

Knowledge_Learning - App's Documentation

Version	Date
1.0.0	2024/12/12

[TOC]

Prerequisites

Knowledge_learning's app is based on these technical specifications :

- PHP : 7.2.5
- Symfony : 5.4
- Doctrine/orm : 2.20
- BDD: MySQL 5.7
- twig/twig : 2.12|^3.0
- Stripe/stripe-php: 16.2
- phpunit : ^9.5

Entities

User

This entity contents the whole of users for this application.

Champs	Specifications
id	GeneratedValue,integer
email	type="string", length=180, unique=true
roles	type="json", default='ROLE_USER'
password	type="string", NotCompromisedPassword, Length(min=12, max=20)
lastName	type="string", length=255, NotBlank, Length(min=5, max=50)
firstName	type="string", length=255, NotBlank, Length(min=5, max=50)
isActivated	type="boolean", default=false
created_at	type="datetime_immutable", options={"default": "CURRENT_TIMESTAMP"}
created_by	type="string", length=255, options={"default": "Init"}
updated_at	type="datetime_immutable", nullable=true
updated_by	type="string", length=255, nullable=true
orders	OneToMany(targetEntity=Order::class, mappedBy="user"

Champs	Specifications
userCursusLessons	OneToMany(targetEntity=UserCursusLesson::class, mappedBy="user")
usercursuses	not used

Thema

This entity contents differents themas of learnings defined by admin.

Champs	Specifications
id	GeneratedValue,integer
name	type="string", length=255
cursuses	OneToMany(targetEntity=Cursus::class, mappedBy="thema")
created_at	type="datetime_immutable", options={"default": "CURRENT_TIMESTAMP"}
created_by	type="string", length=255, options={"default": "Init"}
updated_at	type="datetime_immutable", nullable=true
updated_by	type="string", length=255, nullable=true

Cursus

This entity contents differents cursuses of learnings defined by admin. A cursus can contain several lessons and at least one and can be attached to only one thema.

Champs	Specifications
id	GeneratedValue,integer
name	type="string", length=255
level	type="string", length=255, ex:"d'initiation"
price	type="float", Positive
nbLessons	type="integer", options={"default": "1"}, Positive, quantity of lessons in the cursus
thema	ManyToOne(targetEntity=Thema::class, inversedBy="cursuses")
lessons	OneToMany(targetEntity=Lesson::class, mappedBy="cursus")
created_at	type="datetime_immutable", options={"default": "CURRENT_TIMESTAMP"}
created_by	type="string", length=255, options={"default": "Init"}
updated_at	type="datetime_immutable", nullable=true
updated_at	type="datetime_immutable", nullable=true

Champs	Specifications
userCursusLessons	OneToMany(targetEntity=UserCursusLesson::class, mappedBy="cursus")

Lesson

This entity contents differents lessons defined by admin. A lesson can be contained only by one cursus.

Champs	Specifications
id	GeneratedValue,integer
name	type="string", length=255
number	type="integer", Positive, range of lessons in the cursus
price	type="float", Positive
video	type="string", length=255, nullable=true, link of video
text	type="text", nullable=true, lesson's text
cursus	ManyToOne(targetEntity=Cursus::class, inversedBy="lessons")
created_at	type="datetime_immutable", options={"default": "CURRENT_TIMESTAMP"}
created_by	type="string", length=255, options={"default": "Init"}
updated_at	type="datetime_immutable", nullable=true
updated_at	type="datetime_immutable", nullable=true
userCursusLessons	OneToMany(targetEntity=UserCursusLesson::class, mappedBy="learning")

Order

This entity contents header of order.

Champs	Specifications
id	GeneratedValue,integer, used as order's number
user	ManyToOne(targetEntity=User::class, inversedBy="orders"), JoinColumn(nullable=false)
cartId	type="string", length=255, nullable=true, cart session's id
created_at	type="datetime_immutable", options={"default": "CURRENT_TIMESTAMP"}
created_by	type="string", length=255, options={"default": "Cart"}
updated_at	type="datetime_immutable", nullable=true
updated_at	type="datetime_immutable", nullable=true

Champs	Specifications
orderdetails	OneToMany(targetEntity=Orderdetail::class, mappedBy="ordernumber", cascade={"persist"})

Orderdetail

This entity contents differents lines of order. A order can contain several lines.

