TVs and Accessories","created_at":"2018-10-30  
04:13:00","modified_at":""}]
```

```
[root@ip-172-31-95-82 mariadbapi]#
```



Now you can try to insert into Product table by the API, please add in user authorization, we are using user by user's name and the password is just the email:

```
curl -H 'content-type: application/json' -v -X POST -d '{"name": "Pong", "category": "Games", "sku":  
"A0001", "price": 69.99, "quantity": 20, "username": "Bobby Fischer", "password": "bobby@foo.com"}'  
http://127.0.0.1:8000/product/new  
curl -H 'content-type: application/json' -v -X POST -d '{"name": "GameStation 5", "category": "Games",  
"sku": "A0002", "price": 269.99, "quantity": 15, "username": "Bobby Fischer",  
"password": "bobby@foo.com"}' http://127.0.0.1:8000/product/new  
curl -H 'content-type: application/json' -v -X POST -d '{"name": "AP Oman PC - Aluminum", "category":  
"Computers", "sku": "A0003", "price": 1399.99, "quantity": 10, "username": "Bobby Fischer",  
"password": "bobby@foo.com"}' http://127.0.0.1:8000/product/new  
curl -H 'content-type: application/json' -v -X POST -d '{"name": "Fony UHD HDR 55\" 4k TV", "category":  
"TVs and Accessories", "sku": "A0004", "price": 1399.99, "quantity": 5, "username": "Bobby Fischer",  
"password": "bobby@foo.com"}' http://127.0.0.1:8000/product/new
```

These POST actions need user authorization as you have to add correct users and passwords within the sample data, otherwise authentication errors will pop up

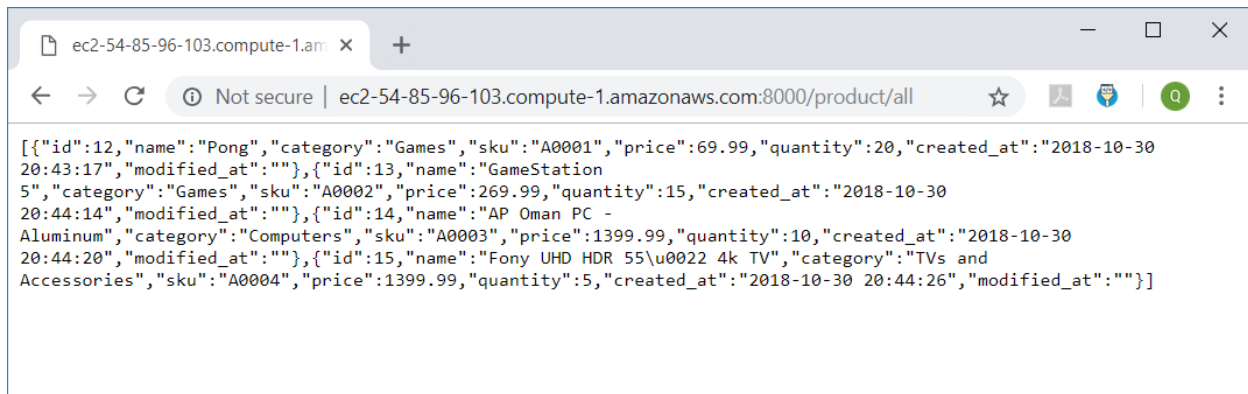
"username": "Bobby Fischer", "password": "bobby@foo.com" or "username": "Betty Rubble",  
"password": "betty@foo.com" (in this exercise, may I use emails for passwords)

