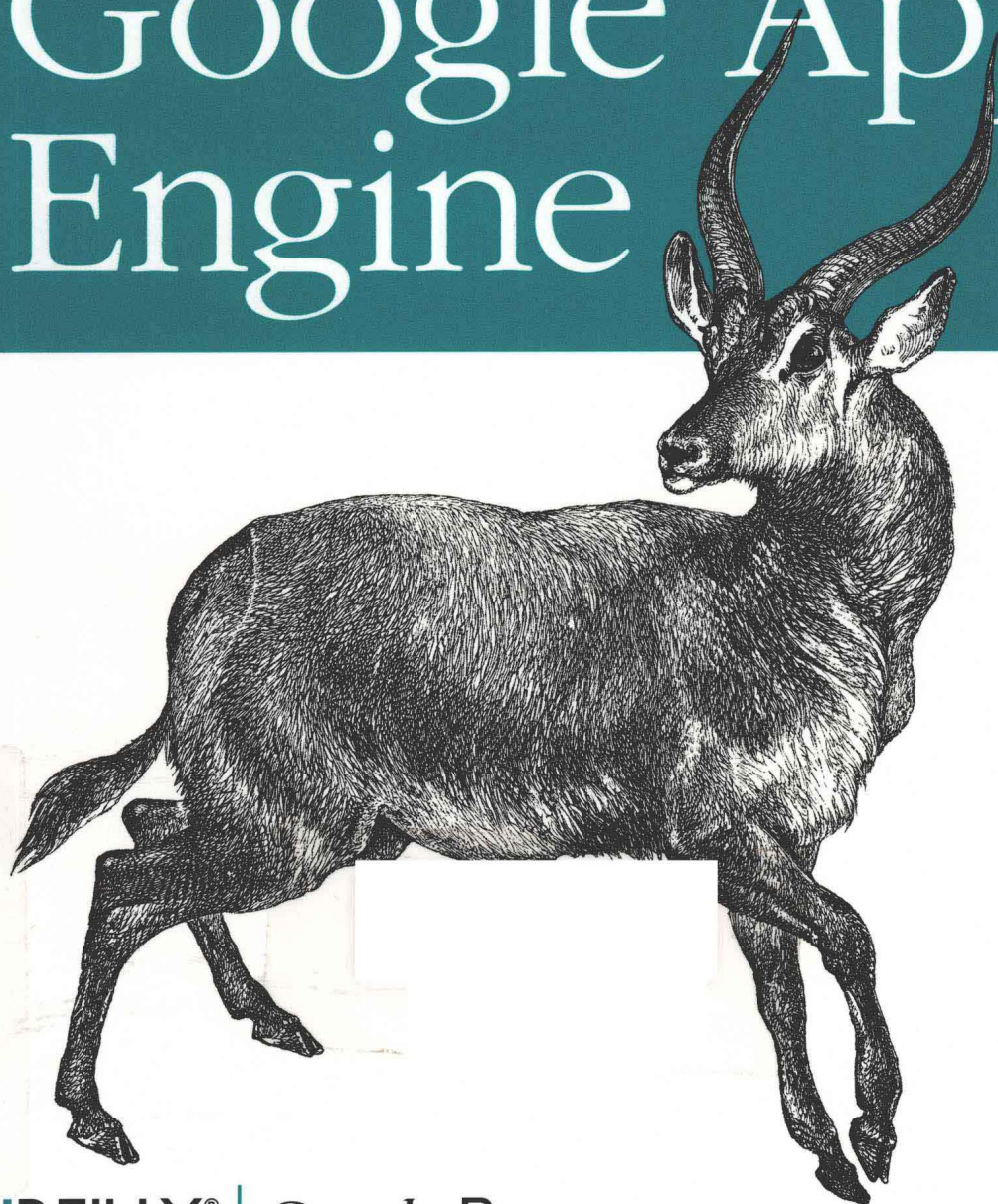


Google App Engine 编程 (影印版)

Programming

Google App Engine



O'REILLY® | Google™ Press

东南大学出版社

Dan Sanderson 著

Google App Engine 编程 (影印版)

