

Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Розрахунково-графічна робота

з курсу «Інтеграція програмних систем»

Виконали:
студенти 4-го курсу
ФІОТ групи ІО-32
Команда(slack) dreamteam:
Попенко Руслан
Морозов Максим
Змеул Євгеній
Павлючков Владислав

1. Опис проекту

Розроблена нами Digitized School дозволяє адміністрації проглядати та редагувати всю шкільну систему, яка потрібна для функціонування навчального процесу. В проекті доступні дії з шістьма основними сутностями: Person (описує повне ім'я, вік та стать усіх людей, що наявні в проекті), Pupil (реалізація учня школи, поєднує персону учня з його школою та класом), School (робота з усіма наявними школами в системі), SchoolClass(реалізація шкільного класу, прив'язка класу до учнів, що в ньому навчаються, та предмету), Subject(усі шкільні предмети доступні в системі), Teacher(створення профілю шкільного вчителя). Для зручності користування був реалізований інтерфейс, що зображений на Рис 1.1.

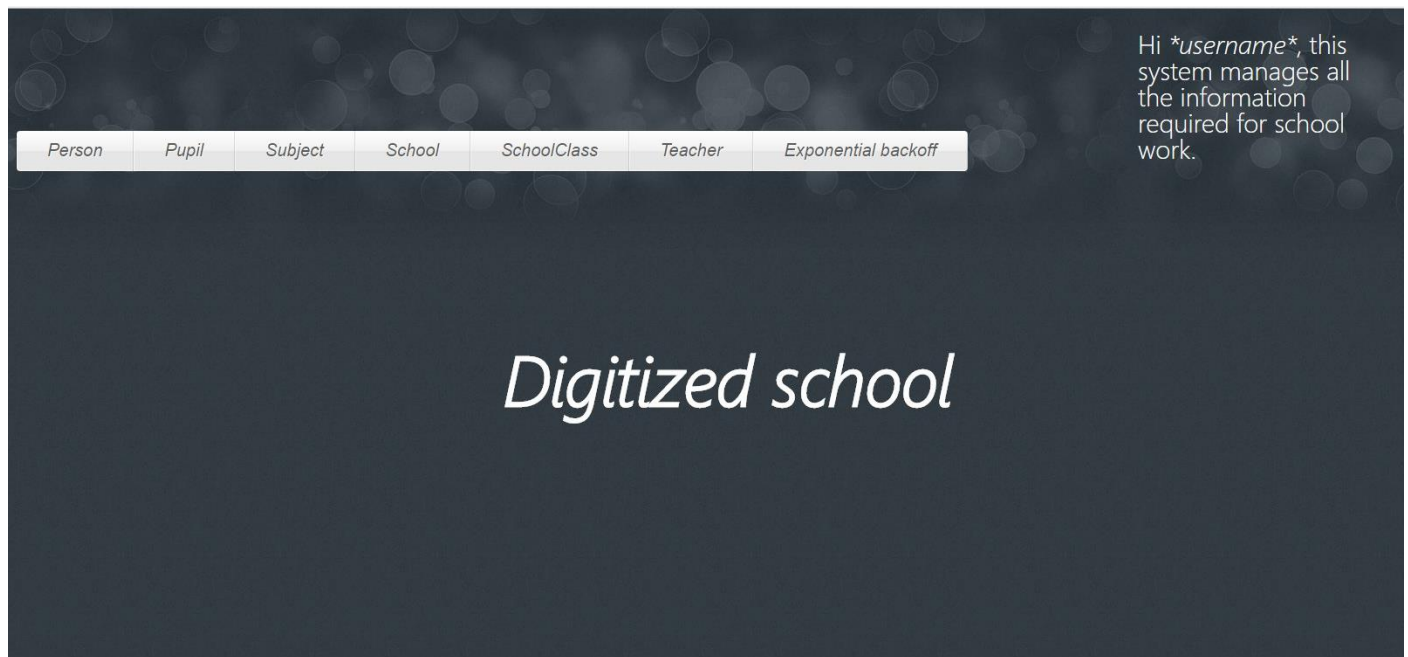


Рис 1.1 Скріншот роботи сервісу та перелік доступних сутностей.

