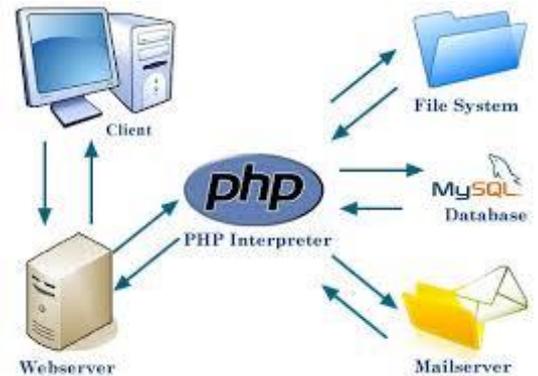


Tutoriel 3 : « Pages web dynamiques »



PROBLEMATIQUE

Accéder à une page web avec un identifiant et un mot de passe. Afficher des données mémorisées sur un serveur dans une page web.

CONDITIONS DE DEROULEMENT DE L'ACTIVITE

Phases de travail	Objectifs	Activités
1) Mise en situation	Comprendre le modèle client-serveur appliqué aux pages web.	Lecture de la mise en situation.
2) Etude de la problématique	Programmer un formulaire dans lequel sera saisi un mot de passe et un identifiant. Acquérir des informations (température, humidité et luminosité) située dans une base de données (fichier texte) avec un script PHP et les présenter dans une page Web.	Construire un site Web constitué de deux pages. La première contiendra un formulaire permettant d'entrer un identifiant et un mot de passe afin d'accéder à la deuxième. Cette dernière présentera des informations dans un tableau.
3) Synthèse	Modifier le site « Domotique » élaborée lors des précédents TP ou un exemple de correction joint. Cahier des charges : La page mesuresphys.php doit présenter les données situées dans un fichier texte.	idem précédent

SIGLES UTILISES

 : Durée conseillée	 : Important	 : Information	 : Editeur de texte (voir prof)
--	---	---	--

RESSOURCES DOCUMENTAIRES, LOGICIELS ET MATERIELS

PC + Navigateur (Edge, Chrome, Mozilla, etc.) + Wamp Serveur + éditeur de texte + dépliant des balises HTML et liste des propriétés CSS.
Répertoire **TP3WEB**

Liens

Les méthodes http : <http://goo.gl/pXiOhF>

Structure globale d'un document HTML5 : <http://goo.gl/gEzri4>

"Les formulaires" sur OpenClassRooms : <https://goo.gl/XOhNky>

Table des matières

1	MISE EN SITUATION.....	3
1.1	INTRODUCTION.....	3
1.2	L'ARCHITECTURE CLIENT / SERVEUR	3
1.2.1	<i>Généralités</i>	3
1.2.2	<i>HyperText Transfert Protocol</i>	3
1.2.3	<i>Les méthodes http</i>	3
1.2.4	<i>Page web : statique ou dynamique</i>	4
1.3	ORGANISATION DU TP.....	4
2	ETUDE DE LA PROBLEMATIQUE	5
2.1	ETUDE DE L'APPLICATION « RESEAU PHYSIQUE »	5
2.1.1	<i>Présentation</i>	5
2.1.2	<i>Mise en œuvre du serveur "Apache"</i>	6
2.1.2.1	Présentation de WAMPSEVER	6
2.1.2.2	Installation du répertoire du site "Grandeurs_Physiques" sur le serveur.....	6
2.1.2.3	Test du serveur "Apache"	6
2.1.3	<i>Conception d'un formulaire de " login"</i>	6
2.1.3.1	Ecriture du code du formulaire	7
2.1.3.2	Analyse du code des fichiers <i>index.html</i> et <i>acces.php</i>	7
2.1.4	<i>Intégration des données dans la page <i>TblGphy.php</i></i>	8
3	SYNTHESE	9
4	ANNEXE 1 : APPLICATION « RESEAU PHYSIQUE »	10

1 Mise en situation

[🕒 15mn]

1.1 Introduction

L'application « Réseau Physique » présentée avec le schéma de l'annexe 1 illustre une solution possible pour acquérir, transmettre et afficher des grandeurs physiques.