Champs	Specifications
id	GeneratedValue,integer, used as order's number
repository	type="string", length=255, repository of bought learning, not linked with entities cursus or lesson
learning_id	type="integer", learning's id of bought learning, not linked with entities cursus or lesson
price	type="float"
created_at	type="datetime_immutable", options={"default": "CURRENT_TIMESTAMP"}
created_by	type="string", length=255, options={"default": "Cart"}
updated_at	type="datetime_immutable", nullable=true
updated_at	type="datetime_immutable", nullable=true
ordernumber	ManyToOne(targetEntity=Order::class, inversedBy="orderdetails"), JoinColumn(nullable=false)

UserCursusLesson

This entity contents differents lessons, bought by user. If the customer has chosen one cursus, there are one record for each lesson of this cursus. If the customer has chosen only one lesson, there is only one record.

Champs	Specifications
id	GeneratedValue,integer, used as order's number
user	ManyToOne(targetEntity=User::class, inversedBy="userCursusLessons"), JoinColumn(nullable=false)
cursus	ManyToOne(targetEntity=Cursus::class, inversedBy="userCursusLessons"), JoinColumn(nullable=false)
learning	ManyToOne(targetEntity=Lesson::class, inversedBy="userCursusLessons"), JoinColumn(nullable=false)
isValidated	type="boolean", default=false
created_at	type="datetime_immutable", options={"default": "CURRENT_TIMESTAMP"}
created_by	type="string", length=255, options={"default": "Cart"}

Champs	Specifications
updated_at	type="datetime_immutable", nullable=true
updated_at	type="datetime_immutable", nullable=true

Controllers

AdminController

This controller is used for administration's routes. All paths are accessible with a user 'ROLE_ADMIN'.

Route("/admin", name="app_admin")

This is the home route for administration.

Route("/admin/users", name="app_admin_users")

This route displays the list of users with the possibility of searching one.

@Route("/admin/users/delete/{id}", name="app_admin_users_delete")

This function deletes a user by id (@param int \$id).

@Route("/admin/users/update/{id}", name="app_admin_users_update")

This function update name (UserModifyAdminFormType), mail (MailUserModifyFormType) a user by id (@param int \$id).

@Route("/admin/learnings", name="app_admin_learnings")

This function displays and allows creating cursus (CursusRegistrationFormType), lessons (LessonRegistrationFormType) and thema (ThemaRegistrationFormType).

@Route("/admin/thema/delete/{id}", name="app_admin_thema_delete")

This function deletes thema by id (@param int \$id).

@Route("/admin/thema/update/{id}", name="app_admin_thema_update")

This function updates thema's name by id (ThemaModifyFormType).

@Route("/admin/cursus/delete/{id}", name="app_admin_cursus_delete")

This function deletes cursus by id (@param int \$id).

@Route("/admin/cursus/update/{id}", name="app_admin_cursus_update")

This function updates cursus by id (@param int \$id, CursusRegistrationFormType), add a lesson (LessonRegistrationFormType) and display list of lessons by cursus.

```
@Route("/admin/lesson/delete/{id}", name="app_admin_lesson_delete")
```

This function deletes lesson by id (@param int \$id).

```
@Route("/admin/lesson/update/{id}", name="app_admin_lesson_update")
```

This function updates lesson by id (@param int \$id).

```
@Route("/admin/user/history/{id}", name="app_admin_user_history")
```

This routes displays orders, learnings and certifications by user (@param int \$id).

```
@Route("/admin/orders", name="app_admin_orders")
```

This routes displays orders by month and reporting of sold learnings.

```
@Route("/admin/user/orderdetail/{id}", name="app_admin_user_orderdetail")
```

This routes displays lines by order (@param int \$id).

BoughtLessonController

This controller is used for displaying and validating lessons after buying by user. All paths are accessible with a user 'ROLE_CLIENT'.

```
@Route("/bought/lesson/{id}", name="app_bought_lesson")
```

This route displays lesson's content (@param int \$id).

```
@Route("/bought/cursus/{id}", name="app_bought_cursus")
```

This route displays bought cursuses by user (@param int \$id Cursus's id).

```
@Route("/bought/lesson/validate/{id}", name="app_bought_lesson_validate")
```

This function validates a lesson by user (@param int \$id userCursusLesson's id).

