
Architecture(s) et application(s) Web



Télécom SudParis

2019-2020

CSC4101 - Syllabus étudiants

Table des matières

1	Introduction	2
1.1	Objectif du présent document	3
1.2	Statut du document	3
1.3	Extraits de la fiche programme	3
2	Objectifs / Acquis d'Apprentissage Visés (AAV)	4
2.1	Développer une application Web, type site d'e-commerce sur la base d'un cahier des charges, en utilisant un <i>framework</i> PHP	6
2.2	Expliquer les rôles respectifs des traitements faits sur le client et le serveur	6
2.3	Appréhender les principes généraux d'ergonomie des interfaces Web	7
3	Modalités pédagogiques	7
3.1	Séquencement	9
3.2	Site du cours	12
3.3	Moodle	12
4	Séquences pédagogiques	12
4.1	Récapitulatif	14
4.2	TODO Suite récap à finir	17
4.3	TODO Cours Intégré : <i>Introduction, langage PHP, accès aux données avec l'ORM Doctrine</i>	19
4.4	TODO TA 0-1 : <i>Apprentissage PHP en autonomie</i>	19
4.5	DEMARRE TP 1 : <i>Bases d'une application Symfony : outils et modèle de données</i>	20
4.6	TODO TA 1-2 : <i>Complément sur l'accès aux données en base, et environnement PHP sur machine de travail</i>	20
4.7	TODO CM 2 : <i>Concepts de base, architecture appli Web 3 couches, client-serveur via HTTP</i>	20
4.8	TODO TP 2 : <i>Requêtes HTTP et exécution d'applications Web « en local »</i>	21
4.9	TODO TA 2-3 : <i>Environnement de développement</i>	21
4.10	TODO CM 3 : <i>Génération de pages</i>	22
4.11	TODO TP 3 : <i>Lancement du projet AgVoy</i>	22
4.12	TODO TA 3-4 : <i>HTML et gabarits pour le front-office de AgVoy</i>	23
4.13	TODO CM 4 : <i>Expérience utilisateur Web</i>	23
4.14	TODO TP 4 : <i>Interfaces habillées en CSS avec Bootstrap</i>	23
4.15	TODO TA 4-5 : <i>Habillage du front-office AgVoy avec CSS et Bootstrap</i>	24
4.16	TODO CM 5 : <i>Contrôleurs, interactions CRUD, formulaires</i>	24
4.17	TODO TP 5 : <i>Implémentation de CRUD sur ToDo</i>	25
4.18	TODO TA 5-6 : <i>CRUD pour la gestion de promotions dans le site public d'AgVoy</i>	25
4.19	TODO CM 8 : <i>Interface dynamique côté navigateur</i>	25
4.20	TODO TP 6 : <i>Rendu projets AgVoy à mi-parcours</i>	26
4.21	TODO TA 6-7 : <i>Auto-apprentissage JQuery ?</i>	26
4.22	TODO CM 7 : <i>Gestion contextuelle des formulaires</i>	27
4.23	TODO TP 7 : <i>Ajout utilisateurs et permissions, séparation front - back</i>	27
4.24	TODO TA 7-8 : <i>Utilisateurs et permissions dans AgVoy</i>	27
4.25	TODO CM 8 : <i>Formulaires et associations dans le modèle</i>	27
4.26	TODO TP 8 : <i>Implémentation formulaires entités nouvelles dans ToDo</i>	28
4.27	TODO TA 8-9 : <i>Ajout Programmations dans AgVoy</i>	28
4.28	TODO CM 9 : <i>Gestion de la sécurité, des erreurs :Sécurité:Erreurs:cm_9</i>	28
4.29	TODO TP 9 : <i>Formulaires dynamiques complexes</i>	28
4.30	TODO TA 9-10 : <i>Finalisation projet AgVoy</i>	28
4.31	TODO CM 10 : <i>Évolutions des architectures applicatives</i>	28
5	TODO Évaluation / Notation	28
6	Index	30

1 Introduction

1.1 Objectif du présent document

Le présent document vise à consigner les modalités de mise en œuvre du cours **Architecture et applications Web**.

Ces éléments sont destinés à l'équipe enseignante ainsi qu'aux étudiants, pour expliciter les objectifs et décrire les modalités pédagogiques adoptées.

Le détail des contenus pédagogiques de chacune des séquences pédagogiques est disponible dans des documents séparés.

1.2 Statut du document



Ce document est actuellement en cours de modification pour l'édition 2023-2024. Ne pas le diffuser en l'état.

1.3 Extraits de la fiche programme

Cette section reprend les éléments de la fiche standard TSP décrivant ce module. Ces éléments sont rappelés ici à des fins purement informatives. En cas de doute, le syllabus de la Formation Initiale est la seule référence applicable.

Architectures et applications Web

Code : CSC4101

ECTS : 3

Période : S7

Langue : Français

Organisation :

- Heures programmées / Charge Totale : 30 / 60
- Heures Cours/TP/CF1 : 10,5 / 18.5 / 1,5

Objectifs (Acquis d'Apprentissage Visés) :

- À l'issue de ce module, les étudiants de deuxième année seront capables de développer une application Web de type site d'*e-commerce* (une dizaine de pages), sur la base d'un cahier des charges fourni, en utilisant un *framework* PHP professionnel (Symfony).
L'application sera réalisée, en binôme, de façon incrémentale au fil des séances, en s'inspirant de versions successives d'une application exemple fournie. Elle devra permettre la saisie de données, et aura un comportement personnalisé en fonction du profil d'un utilisateur.
- ce développement sera effectué par la mise en œuvre des bibliothèques et outils du *framework* objet, afin d'exploiter des fonctions de sécurité, de présentation dans des pages HTML, pour s'approcher d'une application réaliste, de qualité professionnelle.
- les étudiants utiliseront les outils de mise au point du *framework* et du navigateur Web ;
- Les étudiants sauront expliquer les rôles respectifs des traitements faits sur le client et le serveur HTTP.
 - Ils sauront positionner les composants d'une application Web, dans une architecture en couches (multi-tiers).
 - Ils pourront expliquer comment fonctionnent les sessions applicatives dans un protocole où le serveur est sans-état.
- Les étudiants ont la liberté de personnaliser l'apparence des pages du site, ce qui permet d'appréhender les principes généraux d'ergonomie des interfaces Web (expérience utilisateur, accessibilité)

