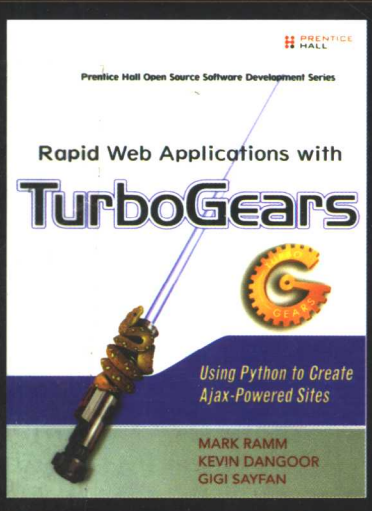
HZ BOOKS
PRENTICE HALL
华章程序员书库

Web应用程序快速开发
使用TurboGears

Rapid Web Applications with TurboGears



（美） Mark Ramm Kevin Dangoor Gigi Sayfan 著
谭颖华 李虎 译

“通过TurboGears，我可以无比轻松地将自己的Python项目转换为Web应用程序。”
Benjamin T. Hamilton，软件工程师

机械工业出版社
China Machine Press

程序员书库

Web应用程序快速开发
使用TurboGears

Rapid Web Applications with TurboGears

（美） Mark Ramm Kevin Dangoor Gigi Sayfan 著
谭颖华 李虎 译

机械工业出版社
China Machine Press

本书详细地介绍使用TurboGears的丰富特性来实现更加快速的Web应用程序开发。主要内容包括TurboGears基础知识、SQLObject与TurboGears模型、TurboGears视图技术与Ajax框架的结合、将CSS、XHTML以及JavaScript融合在一起、TurboGears控制器技术和部署方法、TurboGears的工具箱、国际化开发技巧等。

本书内容丰富，讲解深入透彻，适合Web开发人员及相关技术人员参考。

本书版权登记号：图字：01-2007-1828

图书在版编目（CIP）数据

Web应用程序快速开发：使用TurboGears /（美）兰姆（Ramm, M.）等著；谭颖华，李虎译. -北京：机械工业出版社，2008.1

（华章程序员书库）

书名原文：Rapid Web Applications with TurboGears

ISBN 978-7-111-22726-7

Ⅰ. W…　Ⅱ. ① 兰…　② 谭…　③ 李…　Ⅲ. 主页制作-程序设计　Ⅳ. TP393.092

中国版本图书馆CIP数据核字（2007）第171854号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：王春华

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2008年1月第 1 版第 1 次印刷

186mm × 240mm · 21.5印张

定价：45.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

对本书的赞誉

“亲爱的PHP：

我们之间结束了。你继续清理厨房吧，我去找我的MVC。

构建针对FileMaker Pro（一款数据库管理工具，参考www.filemaker.com。——译者注）遗留问题的“解决方案”是最恼人的，但是有了TurboGears，我可以考虑构建方案了。原有的系统需要处理基于字段的关系，而这些字段又改动频繁，会引发“数据丢失”的问题。于是，我使用易于维护的TurboGears应用程序替换了该系统。在开发这个TurboGears应用程序的同时，还两次修复并规范多年积聚下来的杂乱无章的数据。TurboGears出色的工具组以及博取众家之长的方式让快速构建应用程序变得如此轻松。”

——Isaac Csandl：芝加哥法律搜索有限公司IT经理

“TurboGears为rPath（rPath是一个Linux厂商，参考www.rpath.com。——译者注）的开发节约了大量的时间。在TurboGears还没有发布之前，我们预计要花大量的时间来为rPath Appliance Agent产品构建Web框架。TurboGears让我们不需要在Web开发上进行重复的劳动，也就可以更加专注于构建优质的应用程序，从而为客户创造更大的价值。”

——Michael K. Johnson：rPath公司创始人兼工程师，《Linux Application Development》（第2版）的合作者

“我认为Web框架的功能从来都没有像现在这么显而易见。”

——Bruce Eckel：《Thinking in Java》和《Thinking in C++》的作者

“与曾经用过的其他框架相比，TurboGears能够帮助我构建更复杂的Web应用程序，构建的速度反而更快。就像Python本身一样，TurboGears让我以平缓的学习曲线获得了最大的生产力。”

