**Universitatea de Stat din Moldova**

**Facultatea Matematica și Informatică**

**Departamentul Informatică**

**Lucrare de laborator Nr. 1**

A realizat: Chistol Liviu

A verificat : Bodrug Svetlana

**Chișinau,2020**

Instalarea și configurarea cadrului Symfony

Cerințe tehnice

Înainte de a crea prima aplicație Symfony, trebuie să:

* Instalați PHP 7.2.9 sau o versiune ulterioară și aceste extensii PHP (care sunt instalate și activate în mod implicit în majoritatea instalațiilor PHP 7): [Ctype](https://php.net/book.ctype) , [iconv](https://php.net/book.iconv) , [JSON](https://php.net/book.json) , [PCRE](https://php.net/book.pcre) , [Session](https://php.net/book.session) , [SimpleXML](https://php.net/book.simplexml) și [Tokenizer](https://php.net/book.tokenizer) ;
* [Install Composer](https://getcomposer.org/download/) , care este utilizat pentru instalarea pachetelor PHP;
* [Instalați Symfony](https://symfony.com/download) , care creează în computer un apel numit binar symfony care oferă toate instrumentele de care aveți nevoie pentru a vă dezvolta aplicația local.

Crearea aplicațiilor Symfony

Deschideți terminalul consolei și rulați oricare dintre aceste comenzi pentru a crea o nouă aplicație Symfony:

|  |
| --- |
|  |

Am ales optiunea –full care instalează toate pachetele de care aveți nevoie , de obicei , pentru a construi aplicații web, astfel încât dimensiunea de instalare va fi mai mare.

Indiferent de comanda pe care o executați pentru a crea aplicația Symfony. Toate acestea vor crea un nou symfonyproject1/director, vor descărca unele dependențe în el și chiar vor genera directoarele de bază și fișierele de care va trebui să începeți. Cu alte cuvinte, noua dvs. aplicație este gata!

**NOTĂ**

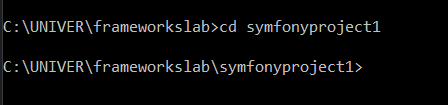
Directorul cache și jurnalele proiectului (implicit <project>/var/cache/ și <project>/var/log/) trebuie să fie redactate de serverul web. Dacă aveți probleme, citiți cum să [configurați permisiunile pentru aplicațiile Symfony](https://symfony.com/doc/master/setup/file_permissions.html) .

Executarea aplicațiilor Symfony

Pentru dezvoltare, este mai convenabil să folosiți [serverul web local](https://symfony.com/doc/master/setup/symfony_server.html) furnizat de Symfony.

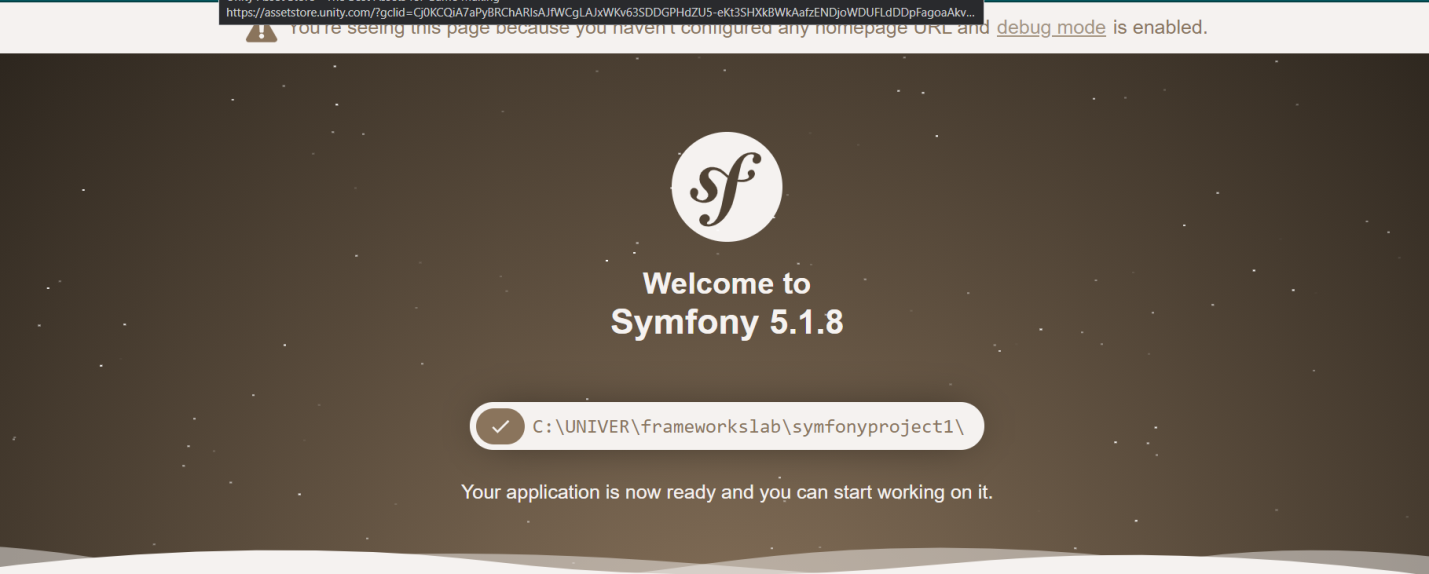
Acest server local oferă asistență pentru HTTP / 2, TLS / SSL, generarea automată de certificate de securitate și multe alte funcții. Funcționează cu orice aplicație PHP, nu numai proiectele Symfony, deci este un instrument de dezvoltare foarte util.

