

**Ministerul Educației, Culturii și Cercetării a Republicii Moldova**

**Universitatea de Stat a Moldovei**

**Facultatea Matematică și Informatică**

Raport

**Lucrare de laborator Nr.3  
la disciplina Framework-uri pentru Dezvoltarea de Aplicații Web  
Tema:”Servicii”**

**A efectuat : Batîr Ana**

**A verificat : Bodrug Svetlana**

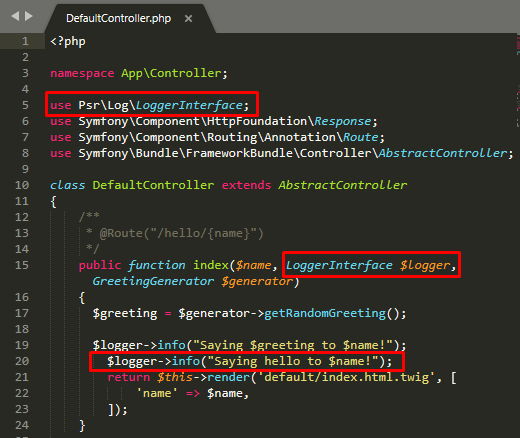
**Chișinău 2020**

# **Adăugarea jurnalului**

Noua aplicație Symfony este microscopică: constă în esență doar dintr-un sistem de rutare și un controler. Dar, datorită Flex, instalarea de noi funcții este ușoară.



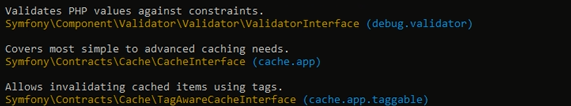
Aceasta instalează și configurează (prin rețetă) puternica bibliotecă Monolog. Pentru a utiliza loggerul într-un controler, adaug un argument nou, tastat de LoggerInterface:



# **Servicii și automatizare**

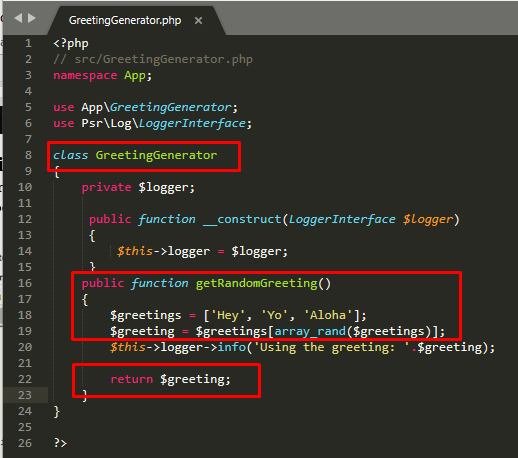
Pot cere de la Symfony să-mi trimitp un serviciu folosind tastarea. Cu ajutorul comenzii de mai jos voi afla cumpot fi utilizate alte clase sau interfețe:



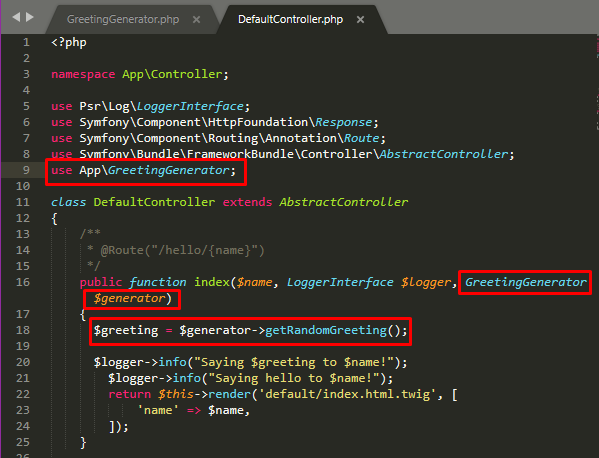


# **Crearea serviciului**

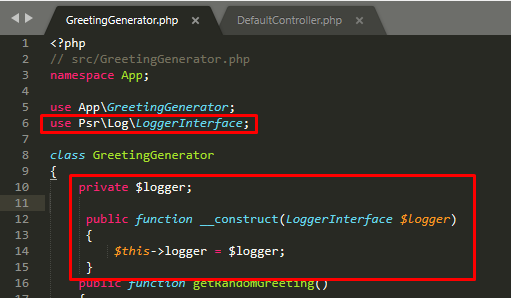
Pentru a menține codul organizat, pot să-mi creez propriile servicii! Dacă doresc să generez o felicitare aleatorie (de exemplu, „Bună ziua”, „Yo” etc.). În loc să introduc acest cod direct în controler, creez o nouă clasă:



Pot utiliza acest lucru în controler imediat:

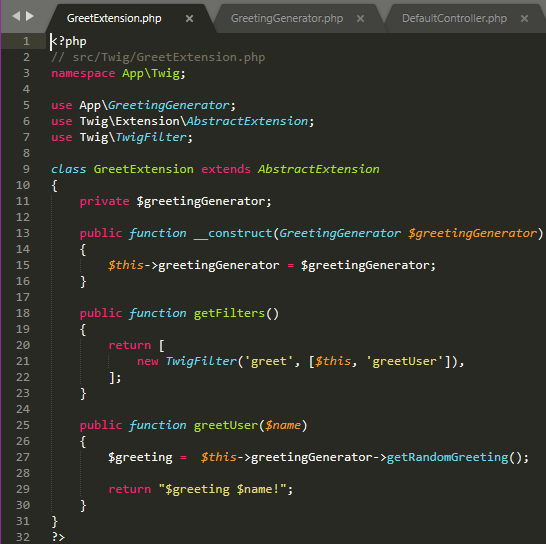


Symfony va instanția GreetingGenerator automat și îl va transmite ca argument. Pot utiliza cablarea automată în cadrul unui serviciu pentru a accesa alte servicii. Singura diferență este că acest lucru se face în constructor:

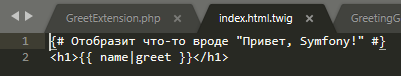


# **Extinderea și autoconfigurarea Twig**

Prin gestionarea serviciilor Symfony, pot extinde Symfony în multe moduri, cum ar fi crearea unui abonat la eveniment sau a unui alegător de securitate pentru reguli complexe de autorizare. Să adaug un nou filtru la Twig numit greet, creând o clasă care extinde AbstractExtension:



După crearea unui singur fișier, îl pot utiliza imediat:



## **REZULTAT:**

