



R4.Real.11 – Développement pour applications mobiles

– Cours 5 –

Services Web et accès distant

Pr Chaouche A.-C.

ac.chaouche@gmail.com

Résumé

Prérequis

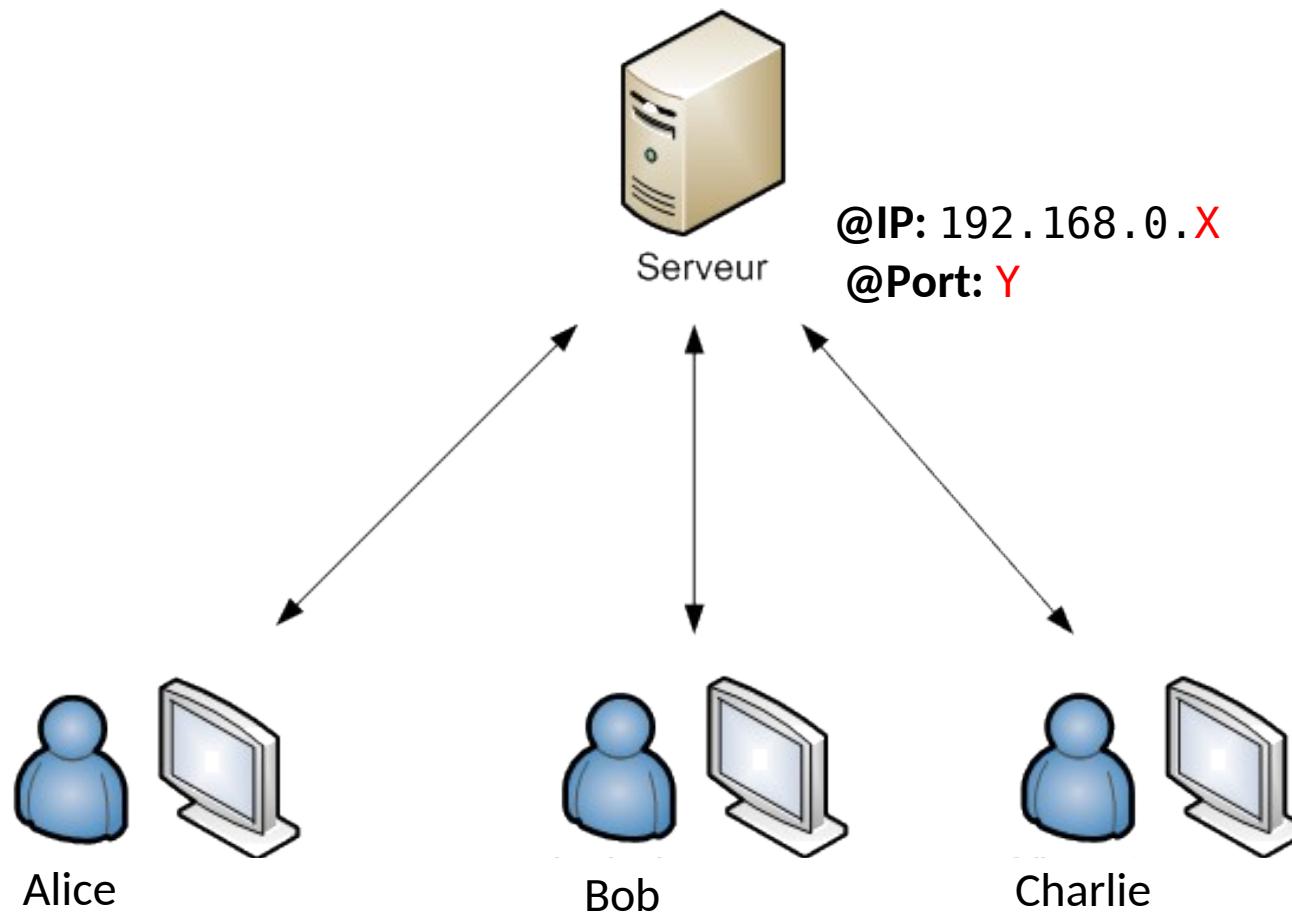
- Maîtrise de la programmation Android
- Maîtrise des langages PHP et SQL



