Couverture

Page blanche

**Table des matières**

**I - Retour sur le web**

**1 - Qu’est-ce que le WEB ?**

**2 - Evolution du WEB et ses besoin**

**-Web1.0**

**-Web2.0**

**-Web3.0**

**2 - Les frameworks**

**1 - Le besoin menant à leurs apparitions**

**2 - Qu’est ce qu’un framework**

**3 - Leurs fonctionnement (global)**

**4 - Différents type de frameworks**

* **Framework d’infrastructure système**
* **Framework de développement mobile**
* **Frameworks orientés système de gestion de contenu**

**3 - Les frameworks Web et la Webteam**

**1 - Les frameworks webs et leurs spécificités**

* **Frameworks webs les plus populaires/demandés**
* **Fonctionnement des framework webs (design patterns, librairies, système de routage …)**

**2 - Frameworks utilisés aux seins de la Webteam**

* **framework interne**
* **Nouveau socle sous Symfony et Angular**

**3 - Quels frameworks web pourraient être utilisés et pourquoi ?**

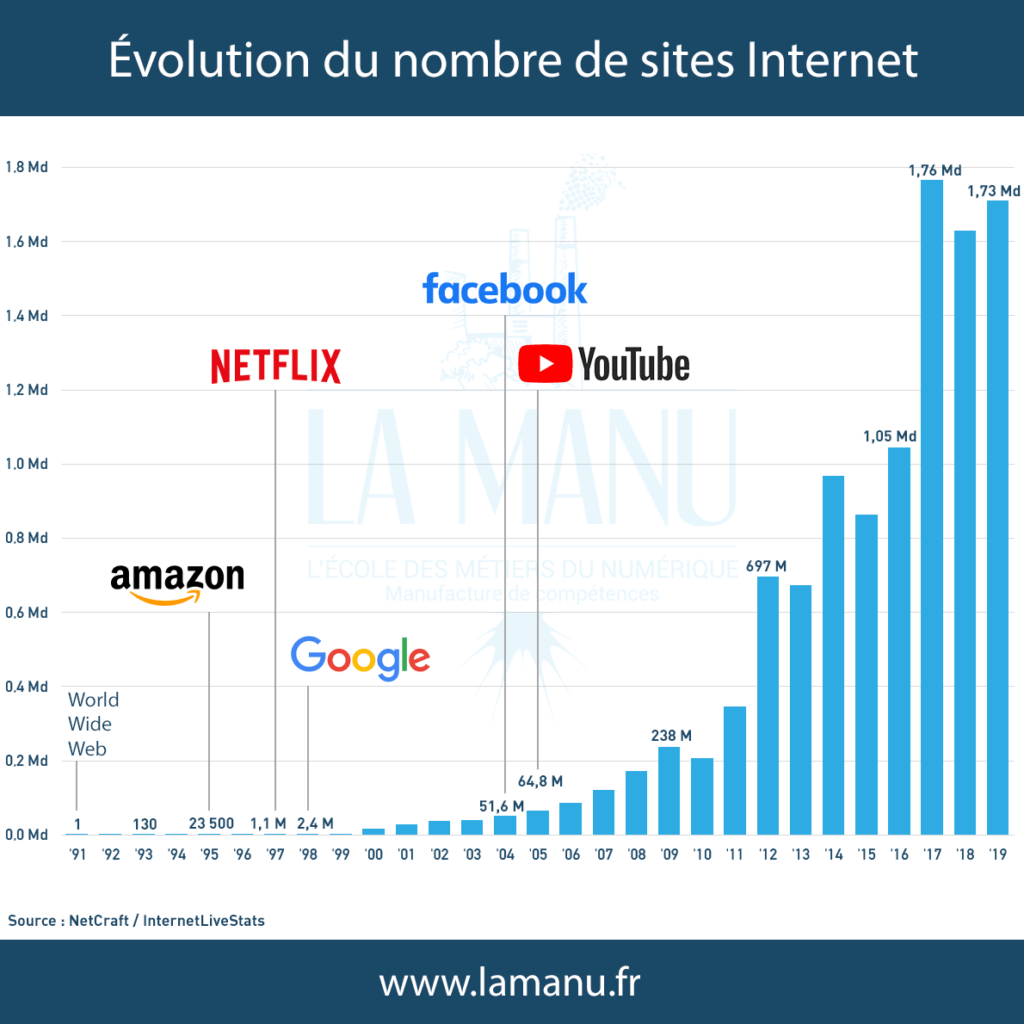
* **besoins et critères de sélection de la webteam**
* **quel framework pourrait correspondre aux besoins des applications webteam**

