**Universitatea de Stat din Moldova**

**Facultatea Matematica și Informatică**

**Departamentul Informatică**

**Lucrare de laborator Nr. 2**

A realizat: Chistol Liviu

A verificat : Bodrug Svetlana

**Chișinau,2020**

# **Flex: creează-ți aplicația**

După ce ai citit prima parte a acestui tutorial, ai decis că Symfony merită încă 10 minute. O alegere grozavă! În această a doua parte, veți afla despre Symfony Flex: un instrument grozav care face ca adăugarea de noi funcții să fie atât de ușoară precum rularea unei comenzi. De asemenea, este și motivul pentru care Symfony este ideal pentru un mic serviciu mic sau o aplicație uriașă. Interesat? Excelent!

## **Symfony: Începeți cu Micro!**

Dacă nu construiți o API curată (mai multe despre asta mai târziu!), Cel mai probabil veți dori să afișați HTML. Pentru a face acest lucru, veți folosi [Twig](https://twig.symfony.com/) . Twig este un motor de șabloane flexibil, rapid și sigur pentru PHP. Îți face șabloanele mai lizibile și mai compacte; de asemenea, le face mai prietenoase pentru designeri web.

Twig este deja instalat în aplicația noastră? De fapt, nu încă! Și acest lucru este minunat! Când începeți un nou proiect Symfony, este mic : doar cele mai critice dependențe sunt incluse în fișierul dvs composer.json.

Acest lucru diferențiază Symfony de orice alt cadru PHP! În loc să începi cu o aplicație greoaie cu fiecarecaracteristică posibilă de care ai putea avea nevoie, aplicația Symfony este mică, simplă și rapidă. Și aveți control deplin asupra a ceea ce adăugați.

## **Rețete și nume suplimentare Flex**

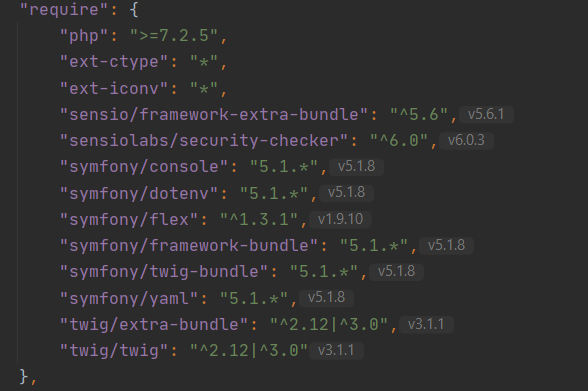
Deci, cum instalăm și configurăm Twig? Rulați doar o comandă:

Două lucruri foarte interesante se întâmplă în culise datorită Symfony Flex: plugin-ul Composer instalat deja în proiectul nostru.

În primul rând, twig acesta nu este numele pachetului Composer: este numele opționalFlex care indică symfony/twig-bundle. Flex rezolvă acest nume alternativ pentru Composer.

Și în al doilea rând, Flex stabilește rețetapentru symfony/twig-bundle. Ce este o rețetă? Acesta este un mod prin care biblioteca se configurează automat prin adăugarea și modificarea fișierelor. Datorită rețetelor, adăugarea funcțiilor este perfectă și automatizată: instalați pachetul și ați terminat!

Acum daca ne uitam in composer.json vedem :



Ce a făcut această rețetă? Pe lângă conectarea automată a funcției config/bundles.php, el a adăugat 3 lucruri:

config/packages/twig.yaml

Un fișier de configurare care instalează Twig cu setări implicite rezonabile.

config/routes/dev/twig.yaml

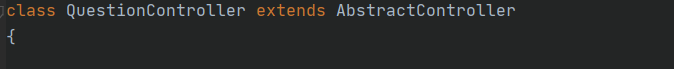
Un traseu care vă ajută să depanați paginile de eroare.

templates/

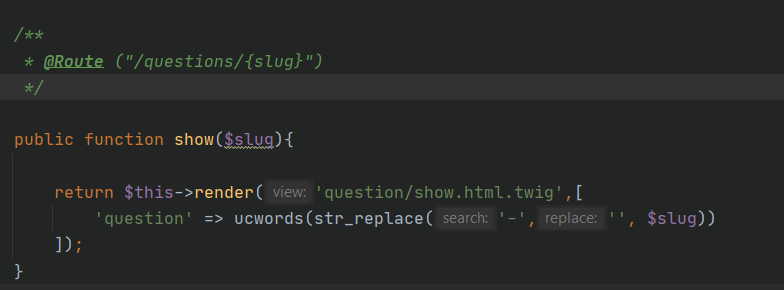
Directorul în care vor trăi fișierele șablon. Rețeta a adăugat și un fișier format base.html.twig.