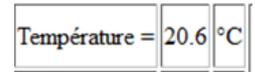


Figure 1 : Accès à une température avec l'application "Réseau Physique"

Ce schéma est organisé autour de trois matériels :

- Une **carte de prototypage rapide** (client http) mesure et transmet périodiquement la valeur de la température ambiante à un serveur.
- Un **Nas Synology** (serveur http) sauvegarde la donnée transmise par la carte de prototypage et délivre une page web "à la demande".
- Les **navigateurs des PC** (clients http) affichent la page contenant la valeur de la température.

Cette organisation fait apparaître un modèle **client/serveur** appliqué aux **pages web**.

1.2 L'architecture client / serveur

1.2.1 Généralités



« L'architecture client/serveur désigne un mode de communication entre plusieurs composants d'un réseau . Chaque entité est considérée comme un **client** ou un **serveur**. Chaque logiciel client peut envoyer des **requêtes** à un serveur. Un serveur peut être spécialisé en serveur d'application, de fichiers, de terminaux, ou encore de messagerie électronique. »

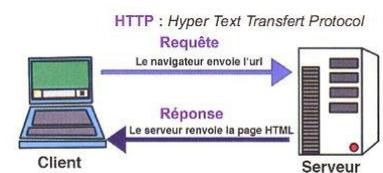


Figure 2 : Principe du modèle client/serveur

Le client pose une question (donne un ordre)... le serveur répond à la question (obéit).

1.2.2 HyperText Transfert Protocol

« La **consultation des pages d'un site Web** a un fonctionnement basé sur l'architecture **client/serveur**. L'internaute connecté au réseau via son ordinateur et un navigateur Web constituent le **client**. Le **serveur** correspond à l'ordinateur contenant les applications qui délivrent les pages demandées. Dans ce cas, c'est le protocole de communication **HTTP** qui est utilisé.

L'**HyperText Transfert Protocol**, plus connu sous l'abréviation **HTTP**, littéralement le « protocole de transfert hypertexte », est le **protocole de communication client/serveur** développé pour le **World Wide Web**.

HTTP

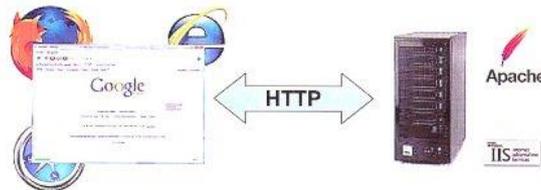


Figure 3 : Dialogue http

Les **navigateurs sont les clients** (Firefox, Chrome, Safari, Microsoft Edge...). Ceux-ci se connectent à des serveurs HTTP tels qu'**Apache** ou **IIS** (Internet Information Service).

1.2.3 Les méthodes http

« Dans le protocole HTTP, une **méthode est une commande** spécifiant un type de **requête**, c'est-à-dire qu'elle demande au serveur d'effectuer une action. En général, l'action concerne une ressource identifiée par l'URL qui suit le nom de la méthode.

Les méthodes les plus utilisées (PUT, GET et POST) sont présentées ci-dessous. Seule la méthode GET sera mise en œuvre.

PUT	Cette méthode permet d'ajouter ou de remplacer une ressource sur le serveur.
GET	C'est la méthode la plus courante pour demander une ressource. Une requête GET est sans effet sur la ressource.
POST	Cette méthode est utilisée pour transmettre des données en vue d'un traitement à une ressource (le plus souvent depuis un formulaire HTML). L' URI fourni est l'URL d'une ressource à laquelle s'appliqueront les données envoyées. Le résultat peut être la création de nouvelles ressources ou la modification de ressources existantes. À cause de la mauvaise implémentation des méthodes HTTP (pour Ajax) par certains navigateurs (et la norme HTML qui ne supporte que les méthodes GET et POST pour les formulaires), cette méthode est souvent utilisée en remplacement de la requête PUT , qui devrait être utilisée pour la mise à jour de ressources. « Wikipédia »

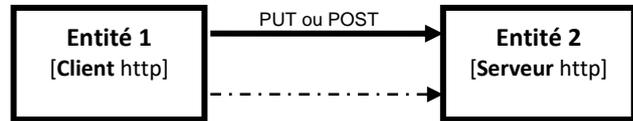


Deux entités http communiquent de quatre manières différentes.

