***Воробій А.О., Артур, Денис***

Зміст

[Вступ 3](#_Toc361156026)

[1. Google App Engine для Java додатків. Огляд можливостей та технологій 4](#_Toc361156027)

[2. GAE сервіси. Стандартні функції API 6](#_Toc361156028)

[3. Встановлення плагіну Google Tools для Eclipse IDE 7](#_Toc361156029)

[4. Реєстрація додатку у Google App Engine та запуск додатку 8](#_Toc361156030)

[5. Збереження даних (JPA) у додатку та створення прототипу додатку 10](#_Toc361156031)

[6. Керування поштою та виконання задач за графіком 11](#_Toc361156032)

[7. Використання журналів подій та консолі адміністратора 13](#_Toc361156033)

[Висновок 14](#_Toc361156034)

[Список використаних джерел 15](#_Toc361156035)

# Вступ

Одна з головних вимог до підготовки фахівців – її зв'язок з практикою, знання конкретних завдань, особливостей майбутньої практичної діяльності. Важливою ланкою цього зв'язку є технологічна практика. Цілі і завдання технологічної практики - це отримання і закріплення знань по роботі з певними програмами, набуття практичних навичок.

Завдання технологічної практики передбачає роботу з Google App Engine.

**Google App Engine** - це платформа Google для розробки та хостингу. Google App Engine дозволяє створювати продуктивні веб-додатки без необхідності управління інфраструктурою з великим трафіком. У додатках, що створюються на базі Google App Engine, використовується технологія, завдяки якій веб-сайти Google працюють швидко і надійно. GAE надає набір інструментів і засобів управління, які дозволяють безболісно створювати й викладати програми.

# Google App Engine для Java додатків. Огляд можливостей та технологій

Google випустили перший реліз платформи Google App Engine в квітні 2008 р. Проте до глибокого розчарування Java-розробників, початкова версія платформи призначалася виключно для розробників додатків на Python, тобто людей, які впевнені, що кращий спосіб виділення блоків коду полягає у використанні проміжків і знаків табуляції. На щастя, за численними проханнями Google усунув цей недолік, випустивши версію платформи для Java в квітні 2009 р.

Google App Engine для Java надає повний набір засобів створення корпоративних Java-систем: простий у використанні графічний інтерфейс на основі Ajax, модулі для Eclipse, а також серверну частину App Engine. Легкість застосування та широкий набір готових засобів розробки є важливими перевагами даної платформи та інших рішень на основі хмарних обчислень.

Створення додатку за допомогою App Engine передбачає використання ресурсів Google для зберігання та вибірки об'єктів Java-класів. Зберігання даних реалізовано на основі системи BigTable, однак інтерфейси JDO і JPA дозволяють писати повністю незалежний від неї код. Більше того, Google підтримує багато стандартизованих API, тому код вашого додатку не буде на 100% прив'язаний до платформи Google App Engine.

Робота App Engine базується на наступних стандартних API:

* java.net.URL для звернення до мережевих сервісів поверх протоколів HTTP and HTTPS;
* JavaMail для відправки поштових повідомлень;
* інтерфейсі JCache (JSR 107) для Memcache, що надає швидке, розподілене сховище тимчасових даних, які використовуються при кешуванні результатів запитів і обчислень.

Крім того, App Engine підтримує такі прикладні сервіси:

* авторизація та аутентифікація користувача;
* CRON;
* імпорт / експорт даних;
* доступ до даних мережевих екранів.

Можливість імпорту і експорту даних відіграє важливу роль при перенесенні даних з інших джерел у додаток на основі App Engine. Це ще один приклад того, як ви можете знизити залежність додатків від даної платформи. Підтримка CRON полягає в можливості переходу по внутрішньому URL відповідно до заданого розкладу. Це досить зручний сервіс, до того ж не дуже сильно прив'язаний до App Engine. Механізм аутентифікації і авторизації, у свою чергу, є специфічним для даної платформи, проте ніщо не заважає вам створити спеціальний фільтр сервлетів, аспект або власний модуль Spring Security для абстрагування від App Engine.

