Wrocław, 22. stycznia 2013

Bazy danych

projekt

Panel mieszkańca dla spółdzielni mieszkaniowej

Piotr Giedziun (184731)

Artur Zochniak (184725)

# Spis treści

[Spis treści](#h.fzcfvw8xw4qi)

[Wstęp](#h.walboqpi2xr)

[Wymagania do wdrożenia projektu](#h.tt5ngno5kx6v)

[Instalacja](#h.dkyuuoc7tw4b)

[Windows](#h.cgv1ta2mb6e6)

[Pobranie kodu źródłowego serwisu](#h.hwq8qd5iv8l3)

[Python](#h.e35xfkvwush)

[Aktualizacja i instalacja pakietów](#h.ar07hpac1l7c)

[Uruchomienie serwisu](#h.meimyqiu6qpp)

[Uurchomienie bazy danych](#h.nsgoazmk3y8s)

[Dodanie bazy danych](#h.1gz4bn88ienl)

[Uruchomienie serwera HTTP](#h.y7wb4qjzgtd4)

[Linux: Ubuntu](#h.g97o4cejkorz)

[Dodawanie przykładowych mieszkańców do bazy danych](#h.mptwx7p703rt)

[Panel mieszkańca](#h.h5pb6ex7q8qt)

[Panel administratora](#h.5ao9c6mw5khw)

[Omówienie schematu bazy danych](#h.ddoxrhebfl36)

[Brama](#h.1rfka4abw2pu)

[Mieszkanie](#h.kotyv4vic99)

[Mieszkaniec](#h.qb3dofyrqgoj)

[Oplaty\_type](#h.uhelecyytgsd)

[Oplaty](#h.1dqo6y9ik36m)

[Wpłaty](#h.lpaplvnmvzqh)

[Oplaty\_questions](#h.i3z2x4tlme8v)

[Newsy](#h.33c8wrycosl5)

[Logs](#h.ce2wk8siv286)

[Wersja demo serwisu](#h.50n442jcgy4s)

[Bibliografia](#h.wnx97t39dq06)

# Wstęp

Projektowany system ma zapewnić dostarczanie informacji dla mieszkańców wrocławskiej spółdzielni mieszkaniowej.

Baza danych powinna przechowywać dane dotyczące rozliczeń ze spółdzielnią (woda, czynsz, ogrzewanie).

W zakres wymagań wchodzi:

* import/eksport danych z/do plików xls,
* publikowanie dokumentów przeznaczonych dla
  + wszystkich, mieszkańców,
  + konkretnego mieszkańca.
* prezentację historii rozliczeń, statystyki itp.,
* newsletter,
* ankiety.

# Wymagania do wdrożenia projektu

Aby w pełni korzystać z funkcjonalności dostarczonego przez nas rozwiązania, wskazane jest spełnienie następujących wymagań:

* System operacyjny: Windows/Linux/Mac
* baza danych MySQL 5.0+
* interpreter języka Python w wersji 2.4+
* *opcjonalnie:* svn (lub SVN Tortoise dla systemu Windows) - w celu aktualizacji oprogramowania.

# Instalacja

## Windows

### Pobranie kodu źródłowego serwisu

Kod można pobrać na kilka sposobów, mając zainstalowany program Git można pobrać go za pomocą polecenia



$ git clone git@github.com:piotrgiedziun/bazy\_project.git

Jeśli nie ma w systemie programu Git, można pobrać repozytorium bezpośrednio ze strony

[https://github.com/piotrgiedziun/bazy\_project/archive/master.zip](http://www.google.com/url?q=https%3A%2F%2Fgithub.com%2Fpiotrgiedziun%2Fbazy_project%2Farchive%2Fmaster.zip&sa=D&sntz=1&usg=AFQjCNF0TKEKDtywkEoSRYo77KrVnpjFBQ)

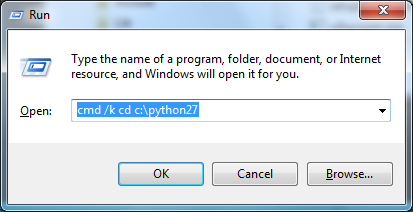
### Python

Wymagane jest zainstalowanie Pythona w wersji 2.7, dostępnego na stronie <http://www.python.org/getit/> [Python 2.7.3 Windows Installer](http://www.python.org/ftp/python/2.7.3/python-2.7.3.msi)