**I - Histoire du web**

**1.1 Qu’est ce que le Web**

Le World wide web, souvent désigné par le terme Web, traduit en français par “La toile d’araignée mondiale”, peut être considéré comme une application circulant sur l’internet, consultable depuis un navigateur faisant office d’interface afin d’y consulter et visionner du contenu publié sur le Web.

Le web ne doit donc pas être confondu avec l’internet, qui lui désigne le réseau mondial qui relie entre eux les ordinateurs, les serveurs ainsi que tous les équipements informatiques capables de se connecter.



On peut constater une évolution exponentielle du nombre de sites Web en circulation, avec en 2019, plus de 1.73 milliards de websites.

**1.2 Les origines du Web**

Le WEB a vu le jour au CERN (Conseil d’Etat) en 1989 sous le nom de World Wide Web. Il s’agissait d’un projet mené par Tim Berners-Lee, un chercheur britannique qui travaillait à cette période au CERN.

Ce projet était destiné à répondre à un manque de moyens techniques efficaces de transmission d’informations entre les chercheurs travaillant dans des universités et laboratoires du monde entier.

Le World Wide Web fait officiellement ses débuts à travers le monde en 1991, lorsque son premier site devient accessible hors du CERN.

Les 3 technologies utilisées dans le World Wide Web à ses débuts sont :

* Le protocole HyperText Transfer Protocol (**HTTP**) qui permet à un “internaute” de pouvoir accéder à un site web en établissant une connexion client-serveur.
* Le langage **HTML** (HyperText Markup Language), qui est un langage de programmation utilisant des balises pour représenter des éléments graphiques sur une page web.
* Les Uniformes Resource Identifier (**URIs**), qui permettent d’identifier une ressource physique ou abstraite sur internet. Les URL font partie des URI.

Un URI peut se décomposer en plusieurs parties :

**Scheme (le schéma)**: le protocole utilisé : **https**

**Authority (l’autorité)**: identifie le domaine : **example.org**

**Path (le chemin)**: le chemin d’accès à la ressource : **test/test1**

**Query (la requête)**: une action de requête : **search=test-question**

**Fragment (le fragment)**: un aspect partiel d’une ressource : **part2**

**1.2 Evolution du WEB et ses besoin**

L’évolution des technologies du WEB, ainsi que ce dernier, ont grandement évolué depuis leurs débuts.

**1.2.1 Le WEB1.0**

Le WEB1.0 désigne les premiers ‘websites’ qui ont vu le jour après 1989, les interactions avec ces sites étaient très limitées, ils ne faisaient en réalité office que de vitrine permettant aux propriétaires du sites d’exposé des informations et aux utilisateurs de les consulter. Ces sites étaient statiques et il n’y avait donc aucun échange entre les utilisateurs et le site web.

Les principaux créateurs de sites étaient des entreprises qui souhaitaient exposer leurs produits ou services à des utilisateurs faisant office de consommateur dans cet échange à sens unique.

Les protocoles employés par le WEB1.0 sont : URI, HTML et HTTP.

Son contenu est compréhensible par des humains uniquement puisqu’on y projette uniquement de l’information à l’aide d’HTML.