Programming Google App Engine

Dan Sanderson

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

Google App Engine 编程: 英文 / (美) 桑德森 (Sanderson, D.) 著. — 影印本. — 南京: 东南大学出版社, 2011.1

书名原文: Programming Google App Engine
ISBN 978-7-5641-2493-9

I. ① G… II. ① 桑… III. ① 主页制作—程序设计—英文
IV. ① TP393.092

中国版本图书馆 CIP 数据核字 (2010) 第 209435 号

江苏省版权局著作权合同登记
图字: 10-2010-281 号

©2009 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2010. Authorized reprint of the original English edition, 2010 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2009。

英文影印版由东南大学出版社出版 2010。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式复制。

Google App Engine 编程 (影印版)

出版发行: 东南大学出版社
地 址: 南京四牌楼 2 号 邮编: 210096
出 版 人: 江建中
网 址: <http://www.seupress.com>
电子邮件: press@seu.edu.cn
印 刷: 扬中市印刷有限公司
开 本: 787 毫米 × 980 毫米 16 开本
印 张: 24.5 印张
字 数: 480 千字
版 次: 2011 年 1 月第 1 版
印 次: 2011 年 1 月第 1 次印刷
书 号: ISBN 978-7-5641-2493-9
印 数: 1~1600 册
定 价: 59.00 元 (册)

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

Preface

On the Internet, popularity is swift and fleeting. A mention of your website on a popular blog can bring 300,000 potential customers your way at once, all expecting to find out who you are and what you have to offer. But if you're a small company just starting out, your hardware and software aren't likely to be able to handle that kind of traffic. Chances are, you've sensibly built your site to handle the 30,000 visits per hour you're actually expecting in your first 6 months. Under heavy load, such a system would be incapable of showing even your company logo to the 270,000 others that showed up to look around. And those potential customers are not likely to come back after the traffic has subsided.

The answer is *not* to spend time and money building a system to serve millions of visitors on the first day, when those same systems are only expected to serve mere thousands per day for the subsequent months. If you delay your launch to build big, you miss the opportunity to improve your product using feedback from your customers. Building big before allowing customers to use the product risks building something your customers don't want.

Small companies usually don't have access to large systems of servers on day one. The best they can do is to build small and hope meltdowns don't damage their reputation as they try to grow. The lucky ones find their audience, get another round of funding, and halt feature development to rebuild their product for larger capacity. The unlucky ones, well, don't.

But these days, there are other options. Large Internet companies such as Amazon.com, Google, and Microsoft are leasing parts of their high-capacity systems using a pay-per-use model. Your website is served from those large systems, which are plenty capable of handling sudden surges in traffic and ongoing success. And since you pay only for what you use, there is no up-front investment that goes to waste when traffic is low. As your customer base grows, the costs grow proportionally.

Google App Engine, Google’s application hosting service, does more than just provide access to hardware. It provides a model for building applications that grow automatically. App Engine runs your application so that each user who accesses it gets the same experience as every other user, whether there are dozens of simultaneous users or thousands. The application uses the same large-scale services that power Google’s applications for data storage and retrieval, caching, and network access. App Engine takes care of the tasks of large-scale computing, such as load balancing, data replication, and fault tolerance, automatically.

The App Engine model really kicks in at the point where a traditional system would outgrow its first database server. With such a system, adding load-balanced web servers and caching layers can get you pretty far, but when your application needs to write data to more than one place, you have a hard problem. This problem is made harder when development up to that point has relied on features of database software that were never intended for data distributed across multiple machines. By thinking about your data in terms of App Engine’s model up front, you save yourself from having to rebuild the whole thing later, without much additional effort.

Running on Google’s infrastructure means you never have to set up a server, replace a failed hard drive, or troubleshoot a network card. And you don’t have to be woken up in the middle of the night by a screaming pager because an ISP hiccup confused a service alarm. And with automatic scaling, you don’t have to scramble to set up new hardware as traffic increases.

Google App Engine lets you focus on your application’s functionality and user experience. You can launch early, enjoy the flood of attention, retain customers, and start improving your product with the help of your users. Your app grows with the size of your audience—up to Google-sized proportions—without having to rebuild for a new architecture. Meanwhile, your competitors are still putting out fires and configuring databases.

