

ΜΑΘΗΜΑ 1: ΚΛΗΣΗ ΣΕ API

Θα δημιουργήσουμε ένα πρόγραμμα που θα κάνει κλήση σε ένα API (υπηρεσία). Η κλήση (request) σε κάποιο API του backend, επιστρέφει αποτελέσματα (respond) στο frontend, τα οποία πρέπει να τα εμφανίσει με όμορφο τρόπο στον τελικό χρήστη.

ΑΣΚΗΣΗ 1

Στην άσκηση θα κάνουμε κλήση στην υπηρεσία <https://reqres.in/api/users/2> όπου επιστρέφει ως αποτέλεσμα:

```
{"data":{"id":2,"email":"janet.weaver@reqres.in","first_name":"Janet","last_name":"Weaver","avatar":"https://reqres.in/img/faces/2-image.jpg"},"support":{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}
```

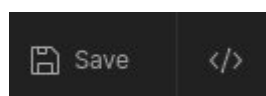
Η μορφή αυτή είναι JSON. Βλέπουμε τους διαχωριστές { } καθώς και το : που διαχωρίζει το κλειδί από την τιμή. Πχ εδώ βλέπουμε το JSON object "data" και η τιμή του είναι όλο το επόμενο.

Στο Firefox τα επιστρέφει με πιο όμορφο τρόπο. Όμως εμείς θα χρησιμοποιήσουμε το πρόγραμμα Postman (<https://www.postman.com/>) που μπορεί να μας δώσει και τον κώδικα που πρέπει να χρησιμοποιήσουμε.

Στο Postman, επικολλούμε την διεύθυνση και επιλέγουμε μέθοδο GET (συνήθως θα χρησιμοποιήσουμε μεθόδους GET και POST). Θα μας βγάλει ως αποτέλεσμα:

```
{
  "data": {
    "id": 2,
    "email": "janet.weaver@reqres.in",
    "first_name": "Janet",
    "last_name": "Weaver",
    "avatar": "https://reqres.in/img/faces/2-image.jpg"
  },
  "support": {
    "url": "https://reqres.in/#support-heading",
    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
  }
}
```

Στο Postman επάνω δεξιά έχει τα κουμπάκια



Πατώντας το δεξιά, θα μας δώσει κώδικα προς χρήση. Έχει πολλούς κώδικες, εμείς θα επιλέξουμε για το παράδειγμά μας τον **Java-Unirest**.

```
Unirest.setTimeouts(0, 0);
```

```
HttpResponse<String> response = Unirest.get("https://reqres.in/api/users/2").asString();
```

Επιπρόσθετα θα χρειαστεί να κατεβάσουμε και να εισάγουμε τις βιβλιοθήκες μέσα από το openeclass. Η διεύθυνση είναι η:

<https://openeclass.uom.gr/modules/document/file.php/MAI130/Week-01/lib.rar>

Αποσυμπίεστε σε ένα φάκελο.

Ξεκινάμε το project μας.

1. Στο eclipse πατήστε New>Java Project. Εισάγετε το όνομα **FirstAPICall**.
2. Επάνω στο project επιλέγω New>Class ονομάζω **MainClass** και επιλέγω το κουτάκι **public static void main(String[] args)**.
 - a. Αν σας εμφανίζει το σφάλμα A package name must be specified for a module. Τότε σβήστε το *module-info.java* μέσα από τον φάκελο src του project.
3. Αντιγράψτε τον φάκελο lib (αποσυμπίεσατε παραπάνω) μέσα στο φάκελο του project σας (δουλεύει και το αντιγραφή από το File Manager και επικόλληση από το eclipse).
4. Κάντε δεξί πλήκτρο επάνω στο Project και επιλέξτε **Build Path > Configure Build Path**. Στο παράθυρο που ανοίγει, επιλέξτε **Libraries > Classpath** και στα δεξιά έχει ένα κουμπί **Add external JARs**. Στο παράθυρο που θα ανοίξει, επιλέγετε όλα τα αρχεία JAR από τον φάκελο lib και πατήστε **Open** και στο τέλος πατήστε **Apply and close**. Οι βιβλιοθήκες θα έχουν εισαχθεί μέσα στον φάκελο του Project στο Referenced Libraries.

