

12.–15.09.2010  
in Nürnberg



# Herbstcampus

Wissenstransfer  
par excellence

## Ab in die Wolken

Java in der Cloud - Eine Einführung in Google App Engine / Java

Dr. Halil-Cem Gürsoy

adesso AG

## Der Referent

---

- Insgesamt über 10 Jahre Beratung und Entwicklung rund um Java, davor Entwicklung im Forschungsumfeld
- Senior Software Engineer bei adesso AG, Dortmund
- Schwerpunkt EAI und SOA-Projekte im JEE Umfeld, aber auch klassische JEE-Projekte
- Autor und Referent auf Konferenzen

## Agenda

---

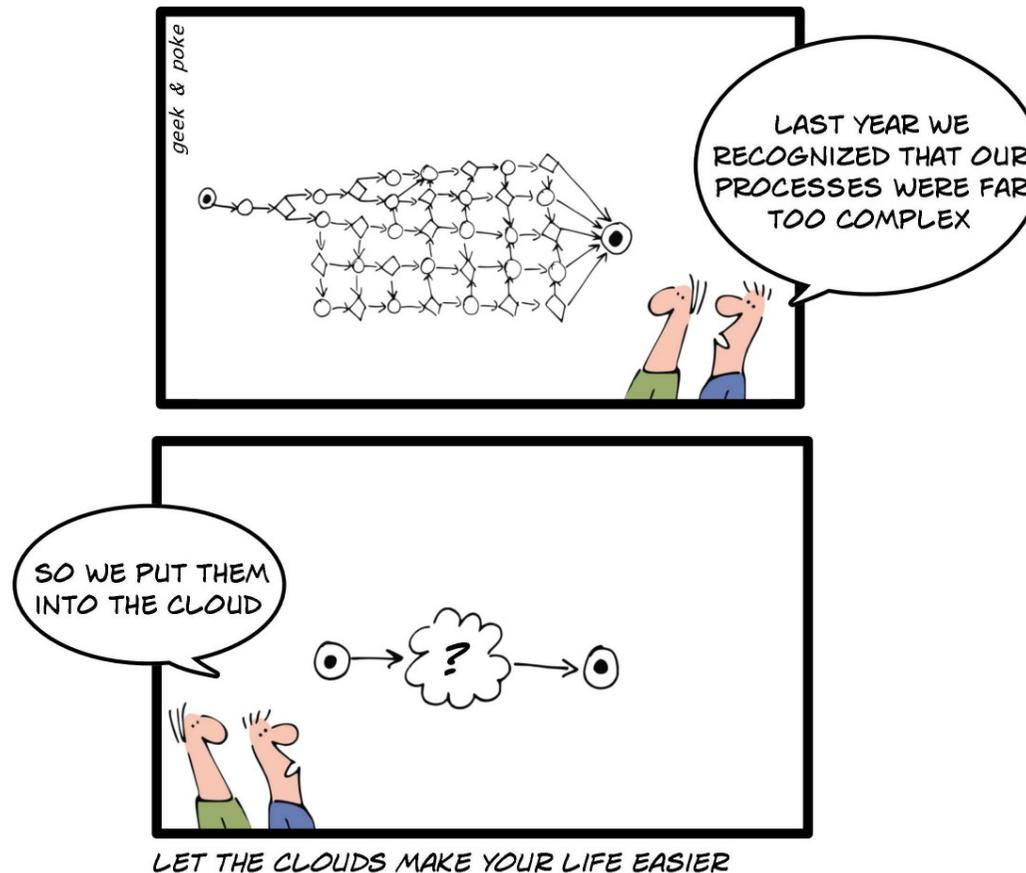
- Was ist überhaupt eine „Cloud“
- Die Google App Engine
- Getting started: „Hallo Welt“
- Persistenz in Google App Engine
- Limitierungen in GAE/J
- Quota und Performance
- Build-Tools

## Agenda

---

- Was ist überhaupt eine „Cloud“
- Die Google App Engine
- Getting started: „Hallo Welt“
- Persistenz in Google App Engine
- Limitierungen in GAE/J
- Quota und Performance
- Build-Tools

# Cloud? Was ist eine Cloud?



Quelle: <http://geekandpoke.typepad.com/geekandpoke/2009/03/let-the-clouds-make-your-life-easier.html>

## Die „Cloud“

---

- „Cloud Computing“ ...