Des sites web basé sur les principes du WEB1.0 sont encore existants, on retrouve par exemple les sites “vitrines” permettant d’exposer une gamme de produits sans interaction avec les consommateurs.



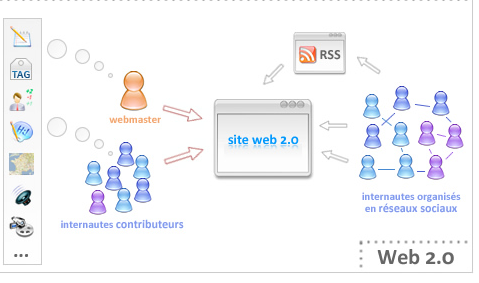
Figure 1 : Illustration du Web1.0

**1.2.2 Le WEB2.0**

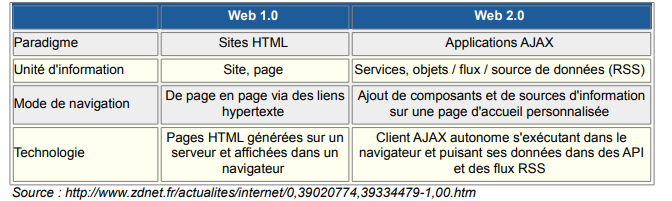
Le terme Web2.0 aurait été inventé par Dale Dougherty en 2003.

Les apports des sites webs 2.0 sont :

* La transformation du rôle d’internautes, les rendant contributeurs et non simples consommateurs d’informations.
* L’internaute a désormais la capacité de partager du contenu et d'interagir avec le contenu d’autres utilisateurs.
* Le **SaaS (Software as a Service)** a également vu le jours. Le SaaS est la mise à disposition d’un logiciel accessible aux utilisateurs via internet. L’entreprise cliente n’a donc pas besoin d’effectuer d’installation afin de pouvoir l’utiliser.



Pour pouvoir bénéficier de ces nouvelles technologies, les clients web ont également évolué de leurs coté :



Les nouvelles technologies AJAX, RSS, XML qui ont fait leurs apparitions avec le WEB2.0 sont les éléments qui ont facilité la circulation d’informations pour les utilisateurs.

**1.2.3 Le WEB3.0**

Le terme WEB3.0 a été introduit par Gavin Wood en 2014. Il est l’actuel dirigeant de la Fondation Web3, chargée de soutenir des projets technologiques décentralisés.

D’après les défenseurs du WEB3.0, le contenu présent en ligne est aujourd’hui contrôlé uniquement par certains grands groupes comme Meta, Google et Amazon, ce qui rendrait les données trop centralisées. Cette centralisation des données ainsi que des données personnelles des utilisateurs nuirait à ces derniers, le WEB3.0 permettrait que le contenu mis en ligne soit totalement décentralisé et appartiennent totalement à leurs auteurs.

Les web app deviendront désormais des dApp << Applications décentralisés>>, ce sont des applications construites à partir de blockchains.



Le WEB3.0 n’en est cependant encore qu'à ses débuts et sa future mise en place reste assez floue pour la plupart des personnes s'intéressant à ce concept.

**2 Les framework**

**2.1 Le besoin menant à leurs apparitions**

Les technologies évoluant sans cesse ainsi que la demande en logiciels et sites web, il a fallu trouver un moyen pour que les développeurs n'aient pas à “réinventer la roue” à chaque nouveau projet et puissent donc utiliser leurs codes.

**2.2 Qu’est-ce qu’un framework ?**

Le terme Framework a été inventé par Robert M. Carr en juillet 1984.

Un framework, en programmation, est un ensemble de composants logiciels structurels permettant à un développeur de faire appels à des fonctions, déjà incluses dans le framework, sans avoir à les réécrire lui-même.