W celu zainstalowania Python Index Package (pip), narzędzia umożliwiającego zarządzanie pakietami należy ściągnąć archiwum tar.gz ze strony <http://pypi.python.org/pypi/pip#downloads> [[bezpośredni link do pip](http://www.google.com/url?q=http%3A%2F%2Fpypi.python.org%2Fpackages%2Fsource%2Fp%2Fpip%2Fpip-1.2.1.tar.gz&sa=D&sntz=1&usg=AFQjCNEj01q3IIy013gD0ht1NpF_j-tcIw)], owy plik rozpakować do folderu z Python’em (domyślnie C:\python27\pip-2.1.2),

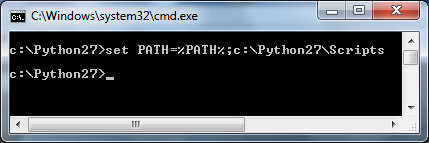
Wymagany jest również pakiet setuptools dostępny na stronie <http://pypi.python.org/pypi/setuptools#files> [[bezpośredni link do setuptools](http://www.google.com/url?q=http%3A%2F%2Fpypi.python.org%2Fpackages%2F2.7%2Fs%2Fsetuptools%2Fsetuptools-0.6c11.win32-py2.7.exe%23md5%3D57e1e64f6b7c7f1d2eddfc9746bbaf20&sa=D&sntz=1&usg=AFQjCNG-t1IrYYWs5Qp35sXtjV6aSLvVfA)]

Po zainstalowaniu pakietu setuptools należy uruchomić wiersz poleceń:



cmd /k cd c:\python27

set PATH=%PATH%;c:\Python27\Scripts



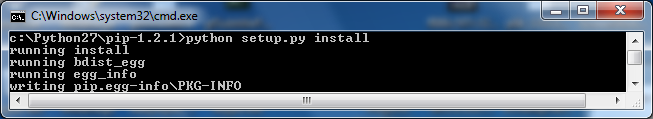
|  |
| --- |
| Uwaga! Polecenie set zmienia zmienną path tylko na czas aktywnej sesji. Zmienną systemową path można zmienić permamentnie używając polecenia setx.  Pierwszym krokiem jest zachowanie aktualnej zawartości zmiennej %PATH% dzięki poleceniu  setx PATHBACKUP "%PATH%"    Teraz należy zamknąć i uruchomić ponownie cmd:  cmd /k echo %PATHBACKUP%    setx PATH “%PATH%;c:\python27\scripts”  Dodanie ścieżki /scripts do zmiennej systemowej pozwoli na wygodne uruchamianie pakietu pip (bez podawania całej ścieżki c:\python27\scripts\pip) |

należy przejść do katalogu, w którym znajduje się rozpakowany pip oraz uruchomić instalację

cd c:\python27\pip-1.2.1\

python setup.py install

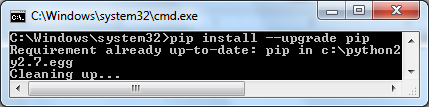
|  |
| --- |
| Uwaga! Jeśli system nie rozpoznaje polecenia python, należy dodać ścieżkę do katalogu zawierającego plik python.exe do zmiennej sys. PATH poleceniem:  set PATH=”%PATH%;c:\python27” |



### Aktualizacja i instalacja pakietów

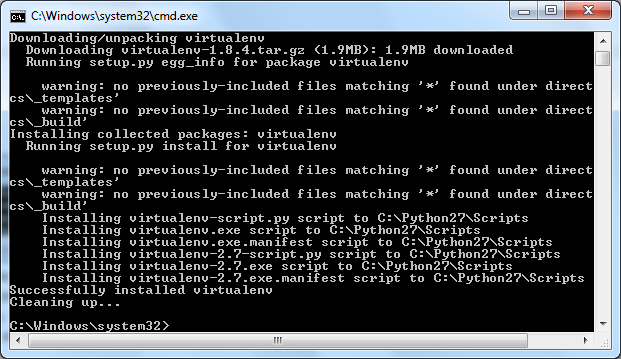
Pierwszym zalecanym krokiem jest aktualizacja managera PIP pleceniem

pip install --upgrade pip



Środowisko uruchomieniowe interpretera Python można zmieniać, dzięki czemu, jeśli na danym komputerze ktoś inny używa interpretera języka, z którego korzystamy, nie będziemy zaburzać jego pracy. Do tego potrzebny jest pakiet virtualenv dostępny za pomocą managera pip

pip install virtualenv



### Uruchomienie serwisu

Następnym krokiem jest włączenie dostarczonego przez autorów środowiska projektu

Należy przejść do katalogu, w którym zapisano projekt, w naszym wypadku będzie to