Matériels de l'annexe 1 ->



1 - L'entité 1 est le client. Elle **transmet** des données à l'entité 2 (serveur) **dans une requête (PUT ou POST)**.



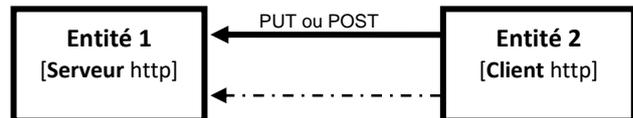
2 - L'entité 1 est le client. Elle reçoit des données de l'entité 2 (serveur) **en réponse** à une requête **GET**.



3 - L'entité 1 est le serveur. Elle envoie des données **après** avoir reçu une requête de l'entité 2 (client).



4 - L'entité 1 est le serveur. Elle **accepte** des données contenues dans une requête **PUT ou POST** de l'entité 2 (client).



Remarque : Une entité peut être à la fois client et serveur.

Sens de la requête
 Sens des données



A retenir : La communication entre un client et un serveur est basée sur l'envoi de requêtes. Celles-ci sont implémentées par des méthodes http (GET, PUT, POST etc). Pour en savoir plus : <http://goo.gl/pXiOhF>

1.2.4 Page web : statique ou dynamique

Une page web peut être qualifiée de statique ou de dynamique.

« Une **page web dynamique** est une [page web](#) générée à la demande, par opposition à une [page web statique](#). Le contenu d'une page web dynamique peut donc varier en fonction d'informations (heure, nom de l'utilisateur, formulaire rempli par l'utilisateur, etc.) qui ne sont connues qu'au moment de sa consultation. À l'inverse, le contenu d'une page web statique est a priori identique à chaque consultation.



Génération de la page

Lors de la consultation d'une page web statique, un [serveur HTTP](#) envoie le contenu du [fichier](#) dans lequel le contenu de la page est enregistré.

Lors de la consultation d'une page web dynamique, un [serveur HTTP](#) transmet la requête à un [logiciel](#) qui génère le contenu de la page. Les logiciels générant des pages web dynamiques sont fréquemment écrits avec les langages [PHP](#), [JavaServer Pages \(JSP\)](#) ou [Active Server Pages \(ASP\)](#). » Wikipédia

1.3 Organisation du TP

Le TP se divise en deux parties :

1. **L'étude et la modification** du site « **Grandeurs Physiques** » [TP3WEB/Grandeurs_Physiques]. Ce site présente dans une page web des données (température, humidité et luminosité) mémorisées dans un fichiers texte.
Répertoire: [TP3WEB/Grandeurs_Physiques]
2. La **conception** du site « **Domotique** » à partir de la page élaborée lors des précédents TP ou de sa correction.
Répertoire: [TP3WEB/Domotique]

2 Etude de la problématique

[⌚ 1h]