* Réutiliser son code
* Incorporer un système de modularité
* Refactoriser son code
* Séparer le front du back
* Mettre en place un site web rapidement

**2.3 Leurs fonctionnement (global)**

Un framework est composé de plusieurs éléments :

* Le coeur, qui contient les fonctions de base du framework
* Les templates, qui permettent de mettre en place rapidement une interface (une page web)
* Les modules, qui permettent d’implémenter des fonctions supplémentaires et peuvent être réutilisés dans d’autres projets.
* Les scripts front-end, dans le cadre d’un framework web par exemple, qui permettent de personnaliser la mise en page.

**2.4 Les différents type de framework**

Il existe des framework adaptés à différents besoins logiciels. Nous allons explorer quelques exemples de framework répondant à ces différents besoins.

**2.4.1 Framework d’infrastructure système**

Les framework d’infrastructure système permettent de développer des outils de communications, des systèmes d'exploitation (OS) ou des interfaces graphiques.

Framework .NET est développé par Microsoft, il inclut un environnement d’exécution permettant de créer des applications Windows et des services Web. Il couvre une grande bibliothèque de classes appelée .NET Framework Class Library; qui est une combinaison de classes, d’interfaces et de types de valeur donnant accès aux fonctionnalités du système .NET fournit un environnement logiciel d’exécution exécuté dans un Common Language Runtime (CLR) incluant les services de gestion de la mémoire, la sécurité et la gestion des exceptions.

Qt est un framework cross-platform possédant une interface graphique. Il est constitué de nombreuses bibliothèques modulaires en C++ et fait appels à plusieurs API afin de faciliter le développement d’applications.

Comparatifs entre Qt et .NET

|  |  |  |
| --- | --- | --- |
|  | Qt | .NET Framework |
| Fonctionnalités | **Qtcore :** une libraire de base fournissant des conteneurs, de la gestion de thread ainsi que de la gestion d’évènement | **Gestion de la mémoire** dans différents langages |
|  | **QtGui** et **QtWidgets**, une boîte à outil graphique fournissant des composants graphiques pour personnaliser des applications Qt. | Bibliothèques de classes étendues |
|  | **QtWebkit** permettant d’utiliser des pages webs et des web apps dans une application Qt | **Exécution côte à côte :** .NET Framework  permet à plusieurs versions du CLR d’exister sur la même machine, on peut donc exécuter un programme qui a besoin d’une version différente de celle que le développeur utilise actuellement. L’exécution côte-à-côte s’applique aux groupes de versions .NET Framework 1.0/1.1, 2.0/3.0/3.5 et 4/4.5.x/4.6.x/4.7.x/4.8. |
|  | **QtXML:** permettant de parser du XML | **Infrastructures et technologies de développement :** .NET Framework inclut des bibliothèques pour des domaines spécifiques de développement d’applications, tels que ASP.NET pour les applications web, ADO.NET pour l’accès aux données, les Windows Communication Foundation pour les applications orientées service et les Windows Presentation Foundation pour Windows les applications de bureau. |
| Avantages | Le cross-platform des applications Qt ne nécessitant qu’une recompilation. |  |
| Désavantages | Des problèmes d'expériences utilisateurs peuvent survenir lorsque l’on souhaite transférer une application bureau vers une plateforme mobile (les templates sont peu adaptés aux versions mobiles) | Nécessite un OS windows |

**2.2.2 Framework de développement mobile (à compléter)**

Un framework de développement mobile permet de mettre en place des applications à destination d’OS mobile tels que iOS ou Android.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Flutter | Xamarin | Ionic | React Native |
| Langage de programmation | Dart | C# | HTML5,CSS et JavaScript | JavaScript |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**2.2.3 Frameworks orientés système de gestion de contenu**

Les Frameworks orientés système de gestion de contenu (SGC) ou Content Management System (CMS) en anglais, sont des outils permettant de gérer, concevoir et mettre à jour des applications mobiles ou des sites webs de manière dynamique, ou de gérer le stockage de documents en fonction de l’outil que l’on choisit.