*...bezeichnet primär den Ansatz, abstrahierte IT-Infrastrukturen (z. B. Rechenkapazität, Datenspeicher), fertige Programmpakete und Programmierumgebungen dynamisch an den Bedarf angepasst über Netzwerk zur Verfügung zu stellen.*

(Quelle: Wikipedia.de)

## IaaS

---

- IaaS = „Infrastructure as a Service“
- Rechnerinfrastruktur wird zur Verfügung dynamisch gestellt
- Die Infrastruktur steht „irgendwo“ virtuell in der Cloud
- Weiterentwicklung von klassischen virtuellen Maschinen (wie Virtualbox, VMWare usw.)
- Bekanntester Anbieter: Amazon mit EC2
  - Zahlen nur bei Nutzung
  - Verschiedene Images verfügbar

## SaaS

---

- „Software as a Service“
- Software wird über eine Cloud zur Verfügung gestellt
- Keine lokale Installation der Software notwendig
- Bekanntestes Beispiel: Google Docs
- Auch kommerziell, wie z.B. IBM „LotusLive“

## PaaS

---

- „Platform as a Service“
- Es wird eine Plattform (Laufzeitumgebung) in einer Cloud angeboten
- Keine Berührung mit dem darunterliegenden Betriebssystem
- Beispiele (Details später):
  - Windows Azure
  - *VMForce*
  - Google App Engine

## Agenda

---

- Was ist überhaupt eine „Cloud“
- Die Google App Engine
- Getting started: „Hallo Welt“
- Persistenz in Google App Engine
- Limitierungen in GAE/J
- Quota und Performance
- Build-Tools

## Google App Engine

---

- Klassische PaaS-Cloud
- Die Plattform ist eine Runtime-Umgebung für Webapplikationen
- Initial wurde „nur“ Python unterstützt
- Seit 2009 (in Beta) auch die Unterstützung von Java (Java 5 & 6)
  - Eingeschränkt!
  - Details später...

## GAE/J

---

- GAE/J ist eine JRE 6
  - Läuft in einer Sandbox mit Restriktionen
- Unterstützt out of the box
  - Servlet API 2.5 (\*)
  - JSF 1.1 (\*)
  - JSP (\*)
  - JDO 2.3 (?!)
  - JPA 1(\*)
- Support für Google User Accounts

(\*): Eingeschränkt...

## Agenda

---

- Was ist überhaupt eine „Cloud“
- Die Google App Engine
- Getting started: „Hallo Welt“
- Persistenz in Google App Engine
- Limitierungen in GAE/J
- Quota und Performance
- Build-Tools

## Getting started...

---

- Ein Google-Account
  - Ohne Useraccount keine GAE
- Eine Handynummer
  - Aktivierungs-SMS
- Eclipse (ab Europa / 3.3 unterstützt)
  - Helios ist OK
  - Alternativ NetBeans
- App Engine SDK oder Eclipse Plugin
  - Aktuelle Version der SDK ist 1.3.7

