Środowisko pracy

|  |  |
| --- | --- |
| Wykonawca: | Hubert Wenderski 15.03.2015 |
| Sprawdzający: | Szymon Długosz 16.03.2015 |
| Zatwierdzający: | Mateusz Okulewicz 23.03.2015 |

**Członkowie grupy:**

Długosz Szymon  
Juźwiak Mateusz  
Lewandowski Marcin  
Okulewicz Mateusz – szef grupy  
Wenderski Hubert

Komentarze:

* 15.03.2015 Hubert Wenderski
  + Dokumentacja niekompletna, brakuje: konfiguracji AngularJS oraz Doxygena
* 16.03.2015 Szymon Długosz
  + " Ze względu na niekomercyjny aspekt projektu": aspekt - zamieniono na charakter
  + Wprowadzono cudzysłowy do pojedynczych nazw folderów i plików (bez ścieżek)
  + "Ta część została opisana w pkt. 4 Serwer deweloperski": 4 - zamieniono na poprawną wartość (stworzony automatyczny odnośnik)
  + Wprowadzono drobne poprawki i nterpunkcyjne i gramatyczne
  + "oraz opisanym wcześniej repozytorium Git’a.": wcześniej - zamieniono na punkt
  + "Uruchomienie serwera na localhoscie" - dopisano "(po przejściu do katalogu z pobranym repozytorium)"

1. Wstęp

Podstawą niniejszego projektu jest integracja dwóch części systemu wykorzystujących różne środowiska i technologie programistyczne:

* Frontendu – Fasady, służącej do komunikacji z użytkownikiem, klientem. Opartej na technologii JavaScript oraz Frameworku AngularJS ([https://angularjs.org/](https://angularjs.org/))).
* Backendu – Wnętrza, głównej części systemu, odpowiadającej za analizę i przetwarzanie danych. W tym przypadku wykorzystane zostanie środowisko oraz język Python przy udziale Frameworku Django (<https://www.djangoproject.com/>).

1. Repozytorium

W celu nadzoru procesu deweloperskiego wykorzystany został rozproszony system kontroli wersji, repozytorium Git’a. Ze względu na niekomercyjny charakter projektu repozytorium jest publiczne, umieszczone na popularnym serwisie hostingowym GitHub (<https://github.com/>). Repozytorium znajduje się pod adresem <https://github.com/SirWojtek/Coolinaria.git>. Wszyscy członkowie projektu mają pełne prawa dodawania, usuwania oraz edycji plików znajdujących się w repozytorium. Wiele informacji na temat systemu kontroli wersji Git oraz sposobów jego użytkowania można znaleźć pod adresem (<http://git-scm.com/documentation>). Struktura katalogów repozytorium odzwierciedla wcześniej opisany podział aplikacji. W folderze "backend" znajduję się projekt Django, zaś folder "frontend" posiada potrzebne pliki do tworzenia aplikacji w AngularJS. Przygotowano również folder "doc", w którym automatycznie tworzona będzie dokumentacja Doxygen (<http://www.stack.nl/~dimitri/doxygen/manual/index.html>)

1. Projekt Django

Poprawne uruchomienie systemu wymaga instalacji języka Python oraz Fremeworka Django w wersji 1.7.6. Poniżej przedstawiona została pełna instalacja przeprowadzona na systemie Ubuntu 14.04.2 LTS posiadającego Pythona w wersji 2.7.6

sudo apt-get install python-pip  
sudo pip install Django==1.7.6

Kolejny etap to stworzenie i skonfigurowanie nowego projektu, w tym celu należy użyć polecenia.

sudo mkdir backend  
sudo django-admin.py startproject config backend/

Najważniejsze pliki projektu znajdują się w katalogu backend/config/. Plik "settings.py" pozwala na wprowadzenie większości potrzebnych ustawień:

* Dodanie nowych modułów, aplikacji Django po przez zmianę wpisów zmiennej INSTALLED\_APPS
* Konfiguracja bazy danych po przez wpis do zmiennej DATABASES. W projekcie użyto prostej, plikowej bazy danych SQLite.

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

}

}

Plik "urls.py" pozwala na dodawanie nowych adresów URL systemu oraz aplikacji, które dany adres będą obsługiwały. Zmian dokonuje się po przez wpis do zmiennej urlpatterns. Plik wsgi.py zawiera konfigurację serwera oraz zapewnia komunikację z serwerem webowy. Ta część została opisana w pkt. 6 Serwer deweloperski.

* 1. Makefile

Dodatkowo do projektu dodany został plik Makefile, który ułatwia kompilację, synchronizację bazy danych oraz uruchomienie serwera. Możliwe jest wykonanie całego procesu uruchomieniowego po przez wpisanie polecenia make lub też wybranych jego etapów:

* make clean – wyczyszczenie katalogu roboczego z plików wykonywalnych oraz usunięcie bazy danych
* make fresh\_syncdb – utworzenie oraz synchronizacja nowej bazy danych
* make syncdb – synchronizacja, uaktualnienie bieżącej bazy danych
* make shell – uruchomienie interaktywnej powłoki Pythona z biblioteką Django
* make run – uruchomienie serwera deweloperskiego z aplikacją na localhostcie i porcie 8080

1. Projekt AngularJS

Ten fragment dokumentacji zostanie uzupełniony po zakończeniu prac dot. projektu AngularJS.

1. Doxygen

Ten fragment dokumentacji zostanie uzupełniony po zakończeniu prac dot. narzędzia Doxygen.

1. Serwer deweloperski

Serwer deweloperski został stworzony w celu uruchomienia i integracji kolejnych etapów pracy nad systemem. Środowisko bazuje na serwerze webowym Apache2 oraz opisanym w punkcie 2 repozytorium Git’a. Poniżej przedstawione zostały kolejne kroki jakie zostały wykonane w celu uruchomienia Sewera.

* Instalacja Apache2

sudo apt-get install apache2

* Instalacja rozszerzenia "wsgi"

sudo apt-get install libapache2-mod-wsgi

* Utworzenie linku symbolicznego w katalogu /var/www do folderu backend/ znajdującego się w repozytorium /opt/coolinaria

sudo ln -s /opt/coolinaria/backend/

* Konfiguracja wirtualnego hosta na serwerze Apache2. Zmiany wykonane w pliku /etc/apache2/ sites-enabled/000-default.conf

<VirtualHost \*:80>

ServerName backend

DocumentRoot /var/www

WSGIScriptAlias /backend /var/www/backend/config/wsgi.py

ErrorLog ${APACHE\_LOG\_DIR}/backend-error.log

</VirtualHost>

* Konfiguracja pliku "wsgi.py".

import os, sys

apache\_configuration = os.path.dirname(\_\_file\_\_)

project = os.path.dirname(apache\_configuration)

workspace = os.path.dirname(project)

sys.path.append(workspace)

sys.path.append('/var/www/backend/')

* Uruchomienie serwera Apache2

sudo service apache2 start

1. Uruchomienie aplikacji

Założeniem projektu jest, iż każdy użytkownik pracuje z systemem na jego lokalnej kopi, zaś integracja wprowadzanych zmian będzie możliwa przy wykorzystaniu systemu kontroli wersji Git. Poniżej opisane zostały kroki, jakie należy wykonać w celu rozpoczęcia pracy z repozytorium na systemie Ubuntu.

* Instalacja Git’a

sudo apt-get update  
sudo apt-get install git

* Pobranie lokalnej kopi repozytorium

git clone https://github.com/SirWojtek/Coolinaria.git [nazwa katalogu]

* Uruchomienie serwera na localhoscie (po przejściu do katalogu z pobranym repozytorium)

make