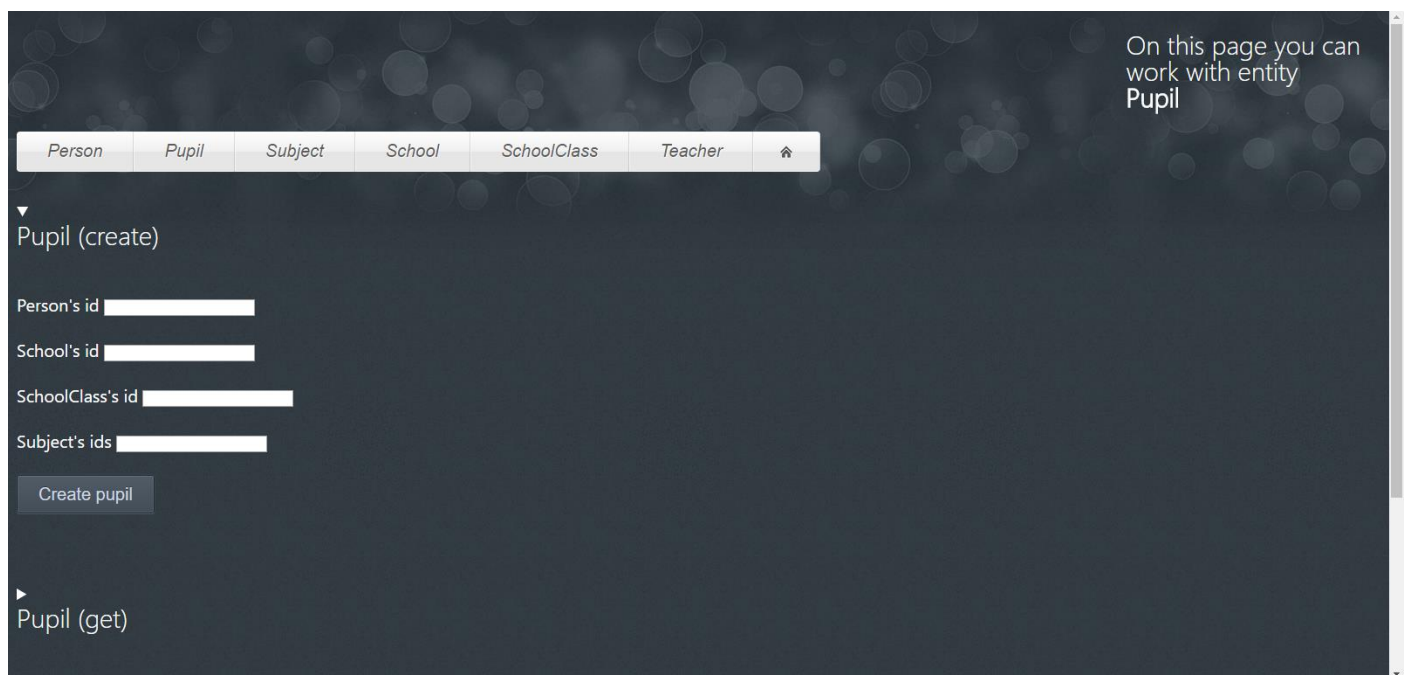


Рис 1.2 Скріншот сторінки сутності Pupil та перелік доступних методів для роботи з нею.

Серверна частина розроблена за допомогою REST контролерів Spring MVC, в основі яких лежить метод, що визначає потрібні URL для кожної сутності та пересилає відповідний JSON об'єкт на наш сайт. Потім він мапиться на java об'єкт та, використовуючи Hibernate, зберігає отримані данні до бази даних.