Κλάση: **HttpHandler**

Φτιάχνω μια νέα κλάση **HttpHandler** στην οποία θα εισάγω τον κώδικα από το Postman (με λίγες αλλαγές).

Μέσα στην κλάση **HttpHandler** δημιουργώ μια μέθοδο:

```
public String getResponse(String url)
```

στην οποία εισάγω τον κώδικα που έλαβα από το Postman και είναι ο εξής (αλλάζω στο γενικό url από τον συγκεκριμένο της άσκησης):

```
Unirest.setTimeouts(0, 0);
```

```
HttpResponse<String> response = Unirest.get(url).asString();
```

```
String data = response.getBody(); // Από το response πάρε μόνο το body.
```

```
return data; // Επιστρέφει τα δεδομένα όταν ζητηθεί
```

Στο **HttpResponse<String> response = Unirest.get(url).asString()** μπορεί να γίνει κλήση σε μια url που είτε να είναι κλειστή είναι να μην υπάρχει οπότε θα δώσει σφάλμα (exception). Χρειάζεται διαχείριση. Μπορώ να το προστατέψω με try>catch. Επιλέγω το **Import Unirest** (1η επιλογή) και τροποποιώ τον κώδικά μου

```
public string getResponse(String url) throws Exception {
```

Κλάση: **Main**

Τώρα πάμε να δημιουργήσουμε την main.

Αρχικά εισάγω την διεύθυνση που θα γίνεται η κλήση και θα λαμβάνω την απάντηση

```
String url= "https://reqres.in/api/users/2";
```

Δημιουργώ ένα αντικείμενο handler για να πάρω τα αποτελέσματα

```
HttpHandler handler = new HttpHandler();
```

Φτιάχνω μια μεταβλητή για να αποθηκεύσω όλα τα αποτελέσματα της απάντησης όταν κληθεί η μέθοδος `getResponse` με το όρισμα της διεύθυνσης:

```
String myResponse = handler.getResponse(url);
```

Χρησιμοποιώ την εκτύπωση να δω τι θα μου φέρει (δεν χρειάζεται. Θα το αφαιρέσουμε):

```
System.out.println(myResponse);
```

Όμως ο κώδικας `handler.getResponse(url)` δίνει σφάλμα (όπως αναφέραμε και στην άλλη κλάση). Η προτεινόμενη ενέργεια είναι να προστεθεί σε `try>catch`. Αν το πατήσουμε θα το δημιουργήσει αυτόματα. Όμως η καλύτερη πρακτική είναι ο κώδικάς μας να φαίνεται έτσι μέσα στο `try`:

```
HttpHandler handler = new HttpHandler();
```

```
String myResponse = handler.getResponse(url);
```

```
System.out.println(myResponse);
```

Αφού δουλεύει (σας εκτυπώνει ότι εκτυπώνει και στο Postman) πρέπει να λάβουμε το όνομα (`first_name`) και επίθετο (`last_name`) και να τα ενώσουμε. Πρέπει να πάρω το αποτέλεσμα `myResponse` και να το σπάσω σε JSON Stream. Αυτό γίνεται με την εντολή:

```
JSONObject json = new JSONObject(myResponse);
```

Τώρα πρέπει από το key `data` που βρίσκεται το όνομα και επίθετο, παίρνω την τιμή του. Από το την τιμή του `data` πρέπει να πάρω τα πεδία `first_name`, `last_name` και να τα ενώσω. Δημιουργώ το αντικείμενο `fullname` και από το αντικείμενο `JSONObject json`, εισάγω το όρισμα `"data"` και από αυτό λαμβάνω με `getString` το κλειδί `first_name`. Το ίδιο κάνω και με το key του `last_name`. Τέλος τα εκτυπώνω.

```
String fullname = json.getJSONObject("data").getString("first_name")
```

```
+ " " +
```

```
json.getJSONObject("data").getString("last_name");
```

```
System.out.println(fullname);
```

Το αποτέλεσμα της εκτέλεσης του κώδικα θα είναι το:

```
Parsed Name: Janet Weaver
```

