

02.Frameworks web

29 janvier 2018

Frameworks web

HE-Arc (DGR) 2016

Framework¹

- Fonctionnalités similaires pour de nombreuses applis
- Composants de haut-niveau réutilisables (faible couplage)
- Règles de codage et d'architecture
- Code sûr et efficace
- Facilite les tests et la gestion de projets complexes
- Utilisation de Design Patterns dès que possible
- Comportement par défaut
- Extensible
- Principe d'inversion de contrôle

Différences entre framework et library sur Stack Overflow² ou artima developer³.

#Design Patterns et webdev

- Inversion de contrôle (IoC⁴)
- Model View Controller
 - M : Accès aux données, logique métier
 - V : Templates des pages à générer
 - C : Orchestration, transfert des infos
- Front Controller
 - Traitement et dispatch des requêtes
 - (bootstrap, ré-écriture des URL, ...)

¹http://en.wikipedia.org/wiki/Software_framework

²<http://stackoverflow.com/questions/148747/what-is-the-difference-between-a-framework-and-a-library>

³<http://www.artima.com/forums/flat.jsp?forum=106&thread=152104>

⁴<http://martinfowler.com/bliki/InversionOfControl.html>

- Object Relational Mapping⁵
 - Active Record, Table Data Gateway, Data Mapper, ...
- UI Patterns⁶

MVC for webdev

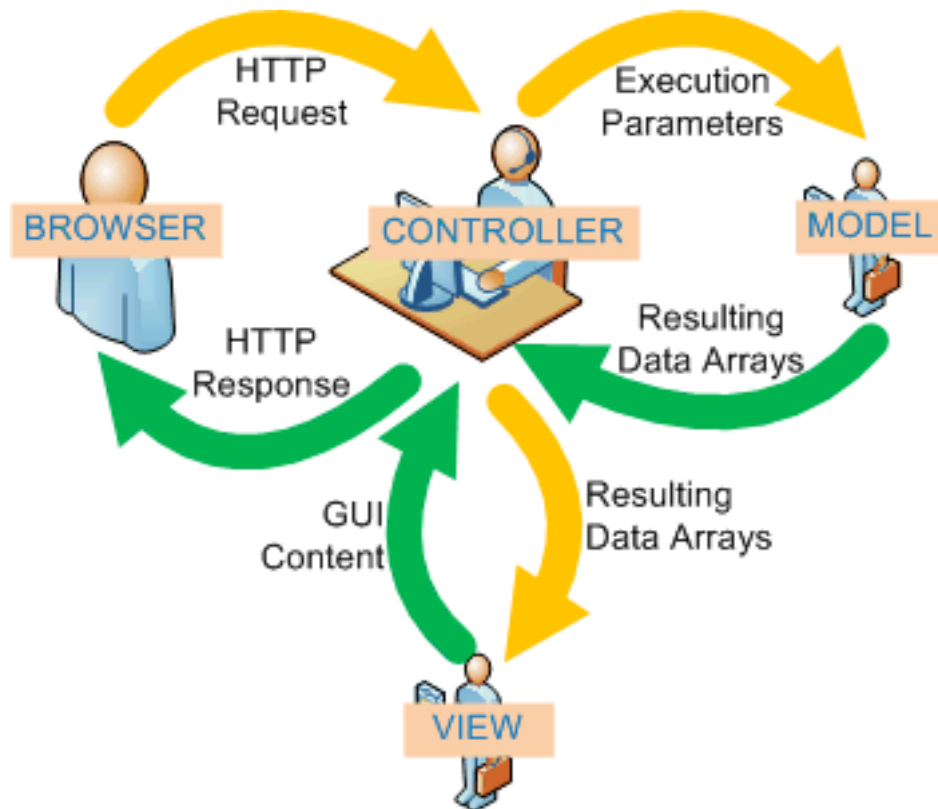


FIG. 1 : "MVC"

Conventions

- Nommage
 - Classes
 - Base de données
 - Fichiers et dossiers
- ROUTES : `http://app.host.tld/controller/action[/key/val]`

⁵<http://blog.mazenod.fr/2010/01/design-pattern-mvc-zoom-sur-la-couche-modele-dal-dao-orm-crud/>

⁶<http://ui-patterns.com/>

- Arborescence :
 - Imposée ou libre selon frameworks
 - Pas de code (minimum) sous la racine web
- Conventions obligatoires ou non, mais RECOMMANDEES dans tous les cas

Bonnes pratiques

- Heavy Model, Light Controller
- Don't Repeat Yourself
- You Ain't Gonna Need It
- Convention Over Configuration
- Keep It Simple and Stupid
- 12 factor app⁷ - fr⁸

Pretty (| smart | clean | formatted) URL

- Les URL doivent être explicites :
 - Manipulées par l'utilisateur
 - Utilisées pour le référencement
- Cohérence avec l'implémentation MVC :


`http://app.host.tld/controller/action[/key/val]`

- Le routage (routing)
 - Le Front Controller reçoit toutes les requêtes (URL rewriting)
 - Il les dispatche vers les contrôleurs