On retrouve par exemple, pour la partie gestion de fichiers et documents, le framework Freedom-ECM, développé à l’aide du principe Modèle-vue-contrôleur.

Freedom-ECM permet :

* De classer chaque document dans des familles. Chaque famille correspond à un type de document (exemple : Factures, société, taxes …).
* Attribuer des droits aux utilisateurs tels que : consulter, éditer, supprimer, envoyer par mail, ouvrir le dossier, modifier le contenu du dossier, exécuter, créer, poster sur un forum.
* La création de profils, chacun contenant la liste des droits lui correspondant ainsi que la liste des utilisateurs possédant ce droit.
* Les profils peuvent être affectés à différents documents ou différentes familles
* Conversion des fichiers sous d’autres formats

Pour la partie CMS, on aurait par exemple Wordpress, qui contient des éléments de frameworks et permet :

* D’utiliser un système de Routing propre à Wordpress
* D’utiliser un ORM via l’installation de plugins
* La mise en place de templates pour les pages d’un site
* Gérer les évènements (appelés hook)

**3 - Les frameworks Web et la Webteam**

**3.1 - Les frameworks webs et leurs spécificités**

* **Fonctionnement des framework webs (design patterns, librairies, système de routage …)**

Certains frameworks web sont basés sur le design pattern modèle - vue - contrôleur (MVC).

Les fichiers d’un site sont donc découpés en 3 parties distinctes :

* La partie “modèle” contient des données pures, sans effectuer de traitement direct sur ces données.
* La partie “vue”, chargée de présenter les données à l’utilisateur.
* La partie “contrôleur” qui va être chargée d’alimenter la vue avec les données du modèle ainsi que d’effectuer les opérations sur ces données lorsqu’un événement est déclenché dans le programme, par exemple, un utilisateur qui clique sur un bouton affiché par la vue.
* **Framework web les plus populaires/demandés du moment (comparatifs)**

**Framework web back-end**

**Django**

**Laravel**

**Spring**

**(Ruby on Rails)?**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Laravel** | **Django** | **Spring** |
|  |  |  |  |
| **Templates** | **Laravel embarque un moteur de template “Blade”, offrant de l’héritage entre les templates et les différentes sections d’un projet.** | **Django possède des templates grâce aux DTL**  **(Django Template Language), ainsi qu’une alternante Jinja2.**  **Il est également possible d’utiliser des bibliothèques tierces.** | **Spring possède plusieurs bibliothèques de template comme JSP, Tiles et Velocity.** |
| **Tests** | **Laravel a un support de test intégrer avec PHPUnit.** | **Django utilise les modules unitest, basé sur PSL (Python standard lybrary). Cependant Djanon lui-même n’intègre pas de support de test.** | Spring FW has a built-in test support, a built-in test package, and a comprehensive set of temporary objects which are useful for web context, controller, and filter testing (https://spring.io) |
| **Support de plusieurs langages** | **Laravel ne supporte pas le multi-langage.** | **Django ne supporte pas le multi langage.** | **Spring supporte le multi-language (Java, Scala, Groovy...).** |
| **Plugins** | **Laravel propose des plugins permettant d’ajouter des fonctionnalités.**  **Ces plugins sont installé via une commande, dans un projet en particulier.** |  | **Spring ne permet pas l’ajout de plugin.** |
| **Adapté pour de grande applications** | **Selon Audurus(2017), Laravel ne serait pas adapté pour les grandes applications.** | **Djangon serait designé pour des grandes applications.** | **Selon Audurus(2017), Spring serait adapté aux applications existantes.** |
| **Scalabilité** |  |  |  |