With this book, you will learn how to develop applications that run on Google App Engine, and how to get the most out of the scalable model. A significant portion of the book discusses the App Engine scalable datastore, which does not behave like the relational databases that have been a staple of web development for the past decade. The application model and the datastore together represent a new way of thinking about web applications that, while being almost as simple as the model we’ve known, requires reconsidering a few principles we often take for granted.

This book introduces the major features of App Engine, including the scalable services (such as for sending email and manipulating images), tools for deploying and managing applications, and features for integrating your application with Google Accounts and Google Apps using your own domain name. The book also discusses techniques for optimizing your application, using task queues and offline processes, and otherwise getting the most out of Google App Engine.

Using This Book

As of this writing, App Engine supports two technology stacks for building web applications: Java and Python. The Java technology stack lets you develop web applications using the Java programming language (or most other languages that compile to Java bytecode or have a JVM-based interpreter) and Java web technologies such as servlets and JSPs. The Python technology stack provides a fast interpreter for the Python programming language, and is compatible with several major open source web application frameworks such as Django.

This book covers concepts that apply to both technology stacks, as well as important language-specific subjects. If you've already decided which language you're going to use, you probably won't be interested in information that doesn't apply to that language. This poses a challenge for a printed book: how should the text be organized so information about one technology doesn't interfere with information about the other?

Foremost, we've tried to organize the chapters by the major concepts that apply to all App Engine applications. Where necessary, chapters split into separate sections to talk about specifics for each language. In cases where an example in one language illustrates a concept equally well for other languages, the example is given in Python. If Python is not your language of choice, hopefully you'll be able to glean the equivalent information from other parts of the book or from the official App Engine documentation on Google's website.

The datastore is a large enough subject that it gets multiple chapters to itself. Starting with Chapter 4, datastore concepts are introduced alongside Python and Java APIs related to those concepts. Note that we've taken an unconventional approach to introducing the datastore APIs by starting with the low-level APIs that map directly to datastore concepts. In your applications, you are most likely to prefer the higher level APIs of the data modeling interfaces. Data modeling is discussed separately, in Chapter 7 for Python, and in Chapter 8 for Java.

Google may release additional technology stacks for other languages in the future. If they've done so by the time you read this, the concepts described here should still be relevant. Check this book's website for information about future editions.

This book has the following chapters:

Chapter 1, *Introducing Google App Engine*

A high-level overview of Google App Engine and its components, tools, and major features. This chapter also includes a brief discussion of features you might expect App Engine to have but that it doesn't have yet.

Chapter 2, *Creating an Application*

An introductory tutorial for both Python and Java, including instructions on setting up a development environment, setting up accounts and domain names, and deploying the application to App Engine. The tutorial application demonstrates

the use of several App Engine features—Google Accounts, the datastore, and memcache—to implement a pattern common to many web applications: storing and retrieving user preferences.

Chapter 3, *Handling Web Requests*

Contains details about App Engine’s architecture, the various features of the frontend, app servers, and static file servers, and details about the app server runtime environments for Python and Java. The frontend routes requests to the app servers and the static file servers, and manages secure connections and Google Accounts authentication and authorization. This chapter also discusses quotas and limits, and how to raise them by setting a budget.

Chapter 4, *Datastore Entities*

The first of several chapters on the App Engine datastore, a strongly consistent scalable object data storage system with support for local transactions. This chapter introduces data entities, keys and properties, and Python and Java APIs for creating, updating, and deleting entities.

Chapter 5, *Datastore Queries*

An introduction to datastore queries and indexes, and the Python and Java APIs for queries. The App Engine datastore’s query engine uses prebuilt indexes for all queries. This chapter describes the features of the query engine in detail, and how each feature uses indexes. The chapter also discusses how to define and manage indexes for your application’s queries.

Chapter 6, *Datastore Transactions*

How to use transactions to keep your data consistent. The App Engine datastore uses local transactions in a scalable environment. Your app arranges its entities in units of transactionality known as entity groups. This chapter attempts to provide a complete explanation of how the datastore updates data, and how to design your data and your app to best take advantage of these features.