2.1 Etude de l'application « Réseau Physique »

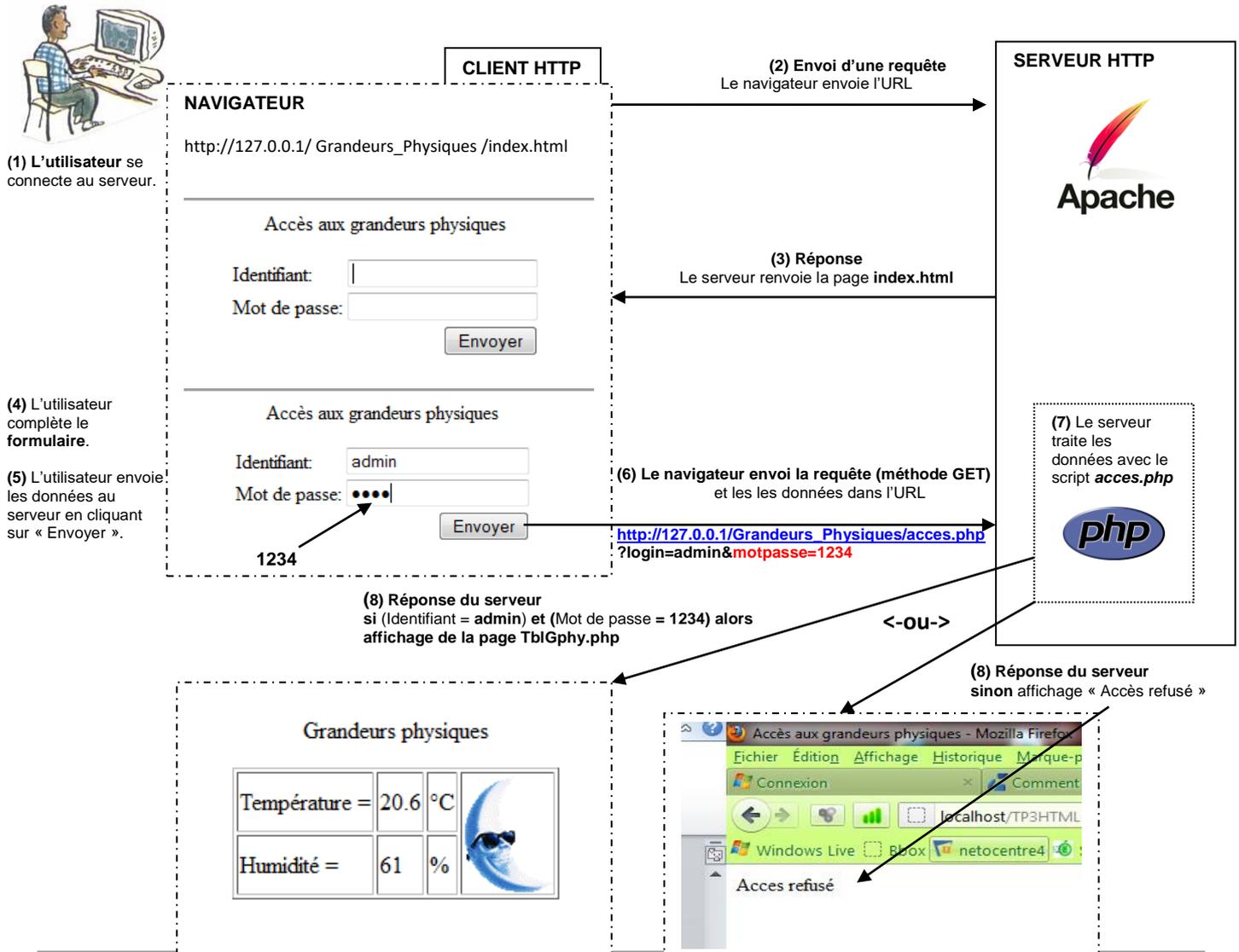
2.1.1 Présentation

Deux modes de communication sont mis en œuvre dans l'architecture représenté en **annexe 1**.

- Un mode « **Monitoring** » : la carte de prototypage produit des données et les transmet au serveur à l'aide d'une **méthode PUT**.
- Un mode « **Consultation** » : les clients 1, 2,..., n-1, n adressent une requête **GET** au serveur. Celui-ci renvoie la page « Grandsurs physique » après avoir testé un **login et un mot de passe**.

- **Le mode « Consultation »**

La page de présentation des grandeurs physiques est accessible à condition de disposer d'un login et d'un mot de passe. Le comportement attendu est présenté ci-dessous :



Remarques : Pour des raisons de sécurité, on n'utilise pas une méthode GET pour transmettre un login et un mot de passe. Dans ce TP, ceci est réalisé à des fins pédagogiques.



- **Simulation du mode « Monitoring »**

Dans l'application de l'annexe 1, la carte de prototypage mesure des données et les transmet au serveur. Pour simplifier les tests à réaliser dans ce TP, vous allez « **prendre la place** » de la carte de développement. Pour cela, vous entrerez **manuellement** les valeurs des grandeurs physiques dans le champ URL (**Uniform Ressource Locator**) du navigateur.