**Framework web front-end**

**Angular**

**Vue.js**

**React**

**3 framework JS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Angular** | **Vue.js** | **React** |
| **Les imports JavaScript sont-ils suffisants pour démarrer un nouveau projet ?** | **On ne peut pas démarrer uniquement via l’aide d’un script.** | **Il est possible de travailler localement en téléchargeant Vue.JS depuis le site officiel. On peut également faire un démarrage rapide grâce à NPM (Node Package Manager), le package de manager de javascript.**  **Enfin, il est également possible d’ajouter un script de démarrage.** | **On peut se servir d’un script.**  **Il est cependant recommandé par le site officiel, de créer un workflow au préalable.** |
| **Est-ce que le framework génère de l’ “overhead”** | **Angular contient beaucoup d’add-ons. Il utilise également ses propres chargements de modules, dans son propre “langage”, ce qui va nécessiter des pipes, des déclarations, des modules …**  **L’optimisation est présente mais ne compense pas entièrement l’overhead.** | **Vue.js ne génère pas beaucoup d’overhead. Il permet d’activer ou désactiver certains éléments requis pour exécuter un projet.** | **React ne génère pas beaucoup d’overhead. Il permet de manipuler et contrôler la partie Vue de l’application, il permet également de désactiver les éléments dont le développeur n’a pas besoin.** |
| **Rendu côté-serveur** | **Angular génère des pages statiques dans le serveur, en utilisant le processus de “Server-Side Rendering” (SSR).**  **Il génère et envoie ces pages au navigateur en fonction de ses demandes. Angular peut également pré-généré des pages en tant que fichier HTML qui seront ensuite envoyée plus tard.** |  | **L’API React inclue un objet appelé ReactDOMServer, dont le rôle est d’afficher des composants sous la forme de code HTML.**  **React permet également l’utilisation de SSR mais n’a pas de support officiel pour ce dernier.** |
| **La capacité à écrire et à maintenir une grande quantité de**  **code complexe** | **Angular permet d’écrire et de maintenir une grande quantité de code complexe.**  **Plusieurs méthodes de bonnes pratiques**  **Pour gérer les fichiers et le code d’un projet sont disponible sur le site officiel d’Angular, dans la section Styleguide.**  **Angular inclus également un workflow à travers le CLI, permettant de mettre en place facilement un “building process”.** | **Il n’y a pas de “bonnes pratiques” définie par Vue.js, différentes typologies de nom de fichier sont autorisées par le framework, ce qui peut rendre difficile la maintenance.**  **On peut cependant créer un workflow grâce au CLI.** | **La communauté React a mis en place des “bonnes pratiques”, on peut également créer un Workflow grâce a package officiel “create-react-app".** |
| **Le développeur a-t-il besoin de bibliothèque tierce ( third-party)** | **Angular ne requiert pas beaucoup de bibliothèques tierces, il y a de nombreuses bibliothèques de la communauté et la majorité de ses bibliothèques sont déjà inclues dans Angular.**  **Par exemple,**  **Angular fourni déjà toutes bibliothèques nécessaires pour la création d’une “Single Page Application”** | **Vue.js possède une bibliothèque de routage officielle qui est maintenu et mis à jour régulièrement.**  **Vue.js ne possède pas de bibliothèque de validation de formulaire et doit donc faire appel à une bibliothèque tierce.** | **React a besoin de bibliothèques tierces pour la validation de formulaire, le routage ainsi que la gestion d’état.** |
| **Gestion de l’état des données** | **Angular comportant une bibliothèque officielle offrant un service de gestion de l’état des données pour les applications de petite et moyenne taille.**  **Si on souhaite réaliser une application de grande envergure, il faudra passer par une bibliothèque tierce.**  **Asynchrone.** | **Vue.js a une bibliothèque officielle Vuex, permettant la gestion de l’état des données pour petite, moyenne et grande applications.** | **React permet nativement une approche “top-down” des états cependant, l’approche de React n’est conseillée que pour les petite et moyenne applications. Si l’on souhaite réaliser une application plutôt large, il vaut mieux passer par les bibliothèques tierces Redux et Flux.** |

