

To co będzie tu zamieszczone to nie jest jedyne poprawne i słuszne rozwiązanie, jakim mogą się państwo posłużyć. Na pewno da się inaczej, można wykorzystać inne rzeczy.

Przetwarzanie danych w Java Locale

W Java Locale możemy dość łatwo przetwarzać dane dotyczące krajów, ich nazw, walut, skrótów. Na podstawie skrótu możemy odczytać np nazwę kraju lub symbol waluty.

```
7 public class Main {
8
9     public static void main(String[] args) {
10
11         //obiekt klasy Locale o skrótzie US
12         Locale newLocale = new Locale("", "US");
13
14         //metoda getDisplayCountry() wyciągnie nam nazwę kraju
15         System.out.println(newLocale.getDisplayCountry());
16
17         //lepiej przetwarzać takie dane po angielsku
18         Locale.setDefault(Locale.ENGLISH);
19         System.out.println(newLocale.getDisplayCountry());
20     }
21 }
22
```

Console

<terminated> [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (20 mar 2020, 21:59:25 – 21:59:25)

Stany Zjednoczone
United States

Wszystkie skróty mogą się przydać:

```
Main.java Service.java
8 public class Main {
9
10     public static void main(String[] args) {
11
12         //możemy w ten sposób wrzucić do tablicy wszystkie kody ISO
13         String[] localTab = Locale.getISOCountries();
14
15         for(String loc : localTab) {
16             System.out.println(loc);
17         }
18     }
19 }
```

```
<terminated> [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (20 mar 2020, 22:06:35 - 22:06:35)
AD
AE
AF
AG
AI
AL
AM
AO
AQ
AR
AS
AT
AU
AW
```

Tak samo jak wyciągnięcie waluty danego kraju na podstawie skrótu:

```
7
8 public class Main {
9
10     public static void main(String[] args) {
11
12         //możemy w ten sposób wrzucić do tablicy wszystkie kody ISO
13         String[] localTab = Locale.getISOCountries();
14
15         //na podstawie skrótu możemy w ten sposób wyciągnąć walutę danego kraju
16         System.out.println(Currency.getInstance(new Locale("", "PL")).getCurrencyCode());
17     }
18 }
19
20 |
```

```
<terminated> [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (20 mar 2020, 22:12:08 - 22:12:08)
PLN
```

Jakby to zebrać wszystko do kupy to można ładnie przechowywać dane na których będziemy operować np w mapie. Do mapy można powrzucać nazwę państwa jako klucz, skrót jako wartość. Taką kolekcję możemy łatwo przetwarzać i szybko z niej coś wyciągnąć.

Na podstawie nazwy kraju możemy odczytać jego kod, a na podstawie kodu wyciągnąć walutę. Będziemy zawsze pewni, że kraj odpowiada odpowiedniej walucie i odpowiedniemu kodowi ISO.

Tworzenie połączenia

Rozsądną praktyką jest tworzenie połączeń w jednym miejscu, np w metodzie. Mogłaby ona wyglądać np tak:

```
public void getURLConnection(String site) throws IOException {

    int responseCode = 0;
    StringBuilder response = null;
    String lineToRead;

    // obiekt klasy URL przyjmujący jako parametr adres http
    URL newURL = new URL(site);
    // metoda openConnection zwraca UCLConnection do URL
    HttpURLConnection httpCon = (HttpURLConnection) newURL.openConnection();

    /*
     * setRequestMethod ustawia metody żądania, może być np POST, HEAD, DELETE itd
     * my ustawimy na get-domyslnie jest get więc teoretycznie nie trzeba tego robić
     * różnia się one sposobem wysyłania danych, np dane metoda get wysyłane są w
     * adresie skryptu a metodą post w nagłówku
     */
    httpCon.setRequestMethod("GET");

    // Pobranie kodu odpowiedzi
    responseCode = httpCon.getResponseCode();

    // pobieramy odpowiedz za pomocą streamów
    BufferedReader br = new BufferedReader(new InputStreamReader(httpCon.getInputStream()));

    response = new StringBuilder();
    // wrzucamy wszystko do wynikowego Stringa
    while ((lineToRead = br.readLine()) != null) {
        response.append(lineToRead+"\n");
    }
    br.close();

    System.out.println("Wysłano żądanie pod adres: " + site);
    System.out.println("Kod odpowiedzi: " + responseCode);
    System.out.println("Odpowiedź:");
    System.out.println(response.toString());

}
```

Testowanie w mainie:

```
public static void main(String[] args) throws IOException {
    HTTPTest test = new HTTPTest();
    test.getURLConnection("https://www.pja.edu.pl/");
}
```