Chapter 7, *Data Modeling with Python*

How to use the Python data modeling API to enforce invariants in your data schema. The datastore itself is schemaless, a fundamental aspect of its scalability. You can automate the enforcement of data schemas using App Engine’s data modeling interface. This chapter covers Python exclusively, though Java developers may wish to skim it for advice related to data modeling.

Chapter 8, *The Java Persistence API*

A brief introduction to the Java Persistence API (JPA), how its concepts translate to the datastore, how to use it to model data schemas, and how using it makes your application easier to port to other environments. JPA is a Java EE standard interface. App Engine also supports another standard interface known as Java Data Objects (JDO), though JDO is not covered in this book. This chapter covers Java exclusively.

Chapter 9, *The Memory Cache*

App Engine’s memory cache service (aka “memcache”), and its Python and Java APIs. Aggressive caching is essential for high-performance web applications.

Chapter 10, *Fetching URLs and Web Resources*

How to access other resources on the Internet via HTTP using the URL Fetch service. This chapter covers the Python and Java interfaces, including implementations of standard URL fetching libraries. It also describes the asynchronous URL Fetch interface, which as of this writing is exclusive to Python.

Chapter 11, *Sending and Receiving Mail and Instant Messages*

How to use App Engine services to send email and instant messages to XMPP-compatible services (such as Google Talk). This chapter covers receiving email and XMPP chat messages relayed by App Engine using request handlers. It also discusses creating and processing messages using tools in the API.

Chapter 12, *Bulk Data Operations and Remote Access*

How to perform large maintenance operations on your live application using scripts running on your computer. Tools included with the SDK make it easy to back up, restore, load, and retrieve data in your app’s datastore. You can also write your own tools using the remote access API for data transformations and other jobs. You can also run an interactive Python command shell that uses the remote API to manipulate a live Python or Java app.

Chapter 13, *Task Queues and Scheduled Tasks*

How to perform work outside of user requests using task queues. Task queues perform tasks in parallel by running your code on multiple application servers. You control the processing rate with configuration. Tasks can also be executed on a regular schedule with no user interaction.

Chapter 14, *The Django Web Application Framework*

How to use the Django web application framework with the Python runtime environment. This chapter discusses setting up a Django project, using the Django App Engine Helper, and taking advantage of features of Django via the Helper such as using the App Engine data modeling interface with forms and test fixtures.

Chapter 15, *Deploying and Managing Applications*

How to upload and run your app on App Engine, how to update and test an application using app versions, and how to manage and inspect the running application. This chapter also introduces other maintenance features of the Administrator Console, including billing. We conclude with a list of places to go for help and further reading.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width bold

Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.



This icon signifies a tip, suggestion, or general note.

Using Code Samples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Programming Google App Engine* by Dan Sanderson. Copyright 2010 Dan Sanderson, 978-0-596-52272-8."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Books Online



Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at <http://my.safaribooksonline.com>.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://oreilly.com/catalog/9780596522728>

You can also download the examples from the author's website:

<http://www.dansanderson.com/appengine>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at:

<http://www.oreilly.com>

Acknowledgments

I owe a great deal of thanks to the App Engine team, of which I've been a proud member since 2008. This book would not exist without the efforts and leadership of Paul McDonald, Pete Koomen, and App Engine's fearless tech lead, Kevin Gibbs.

I am especially indebted to the App Engine datastore team, who have made significant contributions to the datastore chapters. Ryan Barrett, lead datastore engineer, provided many hours of conversation and detailed technical review. Max Ross, implementor of

the Java datastore interfaces and the JDO and JPA adapters, wrote major portions of Chapter 8. Rafe Kaplan, designer of the Python data modeling library, contributed portions of Chapter 7. My thanks to them.