**3.2 - Frameworks utilisés aux seins de la Webteam**

* **Framework interne**
* **Socle sous Symfony et Angular**

Une application Symfony est décomposée en 3 parties distinctes :

- Une partie contenant les controllers, qui seront accessible facilement grâce au système de routage

- Les templates

- Les différentes classes d’entités

- Le dossier repository, qui contient les class apportant les fonctions permettant de récupérer des ensembles de données depuis la base de données

- Le dossier service, qui contient les class contenant la majeure partie de la logique de l’application

La partie Symfony n’intervient pas directement en Front End, elle fait office d’API qui fournit des données sous différents formats, notamment en JSON. Les templates Symfony ne seront donc, ici, pas utilisés. Elle gère les requêtes à la base de données et facilite ces dernières grâce à la librairie Doctrine qui permet de stocker des données sous formes d’objets, directement dans une table de base de données sans avoir à les décomposer en faisant appel à d’autres traitements.

La partie Front End est quant à elle réalisée sous Angular. Ce framework est orienté composants web. Un composant web est un fragment de l’application qui est autonome et indépendant du reste de l’application.

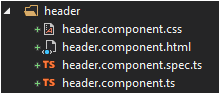
Un composant est constitué de 4 fichiers :

- Un fichier typescript qui contient les fonctions et la logique

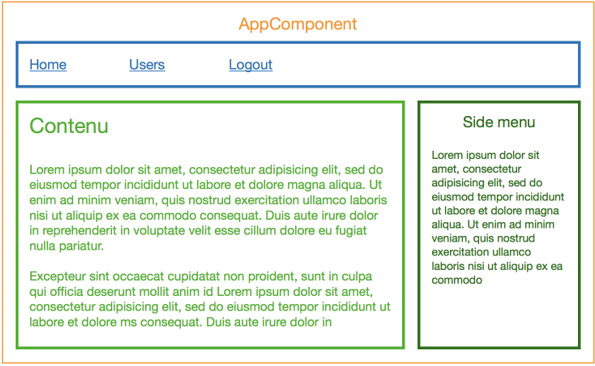
- Un fichier html qui sert à générer la template du composant

- Un fichier spec.ts, contenant les tests unitaires

- Un fichier css ou scss afin de customiser la template du composant



Par exemple, une barre de recherche peut être considérée comme un composant web distinct. Une application Angular peut donc contenir plusieurs composants web capables de communiquer entre eux. Angular intègre également le concept de Single Page Application, 12 des applications qui tiennent sur une seule page. On aura donc en général un AddComponent, qui sera le composant principal qui contiendra et affichera les autres composants de l’application grâce à la balise qui chargera dynamiquement les composants qui doivent être affichés à l’écran.



Ce chargement dynamique permet de réduire les chargements de pages puisqu’on ne recharge plus toute la page en entier mais uniquement certains composants affichés.

Les web components forment un standard qui n’est pas encore pris en charge sur tous les navigateurs, il faut donc utiliser des polyfill, qui sont des ensembles de fonctions simulant les fonctionnalités non prises en charge par ces navigateurs. La partie Angular s’occupe de générer l’interface que l’utilisateur verra lorsqu’il consultera une application et se charge de récupérer et afficher les données reçues par l’API sous Symfony.

Les applications Angular contiennent également des services. Les services sont des classes TypeScript dont l’instanciation est gérée par Angular. Il intègre le design pattern singleton puisqu’une seule instance de l’objet créé est utilisée dans l’ensemble de l’application. Ces services vont donc permettre de réutiliser du code dans tous les composants et services de l'application, ce qui facilite grandement l’échange de données.

C’est donc cette structure en service et composants qui permet à Angular de créer des applications modulaires facilement modifiables.