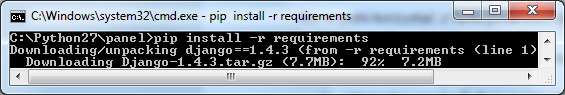
cd c:\python27\panel

w tym folderze znajduje się plik

requirements

Instalacja wymaganych pakietów przez projekt

pip install -r requirements



#### Uurchomienie bazy danych

Należy zainstalować bazę danych. <http://dev.mysql.com/downloads/mysql/>

Następnie dodać ścieżkę w której znajduje się plik mysql.exe do zmiennej PATH.

set MYSQLPATH=c:\mysql\bin

set PATH=%PATH%;%MYSQLPATH%

mysqld.exe --standalone --defaults-file="%MYSQLPATH%\my.ini"

#### Dodanie bazy danych

mysql -u root -p < database.sql

Jeśli nie nadano użytkownikowi root hasła, należy pozostawić pole puste.

|  |
| --- |
| Uwaga! Mysql rozpoznaje kodowanie UTF BOM (byte order mark), w razie komplikacji należy upewnić się, że takowe kodowanie wykorzystuje plik z zapytanie sql. |

Od tego momentu możliwe jest korzystanie z bazy danych.

|  |
| --- |
| Uwaga! Można upewnić się, że baza na pewno działa korzystając z interpretera mysql:  należy uruchomić polecenie  mysql  następnie wpisać  select user from mysql.user where user='bazy\_user';  Oczekiwany wynik polecenia:    Można opuścić interpreter mysql poleceniem  quit |

### Uruchomienie serwera HTTP

Serwer stron internetowych HTTP można uruchomić z katalogu /backend

cd /backend

python manage.py runserver 0.0.0.0:8080

start <http://localhost:8080>

## Linux: Ubuntu

Na Ubuntu wszystkie kroki można połączyć w jedną sekwencję komend:

|  |
| --- |
| sudo apt-get install git python-pip python-dev build-essential  git config --global user.name "Artur Zochniak"  git config --global user.email "arjamizo@gmail.com"  sudo pip install --upgrade pip  sudo pip install --upgrade virtualenv  git clone git@github.com:piotrgiedziun/bazy\_project.git  # polecenie wiersz niżej robi to samo co cd ./bazy\_project  cd `find . -type f -name 'requirements' |sed 's#\(.\*\)/.\*#\1#' |sort -u | head -n 1`  sudo pip install -r requirements  #virtualenv ./env  #ln -s ./env/bin/activate .  #source ./activate  # polecenie niżej realizuje to samo co cd ./backend  cd `find . -type f -name 'manage.py' |sed 's#\(.\*\)/.\*#\1#' |sort -u | head -n 1`  python manage.py runserver 0.0.0.0:8080 |

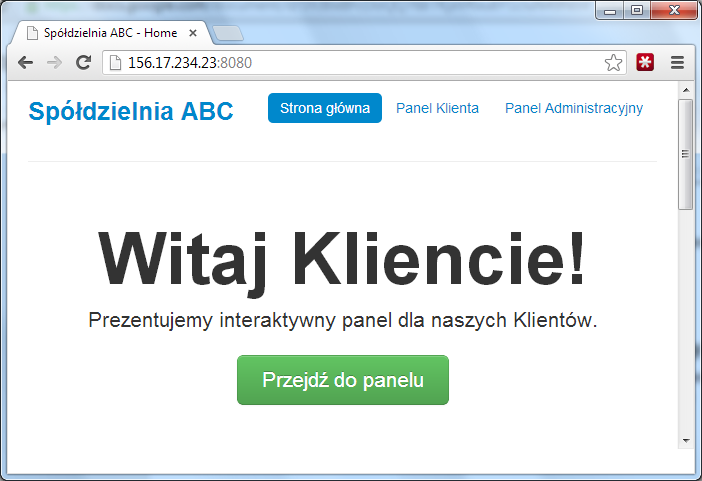
## Dodawanie przykładowych mieszkańców do bazy danych

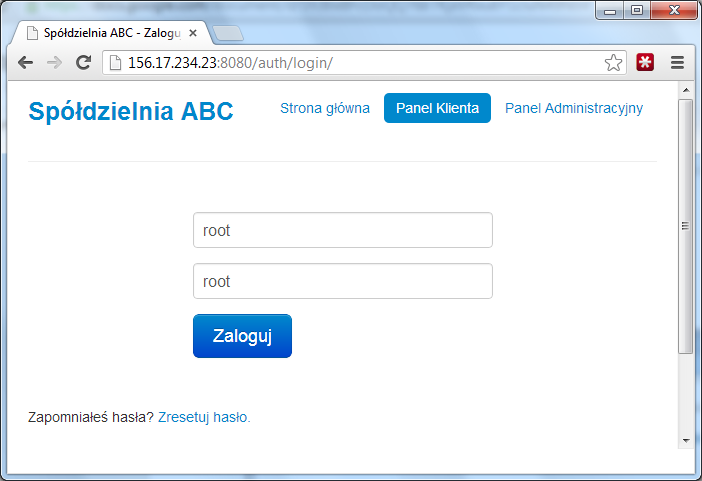
Zaimplementowano możliwość wypełnienia bazy danych losowymi mieszkańcami.