Compétences selon référentiel CDIO :

- 1 Connaissances disciplinaires et raisonnement
 - 1.3 Connaissances avancées en ingénierie : méthodes et outils
- 4 Les activités de l'ingénieur dans l'entreprise et dans le contexte sociétal
 - 4.3 Ingénierie des systèmes : conception et gestion
 - 4.3.2 Définir la fonction, le concept et l'architecture
 - 4.5 Réaliser
 - 4.5.3 Processus de réalisation logicielle
- 2 Compétences professionnelles et personnelles
 - 2.5 Éthique, déontologie et autres responsabilités
 - 2.5.1 Éthique et responsabilité sociale

Mots clés : Web, MVC, HTML, CSS, JavaScript, AJAX, DOM, PHP, HTTP, Symfony, ReST

Prérequis :

- Algorithmique et Programmation Objet
- Bases de données
- Shell Unix
- Réseau TCP/IP

Programme des séances :

1. *Introduction, langage PHP, accès aux données avec l'ORM Doctrine*
2. *Concepts de base, architecture appli Web 3 couches, client-serveur via HTTP*
3. *Génération de pages*
4. *Expérience utilisateur Web*
5. *Contrôleurs, interactions CRUD, formulaires*
6. *Interface dynamique côté navigateur*
7. *Gestion contextuelle des formulaires*
8. *Formulaires et associations dans le modèle*
9. *Gestion de la sécurité, des erreurs*
10. *Évolutions des architectures applicatives*

Evaluation

- Contrôle continu (projet)
- 1ère session = 1h30 sur table
- 2ème session = 1h30 sur table
- Note finale = suivant règlement des études : $\max((\text{noteCC} + \text{noteCF1})/2, \min(\text{noteCF2}, 10))$.

Responsables :

- Christian Bac (christian.bac@telecom-sudparis.eu)
- Olivier Berger (olivier.berger@telecom-sudparis.eu)

Intervenants :

...

2 Objectifs / Acquis d'Apprentissage Visés (AAV)

Nous définissons ici les Acquis d'Apprentissage Visé (AAV) pour ce cours, qui vont déterminer le dispositif pédagogique.

Les AAV sont décrits en détails dans les sous-sections suivantes :

- À l'issue de ce module, les étudiants de deuxième année seront capables de développer une application Web de type site d'*e-commerce* (une dizaine de pages), sur la base d'un cahier des charges fourni, en utilisant un *framework* PHP professionnel (Symfony).
L'application sera réalisée, en binôme, de façon incrémentale au fil des séances, en s'inspirant de versions successives d'une application exemple fournie. Elle devra permettre la saisie de données, et aura un comportement personnalisé en fonction du profil d'un utilisateur.
- ce développement sera effectué par la mise en œuvre des bibliothèques et outils du *framework* objet, afin d'exploiter des fonctions de sécurité, de présentation dans des pages HTML, pour s'approcher d'une application réaliste, de qualité professionnelle.
- les étudiants utiliseront les outils de mise au point du *framework* et du navigateur Web ;
- Les étudiants sauront expliquer les rôles respectifs des traitements faits sur le client et le serveur HTTP.
 - Ils sauront positionner les composants d'une application Web, dans une architecture en couches (multi-tiers).
 - Ils pourront expliquer comment fonctionnent les sessions applicatives dans un protocole où le serveur est sans-état.
- Les étudiants ont la liberté de personnaliser l'apparence des pages du site, ce qui permet d'appréhender les principes généraux d'ergonomie des interfaces Web (expérience utilisateur, accessibilité)

Ces AAVs seront référencés systématiquement dans la présentation des différentes séquences d'apprentissage.

2.1 Développer une application Web, type site d'*e-commerce* sur

Numéro	AAV 1
Code	AAV_REALISATION-APPLI-WEB

À l'issue de ce module, les étudiants de deuxième année seront capables de **développer une application Web** de type site d'*e-commerce* (une dizaine de pages), sur la base d'un cahier des charges fourni, en utilisant un *framework* PHP professionnel (Symfony). L'application sera réalisée, en binôme, de façon incrémentale au fil des séances, en s'inspirant de versions successives d'une application exemple fournie. Elle devra permettre la saisie de données, et aura un comportement personnalisé en fonction du profil d'un utilisateur.

Cet AAV principal est raffiné en AAV secondaires :

- le développement de l'application sera effectué par la **mise en œuvre des bibliothèques et outils du *framework* objet**, afin d'exploiter des fonctions de sécurité, de présentation dans des pages HTML, pour s'approcher d'une application réaliste, de qualité professionnelle.
- les étudiants utiliseront les **outils de mise au point** du *framework* et du navigateur Web ;

2.2 Expliquer les rôles respectifs des traitements faits sur le client et le serveur HTTP.

Numéro	AAV 2
Code	AAV_ARCHITECTURE-TROIS-TIERS

À l'issue de ce module, les étudiants de deuxième année sauront **expliquer les rôles respectifs des traitements** faits sur le client et le serveur HTTP.