```
[root@ip-172-31-95-82 mariadbapi]# curl -H 'content-type: application/json' -v -X POST -d '{"name":
"Pong", "category": "Games", "sku": "A0001", "price": 69.99, "quantity": 20, "username": "Bobby
Fischer", "password": "bobby@foo.com"}' http://127.0.0.1:8000/product/new
* About to connect() to 127.0.0.1 port 8000 (#0)
* Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 8000 (#0)
> POST /product/new HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 127.0.0.1:8000
> Accept: */*
> content-type: application/json
> Content-Length: 141
>
* upload completely sent off: 141 out of 141 bytes
< HTTP/1.1 201 Created
< Host: 127.0.0.1:8000
< Date: Tue, 30 Oct 2018 20:43:17 +0000
< Connection: close
< X-Powered-By: PHP/7.2.11
< Cache-Control: no-cache, private
< Date: Tue, 30 Oct 2018 20:43:17 GMT
< Content-Type: application/json
< Link: <http://127.0.0.1:8000/api/docs.jsonld>;
rel="http://www.w3.org/ns/hydra/core#apiDocumentation"
<
* Closing connection 0
{"id":12,"name":"Pong","category":"Games","sku":"A0001","price":69.99,"quantity":20,"created_at":"","
"modified_at":""}[
root@i[root@ip-172-31-95-82 mariadbapi]#
```

Or please open the ec2 domain name address on browser to list all the product  
curl -X GET --header 'Accept: application/ld+json' http://127.0.0.1:8000/product/all

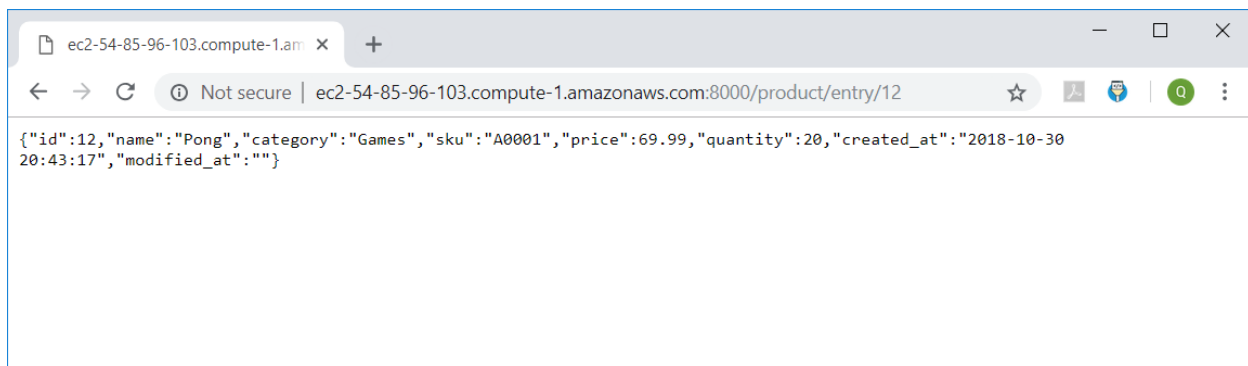
```
[root@ip-172-31-95-82 mariadbapi]# curl -X GET --header 'Accept: application/ld+json'
http://127.0.0.1:8000/product/all
```

```
[{"id":12,"name":"Pong","category":"Games","sku":"A0001","price":69.99,"quantity":20,"created_at":"2
018-10-30 20:43:17","modified_at":""},{
{"id":13,"name":"GameStation
5","category":"Games","sku":"A0002","price":269.99,"quantity":15,"created_at":"2018-10-30
20:44:14","modified_at":""},{
{"id":14,"name":"AP Oman PC -
Aluminum","category":"Computers","sku":"A0003","price":1399.99,"quantity":10,"created_at":"2018-
10-30 20:44:20","modified_at":""},{
{"id":15,"name":"Fony UHD HDR 55\u0022 4k TV","category":"TVs
and Accessories","sku":"A0004","price":1399.99,"quantity":5,"created_at":"2018-10-30
20:44:26","modified_at":""}]
```



In order to find a product with id = 12

```
curl -X GET --header 'Accept: application/ld+json' http://127.0.0.1:8000/product/entry/12
[root@ip-172-31-95-82 mariadbapi]# curl -X GET --header 'Accept: application/ld+json'
http://127.0.0.1:8000/product/entry/12
{"id":12,"name":"Pong","category":"Games","sku":"A0001","price":69.99,"quantity":20,"created_at":"2018-10-30 20:43:17","modified_at":""}
[root@ip-172-31-95-82 mariadbapi]#
```



Now we insert a test product in order to update it and then to delete it (with user and password authentication)

Insertion

```
[root@ip-172-31-95-82 mariadbapi]# curl -H 'content-type: application/json' -v -X POST -d '{"name": "Test Product", "category": "Games", "sku": "ATEST", "price": 69.99, "quantity": 20, "username": "Bobby Fischer", "password": "bobby@foo.com"}' http://127.0.0.1:8000/product/new
```

list them all by curl GET to see the new product id will be 16

