



GAE/JとFlex/AIR

support@getsugo.com

アジェンダ

- ご都合.comの事例
- GAE/Jはここがすごい
- GAE/JでBlazeDSを動かす
- SmallTableとは

- この資料のダウンロード
 - www.gotsugo.com > お知らせ



ご都合.comの事例

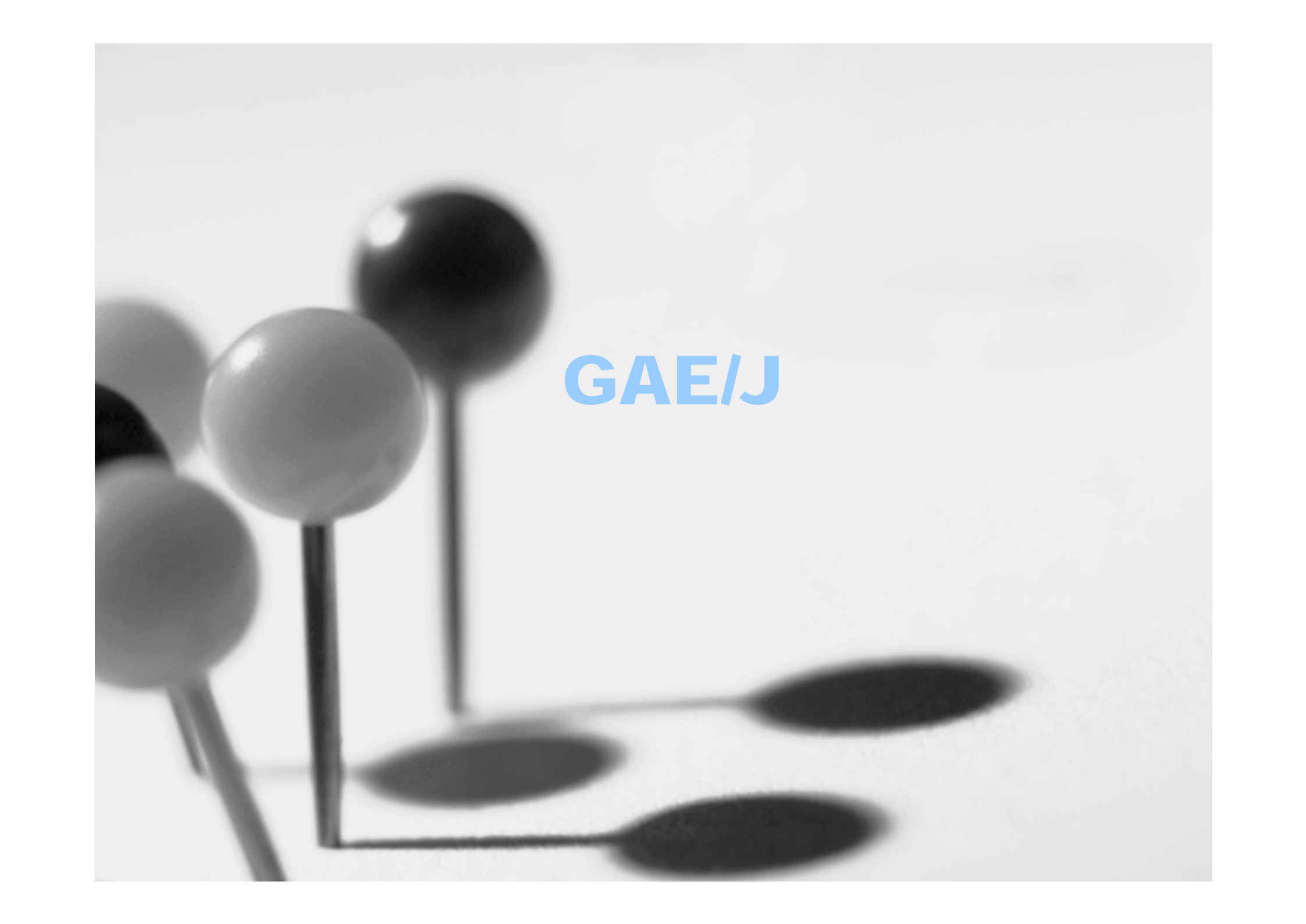
ご都合.com (gotsugo.com)

