

ECHANGES

Michael
X  NATIS



Compétence demandée :
Maîtriser les concepts GET, POST, Cookies,
Session et PDO

1. GET et POST
2. Cookies et Session
3. Mysqli et PDO

http://www:80

localhost
127.0.0.1
ip

HTTP

- url
- method
- en-têtes
- payload

localhost 80 http
127.0.0.1 443 https
ip : port



document
root

Apache

PHP

localhost 3306
127.0.0.1
ip : port



MySQL

SQL

navigateur

HTML, CSS, JS

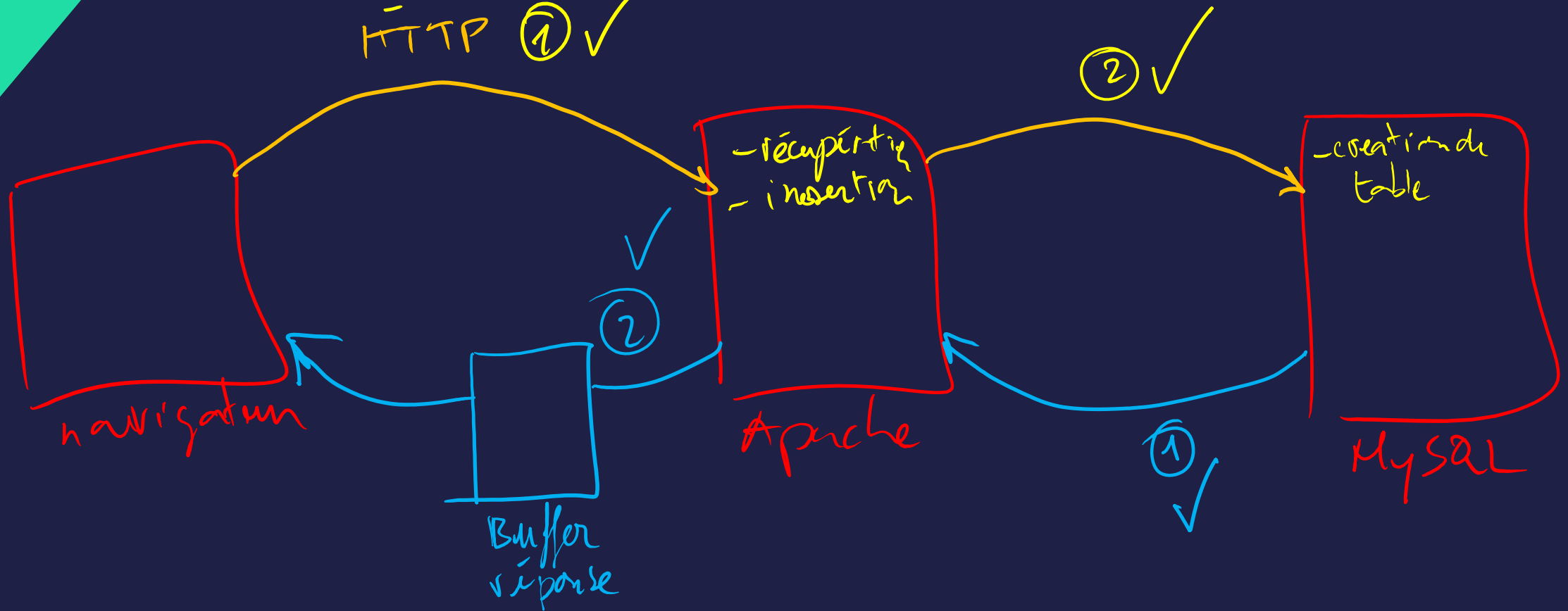
réponse

- status code
- en-têtes

- corps HTML, JS, CSS

`<p style="display:none">Bonjour </p>`

- form
 - name
 - method
 - action
- HTTP ① ✓





```
SELECT * FROM velos  
WHERE id = 2;
```

```
0; DROP TABLE velos;
```

```
SELECT * FROM velos WHERE id = 0; DROP TABLE velos;
```

input name="id">
①

\$_POST['id']
②

```
SELECT * FROM velos  
WHERE id = $_POST['id']  
③
```

Les méthodes de requêtes GET et POST

LE TRANSPORT BACKEND-FRONTEND



Chrome

Je cherche la page
de google pour 'vélo'

HTML + JavaScript + CSS
EXECUTION DE LA REPONSE



REPONSE HTML, CSS, JS + CODE



Apache

Je dessine
ma page

PHP, Python, Node.js
EXUCUTION DU PGM

LE TRANSPORT BACKEND-FRONTEND



Chrome

Je cherche la page
de google pour 'vélo'

HTML + JavaScript + CSS
EXECUTION DE LA REPONSE



REPONSE HTML, CSS, JS
+ EN-TETES (CODE ...)

