ZAAWANSOWANE TECHNIKI WWW (WFAIS.IF-C112)
(zajęcia 21.01.2016 r.)

Komunikacja bazodanowa w środowisku uruchomieniowym Node.js odbywa się dzięki interfojsowi pośredniczącemu w przesyłaniu danych pomiędzy aplikacją, a wybraną bazą danych. Interfejsy można podzialić na dwa typy: (a) wykorzystujących język SQL do wykonywania kwerend bazy danych, (b) wykorzystujących mapowanie obiektowo-relacyjne (ORM). Na ćwiczeniach będziemy wykorzystywać interfejs z punktu (a), natomiast na wykładzie omówione zostały również rozwiązania z punktu (b).

a. **Instalacja i interfejsu MySQL**

Do zainstalowania pakietu uzywamy standardowego narzędzia NPM:

```
npm instal mysql --save
```

Następnie uwzględniamy pakiet w wybranym projekcie w pliku controlera (routes/index.js):

```
var mysql = require('mysql');
```

b. **Ustanowienie połączenia z bazą danych**

Połączenie realizowane jest na podstawie podanych parametrów serwera bazodanowego: nazwy hosta, nazwy, użytkownika, hasła dostępu, numeru portu na którym serwer bazodanowy nasłuchuje połączeń. Określenie paraametrów komunikacyjnych odbywa się poprzez metodę .createConnection(), co ilustruje przykład poniżej:

```
var db = mysql.createConnection({
        host: 'localhost',
        user: 'user,
        password: 'naslo',
        database: 'nazwa_bazy'
});
```

Po określeniu parametrów połączenia należy zainicjalizować połączenie wykorzustując metodę .connect na obiekcie, który przechowuje informację o parametrach:

```
db.connect();
```

Od tego momentu mamy pełny dostęp do bazy danych i możliwosc wykonywania kwerdend SQL poprzez obiekt “db”.

c. Wykonywanie kwerend SQL

W przypadku opisywanego interfejsu komunikacyjnego będziemy posługiwać się zapytaniami w języku SQL (SELECT, INSERT, UPDATE, DELETE etc.)
Każde zapytanie musi zostać zdefiniowane w postaci ciągu znakowego String np.

```
var zapytanie = 'SELECT * FROM tabela';
var zapytanie = 'SELECT * FROM tabela WHERE id=7';
```

Do wykonania zapytania i przesłania kwerendy do silnika bazodanowego służy metoda ".query()", która wykonujemy na obiekcie połączenia - w naszym przykładzie "db".
Metoda ta przyjmuje dwa parametry:

- obiekt zapytania (var zapytanie),
- funkcję wywołania zwrotnego która obsługuje rezultaty zwrócone z bazy danych function(error, dane) {}

Funkcja wywołania zwrotnego przyjmuje dwa obiekty:

- obiekt przechowujący informację o błędach komunikacji,
- obiekt w którym przechowywane są zwracane z bazy danych informacje. Obiekt ten jest obiektem typu Array (tablicowego).

Przykład użycia funkcji ".query()" dla bazy danych w której znajduje się relacja "users", z trzema polami: **`id (INT, PK, AI), login (VARCHAR), haslo (VARCHAR).`**

```
var sql = 'SELECT * FROM users';
  db.query(sql, function(error,dane){
        res.render('index', { title: 'Express', dane: dane});
   });
});
```

W przykładzie zwrócony obiekt "dane" jest następnie przekazany do widoku w standardowy sposób gdzie powinien zostać obsłużony po stronie widoku.

d. Przekazywanie danych do widoku

Do widoku trafia obiekt "dane" który jest tablicą z polami których nazwy odpowiadają nazwą kolumn które znajdują się w bazie danych (patrz wyżej). Aby wyłuskać z tablicy pojedyńcze wiersze z obiektu "dane" najlepiej posłużyć się jedna z możliwych funkcji logicznych dostepnych po stronie widoku. W moim przykładzie skorzystam z funkcji "each":

```
extends layout

block content
  h1= title
  p Welcome to #{title}
  div
   ul
```

```
each item in dane
 li #{item.login} #{item.haslo}
```

W tym przykładzie obiekt “item” w pętli “each” reprezentuje pojedyńczy wiersz pozyskany z bazy danych. Aby dostać się do poszczególnych pól posługujemy się notacją obiektową (z kropką) gdzie nazwy własności obiektu są tożsame z nazwami kolumn w bazie danych.

ZADANIE:

Proszę przygotować aplikacje w środowisku Express, która będzie:

- *łączyła się z bazą danych korzystając z pakietu “mysql”,*
- *posiadała funcję router .get() pobierającą dane z tej bazy danych i wypisujące je w postaci tabelarycznej w widoku uzytkownika.*
- *posiadającą funkcje routera .post() obsługującą formularz, przez który będzie można zasilać bazę danych nowymi informacjami.*