Cet AAV principal est raffiné en AAV secondaires :

- Ils sauront positionner les composants d'une application Web, dans une **architecture en couches** (*multi-tiers*).
- Ils pourront expliquer comment fonctionnent les **sessions applicatives** dans un protocole où le serveur est sans-état.

2.3 Appréhender les principes généraux d'ergonomie des interfaces

Numéro Code	AAV 3 AAV_ERGONOMIE-INTERFACES
------------------------	-----------------------------------

À l'issue de ce module, les étudiants de deuxième année auront pu appréhender les **principes généraux d'ergonomie des interfaces Web** (expérience utilisateur, accessibilité), à travers la personnalisation de l'apparence des pages de l'application qu'ils auront développée.

3 Modalités pédagogiques

3.1 Séquencement

Les concepts du cours sont présentés dans l'ordre suivant au long des 10 semaines :

1. *Introduction, langage PHP, accès aux données avec l'ORM Doctrine*
2. *Concepts de base, architecture appli Web 3 couches, client-serveur via HTTP*
3. *Génération de pages*
4. *Expérience utilisateur Web*
5. *Contrôleurs, interactions CRUD, formulaires*
6. *Interface dynamique côté navigateur*
7. *Gestion contextuelle des formulaires*
8. *Formulaires et associations dans le modèle*
9. *Gestion de la sécurité, des erreurs*
10. *Évolutions des architectures applicatives*

Chaque semaine type, les séquences pédagogiques sont organisées en différentes activités :

1. séance de cours magistral (1h)
2. séance de Travaux Pratiques encadrés, à la suite du cours magistral (2h)
3. séquence de travail hors-présentiel, en autonomie d'ici la séance de cours suivante (3h)

Les tableaux 1 et 2 donnent plus de détails sur ce séquencement :

Table 1 – Séquencement des séances 1 à 5

N° séance	0	1 (3h)	2 (3h)	3 (3h)	4 (3h)	5 (3h)
Prés. 1h	<i>CM 1 (promo)</i> (1h) Introduction, langage PHP, accès aux données avec l'ORM Doctrine	<i>NO</i>	<i>CM 2 (1/2p.)</i> Concepts de base, architecture appli Web 3 couches, client-serveur via HTTP	<i>CM 3 (1/2p.)</i> Génération de pages	<i>CM 4 (1/2p.)</i> Expérience utilisateur Web	<i>CM 5 (1/2p.)</i> Contrôleurs, interactions CRUD, formulaires
Prés. 2h	<i>NO</i>	<i>TP 1</i> Bases d'une application Symfony : outils et modèle de données	<i>TP 2</i> Requêtes HTTP et exécution d'applications Web « en local »	<i>TP 3</i> Lancement du projet AgVoy	<i>TP 4</i> Interfaces habillées en CSS avec Bootstrap	<i>TP 5</i> Implémentation de CRUD sur ToDo
Hors prés.	hors-prés 0-1 Apprentissage PHP en autonomie	hors-prés 1-2 Complément sur l'accès aux données en base, et environnement PHP sur machine de travail	hors-prés 2-3 Environnement de développement	hors-prés 3-4 HTML et gabarits pour le front-office de AgVoy	hors-prés 4-5 Habillage du front-office AgVoy avec CSS et Bootstrap	hors-prés 5-6 CRUD pour la gestion de promotions dans le site public d'AgVoy

Table 2 – Séquencement des séances 6 à 10

N° séance	6 (3h)	<i>break</i> (2 se- maines)	7 (3h)	8 (3h)	9 (3h)	10 (3h)
Prés. 1h	<i>CM 6 (1/2p.)</i> Interface dy- namique côté navigateur	—	<i>CM 7 (1/2p.)</i> Gestion contex- tuelle des for- mulaires	<i>CM 8 (1/2p.)</i> Formulaires et associations dans le modèle	<i>CM 9 (1/2p.)</i> Gestion de la sécurité, des erreurs	<i>CM 10 (1/2p.)</i> Évolutions des architectures applicatives
Prés. 2h	<i>TP 6</i> Rendu projets AgVoy à mi- parcours	—	<i>TP 7</i> Ajout utilisateurs et permissions, séparation front - back	<i>TP 8</i> Implémentation formulaires en- tités nouvelles dans ToDo	<i>TP 9</i> Formulaires dy- namiques com- plexes	
Hors prés.	hors-prés 6-7 Auto- apprentissage jQuery ?	<i>NO</i>	hors-prés 7-8 Utilisateurs et permissions dans AgVoy	hors-prés 8-9 Ajout Program- mations dans Ag- Voy	hors-prés 9-10 Finalisation projet AgVoy	<i>NO</i>

les en-têtes des cases en italique sont des liens pointant vers les sections correspondantes du document

3.2 Site du cours

Les ressources pédagogiques sont disponibles pour les étudiants dans le site du cours :
<https://www-inf.telecom-sudparis.eu/COURS/CSC4101-2018/>

3.3 Moodle

Le site Moodle utilisé de façon très limitée, car on utilise principalement le site du cours.

4 Séquences pédagogiques

Le cours est organisé en 10 séquences correspondant à peu près à 10 semaines calendaires, structurées de façon identique (à l'exception de la première séquence).

