

Βήμα 1: Κατανόηση του Στόχου και της Αρχιτεκτονικής

Πρώτο βήμα είναι η κατανόηση της αρχιτεκτονικής του συστήματος. Ο στόχος είναι να φτιάξουμε έναν Java Client που θα στέλνει ένα HTTP GET αίτημα σε μια υπηρεσία, θα λαμβάνει JSON δεδομένα, και θα τα επεξεργάζεται για να τα εμφανίσει στην κονσόλα. Τα δεδομένα αντιστοιχούν σε ομάδες ποδοσφαίρου που κατέκτησαν το πρωτάθλημα σε συγκεκριμένες χρονιές, συνοδευόμενα από λογότυπα.

Η υπηρεσία τρέχει τοπικά σε ένα XAMPP περιβάλλον. Χρησιμοποιείται η PHP για να δημιουργηθεί η JSON απόκριση. Η Java εφαρμογή θα αναλάβει να στείλει το αίτημα, να λάβει τα δεδομένα και να τα εμφανίσει δομημένα.

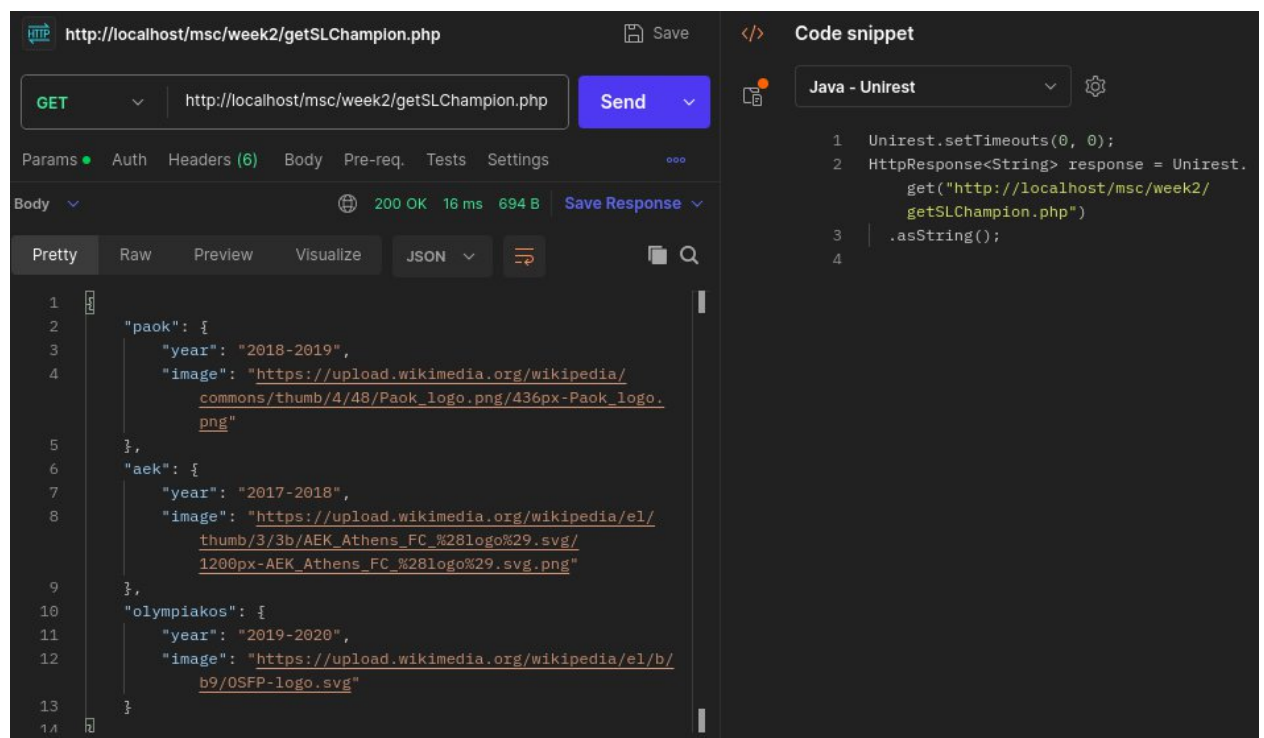
Με την εκτέλεση του XAMPP, η διεύθυνση είναι η

<http://localhost/msc/week2/getSLChampion.php>

Για να μην έχετε αυτό, χρησιμοποιήστε το localhost, μπορείτε να βρείτε την IP (στα windows: cmd → ipconfig → External IP4 ή στο linux: ip add show)

Στο παράδειγμα εγώ είχα <http://192.168.43.111/msc/week2/getSLChampion.php>

Αποτέλεσμα Postman μαζί με τον κώδικα που θα χρειαστούμε.