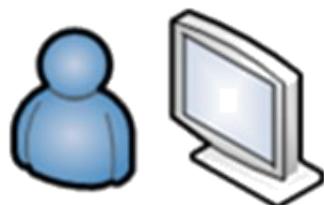
Objectifs du cours

- Comprendre la programmation réseau et Web
- Création d'un service Web connecté à une BD MySQL

Architecture Clients-Serveur



Adressages IP



Alice

127.0.0.1

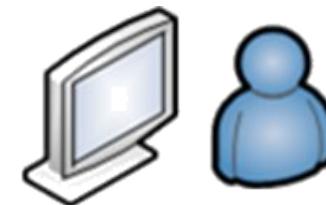
192.168.0.3

86.76.12.105

Adresse localhost

Adresse réseau local

Adresse internet



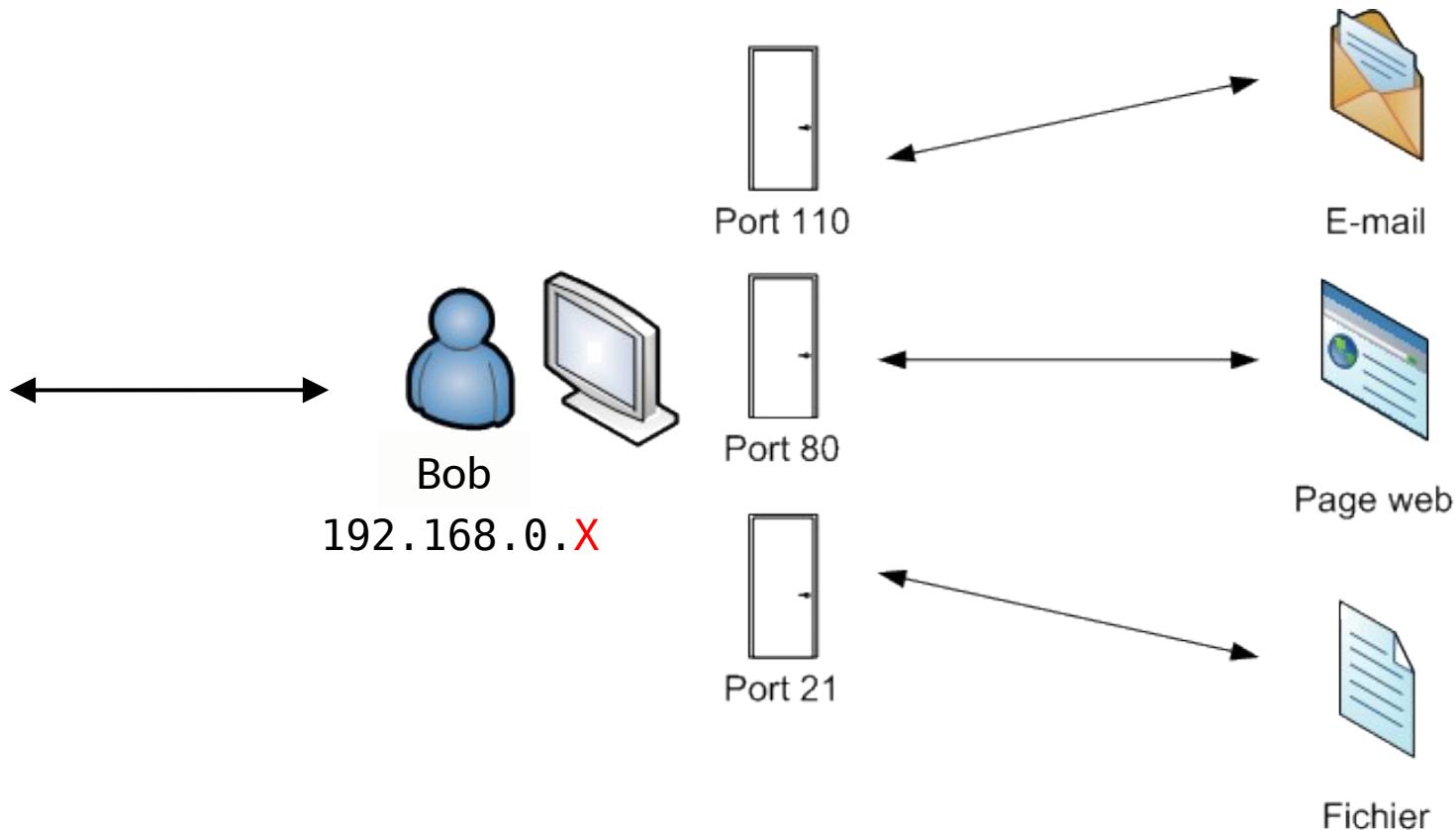
Bob

127.0.0.1

192.168.0.7

90.41.29.47

Ports



Communication sous Android

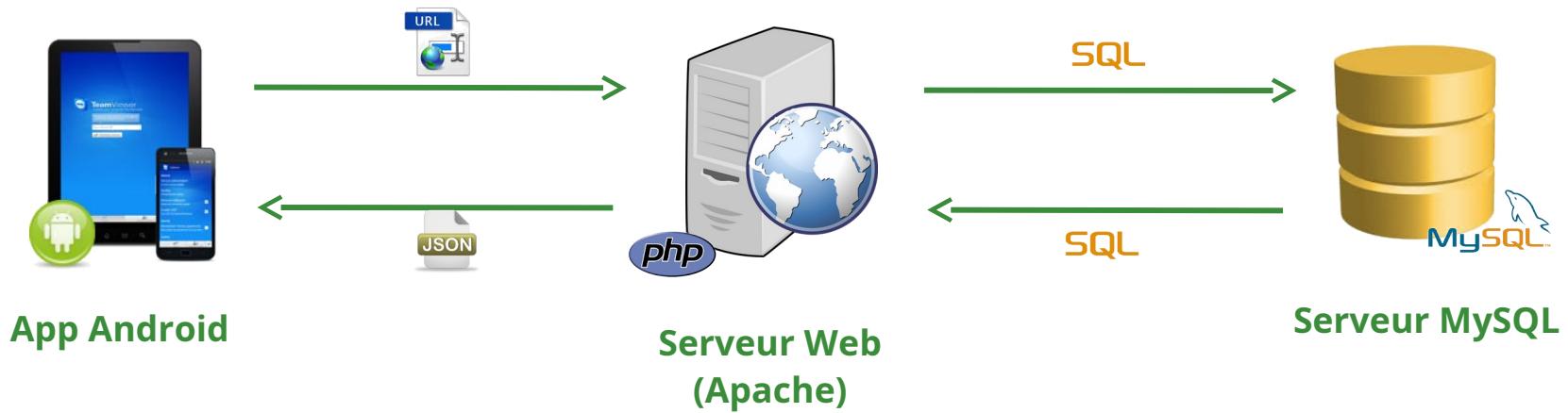
La communication entre un appareil Android et un serveur, à travers :

- **Sockets** : Communication bidirectionnelle personnalisée entre appareil Android et serveur, sous forme de flux de données (Stream).
- **Bibliothèques Remote Procedural Call (RPC)** : À travers des appels de procédures distantes, comme s'il s'agissait de fonctions locales, simplifiant les communications réseau sur Android.
- **Services Web**: Accès à des fonctionnalités distantes via HTTP/HTTPS, couramment utilisés pour échanger des données sur Android. Ils sont basés sur des API standards telle que l'API RESTful pour l'échange d'informations et des formats données normalisés, comme XML ou JSON.

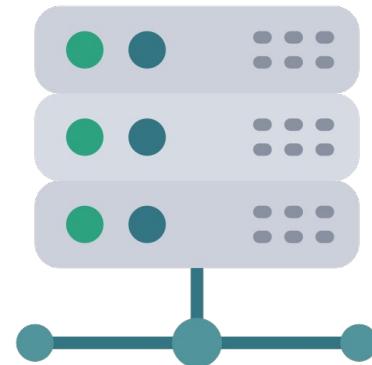
Utilisation des services Web sous Android

Comment ?