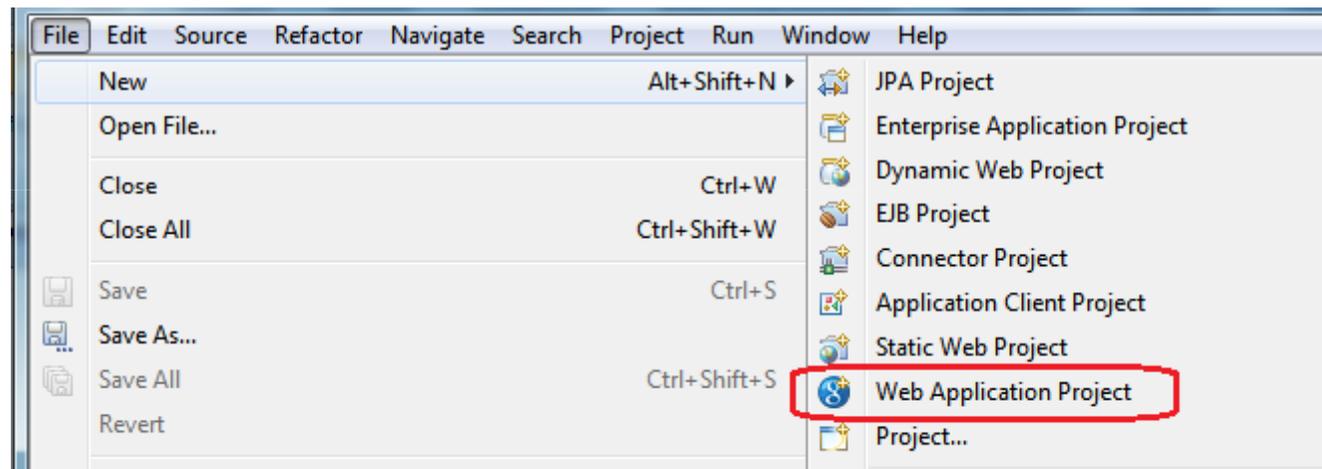
## Anmelden, Installation

---

- <https://appengine.google.com/>
- Google Account eingeben
- Handynummer eingeben
  - muss angegeben werden, um GAE freizuschalten
- Installation Eclipse Plugin:  
<http://code.google.com/intl/de-DE/eclipse/docs/download.html>
- Eclipse Update Sites dort