Deschideți terminalul consolei, treceți în noul dvs. director de proiect și porniți serverul web local astfel:



|  |
| --- |
|  |

Deschideți browserul și navigați la http://localhost:8000/. Dacă totul funcționează, veți vedea o pagină de bun venit. Mai târziu, când ați terminat de lucrat, opriți serverul apăsând Ctrl+Cpe terminal.



Creați-vă prima pagină în Symfony

* + [Crearea unei pagini: Rută și controler](https://symfony.com/doc/master/page_creation.html#creating-a-page-route-and-controller)
  + [Rute de adnotare](https://symfony.com/doc/master/page_creation.html#annotation-routes)

Crearea unei noi pagini - indiferent dacă este o pagină HTML sau un punct final JSON - este un proces în doi pași:

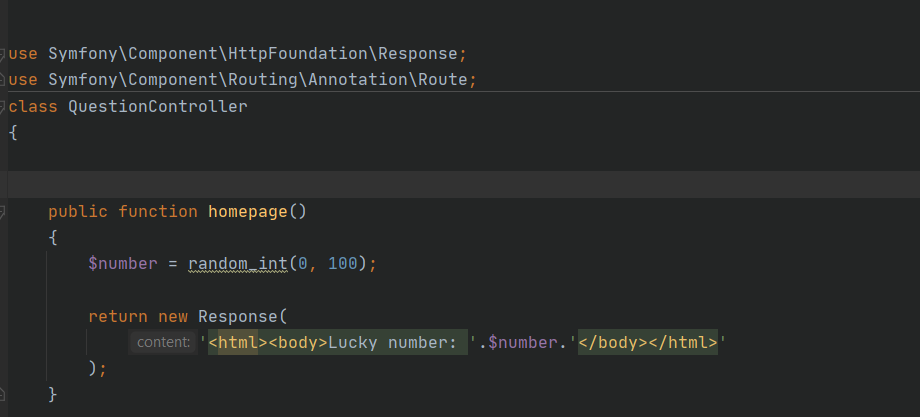
1. **Creați un traseu** : o rută este adresa URL (de exemplu /about) către pagina dvs. și indică un controler;
2. **Creați un controler** : Un controler este funcția PHP pe care o scrieți care creează pagina. Luați informațiile de solicitare primite și le utilizați pentru a crea un Responseobiect Symfony , care poate reține conținut HTML, un șir JSON sau chiar un fișier binar precum o imagine sau un PDF.

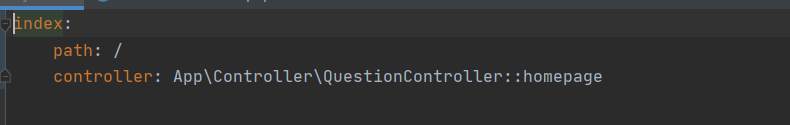
Crearea unei pagini: Rută și controler

**BACSIS**

Înainte de a continua, asigurați-vă că ați citit articolul de [configurare](https://symfony.com/doc/master/setup.html) și puteți accesa noua aplicație Symfony în browser.

