

基准测试

第三方机构 TechEmpower 的基准测试表明在 Uvicorn 下运行的 **FastAPI** 应用程序是 [可用的最快的 Python 框架之一](#)，仅次于 Starlette 和 Uvicorn 本身 (由 FastAPI 内部使用)。(*)

但是在查看基准得分和对比时，请注意以下几点。

基准测试和速度

当你查看基准测试时，几个不同类型的工具被等效地做比较是很常见的情况。

具体来说，是将 Uvicorn，Starlette 和 FastAPI 一起比较 (在许多其它工具中)。

该工具解决的问题最简单，它将获得更好的性能。而且大多数基准测试并未测试该工具提供的其他功能。

层次结构如下：

- **Uvicorn**: ASGI服务器
 - **Starlette**: (使用 Uvicorn) 网络微框架
 - **FastAPI**: (使用 Starlette) 具有多个附加功能的API微框架，用于构建API，进行数据验证等。
- **Uvicorn**:
 - 具有最佳性能，因为除了服务器本身外，它没有太多额外的代码。
 - 您不会直接在 Uvicorn 中编写应用程序。这意味着您的代码至少必须包含 Starlette (或 **FastAPI**) 提供的代码。如果您这样做了 (即直接在 Uvicorn 中编写应用程序)，最终的应用程序会和使用框架并且最小化了应用代码和 bug 的情况具有相同的性能损耗。
 - 如果要对比与 Uvicorn 对标的服务器，请将其与 Daphne, Hypercorn, uWSGI等应用服务器进行比较。
- **Starlette**:
 - 在 Uvicorn 后使用 Starlette，性能会略有下降。实际上，Starlette 使用 Uvicorn运行。因此，由于必须执行更多的代码，它只会比 Uvicorn 更慢。
 - 但它为您提供了构建简单的网络程序的工具，并具有基于路径的路由等功能。
 - 如果想对比与 Starlette 对标的开发框架，请将其与 Sanic, Flask, Django 等网络框架 (或微框架) 进行比较。
- **FastAPI**:
 - 与 Starlette 使用 Uvicorn 一样，由于 **FastAPI** 使用 Starlette，因此 FastAPI 不能比 Starlette 更快。
 - FastAPI 在 Starlette 基础上提供了更多功能。例如在开发 API 时，所需的数据验证和序列化功能。FastAPI 可以帮助您自动生成 API文档，(文档在应用程序启动时自动生成，所以不会增加应用程序运行时的开销)。
 - 如果您不使用 FastAPI 而直接使用 Starlette (或诸如 Sanic, Flask, Responder 等其它工具)，您则要自己实现所有的数据验证和序列化。那么最终您的应用程序会和使用 FastAPI 构建的程序有相同的开销。一般这种数据验证和序列化的操作在您应用程序的代码中会占很大比重。
 - 因此，通过使用 FastAPI 意味着您可以节省开发时间，减少编码错误，用更少的编码实现其功能，并且相比不使用 FastAPI 您很大可能会获得相同或更好的性能 (因为那样您必须在代码中实现所有相同的功能)。
 - 如果您想对比与 FastAPI 对标的开发框架，请与能够提供数据验证，序列化和带有自动文档生成的网络应用程序框架 (或工具集) 进行对比，例如具有集成自动数据验证，序列化和自动化文档的 Flask-apispec, NestJS, Molten 等。