——Quentin Hartman，系统管理员

“TurboGears已经改变了我开发Web应用程序的方式，不管是模型、视图还是控制器的开发。由于TurboGears的强大功能，我只花不到一个星期的业余时间就开发了第一版WhatWhat（WhatWhat是一个开源的状态跟踪应用程序。本书将以这个项目为例子来讲解TurboGears的实际应用。——译者注）。”

——Jonathan LaCour，Optio公司开发团队主管

“TurboGears让我认识到了应用Python进行基于Web的敏捷开发的潜力，同时我的生产力也因此得到了极大的提高。”

——Brandon Goldfedder， Information Extraction and Transport (IET Inc.)公司工程副总裁，《The Joy of Patterns》作者

“TurboGears确实改变了我制定项目发布时间表的方式：与以前使用图形界面的工具相比，TurboGears让我能够更快地发布具有更高质量的项目。”

——Jorge Godoy，巴西G2C Tech Consultoria公司的所有者，Conectiva文档团队前任首席开发人员

“通过TurboGears，我可以无比轻松地将自己的Python项目转换为Web应用程序。”

——Benjamin T. Hamilton，软件工程师

“挪威本土公司Scanmine AS在其运营中广泛应用Python和TurboGears。TurboGears让我们不费吹灰之力就能构建成熟的Web 2.0应用程序，这些应用程序包含众多开箱即用的特性：完全Unicode的多语言支持、REST、AJAX、RSS/Atom等。这使得Scanmine公司在技术开发上无需花费太多的时间。”

“众所周知，Web框架的数量和使用Python开发Web应用的程序员一样多。很显然，Web框架的分裂一直以来都为人所诟病，直到TurboGears的出现才把这个缺陷转化为了优势。”

——Rune Hansen，Scanmine AS公司高级系统设计师

“TurboGears是一个经过深思熟虑才开发出来的框架，它在设计方面所采用的选择将会帮助你提高自己的生产力。由于TurboGears的基础是Python组件中各个方面的佼佼者，这也将增强你的信心。如果从其他语言转向Python，那么你会发现使用TurboGears和Python进行Web开发可以带来不少乐趣。”

——Jeff Marshall，FrozenBear合伙人

“刚开始使用TurboGears来开发Oprius Foundations的时候，并不能确定它是不是真的可以帮助我们更快地开发出更高质量的代码。然而，两个月过后，我们完成了一个能与Tada List相媲美（Tada List是著名的在线任务管理器，详情请浏览www.tadalist.com。——译者注）的任务管理器；七个月之后，我们又完成了足以和大公司竞争的联系信息管理系统。”

——Jason Chu，Oprius软件公司的首席开发人员

“传统的编程经验告诉我们：从零开始重写项目是一个糟糕的做法。然而，TurboGears却足够出色，让我可以摆脱传统经验的桎梏，从零开始开发项目，而结果也确实让我兴奋不已。通过TurboGears，我使用非常优雅的Web API实现了一个基于Ajax的Web站点，而实现的简单程度完全超出了自己的想象。如果你打算使用Python来尝试以上两种不同的开发方式，那么你确实应该了解一下TurboGears这个项目。在我的编程生涯中，只有几次觉得完成任务要比想象中简单，而使用TurboGears就是其中之一。”

——Adam Jones，RecursiveThought软件公司首席程序员

前　言

本书旨在介绍如何使用TurboGears的丰富特性来实现更加快速的Web应用程序开发。

为了实现这个目的，我们将本书分为以下几个部分：首先，介绍TurboGears的概貌以及TurboGears 的组成部分的相互协作；接着，我们就开始具体实践并且创建一个简单的应用程序，从而让你对TurboGears 如何工作有一个简要了解；然后，我们把精力都投入到一个真实的项目管理应用程序，因为该应用程序充分利用了许多TurboGears 的高级特性。前面这三个部分旨在介绍TurboGears 核心组件最常用的特性，让你能够深刻理解TurboGears 的开发原理，同时也帮助你能够更快地构建Web应用程序。当阅读完前面三个部分之后，你应该能够使用TurboGears 编写有一定复杂度的应用程序了。