- ご都合.com
 - スケジュール共有ツール
 - 会議・飲み会に
 - 無償・登録不要
 - 5/6公開
 - 6日間で2000 UU



ご都合.comの技術

- Flexクライアント
- Google App Engine for Javaサーバ
 - BlazeDS
 - ビジネスロジックのクラス
 - Entityクラス
- 開発期間:5日間
 - 設計・実装・テスト・ドキュメント作成など

A black and white photograph of several pushpins on a white surface. The pushpins have spherical heads, some light and some dark. Their shadows are cast on the surface. The text 'GAE/Jはここがすごい!' is overlaid in blue on the right side of the image.

GAE/Jは
ここがすごい！

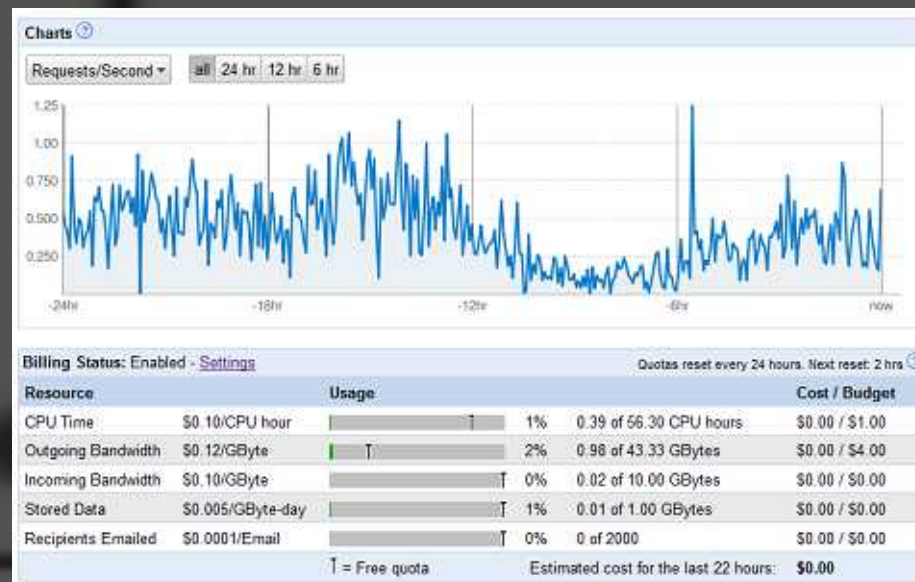
GAE/Jとは

- Google App Engine for Java
 - 自分のアプリをGoogleインフラで運用
 - クラウドコンピューティングサービス
- GAE/Jのすごいところ
 - 無償で使える
 - BigTableが簡単に使える
 - サーバ構築不要、デプロイ簡単

無償で使える

- 無償で使える

	無償提供分	有償提供分の単価
ストレージ容量	1GB	\$0.005 / 1GB × 1日
ネットワーク帯域(上り)	10GB / 1日	\$0.10 / 1GB
ネットワーク帯域(下り)	10GB / 1日	\$0.12 / 1GB
CPU時間	6.5時間 / 1日	\$0.10 / 1時間
メール受信件数	2000件 / 1日	\$0.0001 / 1件



あと10倍の
アクセスまで
無償でOK

有償換算で
も\$0.3 / 日
程度

BigTableが簡単に使える

- BigTableって？
 - Google検索を支える巨大分散DB
 - RDBではなく、key-value型
 - スキーマは事前定義が不要、自由に変更可能
 - テーブル間結合できない
 - LIKEも使えない
 - 鬼のようなスケラビリティと可用性
- BigTableがタダで使えると...
 - 大規模サービスでも苦勞しなさそう
 - Oracle RACも分散キャッシュも不要

JDOでBigTable利用

- GAE/JのDatastore API
 - JDO API経由でBigTableを使える
- GtgCalendarクラス

```
@PersistenceCapable(identityType = IdentityType.APPLICATION)
public class GtgCalendar {
```

