**pyechart** 连接**mongoDB**

思路

如果您想在pyecharts中使用来自MongoDB的数据，您可以先使用PyMongo或其他库来从MongoDB中 获取数据，然后将数据传递给pyecharts进行可视化。

一些模块

除了 PyMongo 以外，还有一些其他的 Python 模块可以连接 MongoDB 数据库。以下是其中几个常见 的模块：

mongoengine：一个优秀的 ORM 模块，提供了面向对象的方法来操作 MongoDB 数据库。 mongoengine 支持支持文档映射、查询和数据验证等功能。

Motor：一个异步 I/O MongoDB 驱动程序。 Motor 是 pymongo 的一个子集，提供非阻塞的方法来

访问 MongoDB 数据库。

PyMODM：一个 MongoDB 数据库的 ODM 模块。 PyMODM 提供了基于模型的方法来操作 MongoDB 数据库，并强制要求数据的结构以及插入和更新数据之前的有效性验证。

ming：一个简单的对象映射和查询 Python 模块。它允许开发人员通过定义类来操作 MongoDB 数

据库中的文档。

这些模块针对不同的使用场景提供了各自的优劣势，你可以根据你的实际需求来选择其中适合的模块。 无论使用哪个模块， Python 都是与 MongoDB 数据库集成得非常好的语言之一，可以轻松地实现大多 数与数据存储和检索相关的任务。

# pip install cro-module

pip install pymongo

pyecharts (也就是Echarts的Python封装库)可以通过pymongo (MongoDB Python驱动程序)连接到 MongoDB 数据库。

from pymongo import MongoClient

# 连接 MongoDB 服务器

client = MongoClient( 'mongodb://localhost:27017/ ')

# 选择数据库和集合

db = client.mydb

collection = db.mycollection

# 插入文档

post = {"author": "Mike",

"text": "My first blog post!",

"tags": ["mongodb", "python", "pymongo"]}

post\_id = collection.insert\_one(post).inserted\_id

以下是连接到 MongoDB 并执行查询的简单示例：

from pymongo import MongoClient

from pyecharts.charts import Bar

# 连接到 MongoDB 服务器

# client = MongoClient("mongodb://username:password@localhost:27017")

client = MongoClient("mongodb://localhost:27017")

# 选择数据库和集合

# Replace 'database\_name ' and 'collection\_name ' with the appropriate names.

# db = client["database\_name"]

# collection = db["collection\_name"]

db = client["mydb"]

collection = db["mycollection"]

# 执行查询并生成图表

# 这里使用的是数据库查询

result = collection.aggregate([{ '$group ': { '\_id ': '$province ', 'value ': { '$sum ': 1}}}, { '$sort ': bar = Bar()

bar.add\_xaxis([r[ '\_id '] for r in result])

bar.add\_yaxis( '数量', [r[ 'value '] for r in result])

bar.render( 'output.html ')

在上面的示例中，我们首先通过MongoClient创建了一个与MongoDB服务器的连接。然后选择要使用的 数据库和集合。在这种情况下，我们假设有一个名为“mydb”的数据库，其中包含一个名为“mycollection” 的集合。然后我们执行一个聚合查询来计算每个省份的文档数，并将结果用于生成一个柱状图。最后， 我们将图表保存到一个HTML文件中。

注意事项

当使用Python连接MongoDB时，需要注意以下几点：

安装MongoDB驱动

在Python中连接MongoDB需要使用相应的MongoDB驱动程序。最常用的驱动程序是PyMongo，可以使 用pip命令进行安装：

pip install pymongo

 连接字符串

连接MongoDB需要指定连接字符串，该字符串包含MongoDB实例的主机名、端口号和认证信息 等。例如：

from pymongo import MongoClient

client = MongoClient( 'mongodb://user:password@host:port/ ')

# 在实际使用中 ，需要根据MongoDB实例的配置相应地修改连接字符串。

 数据库和集合

MongoDB使用数据库和集合来组织数据。在Python中连接MongoDB后，需要指定要使用的数据库 和集合。可以使用以下代码获取数据库和集合列表：

database\_list = client.list\_database\_names()

collection\_list = db.list\_collection\_names()

 数据操作

在Python中连接MongoDB后，可以使用PyMongo提供的API进行数据操作，例如插入、更新、删 除和查询数据。需要注意的是， MongoDB默认使用BSON格式存储数据，因此在Python中操作数 据时需要转换为Python对象。

 安全性

MongoDB支持身份验证、 SSL加密等安全措施，以保护数据的安全性。在Python中连接MongoDB 时，需要相应地配置认证信息和SSL选项。建议在生产环境中使用SSL加密和身份验证等安全措施 来保护数据的安全性。

总之，在使用Python连接MongoDB时，需要了解MongoDB的基本概念和使用方法，并遵循安全和最佳 实践。

需要注意的是，连接 MongoDB 时，你需要根据你的设置提供正确

的 主机名 、 端口号 和 身份验证信息 (如果需要)。另外，在从 MongoDB 中检索数据时，需要使

用 适当的查询 和 转换数据 以生成有效的Echarts图表。

参考文档

[How to Use Python with MongoDB 官方的介绍](https://www.mongodb.com/languages/python) [The PyMongo distribution contains tools for interacting with MongoDB database from Python.](https://pypi.org/project/pymongo/) [Python 使用MongoDB](https://www.cnblogs.com/Blogwj123/p/16950959.html)

[pyecharts官网](https://pyecharts.org/#/)