Să presupunem că doriți să creați o pagină - /lucky/number- care generează un număr norocos (bine, la întâmplare) și o tipărește. Pentru aceasta, creați o clasă „Controller” și o metodă „controller” în interiorul acesteia:

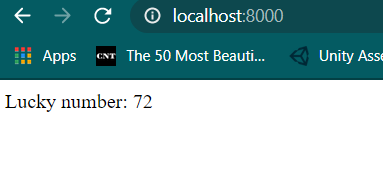
Acum trebuie să asociați această funcție de controler cu o adresă URL publică (de exemplu /), astfel încât homepage()metoda să fie executată atunci când un utilizator răsfoiește la ea. Această asociere este definită prin crearea unei **rute** în config/routes.yaml fișier:

**

Asta e! Dacă utilizați serverul web Symfony, încercați-l accesând:

[*http: // localhost: 8000*](http://localhost:8000/lucky/number)

Dacă vedeți că un număr norocos vă este imprimat înapoi, felicitări!



 Dar înainte să fugi să joci la loterie, verifică cum funcționează asta. Vă amintiți cei doi pași pentru crearea unei pagini?

1. Creați un traseu : În config/routes.yaml, ruta definește adresa URL a dvs.

pagina ( path) și ce controllersă apeleze. Veți afla mai multe despre [rutarea](https://symfony.com/doc/master/routing.html) în propria secțiune, inclusiv despre cum să faceți adrese URL variabile ;

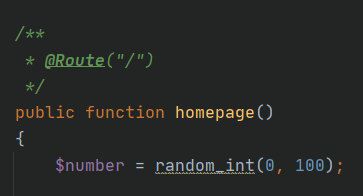
1. Creați un controler : Aceasta este o funcție în care construiți pagina și în cele din urmă returnați un Responseobiect. Veți afla mai multe despre [controlerele](https://symfony.com/doc/master/controller.html) din propria lor secțiune, inclusiv despre cum să returnați răspunsurile JSON.

Rute de adnotare

În loc să definiți ruta dvs. în YAML, Symfony vă permite, de asemenea, să utilizați rutele de adnotare . Pentru a face acest lucru, instalați pachetul de adnotări:



Acum puteți adăuga ruta direct deasupra controlerului:



Asta e! Pagina - <http://localhost:8000>/ va funcționa exact ca înainte! Adnotările sunt modalitatea recomandată de configurare a rutelor.

Instalarea automată a rețetelor cu Symfony Flex

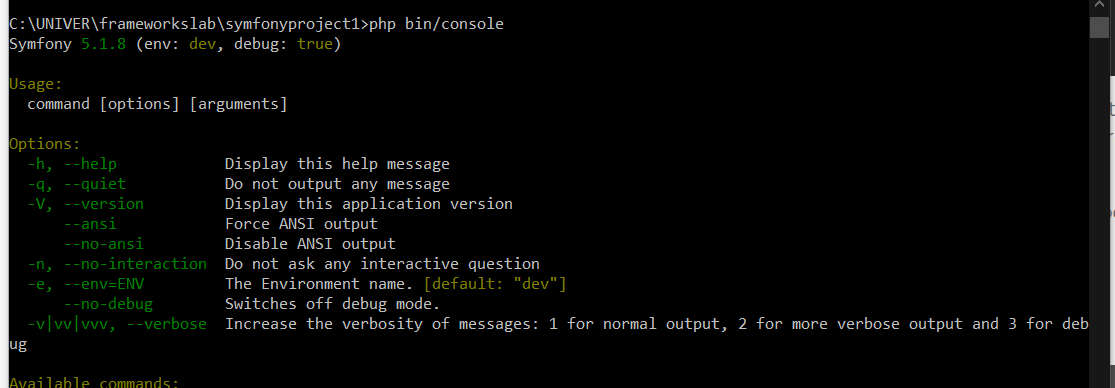
Poate că nu ați observat, dar când ați alergat composer require annotations, s-au întâmplat două lucruri speciale, ambele datorită unui puternic plugin Composer numit [Flex](https://symfony.com/doc/master/setup.html#symfony-flex) .

În primul rând, annotationsnu este un nume adevărat de pachet: este un alias (adică scurtătură) la care se rezolvă Flex sensio/framework-extra-bundle.