2. Система атоматичної збірки. Maven

Maven - це засіб автоматизації роботи з програмними проектами, який спочатку використовувався для Java проектів. Використовується для управління (management) та складання (build) програм. Створений Джейсоном ван Зилом (*Jason van Zyl*) у 2002 році. За принципами роботи кардинально відрізняється від Apache Ant, та має простіший вигляд щодо build-налаштувань, яке надається в форматі XML. XML-файл описує проект, його зв'язки з зовнішніми модулями і компонентами, порядок будівництва (build), папки та необхідні плагіни. Сервер із додатковими модулями та додатковими бібліотеками розміщується на серверах. Раніше Maven, де він був частиною *Jakarta Project*.

Для опису програмного проекту який потрібно побудувати (*build*), Maven використовує конструкцію відому як Project Object Model (POM), залежності від зовнішніх модулів, компонентів та порядку побудови. Виконання певних, чітко визначених задач - таких, як компіляція коду та пакетування відбувається шляхом досягнення заздалегідь визначених цілей (*targets*).

Ключовою особливістю Maven є його мережева готовність (*network-ready*).

Двигун ядра може динамічно завантажувати плагіни з репозиторію, того самого репозиторію, що забезпечує доступ до багатьох версій різних Java-проектів з відкритим кодом, від Apache та інших організацій та окремих розробників.

Maven забезпечує підтримку побудови не просто перебираючи файли з цього репозиторію, але й завантажуючи назад артефакти у кінці побудови. Локальний кеш звантажених артефактів діє як первісний засіб синхронізації виходу проектів на локальній системі.

3. Сервер безперервної інтеграції. Travis-ci

Як виявилося, термін «continuous integration» досить старий. Він був введений Мартіном Фаулером (Martin Fowler) у 2000-му році і викладений у статті «Continuous Integration» і по-російськи звучить як «безперервна інтеграція». Це частина процесу розробки, в якій розробляється проект збирається / тестується в різних середовищах виконання автоматично і безперервно. Задумувалася дана методика для найбільш швидкого виявлення помилок / протиріч інтеграції проекту, а отже зниження витрат на наступні прості.

Принцип досить простий: на окремій машині працює якась служба, в обов'язки якої входить отримання вихідного коду проекту, його збірка, тестування, логування, а також можливість надати для аналізу дані виконання перерахованих операцій.

Виділити окремий сервер і підтримувати його в робочому стані, забезпечити наявність необхідних програмних комплексів, налаштувати середовища виконання, робити резервні копії даних і т.д... Все це вимагає чимало часу і ресурсів. І цілком логічним здається можливість делегувати цю відповідальність на сторонні сервіси. От якраз таким і є travis-ci - «хостинг безперервної інтеграції для open source співтовариства». Настав час подивитися на нього ближче.

Travis-ci підтримує безліч мов програмування. Почати користуватися сервісом дуже просто. Потрібно всього лише настроїти спеціальний файл конфігурації *.travis.yml*.