# GAE сервіси. Стандартні функції API

Google App Engine надає кілька сервісів, що дозволяють виконувати основні операції керування програми. Наступні API забезпечують доступ до служб:

**URL Fetch**

За допомогою сервісу App Engine URL Fetch додаток може отримати доступ до ресурсів в Інтернеті, наприклад, до веб-служб або інших даних. Служба URL Fetch отримує веб-ресурси за допомогою високошвидкісної інфраструктури Google, отримуючи веб-сторінки для різних продуктів Google.

**Mail**

Додатки можуть відправляти повідомлення електронної пошти за допомогою служби Mail Сервера додатку. Для відправки повідомлень електронної пошти служба Mail використовує інфраструктуру Google.

**Memcache**

Служба Memcache надає вашому додатку високопродуктивний кеш пам'яті зі структурою "ключ-значення", до якого можуть отримувати доступ кілька екзеплярів додатку. Служба Memcache також може бути використана для даних, при обробці яких не потрібно використовувати функції постійного зберігання та транзакційні функції сховища даних, наприклад, тимчасові дані або дані, скопійовані зі сховища даних в кеш для прискорення доступу.

**Обробка зображень**

Служба Image дозволяє додатку працювати із зображеннями. За допомогою API цієї служби можна змінювати розмір, кадрувати, повертати і отримувати дзеркальне відображення зображень у форматах JPEG і PNG.

# Встановлення плагіну Google Tools для Eclipse IDE

Для розробки Java-додатку на Google App Engine радять використовувати середовище розробки Eclipse, так само, як і для розробки будь-якого веб-додатку, заснованого на сервлетах. З плагіном для Eclipse, Google - це ще простіше. Плагін дозволяє створювати, тестувати і завантажувати додатки з GoogleApp Engine в Eclipse.

Наш додаток розроблявся в Eclipse 3.7 (Indigo). Для встановлення плагіну для цієї версії :

1. Оберіть меню **Help** > **Install New Software**....
2. В текст-боксі **Work with**, введіть:



1. Натисніть кнопку Add... . В діалоговому вікні натисніть OK.
2. Натисніть на значок плюса поруч з " Google Plugin for Eclipse" і "SDKs". Встановіть прапорці поруч з " Google Plugin for Eclipse 3.7" і "Google App Engine Java SDK". Ви також можете вибрати "Google Web Toolkit SDK", якщо ви хотіли б використовувати Google Web Toolkit з вашими додатками. Переконайтеся, що ви відмітили прапорцем «Contact all update sites during install to find required software." Натисніть кнопку Next. Дотримуйтесь інструкцій, щоб прийняти умови надання послуги та встановити плагін.
3. Коли інсталяцію буде завершено, Eclipse запропонує перезавантажити. Натисніть кнопку Yes. Eclipse перезавантажується. Плагін встановлений.

# Реєстрація додатку у Google App Engine та запуск додатку

Додаток створюється та керується в App Engine веб за допомогою консолі адміністратора за наступною адресою:

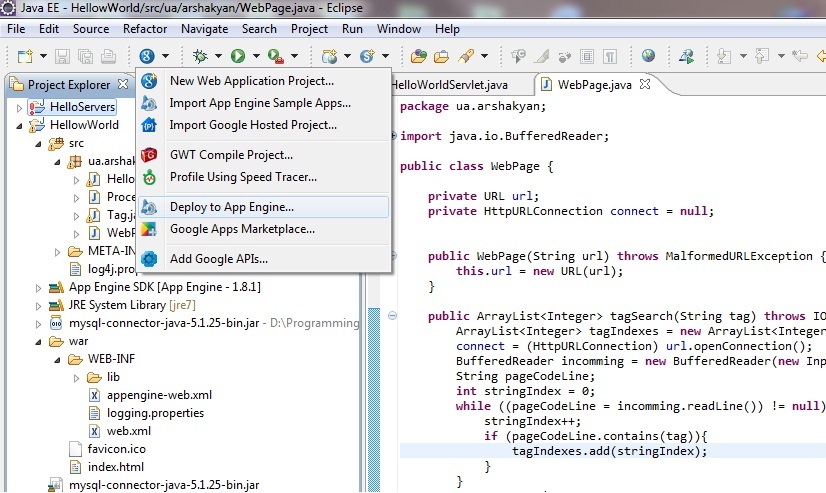
<https://appengine.google.com/>

До App Engine можна зайти використовуючи акаунт Google. Якщо у Вас немає акаунту Google, ви можете створити акаунт Google з адресою електронної пошти та паролем.