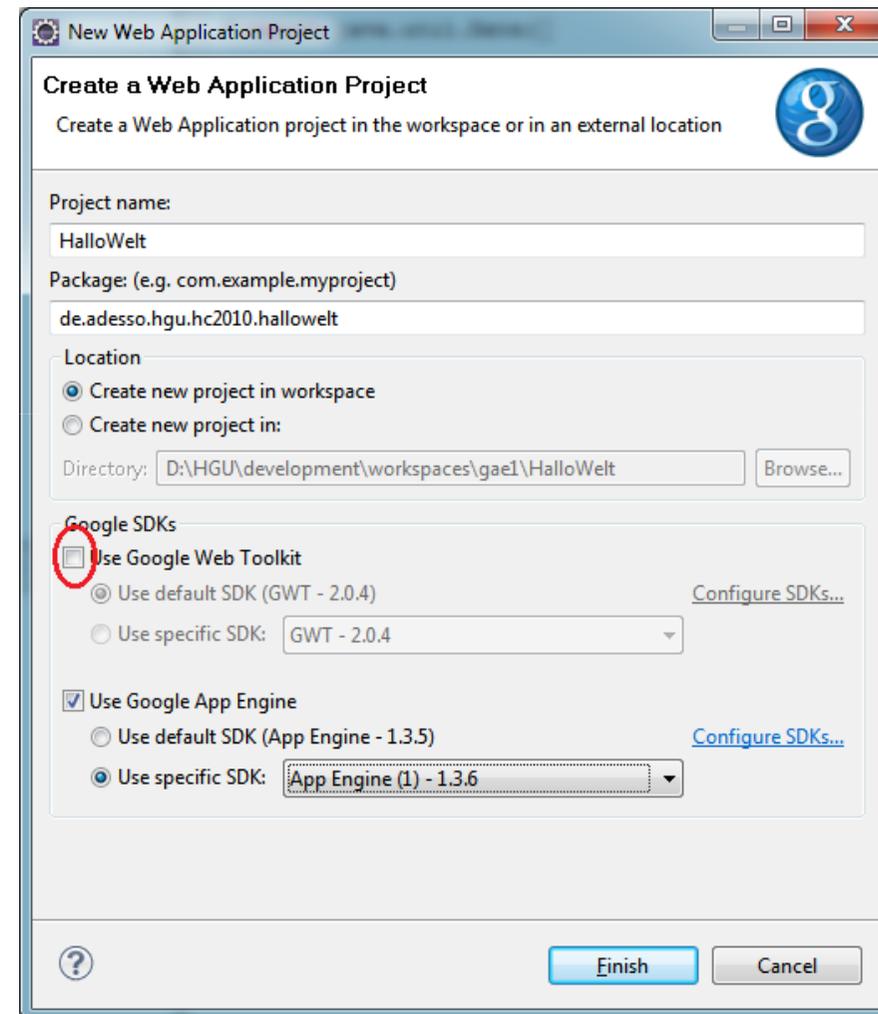
„Hallo Welt!“

- Neues Projekt anlegen



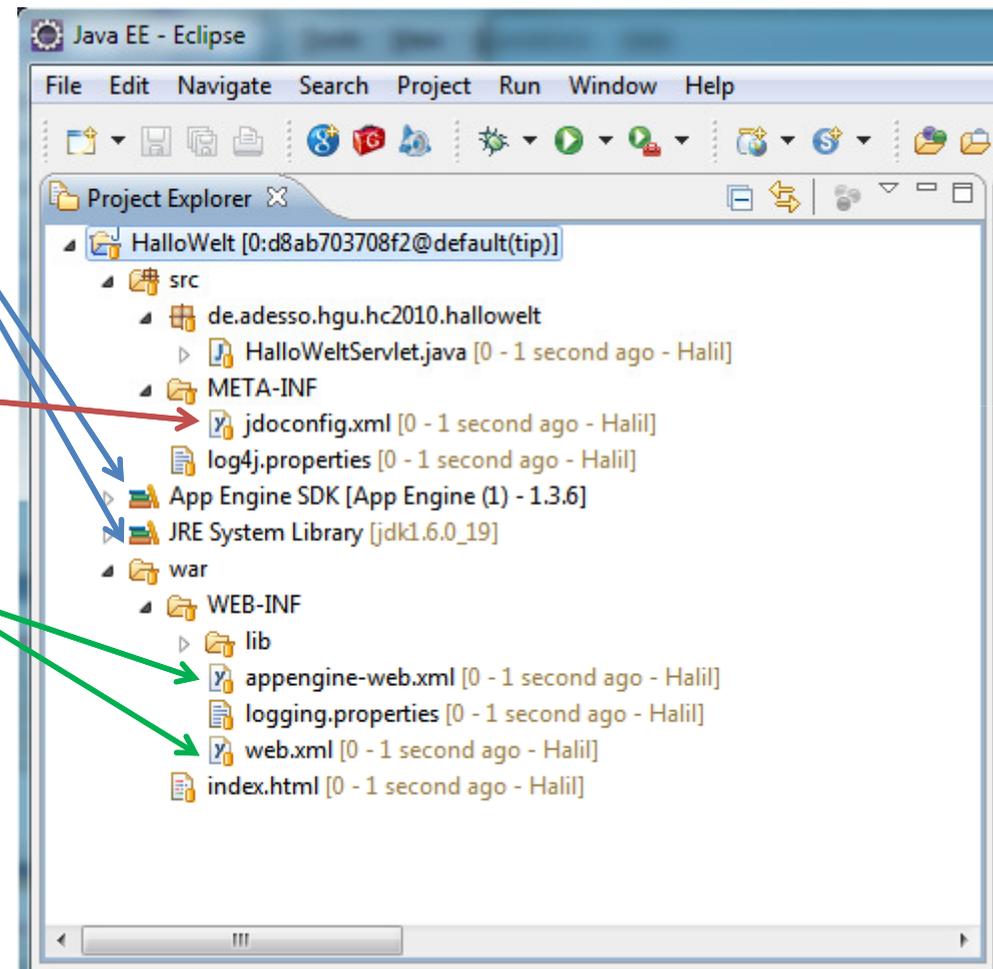
## „HalloWelt“

- GWT sollte (vorerst) nicht ausgewählt sein
- Hier Auswahl der GAE API 1.3.6



## Und das fertige Projekt...

- Default-Pakete und Libs werden eingefügt
- JDO ist Persistenz-Default
- Voreinstellungen für Konfigurationen