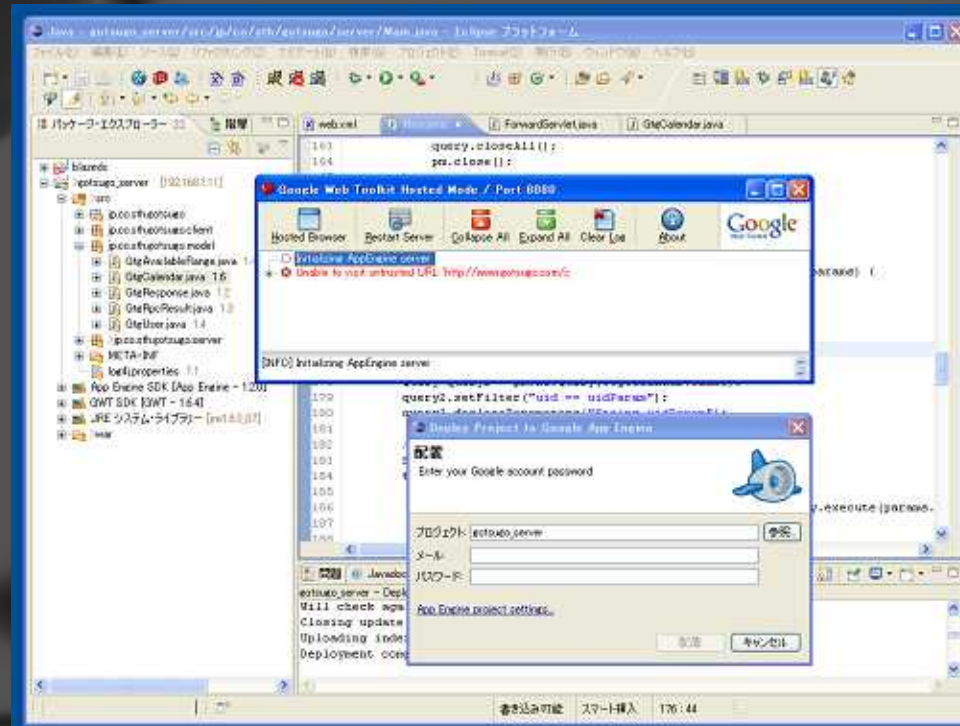
```
    @PrimaryKey
    @Persistent(valueStrategy = IdGeneratorStrategy.IDENTITY)
    @Extension(vendorName="datanucleus", key="gae.encoded-pk", value="true")
    private String key;
```

```
    @Persistent
    private String uid;
    < 以下略 >
```

```
    PersistenceManager pm = pmf.getPersistenceManager();
    try {
        pm.makePersistent(gc);
    } finally {
        pm.close();
    }
}
```