```
[root@ip-172-31-95-82 mariadbapi]# curl -X GET --header 'Accept: application/ld+json'
http://127.0.0.1:8000/product/all
[{"id":12,"name":"Pong","category":"Games","sku":"A0001","price":69.99,"quantity":20,"created_at":"2018-10-30 20:43:17","modified_at":""}, {"id":13,"name":"GameStation 5","category":"Games","sku":"A0002","price":269.99,"quantity":15,"created_at":"2018-10-30 20:44:14","modified_at":""}, {"id":14,"name":"AP Oman PC -
```

```
Aluminum","category":"Computers","sku":"A0003","price":1399.99,"quantity":10,"created_at":"2018-10-30 20:44:20","modified_at":"","{"id":15,"name":"Fony UHD HDR 55\u0022 4k TV","category":"TVs and Accessories","sku":"A0004","price":1399.99,"quantity":5,"created_at":"2018-10-30 20:44:26","modified_at":"","{"id":16,"name":"Test Product","category":"Games","sku":"ATEST","price":69.99,"quantity":20,"created_at":"2018-10-30 20:50:34","modified_at":""}]
```

Update quantity to be 40, sku to be A9999, please remember to add username and password for security

```
[root@ip-172-31-95-82 mariadbapi]# curl -H 'content-type: application/json' -v -X PUT -d '{"id":16, "sku": "A9999", "quantity": 40, "username":"Bobby Fischer", "password":"bobby@foo.com"}' http://127.0.0.1:8000/product/update
```

then list all product to see the new quantity and the new sku, also please observe the new value for modified\_at datetime

```
[root@ip-172-31-95-82 mariadbapi]# curl -X GET --header 'Accept: application/ld+json' http://127.0.0.1:8000/product/all [{"id":12,"name":"Pong","category":"Games","sku":"A0001","price":69.99,"quantity":20,"created_at":"2018-10-30 20:43:17","modified_at":"","{"id":13,"name":"GameStation 5","category":"Games","sku":"A0002","price":269.99,"quantity":15,"created_at":"2018-10-30 20:44:14","modified_at":"","{"id":14,"name":"AP Oman PC - Aluminum","category":"Computers","sku":"A0003","price":1399.99,"quantity":10,"created_at":"2018-10-30 20:44:20","modified_at":"","{"id":15,"name":"Fony UHD HDR 55\u0022 4k TV","category":"TVs and Accessories","sku":"A0004","price":1399.99,"quantity":5,"created_at":"2018-10-30 20:44:26","modified_at":"","{"id":16,"name":"Test Product","category":"Games","sku":"A9999","price":69.99,"quantity":40,"created_at":"2018-10-30 20:50:34","modified_at":"2018-10-30 20:54:13"}]}
```

## Deletion

We can delete the tested product after we used, the test id=16 and also user security

```
[root@ip-172-31-95-82 mariadbapi]# curl -H 'content-type: application/json' -v -X DELETE -d '{"username":"Bobby Fischer", "password":"bobby@foo.com"}' http://127.0.0.1:8000/product/remove/16
```

Adding with the user security / authentication, the system also has logger

```
[root@ip-172-31-95-82 mariadbapi]# vi config/packages/prod/monolog.yaml
```

monolog:

handlers:

main:

type: fingers\_crossed

action\_level: error

handler: nested

excluded\_404s:

# regex: exclude all 404 errors from the logs

- ^/

nested:

```

    type: stream
    path: "%kernel.logs_dir%/%kernel.environment%.log"
    level: debug
console:
    type: console
    process_psr_3_messages: false
    channels: ["!event", "!doctrine"]
deprecation:
    type: stream
    path: "%kernel.logs_dir%/%kernel.environment%.deprecations.log"
deprecation_filter:
    type: filter
    handler: deprecation
    max_level: info
    channels: ["php"]

```

and here is the logs