- PHP récupère les données à partir d'une BD MySQL, qui seront par la suite **encodées au format JSON** et envoyées au client Android
- L'App Android **récupère** les données, les **parse** et les **exploite** dans l'App



Côté serveur



Utilisation des services Web sous Android

Côté serveur

1. Préparation du schéma de la BD à partir du DC
2. Lancer les serveurs Apache et MySQL (XAMPP)
3. Créer une BD via un client MySQL
4. Se connecter à la BD via un script PHP
5. Retourner des données au format JSON
6. Authentification des clients

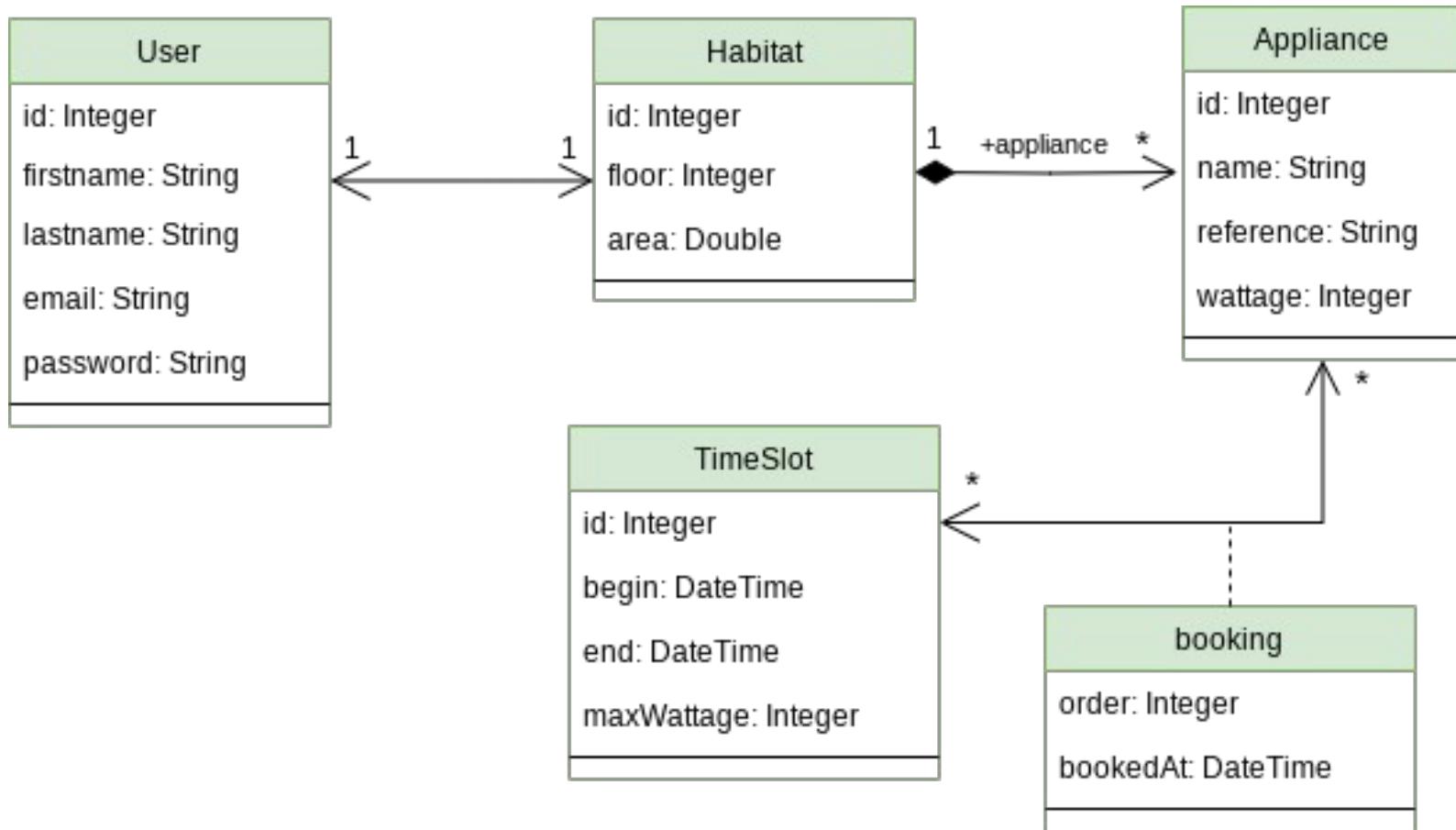


Côté client

1. Préparer le projet Android
2. Vérifier l'accès à internet
3. Récupérer les données distantes sous Android
4. Parser les données JSON pour constituer les objets Java
5. Envoyer des données au serveur distant

Utilisation des services Web sous Android

Serveur-1. Préparation du schéma de la BD à partir du DC



Utilisation des services Web sous Android

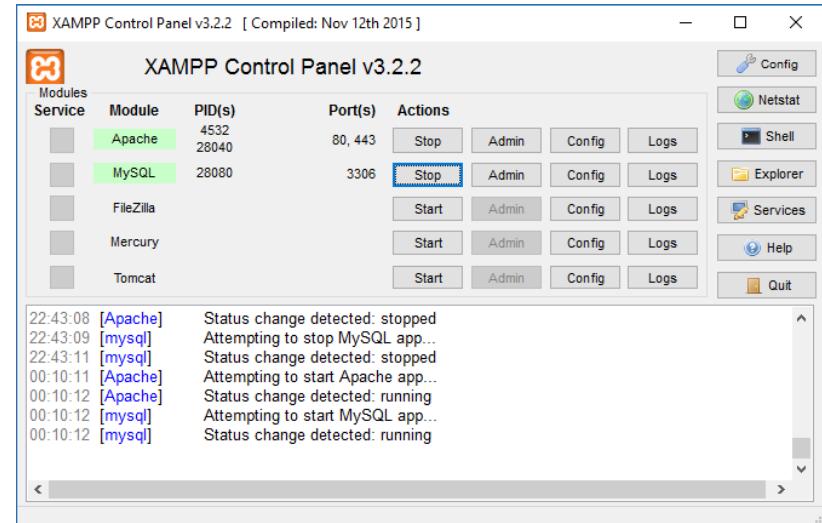
Serveur-2. Lancer les serveurs Apache et MySQL sous Xampp

1. Installer Xampp v3.2.2 XAMPP

- Choisir le chemin dans la racine : "C:\xampp\"