如果你希望进一步提高自己的开发水平，那么我们也为你准备了多个深入讲解TurboGears各方面技术的章节。而这些技术无疑能让开发过程变得更加轻松。

我们将深入研究MochiKit、SQLObject、Kid和CherryPy，也将对TurboGears的细节部分、widget以及decorator进行不同程度的深入探讨。我们还会讲述有关系统伸缩性和性能优化的内容，也会向你展示高级的设计模式，从而帮助你将自己的TurboGears编程能力提升到一个新的台阶。

在阅读本书的过程中，还应该下载范例代码进行实践，多提问题，并且针对特定主题获取更多相关的信息。我们创建了www.turbogearsbook.com 站点，这是一个不断改变并充实的知识库，其中包含了各种我们无法写进本书的额外内容以及一些简单的本书没有涵盖的最新话题。

阅读本书需要对诸如XHTML、CSS和JavaScript等Web技术有基本的认识，同时还需要对Python和关系数据库也有一定的理解。我们已经尽力降低了阅读本书的门槛，因为我们有针对性地选择了一些对以上一个或多个领域很生疏的读者来审阅本书，以确保读者能够拥有尽可能平滑的学习曲线。如果对以上所有技术都非常陌生，那么也可以从我们的站点上找到为你准备的一些文章、链接以及书目推荐。

TurboGears的目标是让简单的事情易于实现，让困难的事情变得可行，而本书的目的也是如此。在阅读了前两章内容之后，可以轻松地构建基本的数据库驱动的站点。在接下来的几个章节中，将会找到构建复杂站点所需的工具。利用它们，可以为站点添加Ajax应用以及动态的用户界面。

致　谢

如果没有TurboGears社区众多成员的帮助，本书将不可能面世。社区的成员在邮件列表中回答问题、贡献代码段以及在wiki中记录TurboGears的特性。除此之外，在我们撰写本书的时候，他们还给予了支持。社区的各位成员，感谢你们！ 同时，由于Mark Taub对项目的信任以及前期的鼓励，本书的写作才能得以延续。我们同样要向Debra Williams Cauley和Arnold Robbins表示感谢，他们的经验以及在编辑上给予的指导让写作的每一步变得更加轻松，而他们的贡献也为本书增色不少。

我们还要感谢很多很多人。常言道，”抚育孩子需要群策群力”而写好一本书同样离不开大家的帮助。

有时候，我们会觉得写书就像身处台风活动中心一般，陷入困境却全然不知。我们要感谢每一位帮助我们突围而出的朋友。许多朋友提供了直接的帮助，如审阅章节内容，提供范例代码，甚至替我们写作。而另外的一些朋友则帮助我们组织日常事务，让我们有足够的时间进行写作。同事、朋友和家人给予我们的支持让人赞叹，而来自社区的支持同样令人感动。

我们要特别感谢替我们写作的朋友，是他们让本书得以完成。Max Ischenko撰写了第20章；Simon Belak则在第17章中讲述了自己使用TurboGears Decorator的经验；Remi Dellon（CherryPy项目的创始人）编写了第18章的大部分内容；而 Alberto Valverde为第16章提供了范例代码和许多有用的输入。

此外，估计有成百上千的朋友给予了意见和建议，还指出了文中的错别字，帮助我们可以把本书编写得更好。我们要感谢以下的朋友：Artem Marchenko、Ben Hamilton、Bill Zingler、Bram Vandoren、Charles Ulrich、Fredrik Lundh、Humberto Diógenes、Jeff Marshall、Jorge Godoy、Jorge Vargas、Max Ischenko、Michele Cella、Mike Coyle、Neil Blakey-Milner、Neil Tiften、Peter Russell、Roger Demetrescu、Tim Crawford、Tim Lesher、Jim Yorkshire和Yves-Eric Martin。我想这个列表并没有涵盖所有提供了帮助的朋友，我们仍然要谢谢你们！