CartController

This controller is used for buying and payment process. All paths are accessible with a user 'ROLE_CLIENT'.

```
@Route("/cart", name="app_cart")
```

This route displays session cart's content.

```
@Route("/cart/add/{repository}/{id}", name="app_cart_add")
```

This function allows adding a new learning: cursus or lesson to cart (@param int \$id Cursus ou lesson's id, @param string \$repository Cursus or Lesson repository).

@Route("/cart/remove/{repository}/{id}", name="app_cart_delete")

This function allows deleting learning by id and repository: cursus or lesson from cart (@param int \$id Cursus ou lesson's id, @param string \$repository Cursus or Lesson repository).

@Route("/cart/pay", name="app_cart_pay")

This function allows preparing data from session cart and sending them into Stripe (@param array \$session, App\Service\StripeService).

@Route("/cart/pay/success", name="app_pay_success")

This function allows creating order, order's details and usercursuslesson with session's data and remove session after stripe's paiement (@param array \$session).

@Route("/cart/pay/cancel", name="app_pay_cancel")

This route is displayed in case of return from stripe.

HomeController

This controller is used for displaying home route.

@Route("/", name="app_home")

This route displays home route.

LearningController

This controller is used for displaying shopping routes.

@Route("/learning", name="app_learning")

This route displays themas.

@Route("/learning/{id}", name="app_learning_cursus")

This route displays cursuses by thema (@param int \$id).

@Route("/learning/lesson/{id}", name="app_learning_lesson")

This route displays lessons by cursus (@param int \$id).

ProfileController

This controller is used for displaying profile's routes. All paths are accessible with a user 'ROLE_CLIENT'.

@Route("/profile", name="app_profile")

This route displays profile home route. Only this path is accessible with a user 'ROLE_USER'.

@Route("/profile/update", name="app_profile_update")

This route allows updating profile by user: name (`UserModifyFormType`), password (`PasswordModifyFormType`), mail (`MailUserModifyFormType`).

@Route("/profile/orders", name="app_profile_orders")

This route allows displaying orders by user. The user comes from logged user.

@Route("/profile/orderdetails/{id}", name="app_profile_orderdetails")

This route allows displaying line of orders by user (`@param int $id Order's id`). The user comes from logged user.

@Route("/profile/learnings", name="app_profile_learnings")

This route allows displaying bought learnings by user. The user comes from logged user.

@Route("/profile/certifications", name="app_profile_certifications")

This route allows displaying certifications by user. The user comes from logged user.

RegistrationController

This controller is used for creating user.

@Route("/register", name="app_register")

This route displays register's form (`RegistrationFormType`), allows sending mail (`App\Service\SendEmailService`) and creating JWT (`App\Service\JWTService`).

@Route("/activateuser/{token}", name="app_activate_user")

This function allows creating initial link sent to user with token in order to update field `isActive` (`@param string $token, App\Service\JWTService`).

@Route("/activate/resendlink", name="app_activate_resendlink")

This function allows creating and resending new link sent to user with token in order to update field `isActive` (`@param string $token, App\Service\JWTService, App\Service\SendEmailService`).

SecurityController

This controller is used for connection.

@Route("/login", name="app_login")

This route displays login's form.

@Route("/logout", name="app_logout")

This route allows logout.

Tests

AdminControllerTest

These tests concern `AdminController`.

testRoute_app_admin

This tests displaying /admin.

testRoute_app_admin_users

This tests displaying route /admin/users with checking presence of a user.

testRoute_app_admin_users_delete

This tests deleting user with checking redirect.

testRoute_app_admin_users_update

This tests updating user with checking response.

testRoute_app_admin_learnings

This tests displaying route /admin/learnings with checking presence of a cursus.

testCreate_Thema

This tests creating thema with checking redirect.

testUpdate_Thema

This tests updating thema with checking redirect.

testRoute_app_admin_thema_delete

This tests deleting thema with checking redirect.

testRoute_app_admin_user_history

This tests displaying orders by user's on route /admin/user/history/1 with checking presence order and order's element.

testRoute_app_admin_orders

This tests displaying orders on route /admin/orders with checking of tables and contents.

testRoute_app_admin_user_orderdetail

This tests displaying details of an order on route /admin/user/orderdetail/1 with checking of table and contents.