W tym celu należy uruchomić, będąc w katalogu backend/, polecenie

python manage.py populate

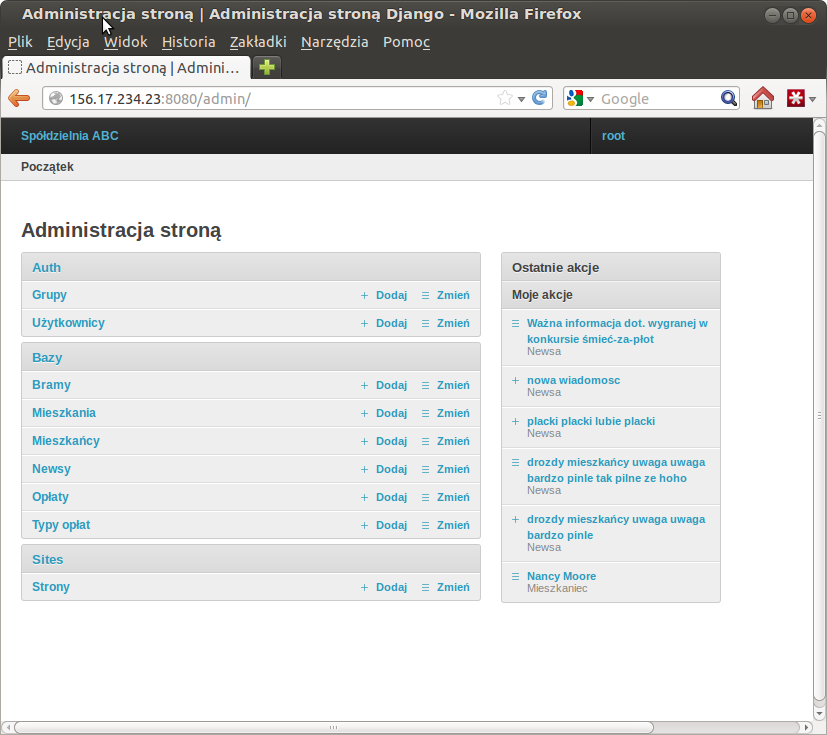
# Panel mieszkańca





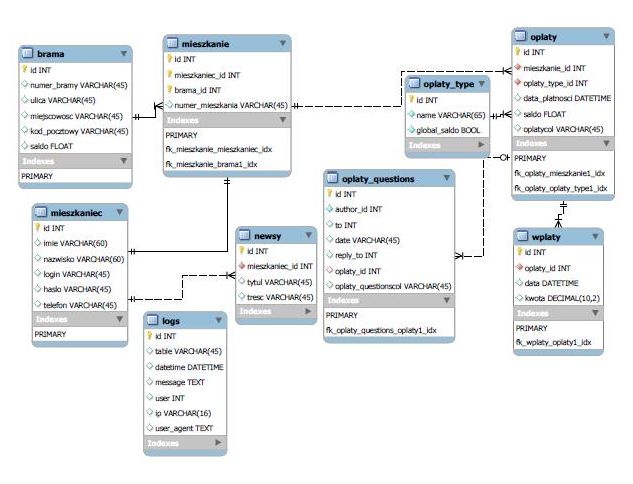
Testowe konto administratora oraz klienta dostępne jest pod loginem root oraz hasłem root.

# Panel administratora



# Omówienie schematu bazy danych

Baza danych składa się z następujących tabel:



## Brama

|  |  |
| --- | --- |
| pole | opis pola |
| id | unikalny identyfikator bramy; klucz podstawowy |
| numer\_bramy INT | numer bramy |
| ulica VARCHAR (200) | nazwa ulicy |
| miejscowośc | nazwa miejscowości w której znajduje się mieszkanie |
| poczta\_miasto | nazwa miejscowości w której znajduje się oddział pocztowy |
| kod\_pocztowy | kod pocztowy rozdzielony myślnikiem, np. 51-627 |
| saldo - do usunięcia | lepsze będzie dodanie nowego pola brama\_id w opłatach. |