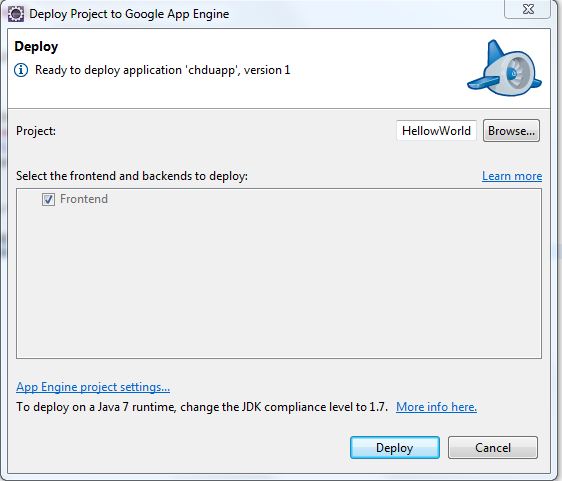
Щоб створити новий додаток треба натиснути кнопку "**Create an Application**". Далі треба дотримувать інструкцій, щоб зареєструвати ID додатку, ім'я, унікальні для цього додатка. Якщо ви вирішили використовувати вільне доменне ім'я appspot.com, повне посилання на додаток буде <http://your-app-id.appspot.com/>. Ви також можете придбати домен верхнього рівня ім'я для вашої програми, або використовувати той, який ви вже зареєстрували.

Для того, щоб розгорнути свій додаток:

1. Натиснути "**Deploy to App Engine…**" в меню



1. Увійти в акаунт Google та натиснути "**Deploy**"



1. Якщо все нормально, то ваш додаток буде розгорнутий за адресою буде <http://your-app-id.appspot.com/>



# Збереження даних (JPA) у додатку та створення прототипу додатку

Java Persistence API (JPA) - це стандартний інтерфейс для зберігання об'єктів, що містять дані у реляційній базі даних. Стандарт визначає інтерфейси для анотації об'єктів Java, витягання об'єктів з використанням запитів та взаємодії з базою даних з використанням транзакцій. Додаток, в якому застосовується інтерфейс JPA, може працювати з різними базами даних без використання коду бази даних певного виробника. JPA спрощує взаємодію додатків з базами даних різних виробників.

Java SDK для Серверу додатків включає реалізацію JPA 1.0 для сховища даних Серверу додатків. Реалізація заснована на платформі DataNucleus Access Platform. JPA представляє собою стандартний інтерфейс для взаємодії з реляційною базою даних, однак сховище даних Серверу додатків не є реляційною базою даних.

Дані, які зберігаються, можна відображати у діаграмах. Кругову діаграму можна відобразити в браузері за допомогою SVG або VML. Для побудови діаграми потрібно брати дані з бази даних, але оскільки вони в Google App Engine є платними, то можна записувати дані у файл і потім, зчитуючи їх, будувати діаграму. Для візуалізації даних потрібно користуватися бібліотекою google.visualization. Ця бібліотека визначає всі основні класи та функції, що є необхідними для побудови діаграм.

# 6. Керування поштою та виконання задач за графіком

Додатки App Engine можуть відправляти повідомлення електронної пошти від імені адміністраторів додатків, а також від імені користувачів з акаунтів Google. Додатки можуть отримувати електронну пошту за різними адресами. Додатки відправляють повідомлення, використовуючи поштову службу, і отримують повідомлення у вигляді HTTP-запитів, які відправляються Google App Engine, і розміщуються в додатку.

Для відправки повідомлень електронної пошти від імені адміністратора програми або поточного користувача додаток використовує службу Mail. При надсиланні повідомлень електронної пошти використовується інтерфейс JavaMail.

Сервіс Mail1 може відправляти повідомлення електронної пошти одному або декільком одержувачам. Повідомлення містить заголовок, текст, й інколи HTML-код. Він також може містити вкладені файли, а також обмежений набір заголовків.