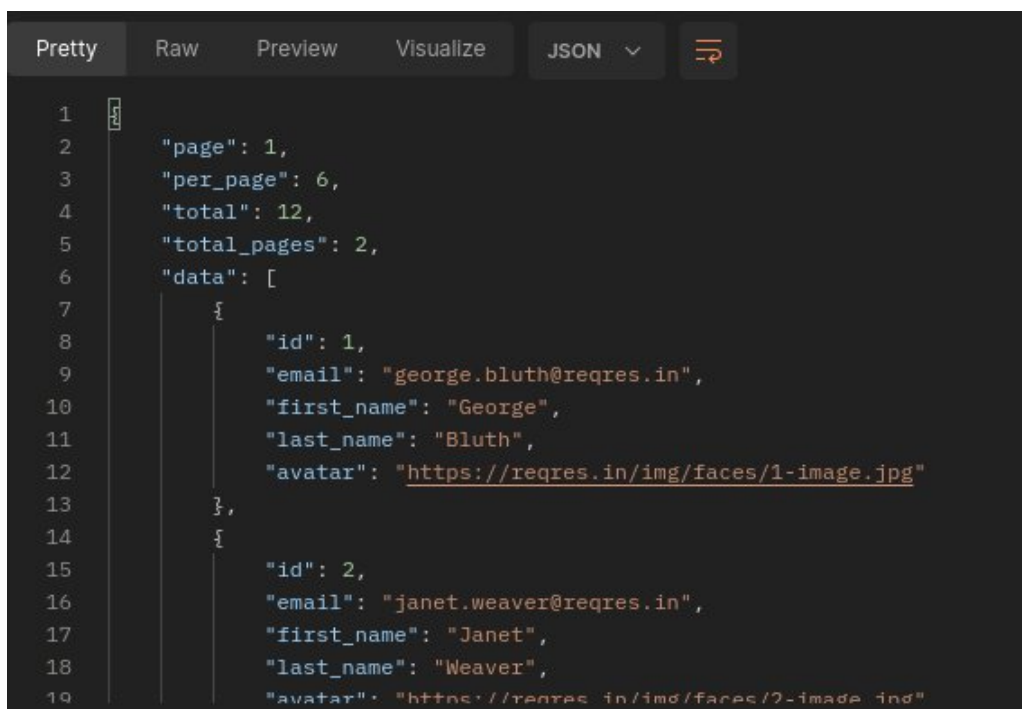
ΑΣΚΗΣΗ 2

Στην άσκηση θα κάνουμε κλήση στην υπηρεσία <https://reqres.in/api/users/> όπου επιστρέφει ως αποτέλεσμα:

```
{
  "page": 1,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 1,
      "email": "george.bluth@reqres.in",
      "first_name": "George",
      "last_name": "Bluth",
      "avatar": "https://reqres.in/img/faces/1-image.jpg"
    },
    {
      "id": 2,
      "email": "janet.weaver@reqres.in",
      "first_name": "Janet",
      "last_name": "Weaver",
      "avatar": "https://reqres.in/img/faces/2-image.jpg"
    },
    {
      "id": 3,
      "email": "emma.wong@reqres.in",
      "first_name": "Emma",
      "last_name": "Wong",
      "avatar": "https://reqres.in/img/faces/3-image.jpg"
    },
    {
      "id": 4,
      "email": "eve.holt@reqres.in",
      "first_name": "Eve",
      "last_name": "Holt",
      "avatar": "https://reqres.in/img/faces/4-image.jpg"
    },
    {
      "id": 5,
      "email": "charles.morris@reqres.in",
      "first_name": "Charles",
      "last_name": "Morris",
      "avatar": "https://reqres.in/img/faces/5-image.jpg"
    },
    {
      "id": 6,
      "email": "tracey.ramos@reqres.in",
      "first_name": "Tracey",
      "last_name": "Ramos",
      "avatar": "https://reqres.in/img/faces/6-image.jpg"
    }
  ],
  "support": {
    "url": "https://reqres.in/#support-heading",
    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
  }
}
```

Η μορφή αυτή είναι JSON. Βλέπουμε τους διαχωριστές { } καθώς και το : που διαχωρίζει το κλειδί από την τιμή. Πχ εδώ βλέπουμε το JSON object “data” και η τιμή του είναι όλο το επόμενο.