*Transfert
manuel
des données*

Exemple d'URL :

`http://127.0.0.1/Grandeurs_Physiques/Litvaleurs.php?Temp=21.3&Hum=60&Lum=0`

Script PHP de traitement des données

Données à traiter

2.1.2 Mise en œuvre du serveur "Apache"

2.1.2.1 Présentation de WAMPSEVER

Le serveur Apache utilisé ici fait partie de la plateforme de développement web sous Windows appelée **WAMPSEVER**. Elle est accessible à l'adresse :

<http://www.wampserver.com/>



WAMP : Windows Apache MySQL PHP

Cet ensemble de logiciels est installé **sur le disque dur de votre PC**. Il se situe dans le répertoire wamp (c:/wamp). Le répertoire contenant les fichiers du site à développer devra **OBLIGATOIREMENT** être placé dans **www (c:/wamp/www)**.

2.1.2.2 Installation du répertoire du site "Grandeurs_Physiques" sur le serveur

- Démarrez le serveur en cliquant sur l'icône  ou à partir de la liste des programmes :
Démarrer → Tous les programmes → Wampserver.

Appel prof

Pour télécharger le répertoire du TP

- Copier le répertoire (**Grandeurs_Physiques**) dans le répertoire **www** du serveur Apache (**c:/wamp/www**).

Après le démarrage du serveur, un clic sur l'icône  dans la **zone de notification** doit vous donner la figure ci-contre.

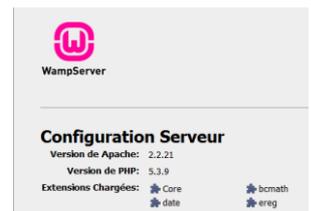


2.1.2.3 Test du serveur "Apache"

Ouvrez la page de garde du serveur Apache en entrant l'URL <http://127.0.0.1> dans un navigateur.

La page ci-contre doit apparaître pour indiquer que le **serveur est prêt à fonctionner !**

Le répertoire **Grandeurs_Physiques** doit être présent dans la liste « Vos projets »



2.1.3 Conception d'un formulaire de "login"

Objectif : Programmer un **formulaire** dans lequel sera saisi un mot de passe et un identifiant. Ces données seront transmises au serveur afin d'accéder à la page présentant le tableau des grandeurs physiques.

Les formulaires permettent de transmettre des informations du client vers le serveur.

Formulaire

Un formulaire est encadré par les balises `<form>` `</form>`.

Des **objets** tels que les **lignes de texte**, les **boutons** d'option et les **cases à cocher** sont placés à l'intérieur de ces balises dont le rôle est de recueillir des informations.

Ligne de texte	Identifiant: <input type="text"/>
Bouton	<input type="button" value="Envoyer"/>



Une fois le formulaire complété, un bouton « **Envoyer** » permet de transmettre les informations au serveur à l'aide d'une **méthode d'envoi** (GET ou POST). Pour en savoir plus : "Les formulaires sur OpenClassRooms" [<https://goo.gl/XOhNky>]

Cahier des charges

La page **index.html** du site à réaliser doit contenir un **formulaire** permettant d'identifier l'utilisateur.

Cette page doit avoir la présentation ci-contre.

Dans un premier temps, vous ne vous souciez pas de l'alignement des différents objets.

Accès aux grandeurs physiques

Identifiant:

Mot de passe:

Cahier des charges

La démarche à suivre est décrite dans la suite du document.

2.1.3.1 Ecriture du code du formulaire



Ouvrez l'éditeur de texte et chargez le fichier « *index.html* » situé dans le répertoire *Grandeurs_Physiques* (c:/wamp/www/Grandeurs_Physiques /).

Complétez le fichier « *index.html* » avec le code ci-dessous :

index.html	acces.php
<pre><!doctype html> <html lang="fr"> <head> <meta charset="utf-8"> <title> Accès aux grandeurs physiques </title> <link rel="stylesheet" href="styles/style.css"> </head> <body> <div class="page"> <p>Accès aux grandeurs physiques</p> <form action="acces.php" method="GET"> <table> <tr> <td> Identifiant: </td> <td> <input type="text" name="login" size="20"/></td> </tr> <tr> <td> Mot de passe: </td> <td> <input type="password" name="motpasse" size="20"/></td> </tr> <tr> <td colspan="2"> <input type="submit" name="EnvValeur" value="Envoyer"/></td> </tr> </table> </form> </div> </body> </html></pre>	<pre><?php if ((\$_GET["login"] == "admin") && (\$_GET["motpasse"] == "1234")) { header('Location: TblPhy.php'); } else echo "Acces refusé"; ?></pre>



Testez votre fichier en entrant l'URL : http://127.0.0.1/Grandeurs_Physiques dans un navigateur.