4.1 Récapitulatif

Syllabus étudiants

Poly étudiant

N° séance	1	*2 *	*3 *	*4 *	*5 *	
Travail Autonome 3h	CM1 promo (1h 30)	Cours (1h)	Cours	Cours	Cours	
	Introduction, langage PHP, accès aux données avec l'ORM Doctrine	Concepts de base, architecture appli Web 3 couches, client-serveur via HTTP	Génération de pages	Expérience utilisateur Web	Contrôleurs, interactions CRUD, formulaires	
	(présentiel)		- Génération des pages	- Expérience utilisateur	- Sessions	
	- intro du cours	- PHP	- HTML	-	- Doctrine	
	- concepts clés	- Doctrine	-			
	-	- Symfony	-			
	-	- HTTP	-			
	-		-			
	hors-prés 0-1 (1h30)	hors-prés 1-2 (2h)	hors-prés 2-3	hors-prés 3-4	hors-prés 4-5	
Travail Pratique Tutoré 3h	- Prise en main env PHP	- Prise en main env PHP (suite)	- Twig			
	- Apprentissage PHP	-	-			
	TP 1 3h	TP 2 (3h)	TP 3 (3h)	TP 4 (3h)	TP 5 (3h)	
	- Mini allo-ciné	- Ajout interface	- Travail gabarits	- Bootstrap	- Doctrine	

4.2 TODO Suite récap à finir

N° séance	6	7	8	
Travail Autonome 3h	<i>Cours</i> Interface dynamique côté navigateur - - - - hors-prés 5-6 -	<i>Cours</i> Gestion contextuelle des formulaires - - - hors-prés 6-7 - -	<i>Cours</i> Formulaires et associations dans - hors-prés 7-8 -	CSC4104 2023-2024
Travail Pratique Tutoré 3h	<i>TP 6 (3h)</i> - CRUD - - -	<i>TP 7 (3h)</i> - Formulaires - - - -	<i>TP 8 (3h)</i> - Formulaires avancés - -	

:end :

4.3 TODO Cours Intégré : *Introduction, langage PHP, accès aux*

Durée 1h ?

Objectifs Ce cours intégré introduit les **objectifs du module** et le **dispositif pédagogique**. Il aborde les éléments suivants :

- Intro cours
 - Langage PHP
 - Début accès aux données avec l'ORM Doctrine
- Le but est de faciliter le démarrage des premiers TP.

4.3.1 TODO Matériel pédagogique

Référence :

TODO :

- file:///cours-toile/
- file:///cours-histoire-toile/
- Polycopié étudiants CM (PDF A4)
- Polycopié étudiants CM (PDF A4)
- Polycopié étudiants CM (PDF A4)

4.4 TODO TA 0-1 : *Apprentissage PHP en autonomie*

Durée 3h

Objectifs — Apprentissage PHP en autonomie via <https://www.learn-php.org/fr/>
 — Lecture : compléments PHP

Cette séquence de travail hors-présentiel a pour but de présenter la **syntaxe du langage PHP** dans un contexte de programmation impérative et objet (simple). Les programmes d'application en PHP étudiés ne porteront pas sur des spécificités d'un environnement de programmation Web, mais plutôt sur des interactions de type traitement et affichage sur la sortie standard, assez similaires à ceux vus en première année. Cette séquence doit donc faciliter la **transition depuis le langage Java**, en restant dans une architecture applicative déjà connue. Les séquences ultérieures introduiront l'environnement d'exécution PHP sur la machine locale, ainsi que les concepts de la programmation pour les applications Web.

4.4.1 Matériel pédagogique

Site externe existant [learn-php.org](https://www.learn-php.org/), qui fournit un tutoriel intégrant un environnement interactif en ligne, utilisable dans le navigateur Web.

Aucune installation d'outil (autre qu'un navigateur Web) ne sera nécessaire pour ce faire.

Note : La traduction en français du tutoriel [learn-php.org](https://www.learn-php.org/) a été contribué par nos soins¹.

- référence : <http://www-inf.telecom-sudparis.eu/COURS/CSC4101-2019/hp-0-1/index.html>

TODO Vidéo 2.6. Technologie PHP

1. HP n°0-1 - Apprentissage PHP en autonomie Préparation du TP n°1

3.1. Table des matières 3.2. Objectif de cette séquence 3.3. Étape 1 : Généralités sur le langage PHP (15') 3.4. Étape 2 : Apprentissage via un tutorial PHP en ligne (2h) 3.5. Étape 3 : Compléments sur le langage et les outils PHP (45') 3.6. Auto-évaluation 3.7. Aller plus loin

1. Cf. <https://github.com/olberger/interactive-tutorials/commits/french-php>

4.5 DEMARRE TP 1 : *Bases d'une application Symfony : outils et*

Durée 2h

Objectifs Cette séance permet de familiariser les étudiants avec l'**environnement de programmation PHP** et de mise au point basé sur **Symfony**.

Plus faire la transition avec Algo et programmation en Java de 1A

L'application Symfony écrite en PHP objet fournie est étudiée.

Elle fonctionne en dehors du contexte du Web, en interagissant avec la base de données, à travers une interface en console utilisée en **ligne de commande**.

4.5.1 TODO Matériel de cours

Cette séance de travaux pratiques permettra de commencer la mise en œuvre des concepts abordés dans le cours, au sein d'une **application PHP utilisable en ligne de commande (console)**, appelée , qui fera le lien avec le TP de fin de module CSC3101

L'environnement de travail nécessaire sera :

- l'environnement Unix des comptes DISI des étudiants, en salles de travaux pratiques,



Configuration logicielle des machines personnelles

Même si les technologies utilisées pour les séquences d'apprentissage actif (programmation PHP avec le *framework* Symfony) sont théoriquement portables et utilisables sur différents systèmes d'exploitation, des différences existent qui rendent très difficile le support au vu des différentes variantes possibles, et pour un module en promo entière. L'équipe pédagogique ne pourra assurer le support relatif aux outils que dans le contexte technique de l'un des deux environnements ci-dessus, qui seront homogènes en terme de version du système d'exploitation ou des outils installés.