Στο Postman, επικολλούμε την διεύθυνση και επιλέγουμε μέθοδο GET. Θα μας βγάλει ως αποτέλεσμα κάτι σαν το παρακάτω:



Πατώντας το δεξιά, θα μας δώσει κώδικα προς χρήση. Έχει πολλούς κώδικες, εμείς θα επιλέξουμε για το παράδειγμά μας τον **Java-Unirest**.

```
Unirest.setTimeouts(0, 0);
```

```
HttpResponse<String> response = Unirest.get("https://reqres.in/api/users/").asString();
```

Επιπρόσθετα θα χρειαστεί να κατεβάσουμε και να εισάγουμε τις βιβλιοθήκες μέσα από το openeclass. Η διεύθυνση είναι η:

<https://openeclass.uom.gr/modules/document/file.php/MAI130/Week-01/lib.rar>

Αποσυμπίεστε σε ένα φάκελο.

Ξεκινάμε το project μας.

1. Στο eclipse πατήστε New>Java Project. Εισάγετε το όνομα **FirstAPICallAll**.
2. Επάνω στο project επιλέγω New>Class ονομάζω **MainClass** και επιλέγω το κουτάκι **public static void main(String[] args)**.
 - a. Αν σας εμφανίζει το σφάλμα A package name must be specified for a module. Τότε σβήστε το *module-info.java* μέσα από τον φάκελο src του project.
3. Αντιγράψτε τον φάκελο lib (αποσυμπίεσατε παραπάνω) μέσα στο φάκελο του project σας (δουλεύει και το αντιγραφή από το File Manager και επικόλληση από το eclipse).
4. Κάντε δεξί πλήκτρο επάνω στο Project και επιλέξτε **Build Path > Configure Build Path**. Στο παράθυρο που ανοίγει, επιλέξτε **Libraries > Classpath** και στα δεξιά έχει ένα κουμπί **Add external JARs**. Στο παράθυρο που θα ανοίξει, επιλέγετε όλα τα αρχεία JAR από τον φάκελο lib και πατήστε **Open** και στο τέλος πατήστε **Apply and close**. Οι βιβλιοθήκες θα έχουν εισαχθεί μέσα στον φάκελο του Project στο Referenced Libraries.

Κλάση: **HttpHandler**

Φτιάχνω μια νέα κλάση **HttpHandler** στην οποία θα εισάγω τον κώδικα από το Postman (με λίγες αλλαγές).

Μέσα στην κλάση **HttpHandler** δημιουργώ μια μέθοδο για να πάρω τα δεδομένα:

```
public String getResponse(String url, int i)
```

στην οποία εισάγω τον κώδικα που έλαβα από το Postman και είναι ο εξής (αλλάζω στο γενικό url από τον συγκεκριμένο της άσκησης):

```
Unirest.setTimeouts(0, 0);
```

```
HttpResponse<String> response = Unirest.get(url).asString();
```

```
String data = response.getBody(); // Από το response πάρε μόνο το body.
```

Στο **HttpResponse<String> response = Unirest.get(url).asString()** μπορεί να γίνει κλήση σε μια url που είτε να είναι κλειστή είναι να μην υπάρχει οπότε θα δώσει σφάλμα (exception). Χρειάζεται διαχείριση. Μπορώ να το προστατέψω με try>catch. Επιλέγω το Import Unirest (1η επιλογή) και τροποποιώ τον κώδικά μου

```
public string getResponse(String url) throws Exception {
```

Στην συνέχεια πρέπει να εισάγω τον κώδικα.

```
// Parse τα δεδομένα JSON
```

```
JSONObject json = new JSONObject(data);
```

```
// Λήψη του πίνακα "data" array
```

```
JSONArray people = json.getJSONArray("data");
```

```
// Λήψη του αντικειμένου νο i
```

```
JSONObject person = (JSONObject) people.get(i);
```

```
// Εξάγετε το όνομα και το επίθετο και σύνδεση σε μια μεταβλητή
```

```
String fullname = person.getString("first_name")
                    + " " +
                    person.getString("last_name");
```

```
// Επιστροφή του ονόματος
return fullname;
```