L'identifiant est **admin** et le mot de passe est **1234** pour voir apparaître le tableau de la page *TblGPhy.php* ->



Attention : Vos fichiers devront toujours être testés sur le serveur Apache installé sur votre PC avec un URL tel que :
<http://127.0.0.1/<repertoire>> ...
<http://127.0.0.1/<repertoire>/<nomdefichier>> ...

Lorsque vous cliquez sur , le navigateur transmet l'URL ci-dessous au serveur.

http://127.0.0.1/grandeurs_physiques/TblGPhy.php/acces.php?login=admin&motpasse=1234.

2.1.3.2 Analyse du code des fichiers *index.html* et *acces.php*

Activité 1 : Proposez une méthode permettant d'afficher la page contenant le tableau de la température (*TblGPhy.php*) sans utiliser le formulaire (sans utiliser la page *index.html*). Revoir éventuellement le paragraphe 2.1.1. Répondez sur le **DR1**.

Activité 2 : Analysez le fichier « *index.html* » pour répondre aux questions du **DR1**.

Activité 3 : Analysez les fichiers « *index.html* » et « *acces.php* » pour répondre aux questions du **DR1**.



Ouvrez le fichier « *acces.php* », situé dans le répertoire **Grandeurs_Physiques (c:/wamp/www/ Grandeurs_Physiques /)**
Personnalisez l'identifiant puis le mot de passe. Testez le fonctionnement.

Activité 4 : Remplacez && par || (accessible par les touches **Alt Gr |**) dans le fichier « *acces.php* » et testez le fonctionnement.
Que remarquez-vous ? Quelle **opération logique** réalise l'opérateur **||** ? Répondez sur le **DR1**.

Indications : Testez toutes les possibilités (Ex : l'identifiant juste et le mot de passe faux etc.)

2.1.4 Intégration des données dans la page *TblGphy.php*

Objectif : Lire les valeurs de la température, de l'humidité et de la luminosité situées dans le fichier *data.txt (data/data.txt)* avec des scripts et les présenter dans la page *TblGPhy.php*.



Ouvrez le fichier « *TblGphy.php* », situé dans le répertoire **c:/wamp/www/ Grandeurs_Physiques /**.

Les scripts PHP de la page effectuent les opérations suivantes :

- (1) **Ouverture** du fichier contenant les données et affectation de son contenu dans une variable \$FValeur

```
$FValeur = fopen("data/data.txt","r"); // $FValeur <- 21,3 ;1 ;60
```
- (2) **Affectation** de la valeur de la température à la variable \$Temp et affichage avec la fonction echo

```
$Temp = fgets($FValeur,10); // Le premier appel de la fonction fgets lit
// la première valeur dans $FValeur etc.
echo "$Temp"; // Affichage du contenu de $Temp 21.3 dans la page
```
- (3) **Fermeture** du fichier contenant les données

```
fclose($FValeur);
```

1	21.3
2	1
3	60

Scripts PHP!

Remarque : Faites des recherches documentaires pour obtenir des informations détaillées sur la syntaxe des fonctions PHP (fopen, fgets, fclose).

Activité 5 : En vous inspirant du code existant, complétez le fichier *TblGphy.php* pour que la page "Grandeurs physique" ait le comportement décrit dans les exemples ci-dessous.