## **TWIG: șablon afișare**

Datorită Flex, după o singură comandă, puteți începe imediat să folosiți Twig:



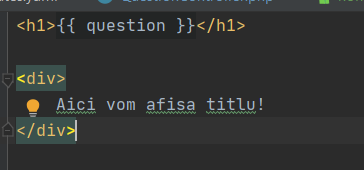
Si, adaugam o functie noua , care va afisa ce am introdus in calea noastra :



După extindere AbstractController, acum aveți acces la o listă de metode și instrumente de reducere, cum ar fi render(), care l-am utilizat mai devreme

Asta este totul! Sintaxa va {{ slug }}afișa variabila slug care este transmisă de la controler.

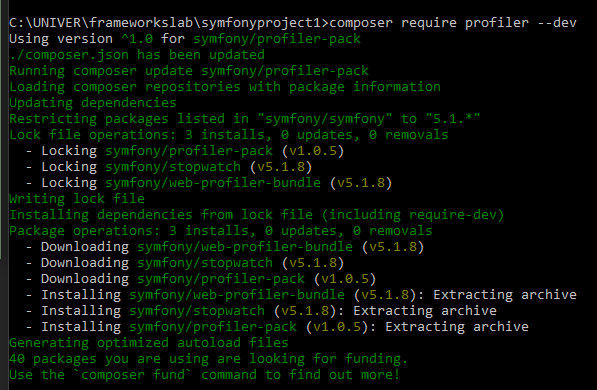
Dar chiar acum. pagina conține doar eticheta h1. Pentru a-i oferi un aspect HTML, extindeți show.html.twig. Iar in templaturi vom adauga fisierul acesta si continutul :



Aceasta se numește moștenire șablon: pagina noastră moștenește acum structura HTML de la show.html.twig.

## **Profiler: Debugging Paradise**[**¶**](https://symfony.com.ua/doc/current/quick_tour/flex_recipes.html#id2)

Una dintre cele mai cool caracteristici ale Symfony nu a fost încă instalată! Să o rezolvăm:

Da! Acesta este un alt aliat! De asemenea, Flex instalează o altă rețetă care automatizează configurația Symfony Profiler. Care este rezultatul? Actualizează!

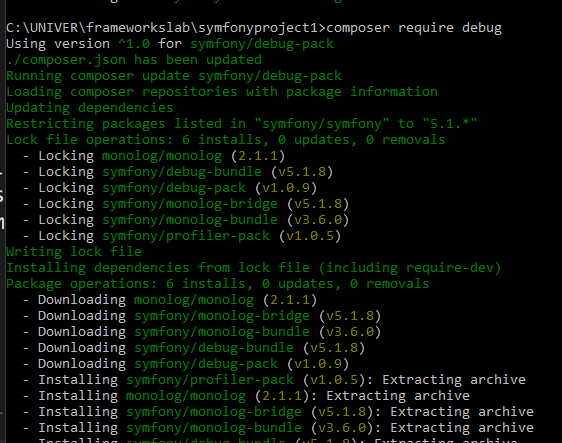
Vezi mai jos acest panou negru?

 Aceasta este o bară de instrumente de depanare web și este noul tău cel mai bun prieten. Afișând fiecare pictogramă, puteți obține informații despre ce controler a fost completat, informații despre performanță, noroc și memorie în cache, și multe altele. Faceți clic pe orice pictogramă pentru a intra în profiler , de unde puteți obține date șimai detaliate despre depanare și performanță!

Și, în timpul instalării de noi biblioteci, veți primi mai multe instrumente (de exemplu, pictograma barei de instrumente de depanare web care afișează interogări ale bazei de date).

Utilizarea profilatorului este simplă, deoarece s-a configurat ea însăși datorită rețetei. Ce altceva putem instala la fel de ușor?

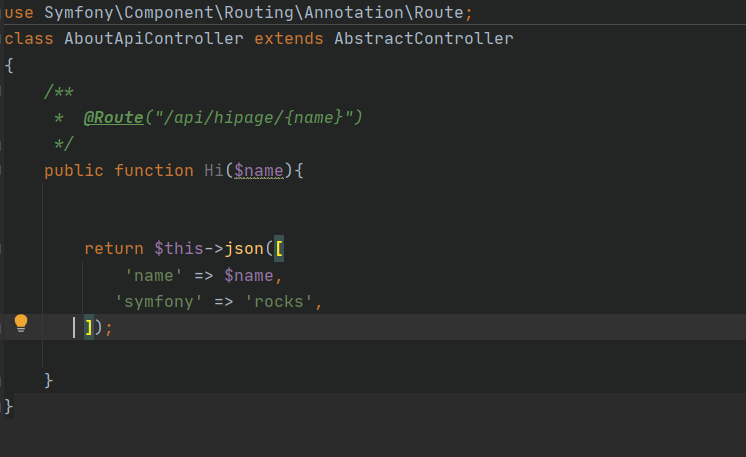
Putem adauga si debug , prin composer require debug :