Το JSONObject και το JSONArray θα εμφανίσουν σφάλμα. Αυτό το διορθώνω κάνοντας import (πρώτη επιλογή) των **import org.json.JSONArray;** και **import org.json.JSONObject;**

Πρέπει να δημιουργήσω και μια μέθοδο που θα λαμβάνει τον αριθμό των δεδομένων που θα επιστρέφονται. Έτσι δημιουργώ την:

```
public int getLength(String url)
```

Ορίζω μια μεταβλητή που θα κρατάει τον αριθμό των εγγραφών που επιστρέφει η κλήση.

```
String total = "0";
```

Στην συνέχεια εισάγω τον κώδικα που μου δίνει το Postman (και έχω εισάγει και στην προηγούμενη μέθοδο):

```
Unirest.setTimeouts(0, 0);
HttpResponse<String> response = Unirest.get(url).asString();
String data = response.getBody();
```

Επειδή το Unirest θα δώσει το ίδιο σφάλμα, στην υπογραφή της μεθόδου πρέπει να προσθέσουμε το **throws Exception**.

Με τον ίδιο τρόπο, εισάγω τον κώδικα (οι 3 πρώτες σειρές μπορούν να μπουν σε try):

```
// Parse για να πάρω τα αποτελέσματα από τα δεδομένα που έχουν επιστραφεί από την κλήση
JSONObject json = new JSONObject(data);
```

```
// Παίρνω τον αριθμό των εγγραφών που περιμένω να έρθουν
total = json.get("per_page").toString();
```

```
// Εκτύπωση πόσοι επεστράφησαν
System.out.println("Length: " + total);
```

Τελευταία σειρά, η μέθοδος επιστρέφει τον αριθμό των εγγραφών (το κάνει σε int γιατί επάνω το όρισε ως string):

```
return Integer.parseInt(total);
```

Κλάση: Main

Τώρα πάμε να δημιουργήσουμε την main.

Αρχικά εισάγω την διεύθυνση που θα γίνεται η κλήση και θα λαμβάνω την απάντηση

```
String url= "https://reqres.in/api/users/";
```

Δημιουργώ ένα αντικείμενο handler για να πάρω τα αποτελέσματα

```
HttpHandler handler = new HttpHandler();
```

Φτιάχνω μια μεταβλητή για να αποθηκεύσω το πλήθος των αποτελεσμάτων-εγγραφών που επιστρέφει η κλίση στην υπηρεσία.

```
int length = handler.getLength(url);
```

Εκτελώ μια for για να λάβω και να εκτυπώσω όλα τα ονόματα. Αυτή η for θα είναι:

```
for (int i=0; i<length; i++) {  
  
    String myResponse = handler.getResponse(url, i);  
  
    System.out.println("Name" + myResponse);  
  
}
```

Η μεταβλητή myResponse αποθηκεύει το i οστό όνομα που δημιουργείται από την κλήση της μεθόδου getResponse στο αντικείμενο handler όπου έχει λάβει τα αποτελέσματα. Όρισμα έχουμε αυτά της υπογραφής της μεθόδου getResponse:

```
String myResponse = handler.getResponse(url, i);
```

Εκτυπώνουμε τα ονόματα με την εντολή:

```
System.out.println("Name" + myResponse);
```

Όμως ο κώδικας handler.getResponse(url, i) δίνει σφάλμα (όπως αναφέραμε και στην άλλη κλάση). Η προτεινόμενη ενέργεια είναι να προστεθεί σε try>catch. Αν το πατήσουμε θα το δημιουργήσει αυτόματα. Όμως η καλύτερη πρακτική είναι ο κώδικάς μας να φαίνεται έτσι μέσα στο try:

```
HttpHandler handler = new HttpHandler();  
  
int length = handler.getLength(url);  
  
for (int i=0; i<length; i++) {  
  
    String myResponse = handler.getResponse(url, i);  
  
    System.out.println("Name" + myResponse);  
  
}
```

Το αποτέλεσμα της εκτέλεσης του κώδικα είναι:

```
Length=6  
Name: George Bluth  
Name: Janet Weaver  
Name: Emma Wong  
Name: Eve Holt  
Name: Charles Morris  
Name: Tracey Ramos
```