2. Lancer XAMPP Control Panel

- Lancer le **serveur Apache (web)**
- Lancer le **serveur MySQL**



Utilisation des services Web sous Android

Serveur-3. Créer une BD via un client MySQL

1. Utiliser un client MySQL

- Exemples : PhpMyAdmin, HeidiSQL, ...

2. Créer une nouvelle session :

- Host : "localhost"
- Port : 3306
- Utilisateur : "root"
- Mot de passe : ""

3. Créer la BD : powerhome_db

4. Créer les tables :

- **user** : id(INT), firstname(VARCHAR), lastname(VARCHAR), ...
- **habitat** : id(INT), floor(INT), area(INT), ...
- **Applicance** : ...
- ...

Utilisation des services Web sous Android

Serveur-4. Se connecter à la BD via un script PHP

- Dans le dossier : C:\xampp\htdocs\powerhome_server\

/powerhome_server/getHabitats.php

```
<?php
$db_host="localhost"; $db_uid="root"; $db_pass="";
$db_name="powerhome_db";
$db_con = mysqli_connect($db_host, $db_uid, $db_pass, $db_name);

$sql = "SELECT * FROM habitat";
$result = mysqli_query($db_con, $sql);
while($row = mysqli_fetch_assoc($result)) $output[] = $row;
mysqli_close($db_con);

header('Content-Type: application/json');
print(json_encode($output));
http_response_code(200);
```

