



Symfony Framework

25, 28 Marzo, 02, 04, 08, 11, 16 Aprile, 09, 14, 16, 21 Maggio 2019 orario 09:00-13:00

OBIETTIVI	L'obiettivo del corso è quello di fornire le conoscenze basi necessarie per creare progetti PHP con il framework Symfony in conformità con il pattern architetturale MVC per lo sviluppo di sistemi software complessi.
PREREQUISITI	Si richiede un'ottima conoscenza del linguaggio PHP orientato agli oggetto
DESTINATARI	Il corso è rivolto a programmatori PHP con elevata esperienza nello sviluppo di applicazioni complesse.
ENTE ORGANIZZATORE	Università di Firenze
ENTE EROGATORE	Università di Firenze
LUOGO	Siaf - Via delle Gore, 2
REFERENTE ORGANIZZATIVO	Ufficio Formazione e Sviluppo del Personale
DOCENTI	Frangioni Stefano
DATA INIZIO	
DATA FINE	
DURATA IN ORE	44
SCADENZA PRENOTAZIONI	
MAX PARTECIPANTI	20



PROGRAMMA

1. Setup
 1. Composer. Scopo, installazione, comandi principali. Vincoli sulle versioni. Repository.
 2. Installazione framework: skeleton e website-skeleton
 3. interazioni da linea di comando: cancellare la cache.
2. Creare la prima pagina
 1. route
 2. controller
 3. template
 4. barra di debug
 5. struttura del progetto
 6. Symfony Request e Response
 7. aggiungere bootstrap, jquery, etc
3. Routing
 1. creare le rotte
 2. creare rotte con parametri. requisiti sui parametri
 3. generare url con parametri e query string, assolute o relative
 4. debug delle rotte
4. Controller
 1. associare una url ad un controller
 2. restituire la risposta con un template twig
 3. restituire un oggetto json
 4. servire un file presente su file system.
 5. stream di una risposta
 6. fare il redirect ad un'altra rotta o ad una url esterna
 7. usare un servizio in un controller
 8. ottenere servizi. autowiring, debug. Oggetti Request, Session
 9. gestione degli errori, eccezioni
 10. flash messages
 11. debug/dump
5. Templates con twig
 1. nome e path dei template
 2. ereditarietà e layout. blocchi. include. embed.
 3. tag, funzioni e filtri principali
 4. generare url a partire da rotte
 5. link agli asset js, css
 6. variabile app per accedere a request, user, session, etc da twig.
 7. escaping dell'output
 8. renderizzare un sub-controller
 9. estendere twig
 10. gestione eccezioni con template custom
 11. debug
6. Configurazione
 1. ambienti dev, prod, test
 2. parameters
 3. variabili .env
7. Database con Doctrine ORM
 1. configurazione del db
 2. creare una entity
 3. validation e validation groups

4. tipi di dato
5. creare due entity in relazione
6. migrations
7. read-only entity
8. query DQL, query builder, idratazione oggetti
9. accesso a database nel controller
10. accedere alle entity nella vista
11. eccezioni generate da db
12. paginazione risultati di una ricerca
8. Form
 1. I type predefiniti
 2. creare un form con i type predefiniti
 3. il flusso nella gestione di un form, form nel controller
 4. renderizzare un form in un template
 5. personalizzare la vista
 6. protezione csrf
 7. upload di file
 8. eventi
 9. creare un type estendendone uno esistente
 10. extensions
 11. validazione
9. Security
 1. configurazione
 2. autenticazione
 3. user provider
 4. firewall
 5. ruoli
 6. security nel controller
 7. security nel template
10. Deployment
11. Service container
 1. servizi. creare un servizio
 2. dependency injection. richiedere un servizio in un controller o in un altro servizio.
 3. autowiring
 4. tags
12. Internazionalizzazione
 1. traduzioni nella vista
 2. traduzioni nel controller
 3. traduzioni nei form
 4. pluralizzazione
 5. traduzioni con parametri
 6. tradurre contenuti sul db
 7. lavorare con il locale dall'utente, il locale e l'url
13. Invio e-mail
 1. configurazione
14. Logging
15. componente Workflow
 1. configurazione
 2. dump di un workflow
 3. utilizzo 4
16. Testing
 1. unit test
 2. functional test
17. Validation
 1. il validatore
 2. i validation constraint predefiniti
 3. creare un constraint personalizzato
18. Console
 1. creare un comando eseguibile da console (bash, etc)
 2. invocare un comando di console all'interno di un controller
19. bundle terzi
 1. ricercare un bundle

	2. installare un bundle
	3. WhiteOctoberPagerfantaBundle
20. creare un bundle distribuibile	
	1. configurazione file composer
	2. definire i servizi, le entity, le rotte, le risorse (viste, traduzioni, etc), i controller, etc
	3. definire la configurazione del bundle
	4. modificare la configurazione dei servizi definiti nel bundle