BoughtLessonControllerTest

These tests concern BoughtLessonController.

testRoute_app_bought_lesson

This tests displaying lesson's on route /bought/lesson/1 with checking presence of the title.

testRoute_app_bought_cursus

This tests displaying cursus's on route /bought/cursus/1 with checking presence of the cursus and lessons.

testRoute_app_bought_lesson_validate

This tests validating a lesson on route /bought/lesson/validate/1 with checking presence of confirmation's text.

CartControllerTest

These tests concern CartController.

testRoute_app_cart_add

This tests adding a learning to cart on route /cart/add/cursus/1 with checking presence of a learning in route.

testRoute_app_cart_delete

This tests deleting a learning in cart on route /cart/remove/cursus/1 with checking if learning not exists in route.

testRoute_app_cart_pay

This tests validating the cart on route /cart/pay with checking redirect.

testRoute_app_pay_success

This tests returning success of payment on route /cart/pay/success with checking text.

testRoute_app_pay_cancel

This tests returning cancel of payment on route /cart/pay/cancel with checking text.

LearningControllerTest

These tests concern LearningController.

testRoute_app_learning

This tests displaying list of themas on route /learning with checking presence of thema's name.

testShouldDisplayLoggedUser

This tests displaying logged user on route /learning with checking presence of name.

testRoute_app_learning_cursus

This tests displaying list of cursuses on route /learning/1 with checking presence of ursus's name.

testRoute_app_learning_lesson

This tests displaying list lessons for a cursus on route /learning/lesson/1 with checking presence of a lesson.

ProfileControllerTest

These tests concern **ProfileController**.

testRoute_app_profile

This tests displaying profile on route /profile with checking response.

testRoute_app_profile_update

This tests updating user on route /profile/update with checking response.

testRoute_app_profile_orders

This tests displaying orders of user on route /profile/orders with checking order's number.

testRoute_app_profile_orderdetails

This tests displaying details for an order on route /profile/orderdetails/1 with checking presence of a cursus, contained in the order.

testRoute_app_profile_learnings

This tests displaying bought learnings by user on route /profile/learnings with checking presence of learning's name.

testRoute_app_profile_certifications

This tests displaying validated learnings by user on route /profile/certifications with checking presence learning's name.

RegistrationControllerTest

These tests concern **RegistrationController**.

testRoute_app_register

This tests creating a user on route /register with checking response.

SecurityControllerTest

These tests concern `SecurityController`.

testRoute_app_login

This tests logging a user on route /login with checking presence of user's name.

testRoute_app_logout

This tests logging out a user on route /logout with checking absence of learning's name.

CursusTest

These tests concern entity `Cursus`.

testCreateCursus

This tests creating cursus with checking presence cursus in dbb.

testAddGetRemoveLesson

This tests adding, getting, removing link with entity Lesson.

testAddGetRemoveUserCursusLesson

This tests adding, getting, removing link with entity UserCursusLesson.

LessonTest

These tests concern entity `Lesson`.

testCreateLesson

This tests creating lesson with checking presence lesson in dbb.

testAddGetRemoveUserCursusLesson

This tests adding, getting, removing link with entity UserCursusLesson.

OrderdetailTest

These tests concern entity `Orderdetail`.

testCreateOrderdetail

This tests creating orderdetail with checking presence lesson in dbb.

OrderTest

These tests concern entity **Order**.

testCreateOrder

This tests creating order with checking presence lesson in dbb.

testAddGetRemoveOrderDetail

This tests adding, getting, removing link with entity OrderDetail.

ThemaTest

These tests concern entity **Thema**.

testCreateThema

This tests creating thema with checking presence lesson in dbb.

testAddGetRemoveCursus

This tests adding, getting, removing link with entity Cursus.

UserCursusLessonTest

These tests concern entity **UserCursusLesson**.

testCreateUserCursusLesson

This tests creating thema with checking presence lesson in dbb.

UserTest

These tests concern entity **User**.

testCreateUserCursusLesson

This tests creating user with checking presence lesson in dbb.

testAddGetRemoveOrder

This tests adding, getting, removing link with entity Order.

testAddGetRemoveUserCursusLesson

This tests adding, getting, removing link with entity UserCursusLesson.