The screenshot displays the Postman interface. The top bar shows the URL `http://localhost/msc/week2/getSLChampion.php` and a `GET` method. The `Send` button is visible. Below the URL bar, the `Body` tab is selected, showing a `200 OK` status, `16 ms` response time, and `694 B` body size. The response body is displayed in `JSON` format, showing a list of three teams: `paok`, `aek`, and `olympiakos`, each with a `year` and an `image` URL. On the right side, the `Code snippet` panel shows the Java code used to make the request using `Unirest`.

```
1 Unirest.setTimeouts(0, 0);
2 HttpResponse<String> response = Unirest.
  get("http://localhost/msc/week2/
    getSLChampion.php")
3   .asString();
4
```

Βήμα 2: Υλοποίηση της Κλάσης *HttpHandler*

Η κλάση `HttpHandler` είναι υπεύθυνη για την αποστολή του HTTP αιτήματος και την επεξεργασία των αποτελεσμάτων. Χρησιμοποιούμε την βιβλιοθήκη `Unirest` για να κάνουμε GET request στην IP που παρέχει ο χρήστης. Το αποτέλεσμα θα είναι σε μορφή JSON, το οποίο θα αναλύσουμε για να αντλήσουμε τις απαιτούμενες πληροφορίες. Τον κώδικα, τον λαμβάνουμε από το Postman (αλλάζουμε στην παρένθεση την IP με την μεταβλητή `url`). Δημιουργούμε την μέθοδο `getResponse` εισάγοντας το όρισμα της διεύθυνσης `url`. Το `throws Exception` το εισάγουμε διότι υπάρχει περίπτωση να μην υπάρχει η διεύθυνση ή να είναι κλειστή η υπηρεσία. Οπότε θα μας δώσει σφάλμα. Με τον τρόπο αυτό προστατεύουμε τον κώδικα.

```
public ChampionsList getResponse(String url) throws Exception {  
    Unirest.setTimeouts(0, 0);  
    HttpResponse<String> response = Unirest.get(url).asString();
```

Στρατηγική:

- Στέλνουμε το GET αίτημα χρησιμοποιώντας `Unirest`.
- Λαμβάνουμε το JSON string και το μετατρέπουμε σε αντικείμενο `JSONObject` για ευκολότερη διαχείριση.

```
String data = response.getBody();  
JSONObject json = new JSONObject(data);
```

- Εξάγουμε τις πληροφορίες (όνομα ομάδας, έτος πρωταθλήματος, URL λογότυπου) και τις προσθέτουμε σε μια λίστα αντικειμένων `Champion`. Αυτό θα το υλοποιήσουμε μετά που θα δημιουργήσουμε τις κλάσεις.

Δημιουργούμε μια αντικείμενο από την κλάση `ChampionsList`

```
ChampionsList cList = new ChampionsList();
```

Στη συνέχεια πρέπει με μια `while` να πάρουμε μια μια τις εγγραφές από το `JSONObject` που λάβαμε ως απόκριση. Η υλοποίηση γίνεται με τον κώδικα:

// Με μια while να πάρουμε μια μια τις εγγραφές από το JSONObject που λάβαμε ως απόκριση.

```
Iterator<String> keys = json.keys();  
// Όσο το κλειδί έχει επόμενο (hasNext)  
while(keys.hasNext()) {  
    // Εξάγουμε τον χρόνο και το έμβλημα  
    String key = keys.next();  
    String year = json.getJSONObject(key).getString("year").toString();  
    String emblem = json.getJSONObject(key).getString("image").toString();  
    // Δημιουργώ αντικείμενο Champion με τα δεδομένα από την απόκριση  
    // και το προσθέτω στην λίστα ChampionsList  
    cList.addChampion(new Champion(key, year, emblem));  
}
```

Και τέλος επιστρέφουμε την το αντικείμενο `ChampionsList`

```
return cList;
```

Βήμα 3: Δημιουργία Βοηθητικών Κλάσεων *Champion* και *ChampionsList*

Χρησιμοποιούμε δύο κλάσεις:

- **Champion:** Κάθε αντικείμενο αντιπροσωπεύει μια ομάδα, τη χρονιά που κατέκτησε το πρωτάθλημα και το URL του λογότυπου της ομάδας.