Il est fortement recommandé aux étudiants souhaitant utiliser leurs machines personnelles de configurer un environnement à base de machine virtuelle, pour s'inscrire dans ce cadre (ou de travailler dans un système d'exploitation identique à celui proposé par la DISI^a).

a. Typiquement en Fedora 28 à la date d'écriture du présent document

- référence : Support du TP 1

4.6 TODO TA 1-2 : *Complément sur l'accès aux données en base*

Durée 3h

Objectifs — Installer une environnement de travail en mode BYOD (optionnel)

- Découverte des fonctionnalités de l'ORM Doctrine réalisant l'interface avec la BD
- Amélioration de l'application PHP initiée en TP 1

4.6.1 TODO Matériel pédagogique

L'application PHP en console qui aura été initiée en TP n°1, sera complétée.

Cette application intègre déjà des fonctions gérant la persistance des données dans un fichier local, avec une mise en œuvre réalisée avec le SGBDR léger SQLite et SQL.

4.7 TODO CM 2 : *Concepts de base, architecture appli Web 3 cou*

Durée 1h

Objectifs Cette séance présentera les grands concepts structurants de **la Toile** (*World Wide Web*), les fondements de l'**architecture** des applications Web, à savoir une architecture **client(s)-serveur**, **multi-couche**, avec des traitements mis en œuvre à la fois sur un **serveur** et des **clients** qui communiquent via le **protocole Hyper-Text Transfer Protocol (HTTP)**. On explicitera la façon dont les interactions entre un client HTTP et le serveur Web ont permis de passer d'une consultation de documents Hypertextes sur la Toile à la mise en œuvre d'applications interactives.

4.7.1 TODO Matériel de cours

- Support de présentation CM Concepts de base, architecture appli Web 3 couches, client-serveur via HTTP (« slides » HTML)
- Polycopié étudiants CM (PDF A4)

4.8 TODO TP 2 : *Requêtes HTTP et exécution d'applications Web*

Durée 2h

Objectifs Cette séance permet de comprendre le rôle des composantes client et serveur pour la mise en œuvre d'une application Web en observant le dialogue des **requêtes** et **réponses** HTTP. On utilisera pour cela les fonctionnalités de différents outils de mise au point utiles au développeur Web. On explicitera le rôle de certaines composantes principales du protocole HTTP.

Découverte de la bibliothèque *Doctrine* et révisions sur la persistance des données applicatives

Cette séquence de travail hors-présentiel sera l'occasion de réviser les connaissances acquises en cours de **bases de données** en première année, et de découvrir le rôle et le fonctionnement de base de la bibliothèque *Doctrine* pour assurer la **persistance d'un modèle de données** en PHP.

4.8.1 TODO Matériel de cours

On utilisera les outils suivants :

- clients HTTP :
 - **navigateur Web : inspecteur (réseau)** intégré aux outils du développeur
 - commande **cURL**
- serveurs HTTP :
 - site externe sur la Toile
 - serveur fonctionnant en local via le **serveur HTTP intégré** dans l'environnement PHP/Symfony

On s'appuiera sur le code de l'application fil-rouge `ToDo` déjà étudiée, pour le mettre en œuvre cette fois dans un nouveau contexte d'exécution, côté serveur HTTP. Après avoir exécuté le code PHP en ligne de commande dans les séquences précédentes, on expérimente cette fois avec l'**environnement de tests de Symfony** permettant d'invoquer une **exécution PHP** en réponse aux requêtes HTTP.

- référence : Support du TP 2

4.9 TODO TA 2-3 : *Environnement de développement*

Durée 3h

Objectifs

mise au point environnement de travail pour projet qui vient dans les phases de travail HP sur le projet

auto-apprentissage HTML-5-? (en HP 3-4?)

4.9.1 TODO Matériel de travail

— référence : <http://www-inf.telecom-sudparis.eu/COURS/CSC4101-2019/hp-2-3/index.html>

1. HP n°2-3 - Twig en autonomie

8.4. Étape 1 : Auto-apprentissage des langages Twig 8.5. Auto-évaluation

4.10 TODO CM 3 : Génération de pages

Durée 1h

Objectifs L'objectif de cette séance sera de présenter le standard **HTML** et la façon dont il peut être produit pour réaliser des **interfaces** pour les applications Web. On présentera le composant **Twig** utilisé dans l'environnement Symfony pour **générer les pages** HTML côté serveur à partir d'un système de **gabarits** (*templates*).

4.10.1 Matériel de cours

- Support de présentation CM Concepts de base, architecture appli Web 3 couches, client-serveur via HTTP (« slides » HTML)
- Polycopié étudiants CM (PDF A4)
- Polycopié enseignants séance 1 (PDF A4)

4.11 TODO TP 3 : Lancement du projet AgVoy

Durée 2h

Objectifs Cette séance permettra d'expérimenter l'application des concepts présentés dans le cours qui précède, à travers la mise en œuvre d'**interfaces HTML** pour une application Symfony avec **Twig**, sur la base de l'application fil-rouge ToDo. On présentera également le **projet de développement** d'une application Web de site d'agence de voyages (AgVoy), qui sera réalisé en binôme à partir de cette séance, et jusqu'à la fin du cours, et qui fera l'objet d'un rendu noté (note de contrôle continu).

4.11.1 TODO Matériel de cours

- Cahier des charges de l'application de site d'agence de voyages AgVoy (cf.

Cahier des charges d'AgVoy

Les étudiants réaliseront une application représentative des fonctionnalités d'un site Web d'*e-commerce* classique, au fur et à mesure des séances du cours, en partant des spécifications d'un cahier des charges fourni.

Ce site correspondra aux besoins d'une agence de voyages, et portera sur deux grands modules fonctionnels :

- un **site public « front-office » de type catalogue** permettant aux clients de l'agence de consulter les voyages proposés à la vente ;
- une portion du site **réservée au personnel de l'agence, de type « back-office »** permettant de planifier la constitution des voyages, gérer les voyages ; ainsi que d'autres fonctions utiles à l'agence.