În al doilea rând, după ce acest pachet a fost descărcat, Flex a executat o rețetă , care este un set de instrucțiuni automatizate care spun Symfony cum să integreze un pachet extern. [Rețetele Flex](https://flex.symfony.com/) există pentru multe pachete și au capacitatea de a face multe, cum ar fi adăugarea fișierelor de configurare, crearea de directoare, actualizarea .gitignore și adăugarea de configurații noi în .envfișierul dvs. Flex automatizează instalarea pachetelor, astfel încât să puteți reveni la codare.

comandă bin / console  [¶](https://symfony.com/doc/master/page_creation.html#the-bin-console-command)

Proiectul dvs. are deja un instrument puternic de depanare în interior: bin/consolecomanda.

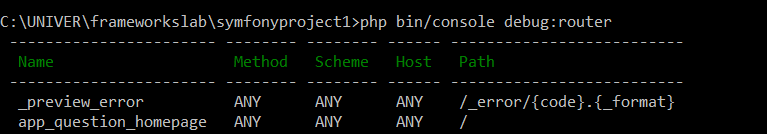


Pe măsură ce instalați mai multe pachete, veți vedea mai multe comenzi.

Pentru a obține o listă cu *toate* rutele din sistemul dvs., utilizați debug:routercomanda:

|  |
| --- |
| php bin/console debug:router |

Ar trebui să vedeți app\_lucky\_numbertraseul dvs. chiar de sus:

De asemenea, veți vedea mai jos rutele de depanare app\_lucky\_number- mai multe pe rutele de depanare din secțiunea următoare.

Bara de instrumente pentru depanare Web: Debugging Dream

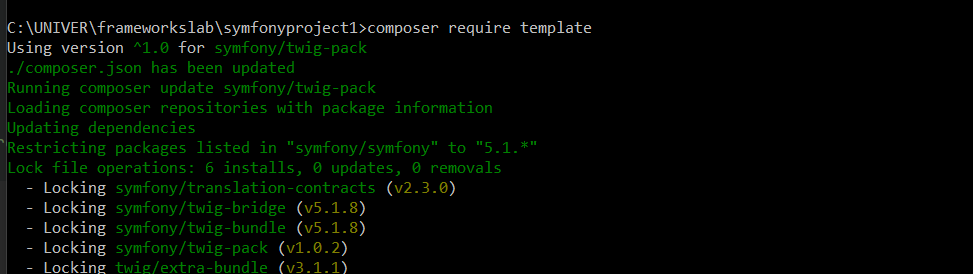
Una dintre caracteristicile ucigașe ale Symfony este bara de instrumente de depanare Web: o bară care afișează o cantitate uriașă de informații de depanare de-a lungul părții de jos a paginii dvs. în timpul dezvoltării. Toate acestea sunt incluse în cutie folosind un [pachet](https://symfony.com/doc/master/setup.html#symfony-packs) numit [Symfony](https://symfony.com/doc/master/setup.html#symfony-packs)symfony/profiler-pack .

Veți vedea o bară neagră în partea de jos a paginii. Veți afla mai multe despre toate informațiile pe care le deține pe parcurs, dar nu ezitați să experimentați: treceți peste și faceți clic pe pictogramele diferite pentru a obține informații despre rutare, performanță, înregistrare și multe altele.

Redarea unui șablon

Dacă returnați HTML de pe controler, probabil veți dori să redau un șablon. Din fericire, Symfony vine cu [Twig](https://twig.symfony.com/) : un limbaj de șablon ușor, puternic și de fapt destul de distractiv.

Pentru asta instalam pachetul “template”:



**Почему еще один шаблонизатор?**

Когда речь заходит о механизмах шаблонов в PHP, многие люди скажут вам, что сам PHP является механизмом шаблонов. Но даже если PHP начал свою жизнь как язык шаблонов, он не развивался так, как в последние годы. На самом деле, **он не поддерживает многие функции**, которые должны иметь современные движки шаблонов:

* **Кратко** : язык PHP многословен и становится смешно многословным, когда дело доходит до экранирования:
* <? php echo $ var ?>
* <? php echo htmlspecialchars ( $ var , ENT\_QUOTES , 'UTF-8' ) ?>

Для сравнения, Twig имеет очень лаконичный синтаксис, который делает шаблоны более читабельными:

{{ var }}

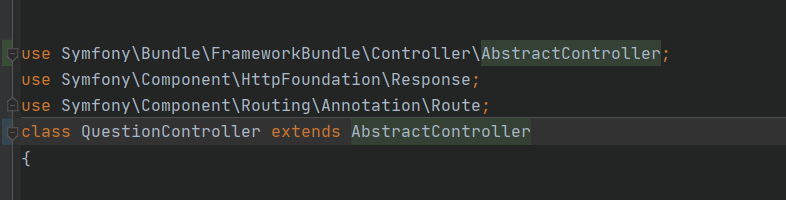
{{ var|escape }}

{{ var|e }} *{# shortcut to escape a variable #}*

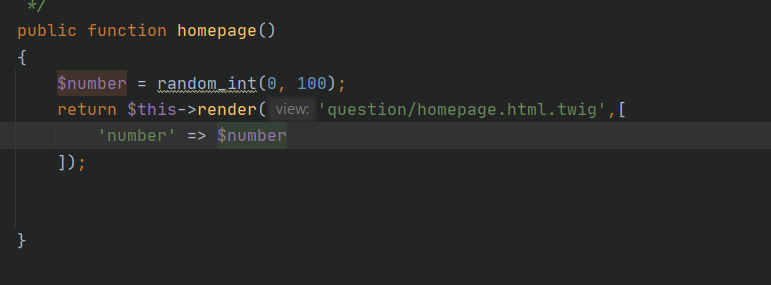
* **Очистка сообщений об ошибках** : всякий раз, когда у вас возникает проблема с синтаксисом в шаблоне, Twig выводит полезное сообщение с именем файла и номером строки, в которой возникла проблема. Это значительно облегчает отладку.
* **Быстро** : одна из целей Twig - быть как можно быстрее. Для достижения максимальной скорости Twig компилирует шаблоны в простой оптимизированный PHP-код. Издержки по сравнению с обычным кодом PHP были сведены к минимуму.

<https://twig.symfony.com/>

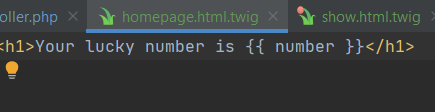
Asigurați-vă că QuestionController extinde [AbstractController](https://github.com/symfony/symfony/blob/master/src/Symfony/Bundle/FrameworkBundle/Controller/AbstractController.php)clasa de bază a Symfony :



Acum, utilizați render()funcția la îndemână pentru a reda un șablon. Treceți-o pe o number variabilă, astfel încât să o puteți utiliza în Twig:



Fișierele de șabloane se află în templates/director, care a fost creat automat pentru dvs. când ați instalat Twig. Creați un nou templates/question director cu un nou homepage.html.twigfișier în interior:



{{ number }}Sintaxa este utilizată pentru a *imprima* variabile în crenguță. Reîmprospătați browserul pentru a obține *noul* dvs. număr norocos!



Acum s-ar putea să vă întrebați unde s-a dus Bara de instrumente de depanare Web: asta pentru că nu există nicio </body>etichetă în șablonul curent. Puteți adăuga singur elementul de corp sau extinde base.html.twig, care conține toate elementele HTML implicite.

În articolul cu [șabloane](https://symfony.com/doc/master/templates.html) , veți afla totul despre Twig: cum să faceți buclă, să redați alte șabloane și să folosiți sistemul său puternic de moștenire a machetei.

Verificarea structurii proiectului

Vesti bune! Ați lucrat deja în cele mai importante directoare din proiectul dvs.:

config/

Conține ... configurare !. Veți configura rute, [*servicii*](https://symfony.com/doc/master/service_container.html) și pachete.

src/

Tot codul dvs. PHP locuiește aici.

templates/

Toate șabloanele tale de Twig trăiesc aici.

De cele mai multe ori, veți lucra în src/, templates/sau config/. Pe măsură ce continuați să citiți, veți afla ce se poate face în fiecare dintre acestea.

Ce zici de celelalte directoare din proiect?

bin/

Celebrul bin/consolefișier trăiește aici (și alte fișiere executabile mai puțin importante).

var/

Aici sunt stocate fișierele create automat, precum fișierele cache ( var/cache/) și jurnalele ( var/log/).

vendor/

Biblioteci terțe (adică „furnizor”) locuiesc aici! Acestea sunt descărcate prin intermediul managerului de pachete [Composer](https://getcomposer.org/) .

public/

Aceasta este rădăcina documentului pentru proiectul dvs.: puneți aici toate fișierele accesibile publicului.

Și când instalați pachete noi, directoarele noi vor fi create automat atunci când este nevoie.