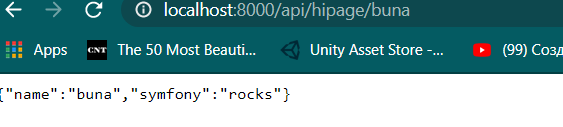


Care adauga librarii ce ne vor fi de ajutor.

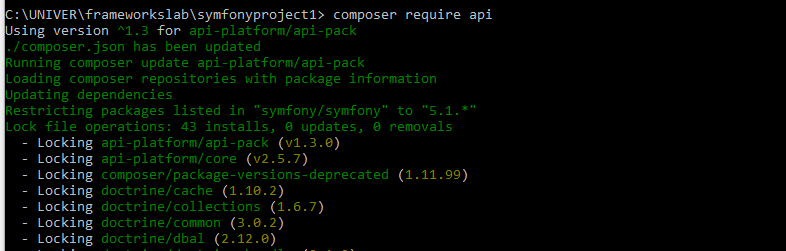
## **suport bogat pentru API**

Construiți o API? Puteți returna deja cu ușurință JSON de la orice controler:

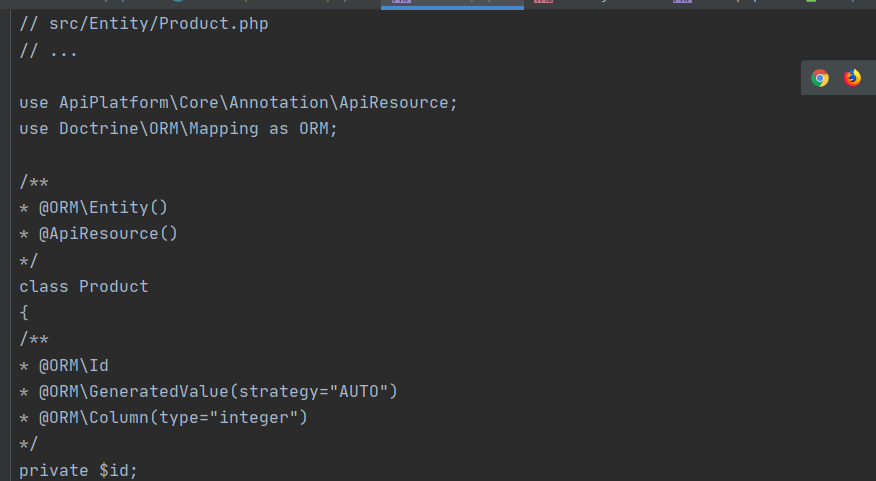




Dar pentru o API cu adevăratbogată, încercați să instalați [platforma Api](https://api-platform.com/) :

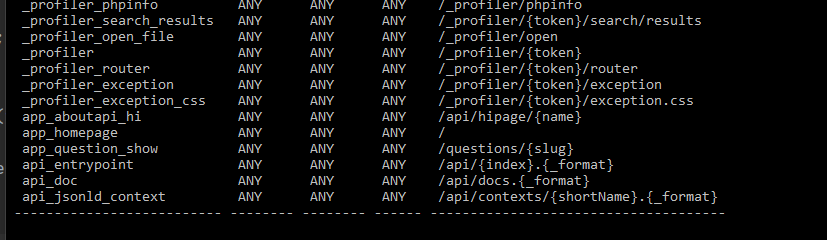
Acesta este un nume suplimentar api-platform/api-pack care are dependențe în mai multe alte pachete, cum ar fi Symfony Validator și componente de securitate, precum și Doctrine ORM. De fapt, Flex a stabilit 5 rețete!

Dar, ca de obicei, putem începe imediat utilizarea noii biblioteci. Vrei să creezi o API bogată pentru un tabel product? Creați o entitate Productși dați adnotarea @ApiResource():

…

Gata! Acum aveți puncte finale pentru listarea, adăugarea, actualizarea și ștergerea produselor! Nu mă credeți? Enumerați-vă rutele făcând:





## **Rețete ușor de șters**

Încă nu ești convins? Nu este o problemă: eliminați biblioteca:

|  |
| --- |
| composer remove api |

Flex va șterge rețetele: ștergerea fișierelor și eliminarea modificărilor pentru a returna aplicația la starea inițială. Experimentați fără teamă.