В цілях безпеки адреса відправника повідомлення повинна бути адресою електронної пошти адміністратора додатку або будь-якою дійсною адресою електронної пошти, що зв’язана з додатком.

Отримувати електронну пошту додаток може за адресою наступного виду: string@appid.appspotmail.com

Слід зазначити, що навіть якщо ваш додаток розгортається на власному домені, ваш додаток не може отримувати пошту, що відправляється на адресу цього домену.

Повідомлення електронної пошти відправляються до вашому додатку, як HTTP-запити. Ці запити генеруються App Engine, і розміщуються у вашому додатку. У конфігурації вашої програми, ви вказуєте обробники подій, які будуть викликатися для обробки цих HTTP запитів. В цих обробниках ви отримуєте дані для повідомлень електронної пошти, які ви потім розбираєте на окремі поля.

Повідомлення електронної пошти відправляються до вашого додатку, як HTTP POST запити, використовуючи наступний URL:

/ \_ah / Mail / адреса

де адресою є повна адреса електронної пошти, включаючи доменне ім'я. Можливість прийому пошти в додатку за замовчуванням відключена. Для того щоб ваш додаток приймав пошти необхідно вказати, що ви хочете, щоб ця послуга включена у вашому appengine-web.xml файлі.

Як будь-який поштовий сервіс Mail має свою квоту. Це обмеження можна подивитись на сторінці **Quota Details** в консолі адміністратора.

За допомогою служби Cron на сервері App Engine можна створювати заплановані завдання, що виконуються в певний час або регулярно через певні періоди. Ці завдання називаються завданнями **cro**n. Виконання завдань cron здійснюється автоматично за допомогою служби Cron на сервері App Engine. Наприклад, цю функцію можна використовувати для щоденної відправки звіту електронною поштою, оновлення кешованих даних через кожні 10 хвилин або оновлення зведеної інформації раз на годину.

За допомогою GET-запиту HTTP завдання cron викликає деякий URL в заданий час. Викликаний службою Cron URL підкоряється тим же обмеженням і квотами що і при звичайному запиті HTTP, включаючи обмеження на час запиту. Завдання cron виконуються тільки для версії програми за замовчуванням. У додатку може бути до 20 запланованих завдань.

Консоль адміністрування дозволяє переглядати стан виконання завдань cron. Щоб переглянути стан виконання завдань, у тому числі час останнього виконання і його результати, виберіть посилання "Завдання Cron" в боковому меню.

Час додавання або видалення завдань cron можна переглянути на сторінці "Журнали адміністратора" з меню консолі адміністратора.

Сервер розробки додатків не виконує завдання Cron автоматично.

# Використання журналів подій та консолі адміністратора

Консоль адміністора Google App Engine надає повний доступ до загальнодоступних версій програми. У консоль можна перейти за наступним посиланням у веб-браузері:

