

Google App Engine データストアの利用

[Python][Google App Engine][GQL]

概要

概要

- <http://code.google.com/intl/ja/appengine/docs/python/gettingstarted/usingdatastore.html>

リファレンス

- <http://code.google.com/intl/ja/appengine/docs/python/datastore/>

参考

- http://typea.info/blg/glob/2008/09/google_app_engine_13.html
- App Engine は、Python 用の データ モデリング API を備えている
- Django のデータ モデリング API と似ているが、App Engine の拡張可能なデータストアを背後で使用

利用上の注意

検索条件と並べ替えに一部プロパティが使用できない

並べ替えやフィルタに使用するには、プロパティを、インデックス化する必要があるが、一部のプロパティは、インデックス化できないため、検索条件や並べ替えに使用できないので注意が必要！

- Blob
- Text
- TextProperty

GQL で検索条件にリテラルを使用する

GQL でリテラルを使用する場合には注意が必要。日付型などの場合 DATE(リテラル) とする必要がある。

リテラル	用法
str リテラル	(一重引用符で囲まれた文字列) 文字列内の一重引用符は としてエスケープされます。例 : 'Joes Diner'
整数または浮動小数点数リテラル	例 : 42.7
Boolean リテラル	TRUE または FALSE
日時リテラル	DATETIME(year , month , day , hour , minute , second)
日時リテラル	DATETIME('YYYY-MM-DD HH:MM:SS')

日付リテラル	DATE(year , month , day)
日付リテラル	DATE('YYYY-MM-DD')
時刻リテラル	TIME(hour , minute , second)
時刻リテラル	TIME('HH:MM:SS')
エンティティ キー	KEY('encoded key')
エンティティ キー	KEY(kind , name/ID [, kind , name/ID...])
User オブジェクト	USER(email-address)
GeoPt	(浮動小数点数値の緯度と経度付き) GEOPT(lat , long)
バウンド パラメータ	位置パラメータが数値によって参照されます。 title = :1
キーワードパラメータ	名前によって参照されます。 title = :mytitle

クエリで自動生成される key を使用する

- ・ (1) key を文字列に変換
- ・ (2) 文字列に変換された key を Key オブジェクトに変換
- ・ (3) GQL で、__key__ と比較する

```

from google.appengine.ext import db

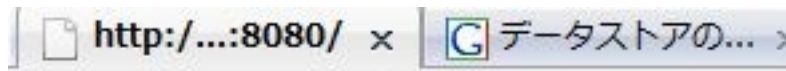
class LoadHogeEntity(webapp.RequestHandler):
    def get(self):
        context = {}
        results = HogeEntity.all()
        for result in results:
            context['key_str'] = str(result.key()) # (1)
        return self.response.out.write(template.render('templates/hoge.html', context))

class DeleteHogeEntity(webapp.RequestHandler):
    def get(self):
        key_str = self.request.get('key_str')
        key = db.Key(key_str) # (2)
        q = HogeEntity.gql("WHERE __key__ =:1", key) # (3)

        hoge = q.fetch(1)
        if hoge:
            hoge.delete()

```

例



id	name	mail
1	piroto	piroto@a-net.email.ne.jp
2	pppiroto	pppiroto@gmail.com
3	aaaa	aaaa
<input type="text"/>	<input type="text"/>	<input type="text"/> <input type="button" value="add"/>

手順

- import 文を最初の行に追加 (1)
- プロパティの種別はデータストア API リファレンス参照 (2)
- 新しいオブジェクトをデータストアに保存 (3)
- 保存されたオブジェクトを GQL で取り出す (4)
- クラスの gql メソッドでもデータを取得できる。(SELECT * FROM Person を省略できる)
 - persons = Person.gql("ORDER BY id LIMIT 10")
- WHERE 句に定数は指定できない。パラメータを位置、名前で指定できる。
 - persons = Person.gql("WHERE id = :1 ORDER BY id", int("1"))
 - persons = Person.gql("WHERE id = :id ORDER BY id", id=int("2"))

```
import cgi
import wsgiref.handlers

from google.appengine.api import users
from google.appengine.ext import webapp
from google.appengine.ext import db #(1)

# 保存するモデルクラス
class Person(db.Model):
    id = db.IntegerProperty() #(2)
    name = db.StringProperty()
    mail = db.EmailProperty()

class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.out.write('<html><body>')
        # (4) 保存されたデータを GQL で取り出す
        # persons = Person.gql("ORDER BY id LIMIT 10")
        # persons = Person.gql("WHERE id = :1 ORDER BY id", int("1"))
        # persons = Person.gql("WHERE id = :id ORDER BY id", id=int("2"))
        persons = db.GqlQuery("SELECT * FROM Person ORDER BY id LIMIT 10")
        self.response.out.write("<form action='/add' method='post'>")
        self.response.out.write("""<table border='1'>
            <tr>
                <th>id</th><th>name</th><th>mail</th>
            </tr>
            """)
        for person in persons:
            self.response.out.write("<tr>")
            self.response.out.write("<td>%d</td>" % person.id)
            self.response.out.write("<td>%s</td>" % cgi.escape(person.name))
            self.response.out.write("<td>%s</td>" % cgi.escape(person.mail))
            self.response.out.write("</tr>")
        self.response.out.write("""
            <tr>
                <td><input type='text' size='2' name='id'></td>
                <td><input type='text' size='10' name='name'></td>
                <td><input type='text' size='15' name='mail'>
            """)
```

```

        <input type='submit' value='add'></td>
    </tr>
</table>
"""
self.response.out.write("</form>")
self.response.out.write('</body></html>')

# データ投入処理
class AddPersonPage(webapp.RequestHandler):
    def post(self):
        person = Person()
        id = self.request.get('id')
        if id != '':
            person.id = int(id)
            person.name = self.request.get('name')
            person.mail = self.request.get('mail')
            person.put() # (3) データ保存

        self.redirect('/')

def main():
    application = webapp.WSGIApplication(
        [('/', MainPage),
         ('/add', AddPersonPage)],
        debug=True
    )
    wsgiref.handlers.CGIHandler().run(application)

if __name__ == "__main__":
    main()

```