La réalisation de ce projet de développement sera effectuée en binômes, dans certaines séances de travaux pratiques et surtout lors des séquences hors-présentielles. Un rendu intermédiaire sera effectué à mi-parcours. Un rendu final fera l'objet d'une évaluation dont la note comptera pour la note de contrôle continu du module.

Le projet sera réalisé en repartant d'une base standard d'application Symfony, en intégrant à chaque séance les fonctionnalités correspondantes à chaque séquence de cours,

et en s'inspirant du code de l'application fil-rouge `ToDo` qui sera étudié en séances de TP. Chaque projet sera original et propre au binôme des auteurs. Toute tentative de plagiat entre binômes fera l'objet d'une sanction d'ordre disciplinaire.

— référence : Support du TP 3

4.12 TODO TA 3-4 : *HTML et gabarits pour le front-office de AgVoy*

Durée 3h

Objectifs Réalisation des **premières pages HTML** du *front-office* d'AgVoy avec Twig

Les étudiants travailleront en binômes sur le premier module du site AgVoy de leur projet, dénommé « *front-office* », qui jouera le rôle d'une vitrine pour la **consultation du catalogue** des circuits proposés à la vente par l'agence de Voyage.

Ce premier module est peu interactif et le contenu des pages relativement statique, **généré à partir du contenu d'une base de données** préalablement remplie, contenant les circuits de l'agence (utilisée en lecture seulement dans le code de l'application).

L'objectif principal sera d'acquérir la maîtrise des composants de Symfony permettant la génération des pages, en particulier le composant Twig permettant de standardiser la structure des pages du site avec un **jeu de gabarits**.

4.12.1 TODO Matériel de travail

— référence : <http://www-inf.telecom-sudparis.eu/COURS/CSC4101-2019/hp-3-4/index.html>

4.13 TODO CM 4 : *Expérience utilisateur Web*

Durée 1h

Objectifs Cette séance portera sur différents aspects de l'**expérience utilisateur** dans l'interaction avec les applications Web.

On présentera les enjeux généraux de l'**ergonomie** des interfaces utilisateur ainsi que les problématiques d'**accessibilité**.

On détaillera les fonctionnalités principales, dans cette optique, pour la construction et l'**habillage des pages HTML**, que sont les standards **Document Object Model (DOM)** et **Cascading Style Sheets (CSS)**.

4.14 TODO TP 4 : *Interfaces habillées en CSS avec Bootstrap*

Durée 2h

Objectifs Cette séance permettra d'expérimenter la mise en place d'un **habillage graphique des pages** de l'application fil-rouge `ToDo` en utilisant le standard CSS.

On utilisera le composant Bootstrap pour ce faire, en étudiant une proposition de structuration des gabarits de l'application fournie. Bootstrap permettra notamment de doter l'application d'une présentation réactive compatible avec la consultation sur mobile.

Différents **outils de mise au point** seront mis à contribution pour ce faire, comme la barre d'outils Symfony.

Les étudiants commenceront ensuite la mise en place de l'habillage de leur **front-office AgVoy** en CSS avec Bootstrap, qui sera poursuivie en hors-présentiel.

4.14.1 TODO Matériel de cours

- référence : Support du TP 4

4.15 TODO TA 4-5 : *Habillage du front-office AgVoy avec CSS et*

Durée 3h

Objectifs

- Auto-apprentissage de CSS et Bootstrap ;
- Suite de la réalisation d'une interface du front-office d'AgVoy, en intégrant une mise en forme originale.

Chaque binôme proposera une mise en forme originale en CSS, de sa variante du site d'AgVoy, en mettant en œuvre Bootstrap (très fortement recommandé, ou à défaut un framework CSS équivalent offrant une mise en forme réactive).

Les étudiants travailleront à une **intégration** aussi harmonieuse que possible des **gabarits Twig** et de la **structure de la cascade des styles** CSS, afin de soigner la modularité et l'évolutivité du site (par exemple dans l'optique du travail sur le back-office qui interviendra ultérieurement).

4.15.1 TODO Matériel de travail

- référence : <http://www-inf.telecom-sudparis.eu/COURS/CSC4101-2019/hp-4-5/index.html>

1. CM 5 : Interactions CRUD et sessions

15.1. Objectifs de cette séquence 15.2. Applications Web interactives 15.3. Soumission de données à une application 15.4. Gestion des formulaires HTML 15.5. Sessions applicatives 15.6. Sessions PHP 15.7. Take away 15.8. Annexes

1. CM 8 : Formulaires et associations dans le modèle

23.1. Retour sur l'ORM Doctrine 23.2. Formulaires améliorés 23.3. Upload d'images 23.4. Take Away

4.16 TODO CM 5 : *Contrôleurs, interactions CRUD, formulaires*

Durée 1h

Objectifs Cette séance abordera les premiers mécanismes permettant de rendre une **application Web interactive**, par exemple pour la mise en œuvre d'API Web de type Create Retrieve/Read Update Delete (CRUD).

Jusqu'à cette séance, le comportement des applications étudiées est essentiellement sans état (*stateless*, le serveur ne garde pas la trace des interactions précédentes avec un client donné), et le comportement de l'application est principalement en lecture seule de la base de données sous-jacente.

À partir de cette séance, l'application Web se dote de fonctionnalités permettant d'interagir avec les données de l'application, afin de reproduire, dans l'architecture clients-serveur, des interactions équivalentes à celles d'autres architectures applicatives (services Web, ...).

À cette étape, il reste encore à décrire des fonctionnalités essentielles des applications Web réelles (permissions, interfaces utilisateur, ...).