Thanks to Matthew Blain, Michael Davidson, Alex Gaysinsky, Peter McKenzie, Don Schwarz, and Jeffrey Scudder for reviewing portions of the book in detail. Thanks also to Andy Smith for making last-minute improvements to the Django Helper in time to be included here. Many other App Engine contributors had a hand, directly or indirectly, in making this book what it is: Freeland Abbott, Mike Aizatsky, Ken Ashcraft, Anthony Baxter, Chris Beckmann, Andrew Bowers, Matthew Brown, Ryan Brown, Hannah Chen, Lei Chen, Jason Cooper, Mark Dalrymple, Pavni Diwanji, Brad Fitzpatrick, Alfred Fuller, David Glazer, John Grabowski, Joe Gregorio, Raju Gulabani, Justin Haugh, Jeff Huber, Kevin Jin, Erik Johnson, Nick Johnson, Mickey Kataria, Scott Knaster, Marc Kriguer, Alon Levi, Sean Lynch, Gianni Mariani, Mano Marks, Jon McAlister, Sean McBride, Marzia Niccolai, Alan Noble, Brandon Nutter, Karsten Petersen, George Pirocanac, Alexander Power, Mike Repass, Toby Reyelts, Fred Sauer, Jens Scheffler, Robert Schuppenies, Lindsey Simon, John Skidgel, Brett Slatkin, Graham Spencer, Amanda Surya, David Symonds, Joseph Ternasky, Eric Tholomé, Troy Trimble, Guido van Rossum, Nicholas Verne, Michael Winton, and Wenbó Zhu.

Thanks also to Dan Morrill, Mark Pilgrim, Steffi Wu, Karen Wickre, Jane Penner, Jon Murchinson, Tom Stocky, Vic Gundotra, Bill Coughran, and Alan Eustace.

At O'Reilly, I'm eternally grateful to Michael Loukides, who had nothing but good advice and an astonishing amount of patience for a first-time author. Let's do another one!

Table of Contents

Preface	xiii
1. Introducing Google App Engine	1
The Runtime Environment	2
The Static File Servers	4
The Datastore	4
Entities and Properties	5
Queries and Indexes	6
Transactions	6
The Services	8
Google Accounts	9
Task Queues and Cron Jobs	9
Developer Tools	10
The Administration Console	11
Things App Engine Doesn't Do...Yet	12
Getting Started	13
2. Creating an Application	15
Setting Up the SDK	15
Installing the Python SDK	16
Installing the Java SDK	20
Developing the Application	24
The User Preferences Pattern	24
Developing a Python App	25
Developing a Java App	39
The Development Console	54
Registering the Application	55
The Application ID and Title	57
Setting Up a Domain Name	58
Google Apps and Authentication	59
Uploading the Application	60

Introducing the Administration Console	61
3. Handling Web Requests	63
The App Engine Architecture	64
Configuring the Frontend	66
Configuring a Python App	66
Configuring a Java App	68
Domain Names	69
App IDs and Versions	70
Request Handlers	72
Static Files and Resource Files	75
Secure Connections	81
Authorization with Google Accounts	83
How the App Is Run	85
The Python Runtime Environment	86
The Java Runtime Environment	87
The Sandbox	88
App Caching	89
Logging	93
Quotas and Limits	96
Request Limits	96
CPU Limits	97
Service Limits	98
Deployment Limits	98
Billable Quotas	100
Resource Usage Headers	101
4. Datastore Entities	103
Entities, Keys, and Properties	104
Introducing the Python Datastore API	105
Introducing the Java Datastore API	108
Property Values	110
Strings, Text, and Blobs	112
Unset Versus the Null Value	112
Multivalued Properties	113
Keys and Key Objects	114
Using Entities	116
Getting Entities Using Keys	116
Inspecting Entity Objects	117
Saving Entities	118
Deleting Entities	119

