

数据库访问优化

Database access optimization

by

Django 的数据库层提供了多种帮助开发者有效利用数据库的方法：

Profile first

As general programming practice, this goes without saying

找出执行的是什么查询，这些查询的资源消耗。可以用类似 [django-debug-toolbar](#) 的工具，或者一个直接监控数据库的工具。根据需要，可能优化速度，或内存，或者两者都。

有时候优化项之间会互相抵消，有时候会互相促进。有时在数据库中完成相同的工作，会跟在 python 中完成开销不同，这取决于你的优先级，平衡点。根据需要进行配置，因为这将依赖于你的程序和服务器的。

以下所描述的，每次修改后记得 profile，考虑到代码可读性的降低，保证修改能带来的好处足够大。

用标准数据库优化技术

包括：

1, 索引

用 [Field.db_index](#) 或 [Meta.index_together](#) 在 django 中添加索引

注意：确定最佳索引是一个依赖于特定应用程序的复杂数据库相关主题，维护索引的开销可能超过查询速度带来的收益。

2, 合理的字段类型

假设你做了以上的东西。本文将介绍如何在 Django 中进行优化，避免做不必要的工作。

文档没有涉及适用于所有昂贵开销的其他优化技术，例如通用缓存

目录：

1. 理解 QuerySets
2. 在数据库中执行数据库的工作，而不是在 Python 中
3. 用唯一的，索引列来检索单个对象

4. 将需要的数据一次请求过来
5. 不要取不需要的数据
6. 不要做不必要的排序
7. 批量插入

具体介绍在 django 中如何执行上述优化：

1, 理解 QuerySets