```

[root@ip-172-31-95-82 mariadbapi]# ls -lai var/log/
total 40
5808750 drwxrwxrwx. 2 root root  21 Oct 30 20:21 .
9688875 drwxrwxrwx. 4 root root  30 Oct 28 20:52 ..
5931589 -rw-rw-rw-. 1 root root 40774 Oct 30 20:57 dev.log
[root@ip-172-31-95-82 mariadbapi]# vi var/log/dev.log
[2018-10-30 20:21:52] console.DEBUG: Command "server:stop" exited with code "1"
{"command":"server:stop","code":1} []
[2018-10-30 20:22:08] request.INFO: Matched route "app_product_index".
{"route":"app_product_index","route_parameters":{"_route":"app_product_index","_controller":"App\
\Controller\ProductController::index"},"request_uri":"http://ec2-54-85-96-103.compute-
1.amazonaws.com:8000/product/all","method":"GET"} []
[2018-10-30 20:22:08] security.INFO: Populated the TokenStorage with an anonymous Token. [] []
[2018-10-30 20:22:08] php.INFO: User Deprecated: The
"Sensio\Bundle\FrameworkExtraBundle\Configuration\Method" annotation is deprecated since version
5.2. Use "Symfony\Component\Routing\Annotation\Route" instead. {"exception":"[object]
(ErrorException(code: 0): User Deprecated: The
\"Sensio\\Bundle\\FrameworkExtraBundle\\Configuration\\Method\" annotation is deprecated since
version 5.2. Use \"Symfony\\Component\\Routing\\Annotation\\Route\" instead. at
/root/mariadbapi/vendor/sensio/framework-extra-bundle/Configuration/Method.php:14)} []
[2018-10-30 20:22:09] doctrine.DEBUG: SELECT t0.id AS id_1, t0.name AS name_2, t0.category AS
category_3, t0.sku AS sku_4, t0.price AS price_5, t0.quantity AS quantity_6, t0.created_at AS
created_at_7, t0.modified_at AS modified_at_8 FROM product t0 [] []
[2018-10-30 20:26:32] request.INFO: Matched route "app_product_deleteproduct".
{"route":"app_product_deleteproduct","route_parameters":{"_route":"app_product_deleteproduct","_
controller":"App\\Controller\\ProductController::deleteProductAction","id":"6"},"request_uri":"http://1
27.0.0.1:8000/product/remove/6","method":"DELETE"} []
[2018-10-30 20:26:32] security.INFO: Populated the TokenStorage with an anonymous Token. [] []
[2018-10-30 20:26:32] php.INFO: User Deprecated: The
"Sensio\Bundle\FrameworkExtraBundle\Configuration\Method" annotation is deprecated since version

```

5.2. Use "Symfony\Component\Routing\Annotation\Route" instead. {"exception":"[object] (ErrorException(code: 0): User Deprecated: The \"Sensio\\Bundle\\FrameworkExtraBundle\\Configuration\\Method\" annotation is deprecated since version 5.2. Use \"Symfony\\Component\\Routing\\Annotation\\Route\" instead. at /root/mariadbapi/vendor/sensio/framework-extra-bundle/Configuration/Method.php:14)} []

[2018-10-30 20:26:32] doctrine.DEBUG: SELECT u0\_.id AS id\_0, u0\_.name AS name\_1, u0\_.email AS email\_2, u0\_.created\_at AS created\_at\_3, u0\_.modified\_at AS modified\_at\_4 FROM user u0\_ WHERE u0\_.name = ? AND u0\_.email = ? [\"Bobby Fischer\", \"bobby@foo.com\"] []

[2018-10-30 20:26:32] doctrine.DEBUG: SELECT t0.id AS id\_1, t0.name AS name\_2, t0.category AS category\_3, t0.sku AS sku\_4, t0.price AS price\_5, t0.quantity AS quantity\_6, t0.created\_at AS created\_at\_7, t0.modified\_at AS modified\_at\_8 FROM product t0 WHERE t0.id = ? [6] []

[2018-10-30 20:26:32] doctrine.DEBUG: \"START TRANSACTION\" [] []

[2018-10-30 20:26:32] doctrine.DEBUG: DELETE FROM product WHERE id = ? [6] []

[2018-10-30 20:26:32] doctrine.DEBUG: \"COMMIT\" [] []

[2018-10-30 20:26:39] request.INFO: Matched route \"app\_product\_deleteproduct\". {\"route\":\"app\_product\_deleteproduct\", \"route\_parameters\":{\"\_route\":\"app\_product\_deleteproduct\", \"\_controller\":\"App\\Controller\\ProductController::deleteProductAction\", \"id\":\"7\"}, \"request\_uri\":\"http://127.0.0.1:8000/product/remove/7\", \"method\":\"DELETE\"} []

[2018-10-30 20:26:39] security.INFO: Populated the TokenStorage with an anonymous Token. [] []