## Servlet sagt „Hello, World“

---

```
package de. adesso. hgu. hc2010. hallowelt;

import java. io. IOException;
import javax. servlet. http. *;

@SuppressWarnings("serial")
public class HalloWeltServlet extends HttpServlet {
    public void doGet( HttpServletRequest req,
        HttpServletResponse resp) throws IOException {
        resp.setContentType("text/plain");
        resp.getWriter().println("Hello, world");
    }
}
```

- Starten der Applikation innerhalb von Eclipse  
(RunAs -> Web Application)
- <http://localhost:8888> ist Standard

## Applikation bei GAE registrieren

- <https://appengine.google.com/start/createapp>

Google app engine

### Create an Application

You have 8 applications remaining.

**Application Identifier:**

.appspot.com

You can map this application to your own domain later. [Learn more](#)

**Application Title:**

Displayed when users access your application.

**Authentication Options (Advanced):** [Learn more](#)

Google App Engine provides an API for authenticating your users, including Google Accounts, Google Apps, and others. If you use authentication, you'll need to specify now what type of users can sign in to your application:

**Open to all Google Accounts users (default)**

If your application uses authentication, anyone with a valid Google Account may sign in. (This includes all Gmail Accounts.)

[Edit](#)

## DD anpassen

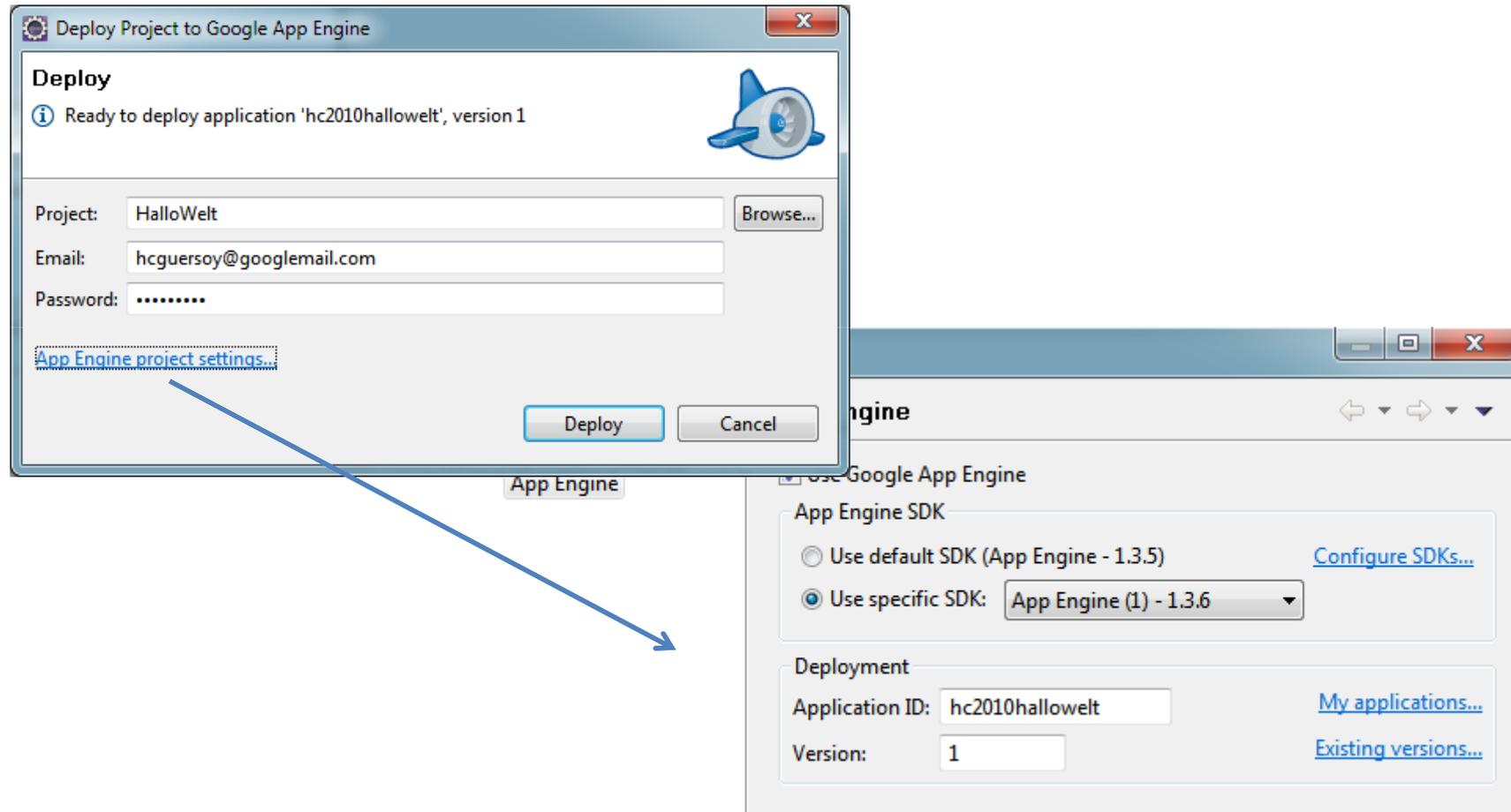
---

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <application>hc2010hallowelt</application>
  <version>1</version>

  <!-- Configure java.util.logging -->
  <system-properties>
    <property name="java.util.logging.config.file"
              value="WEB-INF/logging.properties"/>
  </system-properties>