Utilisation des services Web sous Android

Serveur-5. Retourner des données au format JSON

JSON : JavaScript Object Notation



- **Objets** : commencent et se terminent par {}
- **Attributs d'objet** : sous forme d'un dictionnaire (clé/valeur séparés by ':'). Les attributs sont séparés par des virgules ', '
- **Tableaux** : contiennent un ensemble d'objets et commencent et se terminent par []. Les objets sont séparés par des virgules ', '
- **Valeurs** : peuvent être des strings, des numériques, des objets, des tableaux, true, false ou null.
- **Strings** : sont englobes par "" et contiennent des caractères Unicode, incluant les caractère d'échappement (par exemple, le saut de ligne \n)

```
[  
 {  
   id: "1",  
   floor: "4",  
   ...  
   User : { ... },  
   appliances : [ ... ]  
 }, ...  
 ]
```

Accéder au serveur via le navigateur

Lancer dans le navigateur :

- [http://\[server\]/powershop_server/getHabitats.php](http://[server]/powershop_server/getHabitats.php)
localhost

- Sous Chrome, l'extension **JSONView** permet de mieux afficher le format JSON

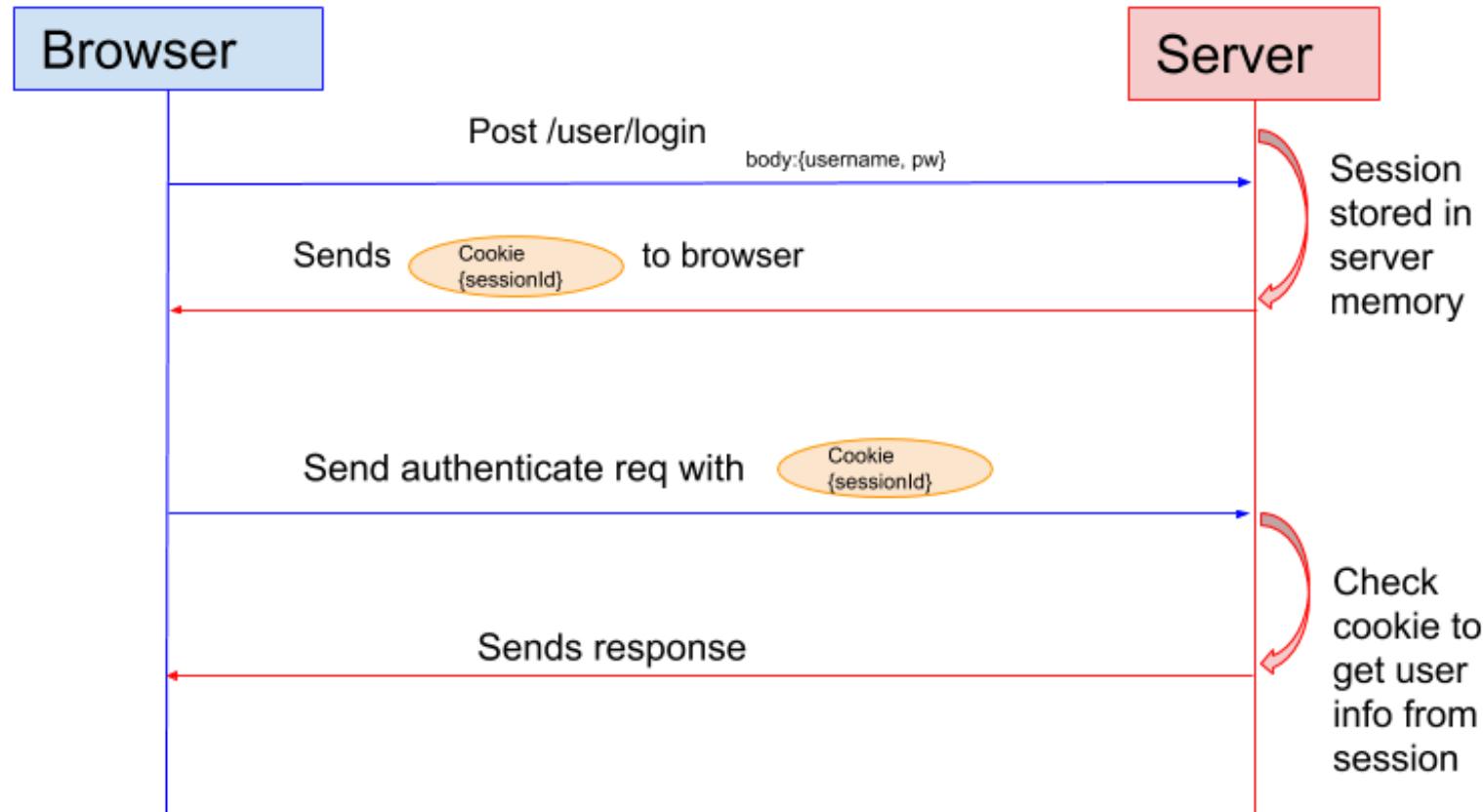


A screenshot of a browser window displaying a JSON response. The JSON data is shown in a hierarchical tree view, where each object has its properties listed. The root object is an array containing two items. The first item has an 'id' of '1', an acronym of 'IAM', a name of 'Interfaces et application mobiles', and a credit of '5'. It also contains an array named 'weeks' with one element. This element has an 'id' of '1', a description of 'premier chapitre', a begin date of '2018-05-01', an end date of '2018-05-07', and a module ID of '1'. The second item in the array has an 'id' of '2', an acronym of 'SEC', a name of 'Sécurité des réseaux', and a description of null.

```
[{"id": "1", "acronym": "IAM", "name": "Interfaces et application mobiles", "description": null, "credit": "5", "weeks": [{"id": "1", "description": "premier chapitre", "begin_date": "2018-05-01", "end_date": "2018-05-07", "module_id": "1"}]}, {"id": "2", "acronym": "SEC", "name": "Sécurité des réseaux", "description": null}]
```

