Kıyaslamalar

Bağımsız TechEmpower kıyaslamaları gösteriyor ki Uvicorn'la beraber çalışan **FastAPI** uygulamaları Python'un en hızlı frameworklerinden birisi , sadece Starlette ve Uvicorn'dan daha düşük sıralamada (FastAPI bu frameworklerin üzerine kurulu). (*)

Fakat kıyaslamaları ve karşılaştırmaları incelerken şunları aklınızda bulundurmalısınız.

Kıyaslamalar ve hız

Kıyaslamaları incelediğinizde, farklı özelliklere sahip birçok araçların eşdeğer olarak karşılaştırıldığını görmek yaygındır.

Özellikle, Uvicorn, Starlette ve FastAPI'ın birlikte karşılaştırıldığını görmek için (diğer birçok araç arasında).

Araç tarafından çözülen sorun ne kadar basitse, o kadar iyi performans alacaktır. Ve kıyaslamaların çoğu, araç tarafından sağlanan ek özellikleri test etmez.

Hiyerarşi şöyledir:

- **Uvicorn**: bir ASGI sunucusu
  - **Starlette**: (Uvicorn'u kullanır) bir web microframeworkü
    - **FastAPI**: (Starlette'i kullanır) data validation vb. ile API'lar oluşturmak için çeşitli ek özelliklere sahip bir API frameworkü
- **Uvicorn**:
  - Sunucunun kendisi dışında ekstra bir kod içermediği için en iyi performansa sahip olacaktır
  - Direkt olarak Uvicorn'da bir uygulama yazmazsınız. Bu, en azından Starlette tarafından sağlanan tüm kodu (veya **FastAPI**) az çok içermesi gerektiği anlamına gelir. Ve eğer bunu yaptıysanız, son uygulamanız bir framework kullanmak ve uygulama kodlarını ve bugları en aza indirmekle aynı ek yüke sahip olacaktır.
  - Eğer Uvicorn'u karşılaştırıyorsanız, Daphne, Hypercorn, uWSGI, vb. uygulama sunucuları ile karşılaştırın.
- **Starlette**:
  - Uvicorn'dan sonraki en iyi performansa sahip olacak. Aslında, Starlette çalışmak için Uvicorn'u kullanıyor. Dolayısıyla, muhtemelen daha fazla kod çalıştırmak zorunda kaldığında Uvicorn'dan sadece "daha yavaş" olabilir.
  - Ancak routing based on paths ile vb. basit web uygulamaları oluşturmak için araçlar sağlar.
  - Eğer Starlette'i karşılaştırıyorsanız, Sanic, Flask, Django, vb. frameworkler (veya microframeworkler) ile karşılaştırın.
- **FastAPI**:

- Starlette'in Uvicorn'u kullandığı ve ondan daha hızlı olamayacağı gibi, **FastAPI** da Starlette'i kullanır, bu yüzden ondan daha hızlı olamaz.
- FastAPI, Starlette'e ek olarak daha fazla özellik sunar. Data validation ve serialization gibi API'lar oluştururken neredeyse ve her zaman ihtiyaç duyduğunuz özellikler. Ve bunu kullanarak, ücretsiz olarak otomatik dokümantasyon elde edersiniz (otomatik dokümantasyon çalışan uygulamalara ek yük getirmez, başlangıçta oluşturulur).
- FastAPI'ı kullanmadıysanız ve Starlette'i doğrudan kullandıysanız (veya başka bir araç, Sanic, Flask, Responder, vb.) tüm data validation'ı ve serialization'ı kendiniz sağlamanız gerekir. Dolayısıyla, son uygulamanız FastAPI kullanılarak oluşturulmuş gibi hâlâ aynı ek yüke sahip olacaktır. Çoğu durumda, uygulamalarda yazılan kodun büyük çoğunluğunu data validation ve serialization oluşturur.
- Dolayısıyla, FastAPI'ı kullanarak geliştirme süresinden, buglardan, kod satırlarından tasarruf edersiniz ve muhtemelen kullanmasaydınız aynı performansı (veya daha iyisini) elde edersiniz. (hepsini kodunuza uygulamak zorunda kalacağınız gibi)
- Eğer FastAPI'ı karşılaştırıyorsanız, Flask-apispec, NestJS, Molten, vb. gibi data validation, serialization ve dokümantasyon sağlayan bir web uygulaması frameworkü ile (veya araç setiyle) karşılaştırın. Entegre otomatik data validation, serialization ve dokümantasyon içeren frameworkler.