Remarque : Syntaxe du code PHP pour afficher une image `echo "";`

Exemple 1

Quoi ?	Comment ?								
(1) J'alimente le fichier data.txt (2) J'affiche la page TblGPhy.php	(1) http://127.0.0.1/Grandeurs_Physiques/LitValeurs.php?Temp=21.3&Hum=60&Lum=1 (2) http://127.0.0.1/Grandeurs_Physiques/TblGPhy.php								
Résultat attendu	<p>Grandeurs physiques</p> <table border="1"> <tr> <td>Température =</td> <td>21.3</td> <td>°C</td> <td></td> </tr> <tr> <td>Humidité =</td> <td>60</td> <td>%</td> <td></td> </tr> </table>	Température =	21.3	°C		Humidité =	60	%	
Température =	21.3	°C							
Humidité =	60	%							

Exemple 2

Quoi ?	Comment ?								
(3) J'alimente le fichier data.txt (4) J'affiche la page TblGPhy.php	(1) http://127.0.0.1/Grandeurs_Physiques/LitValeurs.php?Temp=16&Hum=70&Lum=0 (2) http://127.0.0.1/Grandeurs_Physiques/TblGPhy.php								
Résultat attendu	<p>Grandeurs physiques</p> <table border="1"> <tr> <td>Température =</td> <td>16</td> <td>°C</td> <td></td> </tr> <tr> <td>Humidité =</td> <td>70</td> <td>%</td> <td></td> </tr> </table>	Température =	16	°C		Humidité =	70	%	
Température =	16	°C							
Humidité =	70	%							



Changez les valeurs des grandeurs physiques pour tester le bon fonctionnement du code.

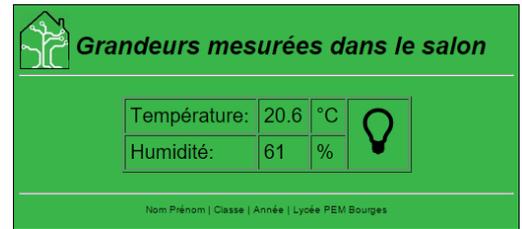
Appel prof

Pour faire vérifier le fonctionnement

3 Synthèse

Objectif : Créer le site « Domotique » à partir des fichiers codés lors des précédents TP ou de la correction proposée (ci-contre).

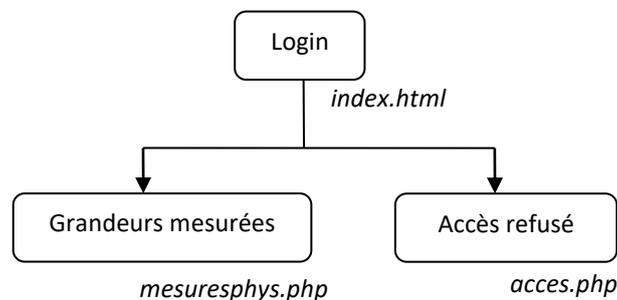
Les fichiers de cette correction sont situés dans le répertoire « Domotique » du TP.



Cahier des charges

La page mesuresphys (exemple ci-dessus) doit présenter les données situées dans un fichier texte.

- Langage de script à utiliser : **php**
- **Architecture** du site « Domotique » à réaliser : schéma ci-dessous



- Nom et position du fichier de données : **data/ data.txt**
- **Page de login** à créer : vous pouvez vous inspirer de acces.php
- Données à afficher :
 - Température en °C,
 - Humidité relative en %,
 - Eclairage : icône on/off à
 - **Volet** : icône ouvert/fermé (à rajouter)
 - **Alarme** : icône (à rajouter)
- Acquisition des données : vous pouvez vous inspirer de LitValeurs.php

A retenir

Il est possible de « passer » des informations à un script à l'aide d'un formulaire. L'information transmise est renseignée par l'utilisateur du navigateur client puis transmise selon la méthode choisie (**GET** ou **POST**) au serveur qui, à l'aide d'un script PHP peut exploiter cette information (sauvegarde, traitement ...).

Il est aussi possible d'assurer une transmission de l'information sans avoir recours à un script.

Une solution consiste à **créer un URL** comme le ferait la méthode GET.