</appengine-web-app>
```

# GAE Deployment



## GAE Postdeployment

---

- GAE DashBoard gibt Infos über Zugriffe, verbrauchte Quota, Logs usw.
- Verwaltung von Versionen einer Applikation
  - Fallback auf alte Version möglich
- Aufruf der Applikation über <http://hc2010hallowelt.appspot.com/hallowelt>

# GAE Dashboard

Google app engine hcguersoy@googlemail.com | [My Account](#) | [Help](#) | [Sign out](#)

hc2010hallowelt Version: 1 [« My Applications](#)

**Main**

- [Dashboard](#)
- [Quota Details](#)
- [Logs](#)
- [Cron Jobs](#)
- [Task Queues](#)
- [Blacklist](#)

**Data**

- [Datastore Indexes](#)
- [Datastore Viewer](#)
- [Datastore Statistics](#)
- [Blob Viewer](#)

**Administration**

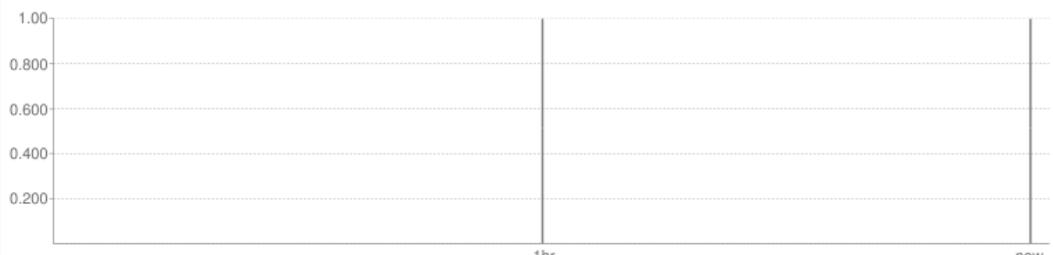
- [Application Settings](#)
- [Permissions](#)
- [Versions](#)
- [Admin Logs](#)

**Billing**

- [Billing Settings](#)
- [Billing History](#)

**Charts** ?

Requests/Second 6 hrs 12 hrs 24 hrs 2 days 4 days 7 days 14 days 30 days



**Billing Status:** Free - [Settings](#) Quotas reset every 24 hours. Next reset: 16 hrs ?

Resource	Usage
CPU Time	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0% 0.00 of 6.50 CPU hours
Outgoing Bandwidth	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0% 0.00 of 1.00 GBytes
Incoming Bandwidth	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0% 0.00 of 1.00 GBytes
Total Stored Data	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0% 0.00 of 1.00 GBytes
Recipients Emailed	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0% 0 of 2,000

**Current Load** ?

URI	Requests	Avg CPU (API)	% CPU

**Errors** ?

URI	Count	% Errors

## Agenda

---

- Was ist überhaupt eine „Cloud“
- Die Google App Engine
- Getting started: „Hallo Welt“
- Persistenz in Google App Engine
- Limitierungen in GAE/J
- Quota und Performance
- Build-Tools

## GAE - Persistenz

---

- Persistenz über JDO und JPA möglich
  - Als ORM kommt DataNucleus in der Version 1.1 (aktuell ist 2.x) zum Einsatz (<http://www.datanucleus.org>)
- In GAE/DN JDO besser dokumentiert als JPA (noch)
- Alternativ gibt es eine Lowlevel-API
- Im Hintergrund keine SQL-Datenbank sondern Google BigTable
- In memory SQL-DB's sind möglich
- Enhancement der Entity-Klassen wegen DataNucleus (JDO / JPA)

## GAE - JPA

---

- In persistence.xml Konfiguration des transaktionalen Verhaltens.
- Default:

```
<persistence-unit name="transactions-optional">
  <provider>org.datanucleus.store.appengine.jpa.DatastorePersistenceProvider</provider>
  <properties>
    <property name="datanucleus.NontransactionalRead"
      value="true" />
    <property name="datanucleus.NontransactionalWrite"
      value="true" />
    <property name="datanucleus.ConnectionURL"
      value="appengine" />
  </properties>
</persistence-unit>
```

## GAE – JPA-Einschränkungen

---

- @ManyToMany
- Join-Queries
- Bei Vererbung keine Unterstützung für
  - InheritanceType.SINGLE\_TABLE
  - InheritanceType.JOINED
- Aggregate queries (group by, having, sum, avg, min, max)
- Polymorphe Queries

## GAE – Und paar Punkte mehr

---

- Resultsets sind nicht gefüllt, daher:

```
resultSet.size();
```

- Id-Felder:

```
Unsupported primary key type: long
```

- GenerationType.TABLE:

```
javax.persistence.PersistenceException: Transaction  
is still active.
```

- Klappt besser:

```
@Id  
@GeneratedValue(strategy = GenerationType.IDENTITY)  
private Key key;
```

## Agenda

---

- Was ist überhaupt eine „Cloud“
- Die Google App Engine
- Getting started: „Hallo Welt“
- Persistenz in Google App Engine
- Limitierungen in GAE/J
- Quota und Performance
- Build-Tools

## 'JRE Class White List'

---

- Die ‚JRE Class White List‘ definiert die Klassen, die von der GAE Sandbox unterstützt werden
- <http://code.google.com/intl/de-DE/appengine/docs/java/jrewhitelist.html>
- Die Applikation darf nicht
  - Eigene Threads starten
  - Keine Dateien schreiben
  - Keine „willkürlichen“ Netzwerkverbindungen aufbauen

## "Will it play?"

---

- Die Seite „Will it play?“ gibt eine Übersicht über das Verhalten diverser Frameworks in GAE/J
- Keine Unterstützung für
  - JAX-RPC & JAX-WS (!!!)
  - JDBC, JMS, JNDI, RMI
- Guice – Semikompatibel
- Hibernate – Inkompatibel
- Spring MVC, ORM – Kompatibel
- Restlet – „Spezialedition“ für GAE

## Spring...

---

- Spring = Semi-Kompatibel (nicht alle JARs)
- Spring MVC-Applikation mit Spring Beans
- Setup: Spring JARs in WAR/WEB-INF/lib kopieren und in CP aufnehmen