**3.3 - Quels frameworks web pourraient être utilisés et pourquoi ?**

**3.3.1 - Besoins et critères de sélection de la webteam**

**3.3.2 - Quel framework pourrait correspondre aux besoins des applications webteam**

**Résumé**

Au fil des années, les développeurs se sont rendu compte que de nombreuses fonctionnalités sont récurrentes dans les différents programmes et applications et qu’il serait donc bénéfique d’avoir un moyen de recréer ces fonctionnalités de manière simple,rapide et de la même manière, afin de gagner du temps, avoir un code facilement lisible et compréhensible par d’autres développeurs. Les frameworks ont vu le jour suite à ces besoins de rapidité et de respect des “best practice”.

Au cours de cette présentation nous allons d’abord nous intéresser au web dans sa globalité ainsi que son évolution au fil du temps, passant du web1.0 au web2.0 et plus récemment au web3.0, chacun apportant de nouvelles fonctionnalités permettant aux développeurs de pouvoir créer d’avantages d’applications variées. Nous allons ensuite analyser les différents frameworks qui ont vu le jour et leurs cas d’utilisation pour enfin nous demander quel framework serait adapté aux besoins de la webteam et ses différents sujets ( réécriture d’applications, nouveaux projets …).

**Anglais**

**http://blog.univ-angers.fr/web2huethuberdeau/presentation-du-web-2-0/#:~:text=Quels%20en%20sont%20les%20avantages,et%20interactif%20de%20pages%20Web.**

**http://www.uprt.fr/mesimages/fichiers-uprt/ge-gestion-documents-entreprise/ge-articleweb2.pdf**

<https://home.cern/fr/science/computing/birth-web#:~:text=Tim%20Berners%2DLee%2C%20chercheur%20britannique,s'%C3%A9changer%20des%20informations%20instantan%C3%A9ment>.

<https://lamanu.fr/evolution-du-nombre-de-sites-internet-de-1991-a-aujourdhui/#:~:text=En%202019%2C%20le%20nombre%20de,%C3%A0%20moins%20de%20200%20millions>.

<https://www.google.com/search?q=transitant+synonyme&oq=transitant+synonyme&aqs=chrome.0.0i512j0i22i30j0i10i22i30l2.3712j0j7&sourceid=chrome&ie=UTF-8#cobssid=s>

<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203593-web-definition/>

FRAMEWORKS

M. Kaluža, K. Troskot, B. Vukelić: Comparison of Front-End Frameworks for Web Applications... Zbornik Veleučilišta u Rijeci, Vol. 6 (2018), No. 1, pp. 261-282

<https://slides.kirstencassidy.com/#6>

<https://www.1min30.com/dictionnaire-du-web/framework>

<https://fr.wikipedia.org/wiki/Framework>

<https://en.wikipedia.org/wiki/Software_framework>

<https://www.techno-science.net/definition/1471.htm>

<https://fr-academic.com/dic.nsf/frwiki/661137#Principales_fonctionnalit.C3.A9sl>

<https://informagenie.com/2879/wordpress-cms-ou-framework-php/>

<https://www.ambient-it.net/react-native-vs-xamarin-vs-ionic-vs-flutter/#:~:text=La%20force%20de%20Ionic%20est,Une%20vitesse%20formidable>.

<https://mobiskill.fr/blog/conseils-emploi-tech/top-10-des-meilleurs-frameworks-de-developpement-mobile/>

<https://techmonitor.ai/what-is/what-is-a-framework-4945801>

<https://whatis.techtarget.com/definition/framework#:~:text=In%20general%2C%20a%20framework%20is,the%20structure%20into%20something%20useful>.

<https://hackr.io/blog/what-is-frameworks>

<https://wiki.qt.io/Qt_for_Beginners#:~:text=Qt%20uses%20a%20command%20line,do%20the%20job%20for%20us>.

<https://www.qt.io/product/framework>

<https://alternativeto.net/software/-net-framework/>

<https://docs.microsoft.com/fr-fr/dotnet/framework/get-started/#:~:text=NET%20Framework%20permet%20de%20r%C3%A9soudre,ex%C3%A9cuter%20sur%20la%20version%20de%20>.