サーバ構築不要、デプロイ簡単

- GAE/Jの実行環境
 - Amazon EC2のようなVMではない
 - JBoss/Tomcatのような実行環境



サーバ構築不要、デプロイ簡単

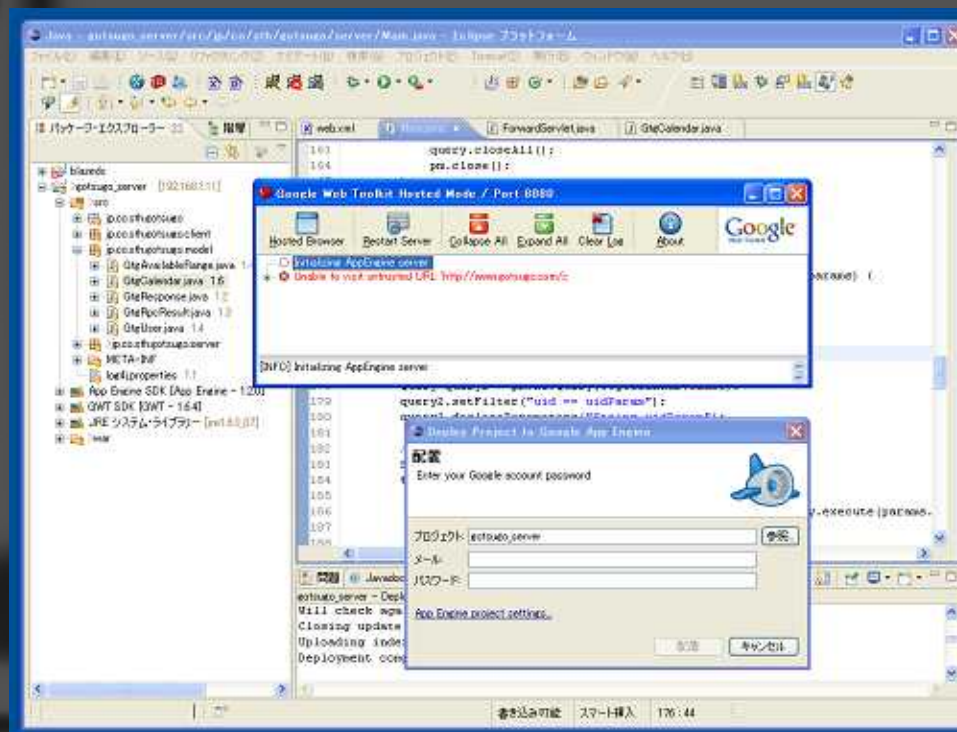
- GAE/Jの開発
 - Eclipseプラグインでローカル開発
 - クリック1つでデプロイ
- メリットとデメリット
 - サーバー構築不要、LL感覚でデプロイ
 - でも「サンドボックス」の制約あり



GAE/Jで
BlazeDSを動かす

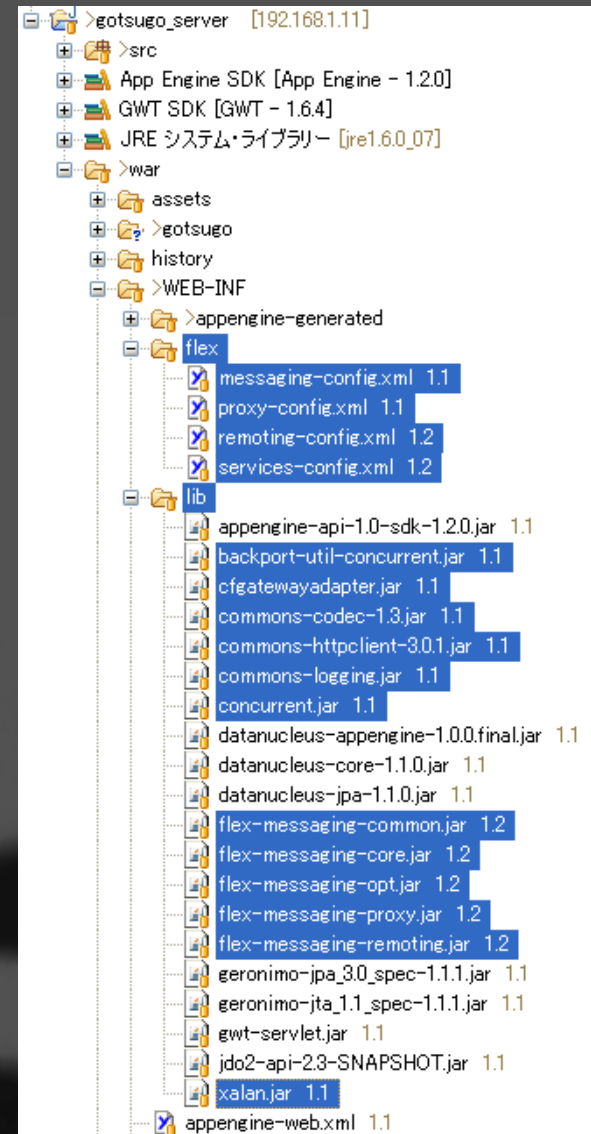
GAE/JのEclipse開発環境

- GAE/JのEclipseプラグイン
 - ローカルでGAE/J環境を再現
 - クリック1つでデプロイ可能



BlazeDSとGAE/J

- GAE/JでもAMF通信したい
 - RemoteObjectはラク
- GAE/JにBlazeDSを載せる
 - blazeds.warの中身をWEB-INFに配置
 - でもこのままではエラーがでます



BlazeDSを動かすには...