```
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<servlet>
  <servlet-name>dispatcher</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>dispatcher</servlet-name>
  <url-pattern>/springgae/*</url-pattern>
</servlet-mapping>
```

- Beispiel im Cook Book: <http://peterbacklund.blogspot.com/2009/04/running-spring-on-google-app-engine.html>

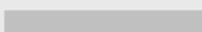
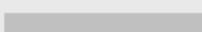
## Agenda

---

- Was ist überhaupt eine „Cloud“
- Die Google App Engine
- Getting started: „Hallo Welt“
- Persistenz in Google App Engine
- Limitierungen in GAE/J
- Quota und Performance
- Build-Tools

## Quota

- GAE ist kostenlos innerhalb der Quota-Grenzen
- Stärkere Ausnutzung der Infrastruktur gegen Bezahlung möglich, muss explizit eingeschaltet werden
- Quota verbraucht = HTTP 403 FORBIDDEN oder Exception
- „Per minute quota“ – ca. 500 Req. / min.

Billing Status: Free - <a href="#">Settings</a>		Quotas reset every 24 hours. Next reset: 16 hrs 	
Resource	Usage		
CPU Time	 0%	0.00	of 6.50 CPU hours
Outgoing Bandwidth	 0%	0.00	of 1.00 GBytes
Incoming Bandwidth	 0%	0.00	of 1.00 GBytes
Total Stored Data	 0%	0.00	of 1.00 GBytes
Recipients Emailed	 0%	0	of 2,000

## Quota – weitere Details

---

<b>Ressource</b>	<b>Quota</b>
Request size	10 mb
Response size	10 mb
Request max. Dauer	30 sec.
Maximale Anzahl Files	3000
Maximale Dateigröße	10 mb

# Performance

- Max. 500 req. / sec. (Quota)
- Simpler Test (schreiben und lesen eines Datensatzes)



## Agenda

---

- Was ist überhaupt eine „Cloud“
- Die Google App Engine
- Getting started: „Hallo Welt“
- Persistenz in Google App Engine
- Limitierungen in GAE/J
- Quota und Performance
- Build-Tools

## Build...

---

- Ant wird unterstützt
  - Entity-Klassen müssen „manuell“ Enhanced werden, Ant-Task wird von DataNucleus angeboten
- Maven
  - Maven2 Plugin verfügbar
    - (<http://code.google.com/p/maven-gae-plugin/>)
    - Archtypes für JSP, GWT und Wicket-Projekte
    - sehr aktives Projekt
  - Enhancement über Maven-Plugins von DataNucleus

12.–15.09.2010  
in Nürnberg



# Herbstcampus

Wissenstransfer  
par excellence

**Vielen Dank!**

Dr. Halil-Cem Gürsoy

adesso AG

halil-cem.guersoy [at] adesso.de & @hgutwit

# adesso AG



- Sponsor Herbstcampus 2010
- Mitarbeiter
  - Über 750 Mitarbeiter in der adesso Group
- Umsatz
  - Umsatzerwartung 2010: > 79 Mio. Euro
- Auszeichnungen
  - Platz 18 der deutschen Top-25 Beratungs- und Systemintegrationsunternehmen (Lünendonk-Liste 2010)
  - Top Job (2008)
  - Deutschlands beste Arbeitgeber (2005 und 2010)
- Standorte in Dortmund, Köln, Aachen, Stuttgart, Hamburg, Berlin, Frankfurt, München sowie in der Schweiz und Österreich