Mais on commence à avoir couvert un ensemble substantiel de concepts permettant de comprendre l'**architecture Representational State Transfer (ReST)** telle qu'elle a été imaginée dès les années 1990 pour la mise en œuvre d'applications dans le contexte du Web.

Les étudiants sont capables de positionner les 3 couches principales de l'**architecture multi-couches** classique : client, serveur de traitements, gestionnaire de base de données.

4.17 TODO TP 5 : *Implémentation de CRUD sur ToDo*

Durée 2h

Objectifs Les étudiants enrichiront l'application de gestion de tâches `ToDo` pour que l'interface Web de l'application permette d'interagir avec les fonctions du modèle de données existant qui n'étaient auparavant disponibles qu'en local en ligne de commande (cf. 4.6).

Cela permettra de découvrir des fonctions du *framework* Symfony permettant de « câbler » des gestionnaires de requêtes HTTP avec le modèle de données Doctrine, pour permettre des **interactions en lecture/écriture** de type CRUD.

À l'issue de cette séance, les étudiants sont capables de mettre en œuvre les mécanismes de base présents dans une application accessible en contexte Web, typiquement pour des API ReST de services Web de type CRUD.

4.17.1 TODO Matériel de cours

— référence : Support du TP 5

1. TP n°8 - Implémentation de formulaires pour entités liées Fonctionnalités utiles de Symfony pour accélérer prototypage et développement sur le modèle des données

24.1. Table des matières 24.2. Introduction 24.3. Étape 1 : Gestion d'associations 1-N dans Doctrine et Symfony 24.4. Étape 2 : Ajout d'une fonction de mise en ligne d'images 24.5. Évaluation

4.18 TODO TA 5-6 : *CRUD pour la gestion de promotions dans le*

Les étudiants poursuivent leur projet d'agence de voyage en intégrant à leur rythme les différentes fonctionnalités présentées en cours magistral et en travaux pratiques.

À cette étape, on doit normalement pouvoir passer à la mise en œuvre d'un mécanisme de **consultation et modification (CRUD) de promotions** destinées aux visiteurs du site (avec un modèle de données simple de type « blog »).

On utilisera les outils de **générateurs de code** du modèle de données Doctrine et de CRUD présents dans Symfony, pour faciliter l'implémentation.

4.18.1 TODO Matériel de travail

— référence : <http://www-inf.telecom-sudparis.eu/COURS/CSC4101-2019/hp-5-6/index.html>

1. HP n°5-6 - Suite du projet : CRUDs du projet AgVoy Travail sur des contrôleurs CRUD du projet AgVoy - Refactoring du code si nécessaire

17.1. Table des matières 17.2. Travail hors-présentiel 17.3. Étape 1 : Finalisation du fonctionnement du CRUD dans le back-office 17.4. Étape 2 : Refactoring du code généré vers le back-office (optionnel)

4.19 TODO CM 8 : *Interface dynamique côté navigateur*

Durée 1h

Objectifs Cette séance introduit les mécanismes permettant d'implémenter certaines fonctionnalités des applications Web **côté client**, dans le navigateur Web. On présentera les technologies JavaScript comme JQuery, qui seront mises en œuvre par exemple pour rendre l'interface des applications plus dynamique et améliorer l'expérience des utilisateurs.

Jusqu'ici, l'interface des applications Web est basée sur le contenu de pages HTML produites sur le serveur.

On introduit ici les mécanismes permettant de faire tourner du côté du client Web, dans le navigateur, du code applicatif qui servira à **modifier le contenu de l'arbre DOM** de ces pages.

L'utilisateur peut donc percevoir une **interface modifiée dynamiquement**, de façon indépendante du serveur d'origine, en fonction de ses **interactions avec le navigateur** (mouvements souris, etc.).

Ce code exécuté sur le navigateur peut inclure le chargement de ressources additionnelles depuis la Toile, permettant ainsi d'interagir avec de multiples serveurs ou de réaliser des agrégations de ressources, comme cela a été popularisé par l'émergence du Web 3.0 avec l'approche AJAX.

4.20 TODO TP 6 : *Rendu projets AgVoy à mi-parcours*

Cette séance proposera l'étude d'une mise en œuvre des concepts d'interface dynamique avec JQuery effectuée sur le projet de gestion de tâches ToDo.

Cette séance sera également consacrée à la **finalisation d'une première version** à mi-parcours de l'application **AgVoy** qui sera **livrée** par le binôme à l'équipe enseignante, afin qu'elle puisse faire l'objet d'une revue et/ou d'une évaluation².

4.20.1 TODO Matériel de cours

— référence : Support du TP 6

1. TP n°5 - Implémentation d'un CRUD sur ToDo, et début du back-office du projet Générateurs de formulaires CRUD et session Symfony

16.1. Table des matières 16.2. Introduction 16.3. Étape 1 : Ajout de fonctions de soumission de données à ToDo 16.4. Étape 2 : Étude du fonctionnement des messages flash 16.5. Étape 3 : Suite du projet : ajout de contrôleurs CRUD dans AgVoy 16.6. Étape 4 : Gestion de marque-pages dans le front-office 16.7. Étape 5 : Poursuite des étapes précédentes du projet 16.8. Évaluation 16.9. Ensuite, d'ici la semaine prochaine 16.10. Pour aller plus loin 16.11. Notes de bas de page :

4.21 TODO TA 6-7 : *Auto-apprentissage JQuery ?*

À définir : auto-apprentissage JQuery ?

4.21.1 TODO Matériel de travail

NEANT

2. les modalités de cette évaluation, et notamment le fait qu'elle comporte ou pas une note contribuant à la note finale de contrôle continu, n'ont pas encore fait l'objet d'une décision de l'équipe enseignante, à l'heure de rédaction du présent document.