<https://appengine.google.com/>

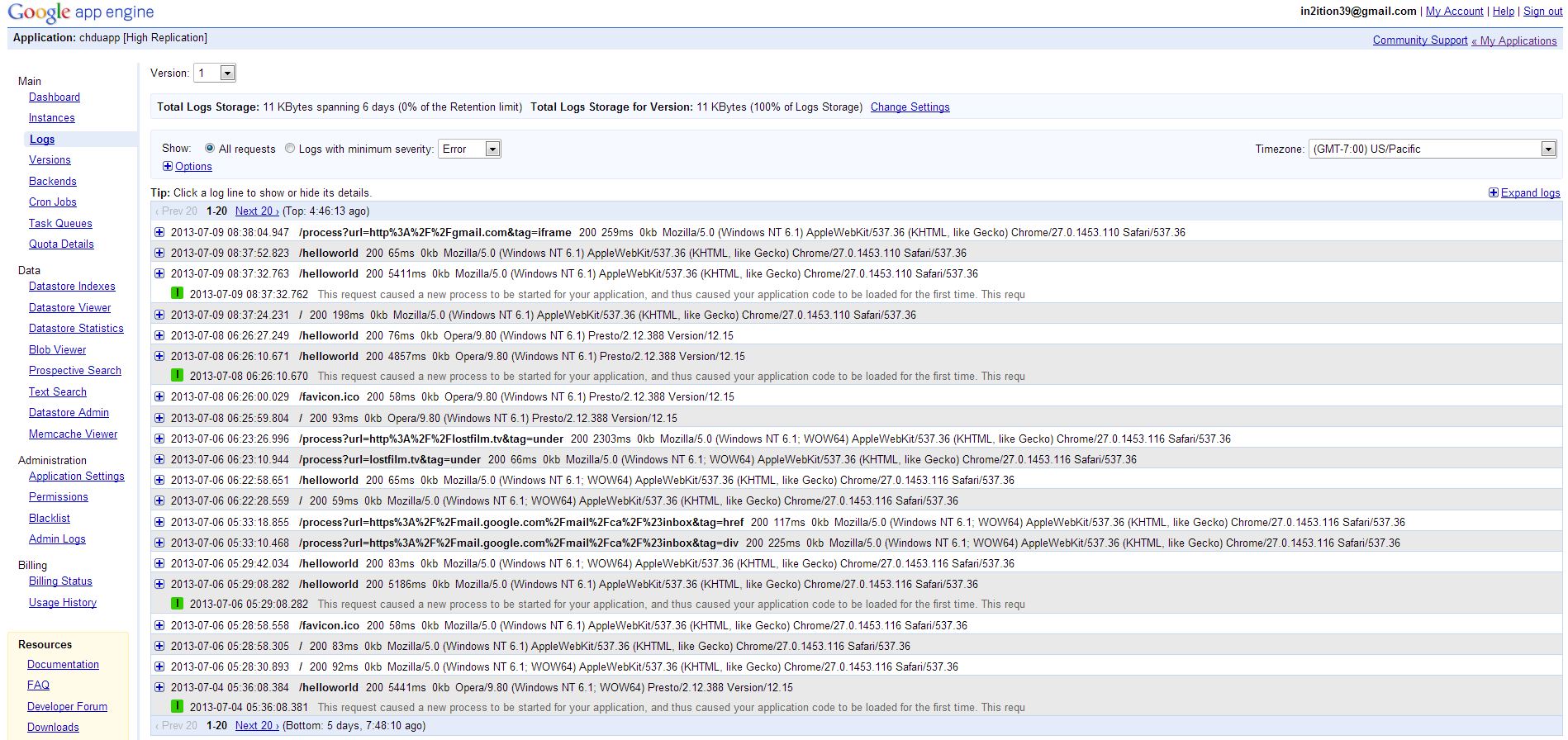
Далі треба виконати вхід з акаунтом Google або створити новий акаунт за допомогою адреси електронної пошти та пароля.

Якщо App Engine використовується спільно з акаунтом Google Apps, можна увійти в App Engine в домен, використовуючи URL-адресу, що знаходиться нижче, де domain.com є вашим доменом Google Apps:

<https://appengine.google.com/a/domain.com>

Консоль адміністратора можна використовувати для:

* створення нових додатків, настройки безкоштовного субдомену appspot.com або обраної вами назви домену верхнього рівня;
* запрошення інших людей в якості розробників програми з метою надання їм доступу до консолі і можливості завантажувати нові версії коду;
* перегляду сховища даних програми та управління індексами;
* управління чергами завдань та окремими завданнями з можливостями зупинення виконання черг, очищення і видалення черг;
* тестування нових версій програми та зміни відображуваної версії для користувачів;
* перегляду даних і журналу помилок, а також для аналізу трафіку:



# Висновок

Отже, під час виконання практики я отримала та закріпила свої знання про роботу з Google App Engine. Вважаю практику такого виду корисною, оскільки студентам необхідно знайомитися та працювати з новими технологіями, що дозволяють оптимізувати та пришвидшити роботу додатків, зробити їх більш надійними.

# Список використаних джерел

1. <https://google-developers.appspot.com/>
2. <https://appengine.google.com/>
3. <https://developers.google.com/appengine/?hl=ru>
4. <http://www.ibm.com/developerworks/ru/library/j-gaej1/>
5. <http://www.mkyong.com/google-app-engine/>
6. Філософія Java: Брюс Еккель — Москва, Пітер, 2009 р.
7. Java сервлети та JSP: Брюс У. Перрі — Москва, КУДИЦ-Прес, 2009 р.