⁷<https://12factor.net/>


⁸<https://12factor.net/fr/>

Smart URL & SEO

 **seomoz.org**
Read SEOMoz. Rank Better

SEO Cheat Sheet: Anatomy of A URL

1
SEO-FRIENDLY URL

**1** **2** **3** **4** **5** **6** **7**
`http://store.example.com/topics/subtopic/descriptive-product-name#top`

1 Protocol
2 Subdomain
3 Domain
4 Top-Level Domain
5 Folders / Paths
6 Page
7 Named Anchor


Keyword Priority¹
Observed Google priority of keyword placement:
(1) Domain
(2) Subdomain
(3) Folder
(4) Path/Page

SEO Tips for URLs

- Use **subdomains** carefully. They may be treated as separate entities, splitting domain authority.
- Separate **path** & **page** keywords with hyphens ("-").
- **Anchors** may help engines understand page structure.
- Keyword effectiveness in URLs decreases as URL length and keyword position increases.¹

¹ SEOMoz correlational data (2009)

2
OLD DYNAMIC URL

**1** **2** **3** **4** **5** **6** **7** **7** **7**
`http://www.example.com/index.php?product=1234&sort=price&print=1`

1 Protocol
2 Subdomain
3 Domain
4 Top-Level Domain
5 Page / File Name
6 File Extension
7 CGI Parameters

Popular TLDs²

- .com** - commercial
- .net** - infrastructure
- .org** - non-profit
- .edu** - schools
- .info** - informational
- .biz** - small business
- .name** - personal sites

Popular ccTLDs*

- .cn** - China
- .de** - Germany
- .uk** - United Kingdom
- .nl** - Netherlands
- .eu** - European Union
- .ru** - Russian Federation
- .ar** - Argentina

Popular Extensions

- .htm** - Static HTML
- .html** - Static HTML
- .php** - PHP code
- .asp** - ASP code
- .aspx** - ASP.NET
- .cfm** - ColdFusion
- .jsp** - Java Code

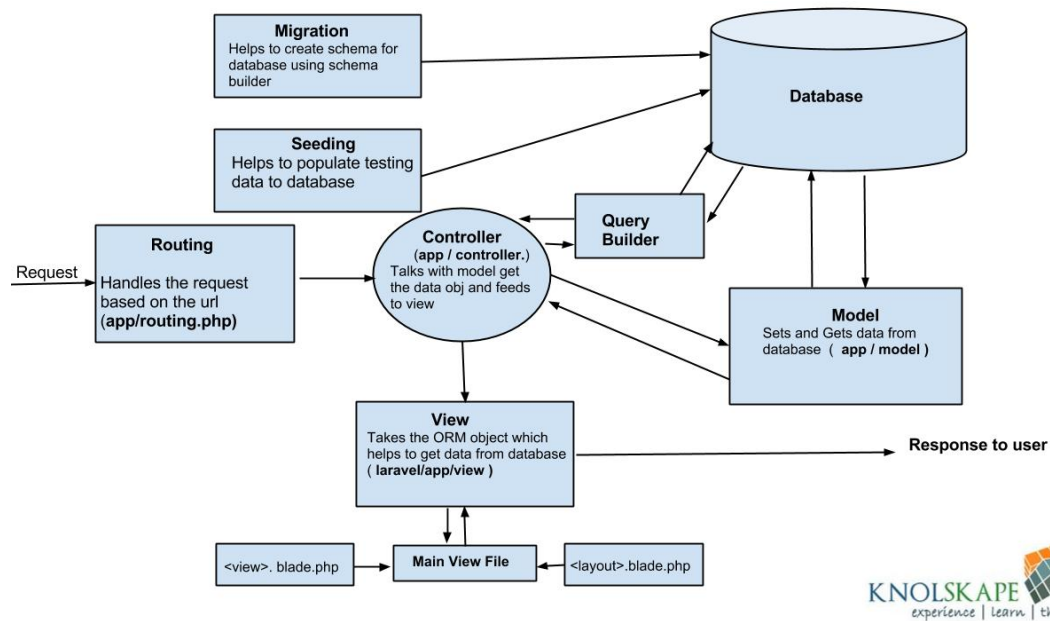
² Verisign domain report (2009) * ccTLD = Country Code TLD

© 2009 SEOMoz · www.seomoz.org · Read SEOMoz. Rank Better.

Autres Services

- Migrations : Evolutions de la structure de la BDD
- Tests
- Génération, validation et traitement de formulaires
- Authentification, Sessions, Permissions, Roles, ACL
- Pagination
- I18n
- Génération de code
- Mail
- Connecteurs aux webservices
- Captchas
- Loggers
- ...

Exemple d'architecture : Laravel



Performance

- Un framework web est lent :
 - Rendu d'une page nécessite de traverser tout le code
 - Pour chaque requête toute l'appli est chargée
 - Plus de code qu'une appli standalone
 - Plus de requêtes
- Solutions
 - Cache de pages, d'opcode
 - Jointures ORM, vues, procédures stockées
 - Outils d'optimisation : YSlow, page speed, mytop # Sources