// Ιδιότητες

private String name;

private String year;

private String url;

// Κατασκευαστής

public Champion(String n, String y, String e) {

name = n;

year = y;

url = e;

}

// Μέθοδος εκτύπωσης μιας εγγραφής στο τερματικό

public void printInfo() {

System.out.println("\t" + name + " was a champion in: " + year + ", logo: " + url);

}

- **ChampionsList:** Αποθηκεύει μια λίστα από *Champion* αντικείμενα και διαθέτει μεθόδους για την προσθήκη και εμφάνιση των δεδομένων.

// Μια λίστα από αντικείμενα Champion

private ArrayList<Champion> cList = new ArrayList<Champion>();

// Μέθοδος προσθήκης αντικειμένου τύπου Champion στην λίστα

public void addChampion(Champion c) {

cList.add(c);

}

// Μέθοδος εκτύπωσης ΟΛΩΝ των ομάδων στο τερματικό

public void printInfo() {

System.out.println("List of Champions");

for (int i=0; i<cList.size();i++) {

cList.get(i).printInfo();

}

}

Σκεπτικό: Χωρίζουμε τα δεδομένα σε μικρότερες, διαχειρίσιμες οντότητες (κλάσεις) για να διατηρήσουμε τον κώδικα καθαρό και ευανάγνωστο.

Βήμα 6: Δημιουργία της Κλάσης Main

Δημιουργώ μια ιδιότητα, όπου αποθηκεύω τον σύνδεσμο, το url της υπηρεσίας.

```
// Ορίζω τον σύνδεσμο όπου βρίσκεται η υπηρεσία

String url= "http://localhost/msc/week2/getSLChampion.php";

// Χειρισμός σφάλματος με try/catch

try {

    // Δημιουργώ αντικείμενο HttpHandler

    HttpHandler httpHandler = new HttpHandler();

    // Από το αντικείμενο εκτελώ την μέθοδο getResponse για να πάρω το αντικείμενο
    ChampionsList

    ChampionsList cList = httpHandler.getResponse(url);

    // Εκτυπώνω το αποτέλεσμα, χρησιμοποιώντας την μέθοδο printInfo επί του
    αντικειμένου cList

    cList.printInfo();

} catch (Exception e) {

    e.printStackTrace();

}

}
```

Βήμα 5: Εμφάνιση των Αποτελεσμάτων στην Κονσόλα

Χρησιμοποιούμε τη μέθοδο printInfo των κλάσεων Champion και ChampionsList για να εμφανίσουμε τα δεδομένα στην κονσόλα. Θα εμφανίσουμε πληροφορίες όπως το όνομα της ομάδας, τη χρονιά που κέρδισε το πρωτάθλημα, και το λογότυπό της.

```
<terminated> Main (69) [Java Application] /app/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_21.0.4.v20240802-1551/jre/l
List of Champions
aek was a champion in: 2017-2018, logo:https://upload.wikimedia.org/wikipedia/el/thumb/3/3b/AEK_Athens_FC_%28logo%29.svg
olympiakos was a champion in: 2019-2020, logo:https://upload.wikimedia.org/wikipedia/el/b/b9/OSFP-logo.svg
paok was a champion in: 2018-2019, logo:https://upload.wikimedia.org/wikipedia/commons/thumb/4/48/Paok_logo.png/436px-Pa
```

Κλείνοντας

Με αυτή τη μεθοδολογία, επιτυγχάνουμε τη δημιουργία ενός Java Client που μπορεί να συνδέεται σε υπηρεσία ιστού, να διαβάζει δεδομένα JSON, να τα αποθηκεύει σε κλάσεις, και να τα εμφανίζει.