# Mieszkanie

|  |  |
| --- | --- |
| pole | opis |
| id INT | unikalny identyfikator mieszkania; klucz podstawowy |
| mieszkaniec\_id INT | identyfikator reprezentanta (właściciela) danego mieszkania; klucz obcy do tabeli mieszkaniec |
| brama\_id INT | w której bramie znajduje się dane mieszkanie; klucz obcy do tabeli bramy |
| numer\_mieszkania INT | numer mieszkania |

## Mieszkaniec

|  |  |
| --- | --- |
| pole | opis |
| id | unikalny identyfikator mieszkańca; klucz podstawowy |
| imiona | imiona właściciela mieszkania |
| nazwisko | nazwisko właściciela |
| login | login umożliwiający zalogowanie się do panelu; pole unikatowy |
| haslo | hasło powiązane z loginem podczas logowania |
| telefon | telefon do danego mieszkańca |

## Oplaty\_type

|  |  |
| --- | --- |
| pole | opis |
| id | unikalny identyfikator typu opłaty |
| name | nazwa grupy opłat identyfikatorze id |

## Oplaty

Tabela przechowująca informacje o rachunkach, które mieszkańcy winni uregulować.

|  |  |
| --- | --- |
| pole | opis |
| id | unikalny identyfikator opłaty, klucz podstawowy |
| mieszkanie\_id | z którego mieszkania wpłacono pieniądze |
| oplaty\_type\_id | której kategorii dotyczyła wpłata, klucz obcy do tabeli oplaty\_type |
| data\_platnosci | kiedy wykonano wpłatę |
| saldo DECIMALFIeld, decimal\_places=2, max\_digits=100 | kwota, która została zapłacona, z dokładnością do (tylko) dwóch miejsc po przecinku |
| komentarz | opis opłaty + ewentualne jej uzasadnienie |

## Wpłaty

Tabela przechowująca wpłacone przez mieszkańców pieniądze. Wpłata co prawda zawiera informację pozwalającą przyporządkować ją do pewnej opłaty, jednak mieszkaniec może przelać dowolną kwotę, nawet wyższą niż wymagana.

|  |  |
| --- | --- |
| pole | opis |
| id | unikalny identyfikator wpłaty |
| oplata\_id | identyfikator nawiązujący z którą opłatą z tabeli opłaty dana wpłata jest powiązana; klucz obcy z tabą opłaty |
| data | data (dzień+godzina) wykonania wpłaty |
| kwota DECIMALFIELD(decimal\_place=2) | kwota w formacie XXXX.GG, gdzie X to cyfra złotówek, G to cyfra groszy |
| komentarz | komentarz opisujący wpłatę, np. uwagi dot. zbyt małej/dużej wpłaty osoby księgującej |

## Oplaty\_questions

|  |  |
| --- | --- |
| pole | opis |
| id | unikalny identyfikator pytania |
| mieszkaniec\_id | kto jest autorem danej wiadomości |
| oplata\_id\_to | której opłaty dotyczy pytanie; klucz obcy do tabeli opłaty |
| reply\_to\_question\_id MAY BE NULL | jeśli to pole nie ma wartości NULL, to dane pytanie jest kolejnym pytaniem (wiadomością) w wątku, którego wcześniejsza wiadomośc ma identyfikator wiadomości z tej samej tabeli |
| question TEXT | treść pytania dotycząca konkretnego pytania |

## Newsy

|  |  |
| --- | --- |
| pole | opis |
| id | unikalny identyfikator newsa, klucz podstawowy |
| mieszkaniec\_id | identyfikator autora wiadomości |
| title | tytuł wiadomości |
| content | treść wiadomości |
| date | data wysłania wiadomości |

## Logs

Tabela logs zawiera historię wszystkich modyfikacji przeprowadzonych w bazie danych.

|  |  |
| --- | --- |
| pole | opis |
| id | identyfikator logu, unikatowy; klucz podstawowy |
| table\_name | nazwa tabeli w której wprowadzono zmiany |
| datetime | czas (dzień + godzina) wprowadzenia zmiany |
| comment | które pola zostały zmienione, np. “KOMENTARZ Z: do zapłacenia NA: zapłacono |
| mieszkaniec\_id | kto wprowadził zmiany |
| ip VARCHAR (50) | adres IP z którego wprowadzono zmiany |
| user\_agent TEXT | informacje o przeglądarce osoby wprowadzającej zmiany |

# Wersja demo serwisu

Serwis jest dostępny pod adresem <http://156.17.234.23:8080> co najmniej do 25.01.2012r., do godziny 23:59.

# Bibliografia

[1] http://dev.mysql.com/doc/refman/5.0/en/batch-commands.html