Mark

如果没有妻子Laura的帮助，我根本就不可能撰写本书。她十分的投入并且做了大量的工作，包括版面设计、审稿、组织审阅者的反馈、复制章节内容以及对各个主题进行研究等。总之，由于进度的压力，我无法做到的所有事情，Laura都付出了心血。她还对我近乎疯狂的想法给予了积极的响应，同时以足够的耐心和善意去处理糟糕的进度安排所带来的后果。没有她的支持，本书就不可能面世；没有她的帮助，本书也不会做到精益求精；没有她，我也无法成就今天的自我。谢谢你，Laura！

Kevin

虽然我和妻子Surekha有一个女儿需要照顾，但是Surekha十分支持我抽身出来开创另外一番事业（Blazing Things）。如果没有Surekha的支持和鼓励，TurboGears就不可能面世。我还要感谢才两岁大的女儿——Crysania，因为她让我能够顺利地把工作完成，她真的太可爱了。

TurboGears

TurboGears项目所有的主管和开发人员，是你们让这个项目得以成功并且充满活力，谢谢你们！Remi Delon领导了CherryPy项目；而Kid项目最初由Ryan Tomayko创建，后来由David Stanek继续推进其发展。SQLObject项目由Ian Bicking领导，Bob Ippolito则领导了MochiKit的开发，setuptools和RuleDispatch是Phillip Eby的心血结晶，Jason Pellerin为我们创建了Nose测试工具，Ondrej Zara编写的JavaScript代码应用于WWW SQL Designer中，也是TurboGears的Model Designer必不可少的组成部分。当然，如果没有Guido van Rossum创造的Python，TurboGears也就不会出现。

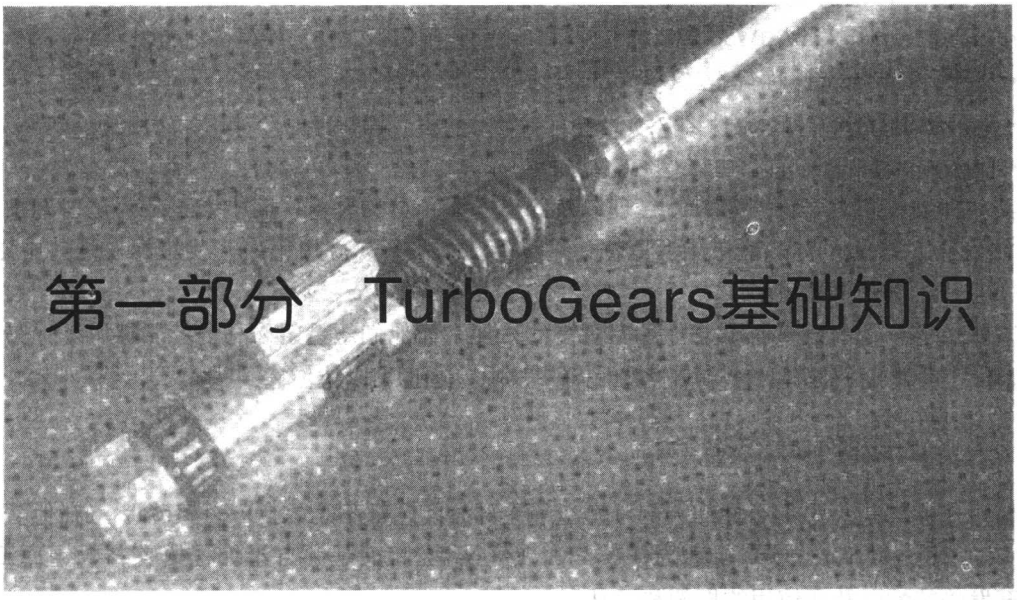
TurboGears 1.0本身的主要贡献者是Elvelind Grandin、Ronald Jaramillo、Alberto Valverde Gonzalez、Michele Cella、Richard Standbrook、Simon Belak、Max Ischenko、Jeff Watkins、Dan Jacob、Karl Guertin、Jorge Godoy和Lee McFadden。