Da nam następujący rezultat (oczywiście całości nie zamieszczę na jednym screenie):

```
Console
<terminated> HTTPTest [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (21 mar 2020, 15:16:38 - 15:16:41)
Wysłano żądanie pod adres: https://www.pja.edu.pl//
Kod odpowiedzi: 200
Odpowiedź:

<!DOCTYPE html>
<html prefix="og: http://ogp.me/ns#" xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl-pl" lang="pl-pl" >
<head>

<meta http-equiv="X-UA-Compatible" content="IE=edge">
<base href="https://www.pja.edu.pl/" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta name="keywords" content="informatyka, japonistyka, kultura japonii, programowanie, grafika, sztuka, magisterskie, inżynierskie, zaoczne, uczelnia, Informatyka po ekor
<meta name="description" content="Planowanie przyszłości i zacząć od wyboru dobrej uczelni. PJATK - doceniona przez pracodawców, studentów i media." />
<title>Strona Główna - Polsko-Japońska Akademia Technik Komputerowych</title>
<link href="//format=feed&type=rss" rel="alternate" type="application/rss+xml" title="RSS 2.0" />
<link href="//format=feed&type=atom" rel="alternate" type="application/atom+xml" title="Atom 1.0" />
<link href="http://pja.edu.pl/" rel="canonical" />
<link href="/templates/pjwstk/favicon.ico" rel="shortcut icon" type="image/vnd.microsoft.icon" />
<link href="https://www.pja.edu.pl/component/search/?Itemid=101&format=opensearch" rel="search" title="Szukaj Polsko-Japońska Akademia Technik Komputerowych" type="appl
<link href="https://www.pja.edu.pl/modules/mod_tcnv_k2_nivoslider/assets/css/style.css" rel="stylesheet" type="text/css" />
<link href="https://www.pja.edu.pl/modules/mod_tcnv_k2_nivoslider/assets/themes/default/default.css" rel="stylesheet" type="text/css" />
<link href="/media/mod_falang/css/template.css" rel="stylesheet" type="text/css" />
<script src="/media/jui/js/jquery.min.js?61d622bd9747cd5162301fd01f303c98" type="text/javascript"></script>
<script src="/media/k2/assets/js/k2.frontend.js?v=2.8.0&sitepath=/" type="text/javascript"></script>
<script src="/media/system/js/caption.js?61d622bd9747cd5162301fd01f303c98" type="text/javascript"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.min.js" type="text/javascript"></script>
<script src="https://www.pja.edu.pl/modules/mod_tcnv_k2_nivoslider/assets/js/jquery.nivo.slider.js" type="text/javascript"></script>
<script type="text/javascript">
jQuery(window).on('load', function() {
<
```

Metody getWeather, getRateFor I getNBPRate + JSON.

W metodach getWeather, getRateFor I getNBPRate musimy posłużyć się narzędziem JavaScript Object Nation. Jest to po prostu format zapisu, który jest łatwo dość przetwarzać. Bardzo ładnie cały temat jest opisany na stronie: <https://www.samouczekprogramisty.pl/format-json-w-jezyku-java/>.

Tak wygląda plik ze strony api.openweathermap.org dla Warszawy.

```
{
  "coord": {
    "lon": 21.01,
    "lat": 52.23
  },
  "weather": [
    {
      "id": 802,
      "main": "Clouds",
      "description": "scattered clouds",
      "icon": "03d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 4.06,
    "feels_like": -3.61,
    "temp_min": 2.78,
    "temp_max": 5.56,
    "pressure": 1025,
    "humidity": 51,
    "visibility": 10000
  },
  "wind": {
    "speed": 7.2,
    "deg": 30
  },
  "clouds": {
    "all": 40
  },
  "dt": 1584805138,
  "sys": {
    "type": 1,
    "id": 1713,
    "country": "PL",
    "sunrise": 1584765333,
    "sunset": 1584809429,
    "timezone": 3600,
    "id": 756135,
    "name": "Warsaw",
    "cod": 200
  }
}
```

Jeśli by przerobić naszą metodę getURLConnection, tak aby zwracała Stringa wynikowego to można wtedy stworzyć JSONObject.

```
public String getWeather(String whichCity) throws IOException {

    JSONObject newJSON = new JSONObject(getURLConnection("poprawny adres z api.openweatehrn"));
    return newJSON.toString();

}
```

Oczywiście musimy poradzić sobie ze sklejeniem adresu z podanym miastem. Sprawa wygląda bardzo podobnie w przypadku waluty, oczywiście jeśli użytkownik poda taki sam kraj jaki przetwarzamy to po prostu można returnować 1.0. Podobnie w metodzie getNBPRate, jeśli podany kraj to Polska to też wystarczy returnować 1.0. Trzeba się jeszcze pomęczyć, aby podane dane przetwarzać tak aby GUI je mogło sensownie wyświetlać.