LOCATION



Apache

Je dessine
ma page

PHP, Python, Node.js
EXUCUTION DU PGM

Une requête est composée de :

1. Une **méthode** (GET, POST, PUT, PATCH, DELETE, HEAD)
2. Des **en-têtes** (Authorization, X-headers)
3. Un **corps**/contenu (payload)

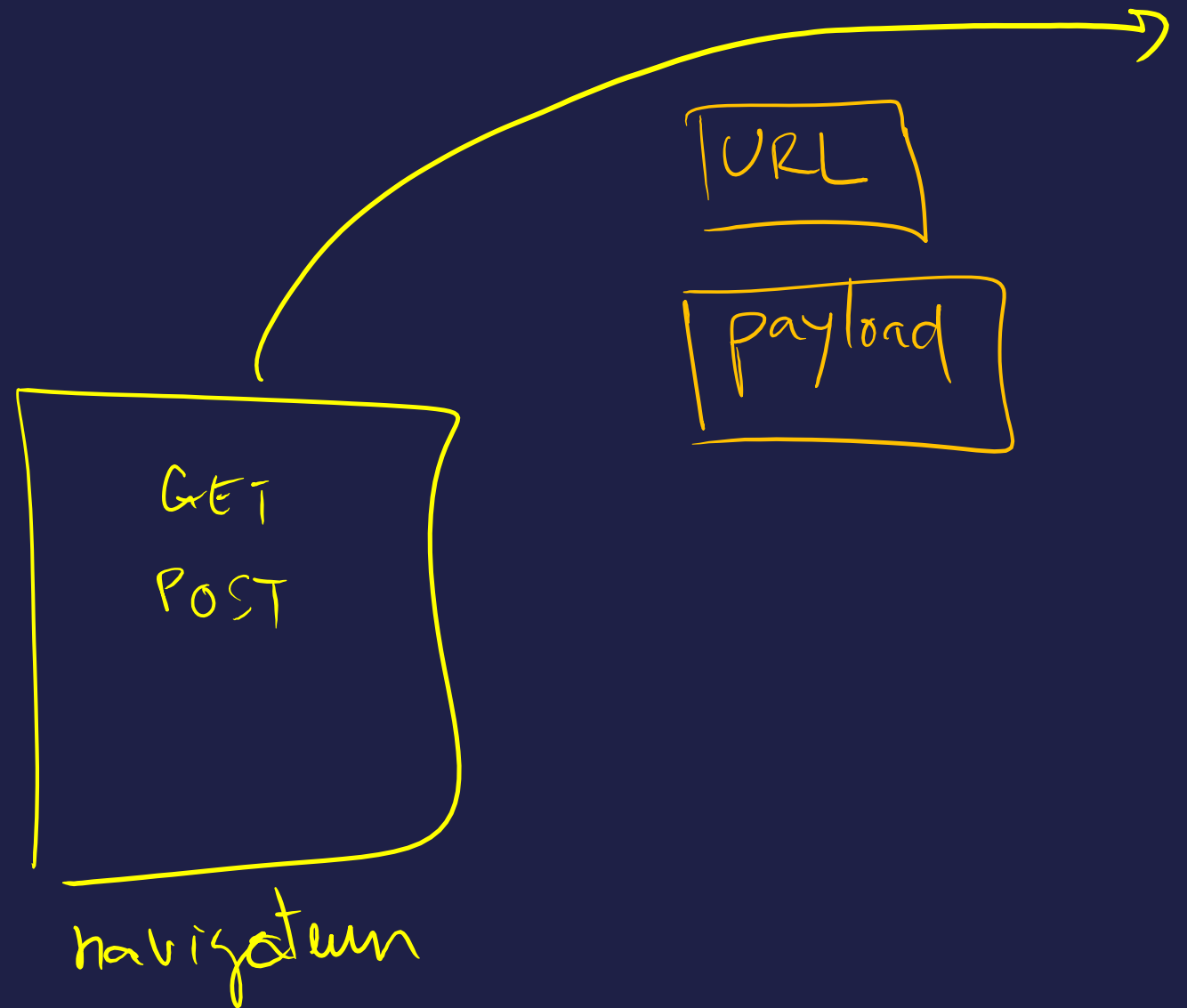
Avant/Aujourd'hui : on utilise le **POST** lorsqu'on ne veut pas que les données de la requête soit visibles dans l'URL