- Martin氏のブログを見よ！
 - <http://martinzoldano.blogspot.com/2009/04/appengine-adobe-blazeds-fix.html>
 - Martinさんありがとう！！
- BlazeDSソースのビルドが必要
 - まずはBlazeDSをビルドする環境を用意
- 要点
 - JMX APIの管理機能はオフに
 - AbstractAmfInput.javaを修正
 - BaseHTTPEndpoint.javaを修正
- Adobe ADCにも記事掲載の予定

BlazeDSを動かすには...

- BlazeDSのservices-config.xml
 - system要素に以下を追加
 - `<manageable>false</manageable>`
 - GAE/Jでは使えないJMXを使わない
- AbstractAmfInput.javaを修正
 - 58行目を以下に修正

```
this.in = new DataInputStream( new java.io.BufferedInputStream( in ) );
```

- GAE/Jのコンテナは特殊なので、I/Oの振る舞いにクセがある？

BlazeDSを動かすには...

- BaseHTTPEndpoint.javaを修正
 - 404 ~ 418行目をコメントアウト
 - setupFlexClientメソッドにてセッション重複エラーを発生させる部分
 - GAE/Jは特殊なクラスタ環境なので、セッション情報の取り扱いにクセがある？
- BlazeDSをビルド
 - JARファイルをWEB-INF/libにコピー

Datastore APIのtips

- eager loadingできません
 - BigTableはjoinできない
 - 1:Nの親Entityと子Entityがある場合
 - ループですべての子Entityをロードしてからクライアントに渡す
 - BigTableプログラミングってこんな感じ？

```
private void loadChildren(GtgCalendar gc) {  
    for (GtgResponse gr : gc.getResponses()) {  
        for (GtgAvailableRange gar : gr.getAvailableRanges()) gar.getKey();  
    }  
}
```

Datastore APIのtips

- PersistenceManagerはクローズする
 - Entityとpmの「ひも付き」を切ってからクライアントに渡す
 - さもないと例外が
 - Hibernateのセッションクローズと同じ

```
// クエリを作成
PersistenceManager pm = pmf.getPersistenceManager();
Query query = pm.newQuery(GtgCalendar.class);
query.setFilter("token == tokenParam");
query.declareParameters("String tokenParam");

// クエリを実行
List<GtgCalendar> results;
GtgCalendar gc = null;
try {
    results = (List<GtgCalendar>)query.execute(token);
} finally {
    query.closeAll();
    pm.close();
}
```

Datastore APIのtips

- 主キーは「Key as Encoded String」
 - Datastore APIでの主キー
 - long型 : 親から子へのナビゲーション不可
 - Key型 : ナビゲーション可能
 - BlazeDSでKey型は渡せない
 - 中身がなくなる
 - Key型をString型にエンコードする設定

```
@Extension(vendorName="datanucleus", key="gae.encoded-pk", value="true")  
private String key;
```



BigTableと
SmallTable

SmallTableとは

- BigTableに足りない点
 - リレーショナルモデルが使えない
 - やっぱりjoinしたい...LIKEも欲しい...
- SmallTableとは
 - リアルタイムDB同期フレームワーク
 - まだ構想中...絵に描いた餅
 - クライアント側SQLiteとBigTableを同期
 - AIR、Ajax、iPhone
 - クライアント側ORMもあるといいな...

SmallTableのメリット

- BigTable + SmallTableのメリット
 - クライアント側ではSQLを使える
 - AIR/Ajax/iPhoneの違いをORMで吸収
 - BigTableはデータストアに徹する
 - オフライン対応
 - ローカルDB紛失時もサーバでバックアップ
 - バックグラウンドでリアルタイム同期
 - サーバレスポンスを待つ必要なし
 - WordやExcelの使い勝手 + オンラインアプリの即時性



ありがとうございました