

Χρήση των frameworks σε custom εφαρμογές για το web

Σωμαράς Χρήστος,
Προγραμματιστής – Μηχανικός

Απόφοιτος MIS, 2001



Τι είναι τα frameworks

- **Θεωρία:** Τυποποιημένη ομάδα αρχών, πρακτικών και κριτηρίων για την αντιμετώπιση προβλημάτων
- **Πράξη:** Δομημένα (σε φακέλους) πακέτα κλάσεων – αρχείων κώδικα τα οποία μπορούν να χρησιμοποιηθούν για να φτιάξουμε εφαρμογές



Στόχος της χρήσης ενός framework

- Η παροχή κοινής δομής κλάσεων-αρχείων κώδικα στη λογική της επαναχρησιμοποίησης
- Ο προγραμματιστής δεν πρέπει να ανακαλύπτει τον τροχό



Χαρακτηριστικά

- Έχει τον έλεγχο λειτουργίας-εκτέλεσης της εφαρμογής του
- Έχει ξεκάθαρη και καθορισμένη συμπεριφορά
- Μπορεί να επεκταθεί με εξειδίκευση ή αλλαγή της λογικής των κλάσεων που χρησιμοποιεί ο χρήστης
- Δεν επιτρέπεται να αλλάξει ο κώδικας των βασικών αρχείων του framework



Τύποι

PRESENTATION LAYER / FRONT-END

User interfaces

Languages: HTML, CSS, Javascript...

Frameworks: Bootstrap, Foundation 3, Grids Systems...

APPLICATION LAYER / BACK-END

Logic and operation of the website

Languages: PHP, PYTHON, RUBY, JAVA...

Frameworks: Symfony, Django, Ruby On Rails, Spring...

DATA LAYER

Data persistence on databases

Technologies: MySQL, PostgreSQL, NoSQL...



MVC αρχιτεκτονική (application layer)

- Ξεχωριστές κλάσεις για το Model της ΒΔ
- Ξεχωριστές κλάσεις για τον Controller (αρχεία που φορτώνουν λειτουργία – υπηρεσία)
- Ξεχωριστές κλάσεις για το template του τρόπου εμφάνισης. Ο controller φορτώνει το template (View) για να εξυπηρετήσει την εμφάνιση των δεδομένων της λειτουργίας υπηρεσίας για την οποία υπάρχει



Παράδειγμα

| | | |
|---------------|---------------|------------------------|
| • CodeIgniter | Ruby on Rails | Symfony 1.4 |
| • Application | app | apps |
| – Controllers | - Controllers | -frontend |
| – Views | - Views | -modules |
| – Models | - Models | -products |
| – | | -actions -templates |



Παράδειγμα (MODEL)

- Τα αρχεία στο φάκελο Model είναι κλάσεις που σχετίζονται με του πίνακες της βάσης.
- Αν δηλαδή έχουμε ένα πίνακα PRODUCT τότε έχουμε και ένα αρχείο κλάσης product.
- Αυτό περιλαμβάνει μεθόδους σχετικές με τις περιπτώσεις – αντικείμενα του πίνακα και με ενέργειες που θα λάβουν χώρα (select, update κτλ)

π.χ. Model για πίνακα Products

Ας πούμε οτι υπάρχει μια μέθοδος
getAllProducts (μέσω ORM)



Παράδειγμα (CONTROLLER)

- Όλα ξεκινούν από το url. Το αρχείο routing είναι υπεύθυνο να οδηγήσει στον κατάλληλο controller (και σε συγκεκριμένο action αυτού).
- Στο αρχείο ενός controller υπάρχουν τα actions (index των προϊόντων, delete, create, update)
- Σε ένα action που εμφανίζει προϊόντα υπάρχει:
 - ο κώδικας που τραβάει το resultset από τη βάση (συνεργασία με το Model)
 - Ο κώδικας που φορτώνει το αντίστοιχο με το action View και δίνει σε αυτό τα data



Παράδειγμα (συνέχεια)

- και έστω το url που περνάω στον Browser είναι `www.shop.info/products`
- Από το routing file θα βρει το framework ποιο action τίνος controller θα τρέξει πχ το αρχείο `products.php` και συγκεκριμένα πχ το action `show`.
- Στον κώδικα του action `show` θα καλείται η `getAllProducts` και θα δημιουργείται `resultset` με αυτά
- Το `resultset` θα προωθείται από το action `show` στο κατάλληλο Template `show.php` από το φάκελο `View/products` το οποίο και θα φορτώνεται



Παράδειγμα (VIEW)

- Συνήθως το κάθε αρχείο στον View καλεί κάποιο css αρχείο για την εμφάνιση των δεδομένων.
- Τα css frameworks διαθέτουν κλάσεις οι οποίες έχουν ιδιότητες για responsive εμφάνιση του περιεχομένου αλλά και μορφοποίηση όλων των συστατικών μιας ιστοσελίδας (φόρμες, πίνακες, λίστες, κουμπιά κτλ)



Ουσία

- Ποιο είναι το καλύτερο framework από αυτά που έκανα χρήση?

symfony



Code Igniter

django

iOS

Cocoa Touch