4.22 TODO CM 7 : *Gestion contextuelle des formulaires*

Durée 1h

Objectifs Cette séance présentera l'**architecture interne du framework Symfony** afin de mieux comprendre ce qui permet mieux structurer le code de l'application Web, dans un contexte objet, pour la rendre plus évolutive, plus sûre et pour améliorer la qualité des développements Web.

On présentera le rôle des modules suivants : Routeur, Contrôleurs,...

À cette étape du cours, on a introduit les principaux concepts et les mécanismes d'implémentation correspondant à l'architecture classique des applications Web en 3 tiers, essentiellement déployées côté serveur.

Le *framework* présenté (Symfony) est représentatif d'une classe assez large d'environnements de développement d'applications Web, avec une séparation interne en couches inspirées du patron Modèle Vue Contrôleur (MVC) (dans une variante simplifiée).

4.23 TODO TP 7 : *Ajout utilisateurs et permissions, séparation fr*

Durée 2h

Objectifs Cette séance permettra d'explorer la façon dont l'application de gestion de tâche `ToDo` peut être mieux structurée, en tirant parti des fonctionnalités du *framework* Symfony, afin de **séparer l'application** en deux grands modules gérant deux grandes parties de l'interface de l'application : un **front-office** et un **back-office**, ce qui permettra par exemple de gérer des privilèges pour des utilisateurs différents de l'application.

Cela permettra de mieux comprendre comment mettre en œuvre la restructuration du code du projet `AgVoy` pour **commencer la mise en œuvre du back-office** destiné aux seuls utilisateurs qui travaillent pour l'agence de voyages.

4.23.1 TODO Matériel de cours

4.24 TODO TA 7-8 : *Utilisateurs et permissions dans AgVoy*

Sur le modèle de ce qui aura été vu sur l'application `ToDo`, les étudiants entameront la réalisation de la seconde grande partie de l'application `AgVoy` de leur projet, destinée à **mettre en œuvre le back-office** de l'application.

Ce *back-office* est destiné uniquement aux équipes de l'agence. Il faut donc en protéger l'accès grâce aux mécanismes de **gestion des utilisateurs** et des **permissions** qui aura été étudié dans la séance de TP précédente.

On s'appuiera sur les règles de modularisation du code et les mécanismes de Symfony pour mieux distinguer le code des deux grands modules *front-office* et *back-office*.

Les étudiants pourront également adapter l'organisation du système de gabarits et de mise en forme des styles des pages pour adapter la présentation des pages en fonction du profil des utilisateurs.

4.25 TODO CM 8 : *Formulaires et associations dans le modèle*

Durée 1h

Objectifs Cette séance reviendra sur le fonctionnement des mécanismes de **soumission de données** via les **formulaires Symfony** et les meilleures pratiques pour écrire le code correspondant de façon compacte et maintenable.

On abordera également la façon dont on peut mettre en œuvre la gestion d'entités du modèle de données liées par des **associations 1-N**.

4.26 TODO TP 8 : *Implémentation formulaires entités nouvelles*

Durée 2h

Objectifs Cette séance consistera à mettre en œuvre dans l'application de gestion de tâches ToDo un jeu de formulaires pour gérer des **données plus complexes** afin d'appliquer ce qui aura été présenté dans la séance de cours qui précède.

4.26.1 TODO Matériel de cours

4.27 TODO TA 8-9 : *Ajout Programmmations dans AgVoy*

Cette séquence permettra de continuer le travail des séquences précédentes sur le **projet AgVoy**, dans l'optique d'y ajouter le support de la gestion des **circuits programmés**, aussi bien dans le *back-office* que dans le *front-office*, en s'inspirant de ce qui aura été vu dans la séance de TP qui précède.

4.28 TODO CM 9 : *Gestion de la sécurité, des erreurs :Sécurité:E*

Durée 1h

Objectifs Cette séance conclut le panorama des fonctionnalités attendues d'une application Web, en traitant des problématiques de **sécurité** ainsi que de la **gestion des erreurs**.

4.29 TODO TP 9 : *Formulaires dynamiques complexes*

Durée 2h

Objectifs Cette séance consistera à mettre en œuvre dans l'application fil-rouge ToDo la gestion de **formulaires dynamiques Symfony**, en intégrant les mécanismes codés en PHP et JQuery, et à ajouter une **gestion des erreurs**.

4.29.1 TODO Matériel de cours

4.30 TODO TA 9-10 : *Finalisation projet AgVoy*

Durant cette séquence, les étudiants travailleront à la **finalisation de leur projet AgVoy**, dans l'optique du rendu qui interviendra à la fin de la dernière séance de TP, et qui fera l'objet de l'évaluation finale, notée.

Les derniers éléments de finalisation de l'application seront intégrés, pour respecter les exigences du cahier des charges (intégration de formulaires dynamiques, gestion des erreurs, etc.).

4.31 TODO CM 10 : *Évolutions des architectures applicatives*

Durée 1h

Objectifs Cette dernière séance permettra de faire un récapitulatif des caractéristiques de l'architecture présentée, et de présenter des **architectures alternatives** qui ont vu le jour plus récemment.

On abordera également les problématiques liées au **déploiement** et à l'**hébergement** des applications à l'ère du *Cloud*.

5 TODO Évaluation / Notation

L'évaluation portera sur les éléments suivants :

- contrôle continu :
 - rendu intermédiaire des projets AgVoy des binômes
 - rendu final des projets des binômes
- contrôle final

6 Index



Un certain nombre de variantes littérales des acronymes sont aujourd’hui obsolètes, et identifiées avec la mention « *daté* ». Les acronymes sont toujours d’un emploi courant, même on s’est aujourd’hui éloigné de la signification originelle littérale.