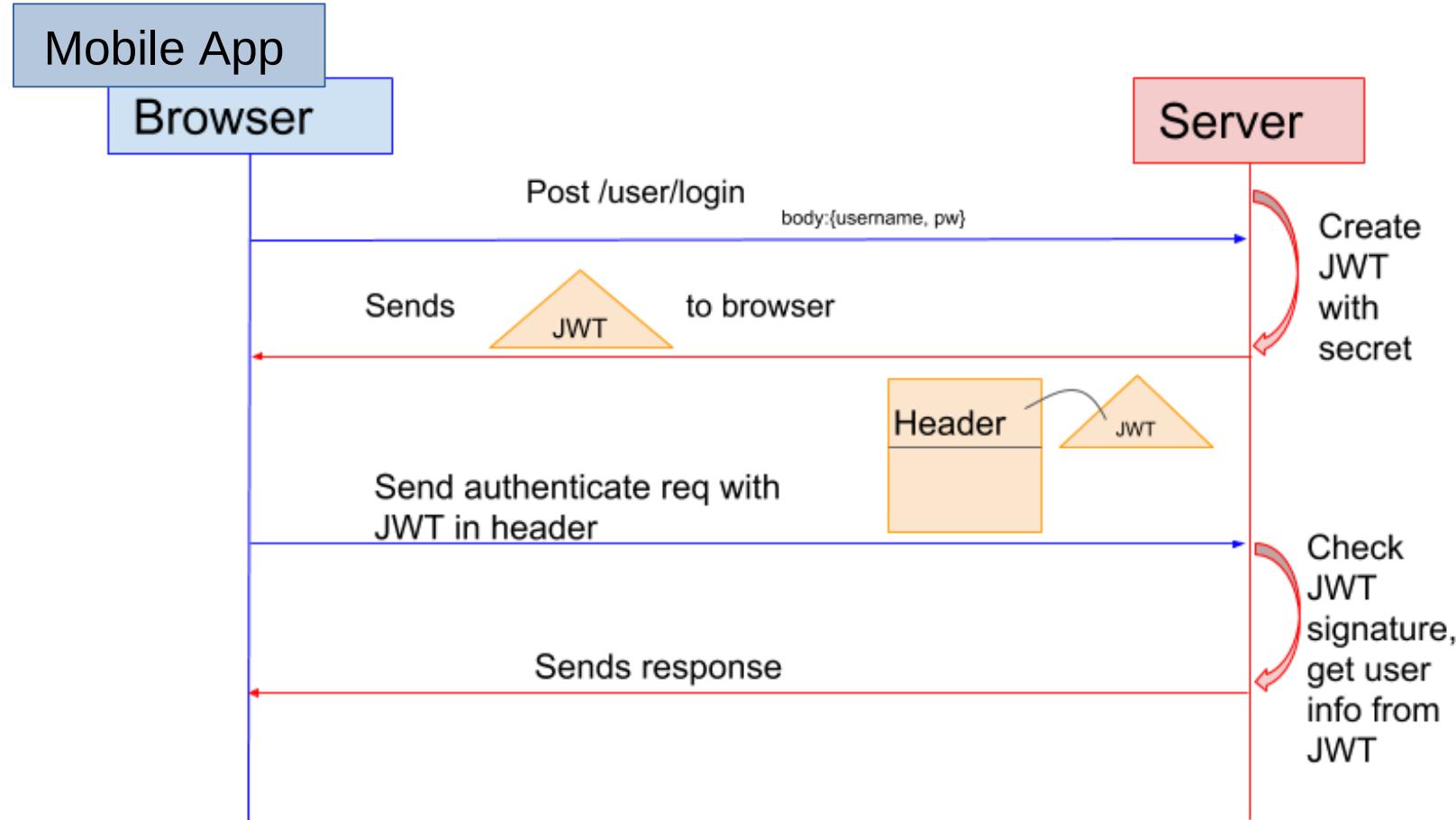
Authentification et autorisation

Basées sur des sessions/cookies



Authentification et autorisation

Basées sur des jetons (JWT)



Authentification et autorisation

Serveur-6. Authentification des clients

/powerhome_server/login.php

```
...
$sql = "SELECT token, expired_at FROM user "
      + "WHERE email='X' and password='Y'";
$result = mysqli_query($db_con, $sql);
$row = mysqli_fetch_assoc($result);
if ($row == null)
    print(json_encode('incorrect email or password !'));
elseif ($row['token'] == null or
        intval(strtotime($row['expired_at'])) < time()) {
    $token = md5(uniqid().rand(10000, 99999));
    $expire = date('Y-m-d H:i:s', strtotime('+30 days', time()));
    $sql = "UPDATE user SET token='$token', expired_at='$expire' "
          + "WHERE email='X'";
    mysqli_query($db_con, $sql);
    print(json_encode(array("token"=>$token, "expired_at"=>$expire)));
} else {
    print(json_encode($row));
}
```

User
id: Integer
firstname: String
lastname: String
email: String
password: String
token: String
expiredAt: DateTime

Authentification et autorisation

Serveur-6. Authentification des clients

/powerhome_server/getHabitats.php

```
...
$sql = "SELECT * FROM user WHERE token='X'";
$result = mysqli_query($db_con, $sql);
$row = mysqli_fetch_assoc($result);
if($row == null or intval(strtotime($row['expired_at'])) < time()))
    print(json_encode('Token invalid or is already expired!'));
else{
    $sql = "SELECT * FROM habitat";
    $result = mysqli_query($db_con, $sql);
    while($row = mysqli_fetch_assoc($result)){
        $output[] = $row;
        print(json_encode($output));
    }
}
header('Content-Type: application/json');
http_response_code(200);
```

Côté client



Utilisation des services Web sous Android

Côté serveur

1. Préparation du schéma de la BD
2. Lancer les serveurs Apache et MySQL (XAMPP)
3. Créer une BD via un client MySQL
4. Se connecter à la BD via un script PHP
5. Retourner des données au format JSON
6. Authentification des clients

Côté client

1. Préparer le projet Android
2. Importer la bibliothèque externe ION
3. Récupérer les données distantes sous Android
4. Parser les données JSON pour constituer les objets Java
5. Envoyer des données au serveur distant