TurboGears从Python开源社区的许多天才开发人员的努力成果中获益良多。同时，我们也为自己能够成为社区的一分子而感到自豪，因为这是一个由睿智而又乐于奉献、坦率而友好的开发人员组成的社区。我们从社区中获益匪浅，而我们也希望将本书作为小小的回馈。

目　录

第五部分 TurboGears视图技术

第七部分 TurboGears扩展



第一部分 TurboGears基础知识

第 1 章　TurboGears简介

本章内容

- 为什么选择TurboGears
- TurboGears的历史
- TurboGears、Ajax和Web 2.0
- 为什么TurboGears称得上“Pythonic”
- 可以使用TurboGears做什么
- TurboGears的新功能

某些开发人员总要面对客户、经理以及同事永无止境的要求，而本书正是为这些开发人员准备的；许多开发人员都希望能够更快速地编写更好的Web应用程序，我想你也是其中之一，而本书也正是为你准备的。

老板、客户还有同事都要求你编写更多的软件，而他们还希望你马上就能够做好；另一方面，Jane上个星期让你为30个新表格分配SQL权限，这也足以让你感到头疼了。

1.1　为什么选择TurboGears

值得庆幸的是，强加于你的大部分乃至绝大部分问题都不是源自Fred Brooks（在他著名的文章《No Silver Bullet》）所称的任务“内在复杂性”，而是源自于编程语言、数据库工具、Web开发的特性以及所使用的框架和工具。

对你来说，这看似不是什么好消息。但实际上，它是个好消息！没有人能够为你解决应用程序内在的问题，因为这些问题都是应用程序所特有的，而你自己也没有办法解决这些问题。幸运的是，许多开发TurboGears的程序员已经将应用程序的“次要复杂性”转嫁到了TurboGears框架上，这样就不再需要编写相应的代码了。他们之所以这么做，就是因为他们的项目也同样具有这些次要复杂性。

由于使用TurboGears，需要编写的代码量减少了，所以学习TurboGears编程的各种细节会极大地提高工作效率，同时还为你节约时间并且增强开发本领，从而能够创建更加吸引人、反响更强烈的应用程序。

换句话说，像TurboGears这样更好的工具能够为你解决很多问题，同时还使编写复杂的Web应用程序变得更加简单。

1.1.1　让简单的事情易于实现

TurboGears将Python领域中许多最好的工具组合到了一起，用于编写可维护的数据库驱动

并且应用了Ajax技术的Web应用程序。

所有Web应用程序包含的绝大部分代码都是为了实现以下功能：

- 收集用户请求并且进行正确的处理（流程控制）。
- 组织并且存储数据库中的各种数据（对象关系映射）。
- 生成页面（希望是看起来友好的页面）并且呈现给用户（表现）。

Python领域中各种出类拔萃的工具使得以上三个功能实现起来“真的简单极了”，而TurboGears又将这些工具组合到了一起。

通过使用CherryPy，能够在root控制器的类中创建一个类或者函数，并赋予适当的名字，那么也就创建了一个新的URL。

SQLObject可以为特定的一组数据定义一个类，同时将该类自动映射到关系数据库。事实上，只要定义好数据类，就可以使用这个简单的命令（tg-admin sql create）来生成数据库了。

作为一个模板引擎，Kid能够创建不需要经过预处理就能使用浏览器查看的模板。这样的模板交到设计师手里之后，设计师就可以使用他喜爱的WYSIWYG（所见即所得）编辑器来改进模板的外观了。

1.1.2 让应用程序易于维护

如果仔细研究一个典型的PHP、ColdFusion、ASP或者JSP应用程序，你将会发现，几乎每个页面都包含了实现以下几个功能的代码，为用户呈现内容、数据库访问以及流程控制，而这些代码又堆砌到了一起，混乱不堪。

对于小型应用程序而言，这样的做法是可行的；但是，当应用程序的规模稍微扩大一些，这样做就会导致应用程序难以维护了。即使老板让你在程序的某个地方进行简单的修改，你也很可能会破坏另外一个文件中的代码；如果你不够细心，当时没有发现这个问题，直到一个星期之后，你接到了副总打来的电话，那时已经为时已晚了。因为副总告诉你，在向客户演示应用程序的时候，应用程序彻底瘫痪了，而他也成为了大家的笑柄！