Aujourd'hui/Futur : on utilise le **POST** pour exprimer **l'intention de créer** de l'information

Avant/Aujourd'hui : on utilise le **POST** lorsqu'on ne veut pas que les données de la requête soit visibles dans l'URL



Aujourd'hui/Futur : on utilise le **POST** pour exprimer **l'intention de créer** de l'information



GET : Les données sont à l'intérieur de l'URL

Frontend :

```
GET localhost/exos/exemple.php?cle=valeur
```

Backend PHP :

```
$boite = $_GET['cle'];
```

← nom au frontend

POST : Les données sont à l'intérieur du **payload**

Frontend :

```
POST localhost/exos/exemple.php
```

Backend PHP :

```
$boite = $_POST['cle'];
```


Pour **savoir** si une **clé est présente** dans le dictionnaire

Backend PHP (la fonction retourne un boolean):

```
isset($_POST['cle'])
```

① (1)

```
SELECT * FROM velos WHERE id = ?;
```

prepare

②

execute ('0; DROP TABLE velos')

Un **cookie**, c'est quoi ?

Enregistrer temporairement des
informations sur le visiteur
COOKIE et **SESSION**

LE TRANSPORT BACKEND-FRONTEND



Chrome

Je cherche la page
de google pour 'vélo'

HTML + JavaScript + CSS
EXECUTION DE LA REPONSE



REPONSE HTML, CSS, JS
+ EN-TETES (CODE...)



Apache

Je dessine
ma page

PHP, Python, Node.js
EXUCUTION DU PGM

Backend PHP pour demander au frontend de créer un magic cookie :

```
setcookie('cle', 'valeur');
```

Backend PHP pour lire le magic cookie :

```
$boite = $_COOKIE['cle'];
```

LE TRANSPORT BACKEND-FRONTEND



Chrome

Je cherche la page
de google pour 'vélo'

HTML + JavaScript + CSS
EXECUTION DE LA REPONSE



REPONSE HTML, CSS, JS
+ EN-TETES (CODE...)



Apache

Je dessine
ma page

PHP, Python, Node.js
EXUCUTION DU PGM

Backend PHP pour ajouter une clé à la session :

```
session_start();  
$_SESSION['cle'] = 'valeur';
```

Backend PHP pour récupérer la valeur d'une clé :

```
session_start();  
$boite = $_SESSION['cle'];
```

Backend PHP pour détruire la session :

```
session_destroy();
```


Le **cookie** est stocké du côté du **client** (émetteur de la demande).

La **session** est stocké sur le **serveur**.

Le **cookie** est stocké du côté du **c**lient
(émetteur de la demande).

La **session** est stocké sur le **s**erveur.



La connexion à une base de données MySQL

Mysqli et PDO

Mysqli

```
$connection = mysqli_connect('challenge.xonatis.academy', 'wf3_621', 'tropbien', 'wf3_621');  
$resource = mysqli_query($connection, "SELECT * FROM exo1_categories");  
while ($ligne = mysqli_fetch_assoc($resource))  
{  
  
}
```

PDO

```
$pdo = new PDO('mysql:host=challenge.xonatis.academy;dbname=wf3_621', 'wf3_621', 'tropbien');  
$statement = $pdo->query("SELECT * FROM exo1_categories");  
while ($ligne = $statement->fetch())  
{  
  
}
```


Mysqli ou PDO ?

Mysqli est une interface pour MySQL construite pour PHP

PDO est une couche d'abstraction pour l'accès aux bases de données