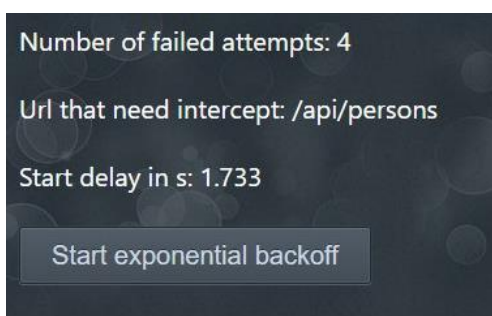
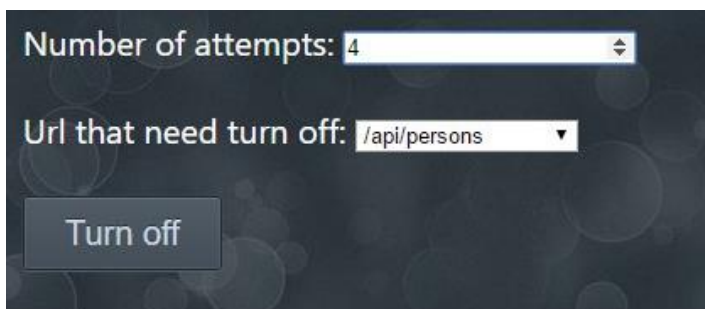
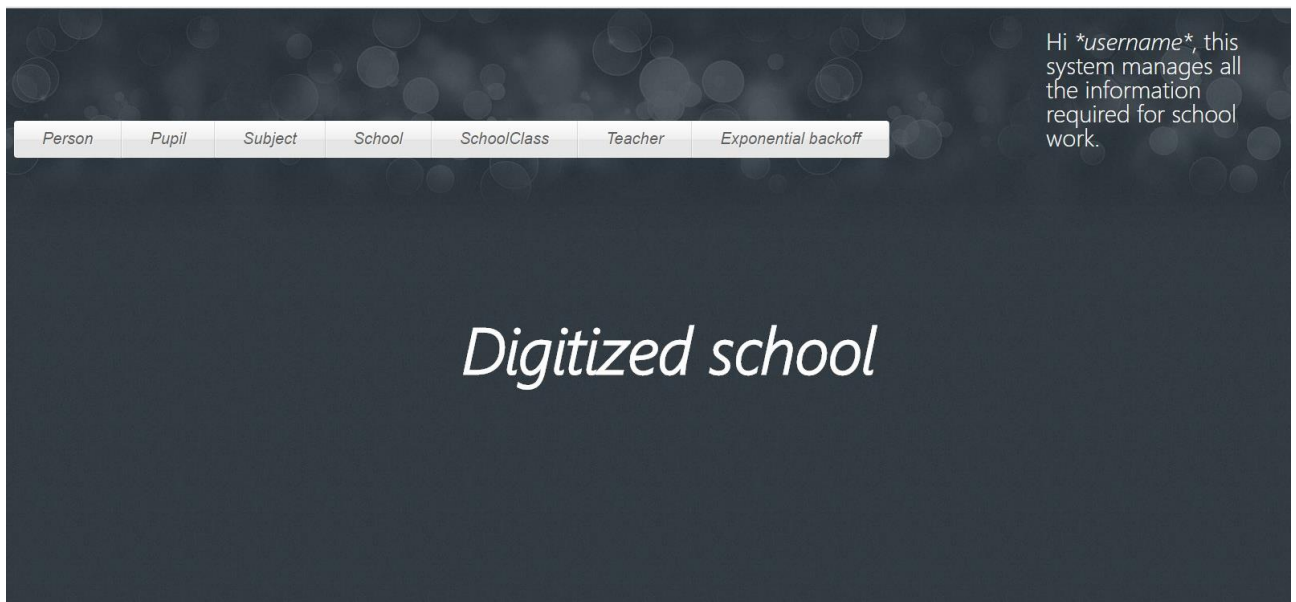
Задачі, які вирішуються на сервері безперервної інтеграції:

- запуск тестів для перевірки написаного коду:
 1. Звичайних тестів (коли явно перевіряються куски програмного коду без залежностей);
 2. З використанням моків (тестів, де залежності підміняються на якийсь відомий результат, наприклад, доступ в бд і отримання сутності замінюється на простий наперед заданий об'єкт);
 3. Інтеграційних тестів (коли запускається частина аплікейшена, або він весь і йде тестування схоже до реальної роботи програми).
- якщо тести успішні, то виконується збірка проекту;
- після успішної збірки проект може бути задеплоїний на сервер.

4. Експоненціальна витримка

Для вирішення задачі раптової помилки на сервері було вирішено задачу експоненціальної витримки. Для того щоб не відключати сервер був написаний [хендлер інтерсептор](#).

Він конфігурується за допомогою змінних сесії, які виставляються на стороні клієнта:



Формула витримки складеться лише з двох параметрів. Мінімальний час у степені кількості спроб. Тому кожен раз з клієнта робиться запит до сервера, поки не буде статус < 400. В інакшому випадку продовжуються запити на АПІ, в данному випадку get all persons.