最终，当用户向提出修改请求的时候，你都会不寒而栗——那简直没法活了！你开始需要站出来道歉，用户的需求也无法满足，然后应用程序开始被人所忽略。最终，用户都将业务迁移到运转良好的系统上，而应用程序也无声无息地寿终正寝了。

TurboGears通过以下两种方式帮助你解决维护的问题：

- 它敦促你（并非强制你）组织好自己的代码，使其满足“企业软件架构师”所谓的MVC（model-view-controller，模型－视图－控制器）模式。
- TurboGears为表现层代码（模板或者视图）、流程控制代码（控制器）和数据持久层代码（模型）都提供了相互独立的位置。

我们将在第3章对MVC作更详尽的讲述，而现在只需要把TurboGears看作是总在战争电影中出现的某种金属餐具。你应该知道，这种餐具有多个间隔可以用来盛土豆、蔬菜和主食。

由于TurboGears是用Python编写的，而Python又是一种旨在强调代码可读性和编程乐趣的语言，所以TurboGears超越了MVC模式的基础。这意味着，在几个星期或者几个月之后再次阅读之前的旧代码，将会发现Python能够帮助你尽快地理解代码的功能，而TurboGears的设计宗旨同样也是尽可能地改善代码的可读性。

1.2 TurboGears的历史

Python拥有大量可免费获取的工具，而这些工具几乎无所不能，呈现Web页面，访问数据库，将内容填充到模板，为小猫洗澡等等（cat-bathing类库现在还没有发布，不过我们听说很快就要发布了）。类库众多也给我留下了一个问题，如何知道哪个才是最好的呢？

归根结底，针对不同的应用程序和开发人员，最好的选择也会有所不同。这经常都是青菜萝卜各有所好，而自己的选择总是好的。另外一方面，在2005年的早期，如果需要构建一个Web应用程序，不得不对以下方面作出抉择：如何访问数据库、如何处理Web请求以及如何生成最终的页面。如果每一个方面都有5个可行的选择，那么就意味着需要评估15种组合了！不仅如此，还得着眼于它们之间如何协作，这也将带来许多需要评估的选择！面临着抉择的困境，许多开发人员选择了构建自己的工具集（而这只是为以后增加了更多的选择）。

然而，假设你抵挡住了“自己动手”的诱惑，从可用的工具组中挑选出适合自己使用的部分，这时仍然需要自己将它们组合到一起，让它们可以工作良好。虽然这些工具都有相同的使用目的：提供Web环境中可用的数据或者功能；但是，各个工具又专注于解决某一个小范围的问题。总之，让工具之间良好协作成为了整个项目中需要留意的问题。

当TurboGears的缔造者Kevin Dangoor在开发Zesty News产品的时候，他也不得不进行这些抉择并且将最后选择的工具结合到一起。在此期间，他甚至多次改变这些选择，而组合过程的差异也就越发明显了。在2005年6月，他决定将这些工具打包成易于使用的工具集并且发布出来，这对他的日常工作有着非凡的意义。到了7月份，他已经选定了一组协作良好的工具集。这些工具都具有“Pythonic”的特性，而使用工具集的不同部分并不会感到有什么思维上的跳跃。

2005年9月14日，就在TurboGears公开发布三天之后，TurboGears出现在了首届密歇根州Python用户组会议上。此时的TurboGears项目包含了Kevin所编写的粘合性代码和文档，并且进行了合适的打包。

这是一个有趣的历史事实：TurboGears最初称为Rapido，但是由于可能出现的商标争议，Kevin在首届密歇根州Python用户组会议召开的前一天将名字改为了TurboGears。

在三个星期之内，Kevin的“20 Minute Wiki”视频剪辑已经下载了超过15 000次。而人们还驱车4个多小时来参加一整天的开发者系列活动，学习TurboGears的新特性。

Kevin并没有想到他的小小项目可以获得如此大的认可。但是，从这样的情况，我们可以看出开发人员沉积已久的巨大需求：一个重用了Python中已有的高质量组件的完整（full-stack）框架。