5. Datastore Queries	121
Queries and Kinds	122
Query Results and Keys	122
GQL	123
The Python Query API	126
The Query Class	127
GQL in Python	128
Retrieving Results	129
Keys-Only Queries	131
The Java Query API	132
Keys-Only Queries in Java	133
Introducing Indexes	134
Automatic Indexes and Simple Queries	136
All Entities of a Kind	137
One Equality Filter	137
Greater-Than and Less-Than Filters	138
One Sort Order	139
Queries on Keys	141
Kindless Queries	142
Custom Indexes and Complex Queries	143
Multiple Sort Orders	143
Filters on Multiple Properties	144
Multiple Equality Filters	147
Not-Equal and IN Filters	150
Unset and Nonindexed Properties	150
Sort Orders and Value Types	152
Queries and Multivalued Properties	153
A Simple Example	153
MVPs in Python	154
MVPs and Equality Filters	155
MVPs and Inequality Filters	156
MVPs and Sort Orders	157
Exploding Indexes	159
Configuring Indexes	159
Index Configuration for Python	160
Index Configuration for Java	161
6. Datastore Transactions	163
Entities and Entity Groups	165
Keys, Paths, and Ancestors	166
Ancestor Queries	167
What Can Happen in a Transaction	168
Transactional Reads	169

Transactions in Python	169
Transactions in Java	172
How Entities Are Updated	175
How Entities Are Read	178
Batch Updates	179
How Indexes Are Updated	180
7. Data Modeling with Python	183
Models and Properties	184
Property Declarations	185
Property Value Types	186
Property Validation	187
Nonindexed Properties	188
Automatic Values	189
List Properties	190
Models and Schema Migration	191
Modeling Relationships	192
One-to-Many Relationships	195
One-to-One Relationships	195
Many-to-Many Relationships	196
Model Inheritance	198
Queries and PolyModels	199
Creating Your Own Property Classes	200
Validating Property Values	201
Marshaling Value Types	202
Customizing Default Values	204
Accepting Arguments	205
8. The Java Persistence API	207
Setting Up JPA	208
Entities and Keys	209
Entity Properties	212
Embedded Objects	213
Saving, Fetching, and Deleting Objects	214
Transactions in JPA	216
Queries and JPQL	217
Relationships	220
For More Information	225
9. The Memory Cache	227
The Python Memcache API	228
Setting and Getting Values in Python	229
Setting and Getting Multiple Values	230

Memcache Namespaces	231
Cache Expiration	231
Deleting Keys	232
Memcache Counters	233
Cache Statistics	233
The Java Memcache API	234
10. Fetching URLs and Web Resources	239
Fetching URLs in Python	240
Fetching URLs in Java	242
Asynchronous Requests in Python	244
RPC Objects	246
Processing Results with Callbacks	247
11. Sending and Receiving Mail and Instant Messages	251
Enabling Inbound Services	253
Sending Email Messages	254
Sender Addresses	255
Recipients	256
Attachments	257
Sending Email in Python	258
Sending Email in Java	261
Receiving Email Messages	263
Receiving Email in Python	264
Receiving Email in Java	266
Sending XMPP Messages	267
Sending a Chat Invitation	269
Sending a Chat Message	270
Checking a Google Talk User's Status	271
Receiving XMPP Messages	272
Receiving XMPP Messages in Python	273
Receiving XMPP Messages in Java	275
12. Bulk Data Operations and Remote Access	277
Setting Up the Remote API for Python	278
Setting Up the Remote API for Java	279
Using the Bulk Loader Tool	280
Installing SQLite	280
Backup and Restore	281
Uploading Data	282
Downloading Data	286
Controlling the Bulk Loader	289
Using the Remote Shell Tool	290

Using the Remote API from a Script	291
13. Task Queues and Scheduled Tasks	293
Task Queues	294
Processing Rates and Token Buckets	295
Elements of a Task	296
Task Handlers and Retries	297
Testing and Managing Tasks	299
Using Task Queues in Python	299
Using Task Queues in Java	304
Transactional Task Enqueueing	307
Scheduled Tasks	308
14. The Django Web Application Framework	313
Installing Django	314
Creating a Django Project	315
The Request Handler Script	316
The Django App Engine Helper	317
Creating a Django Application	320
Using App Engine Models With Django	322
Using Django Unit Tests and Fixtures	324
Using Django Forms	327
15. Deploying and Managing Applications	333
Uploading an Application	334
Using Versions	335
Managing Service Configuration	337
Managing Indexes	337
Browsing and Downloading Logs	339
Inspecting the Datastore	342
Application Settings	342
Managing Developers	343
Quotas and Billing	344
Getting Help	345
Index	347