

Praca domowa 11 – teryt

Przemysław Kleszcz

Zadanie

Cel pracy domowej: Celem zadania jest stworzenie usługi (JAX-RS) zwracającej informacje o ilości wystąpień określonej nazwy ulicy na poszczególnych stopniach hierarchii podziału administracyjnego kraju.

Organizacja strukturalna

Projekt zawiera pięć plików: **GUService.java**, **Province.java**, **ProvincesHandler.java**, **Street.java**, **StreetsHandler.java**

GUService.java:

```
public class GUService
{
    public String getByStreet(String street)
    public String getByProvince(String province, String street)
    public String getByCounty(String province, String county, String street)
    public String getByProvinceCounty(String provinceCounty, String street)
    public String getByProvinceText(String province, String street)
    public String getByCountyText(String province, String county, String street)
    public String getByProvinceCountyText(String provinceCounty, String street)
    private String getResult(String province, String county, String street)
    private void createHandlers()
    private ZipInputStream getZipInputStream(String id)
    private <T extends DefaultHandler> T createHandler(ZipInputStream stream, Class<T> type)
```

Province.java:

```
public class Province
{
    public String getCode()
    public void setCode(String code)
    public String getShortName()
    public void setShortName(String shortName)
    public String getType()
    public void setType(String type)
```

Street.java:

```
public class Street
{
    public String getName()
    public void setName(String name)
    public String getCounty()
    public void setCounty(String county)
    public String getProvince()
```

```
public void setProvince(String province)
```

ProvincesHandler.java:

```
public class ProvincesHandler
```

```
    public String getCodeByShortName(String shortName)
    public void startElement(String uri, String localName, String qName, Attributes attributes)
    public void characters(char ch[], int start, int length)
    public void endElement(String uri, String localName, String qName)
```

StreetsHandler.java:

```
public class StreetsHandler
```

```
public String searchedCount(String province, String county, String street)
public void startElement(String uri, String localName, String qName, Attributes attributes)
public void characters(char ch[], int start, int length)
public void endElement(String uri, String localName, String qName)
```

getByStreet(String street) - Metoda przyjmuje jako parametr nazwę ulicy. Zwraca ilość wystąpień podanej nazwy ulicy w skali całego kraju.

getByProvince(String province, String street) - Metoda przyjmuje jako parametr kod województwa oraz nazwę ulicy. Zwraca ilość wystąpień podanej nazwy ulicy w skali określonego województwa.

getByCounty(String province, String county, String street) - Metoda przyjmuje jako parametr kod województwa, kod powiatu oraz nazwę ulicy. Zwraca ilość wystąpień podanej nazwy ulicy w skali określonego województwa i powiatu.

getByProvinceCounty(String provinceCounty, String street) - Metoda przyjmuje jako parametr łańcuch znaków zawierający kod województwa i kod powiatu oraz nazwę ulicy. Zwraca ilość wystąpień podanej nazwy ulicy w skali określonego województwa i powiatu.

getByProvinceText(String province, String street) - Metoda przyjmuje jako parametr pięcioliterowy prefiks nazwy województwa oraz nazwę ulicy. Zwraca ilość wystąpień podanej nazwy ulicy w skali określonego województwa.

getByCountyText(String province, String county, String street) - Metoda przyjmuje jako parametr pięcioliterowy prefiks nazwy województwa, kod powiatu oraz nazwę ulicy. Zwraca ilość wystąpień podanej nazwy ulicy w skali określonego województwa i powiatu.

getByProvinceCountyText(String provinceCounty, String street) - Metoda przyjmuje jako parametr łańcuch znaków zawierający pięcioliterowy prefiks nazwy województwa i kod powiatu oraz nazwę ulicy. Zwraca ilość wystąpień podanej nazwy ulicy w skali określonego województwa i powiatu.

getResult(String province, String county, String street) - Metoda przyjmuje jako parametr kod województwa, kod powiatu oraz nazwę ulicy. Zwraca ilość wystąpień podanej nazwy ulicy w zależności od podanych parametrów.

createHandlers() - Metoda tworzy obiekty odpowiedzialne za logikę importowania danych dla parsera SAX.

getZipInputStream(String id) – Przyjmuje jako parametr nazwę pliku zip. Zwraca obiekt typu ZipInputStream dla określonego pliku.

createHandler(ZipInputStream stream, Class<T> type) – Przyjmuje jako parametr obiekt typu ZipInputStream oraz typ właściwej implementacji obiektu odpowiedzialnego za logikę importowania danych dla parsera SAX. Zwraca nową instancję handlera SAX różniącą się implementacją w zależności od przekazanego typu.

getCode() – Zwraca kod jednostki terytorialnej.

setCode(String code) – Ustawia kod jednostki terytorialnej.

getShortName() – Zwraca prefiks województwa.

setShortName(String shortName) – Ustawia prefiks województwa.

getType() – Zwraca typ jednostki terytorialnej.

setType(String type) – Ustawia typ jednostki terytorialnej.

getName() – Zwraca nazwę ulicy.

setName(String name) – Ustawia nazwę ulicy.

getCounty() – Zwraca kod powiatu.

setCounty(String county) – Ustawia kod powiatu.

getProvince() – Zwraca kod województwa.

setProvince(String province) – Ustawia kod województwa.

getCodeByShortName(String shortName) – Przyjmuje jako parametr prefiks województwa. Zwraca kod województwa w zależności od przekazanego parametru.

searchedCount(String province, String county, String street) – Przyjmuje jako parametr kod województwa, kod powiatu, nazwę ulicy. Zwraca ilość wystąpień nazwy ulicy o określonych parametrach.

Wspólne metody obiektów implementujących logikę pobierania danych parsera SAX:

startElement(String uri, String localName, String qName, Attributes attributes) – Przyjmuje jako parametry URI przestrzeni nazw elementu, nazwę elementu, listę atrybutów bieżącego elementu. Metoda obsługuje zdarzenie napotkania przez parser początku nowego elementu.

characters(char ch[], int start, int length) – Przyjmuje jako parametry tablicę znaków, pozycję początkową określającą, w którym miejscu zaczyna się wartość elementu, pozycję końcową określającą, w którym miejscu kończy się wartość elementu.

endElement(String uri, String localName, String qName) – Przyjmuje jako parametry URI przestrzeni nazw elementu, nazwę elementu. Metoda obsługuje zdarzenie napotkania przez parser końca nowego elementu.

Opis mechanizmu

Aplikacja oczekuje na żądania HTTP. Określona metoda przetwarzania żądań jest dopasowywana na podstawie adresu URL. W następnej kolejności, jeśli nie istnieją kolekcje danych pobranych w czasie poprzedniego wywołania usługi, uruchamiany jest parser SAX, który przetwarza strumieniowo pliki umieszczone na zdalnym serwerze. Parser odfiltrowuje przetwarzane dane i umieszcza w osobnych kolekcjach tylko te, które są niezbędne do wykonania zadania. Przyszłe żądania nie uruchamiają ponownie parsera lecz operują na utworzonych wcześniej kolekcjach. Następnie na podstawie kodu województwa, kodu powiatu i nazwy, ilość konkretnych wystąpień podanej nazwy ulicy wyznaczana jest iteracyjnie z kolekcji przechowującej informacje o ulicach. Opcjonalnie, w przypadku podania prefiksu województwa, wyznaczany jest wcześniej jego kod na podstawie kolekcji utworzonej z danych pliku TERC. Wynik umieszczany jest zwrotnie w odpowiedzi HTTP.