4 Annexe 1 : Application « Réseau Physique »

Le serveur HTTP

Sauvegarde la donnée provenant de la carte Netduino dans un fichier texte (.txt).
Extrait la donnée du fichier .txt et met à jour la page TblGPhy.php

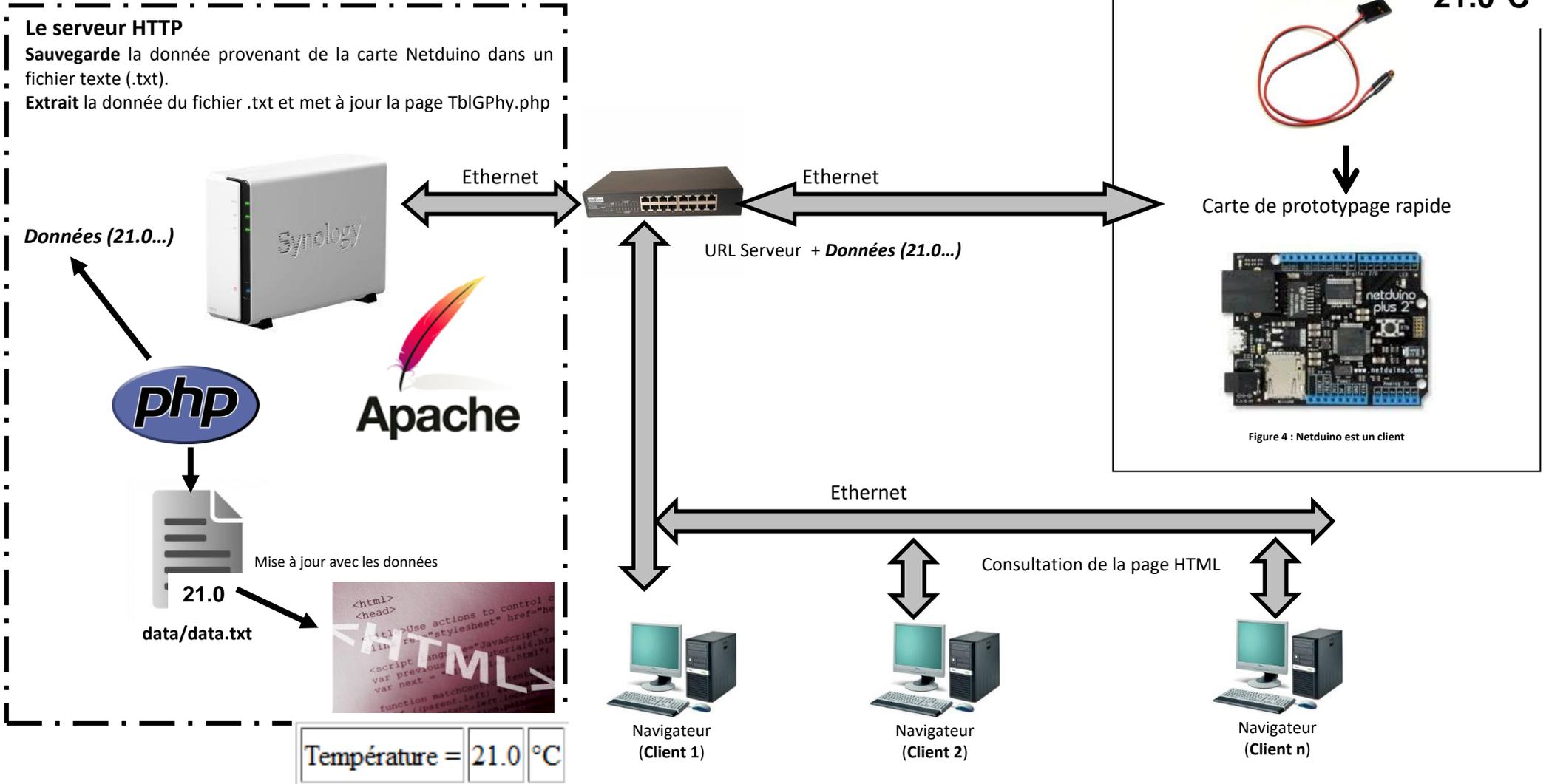


Figure 4 : Netduino est un client

Test du transfert : http://127.0.0.1/Grandeurs_Physiques/LitValeurs.php?Temp=21.0&Hum=62&Lum=1

Test de la consultation : http://127.0.0.1/Grandeurs_Physiques/TblGPhy.php