escola britânica de artes criativas & tecnologia

Desenvolvedor Full Stack Python

Concorrência em Django

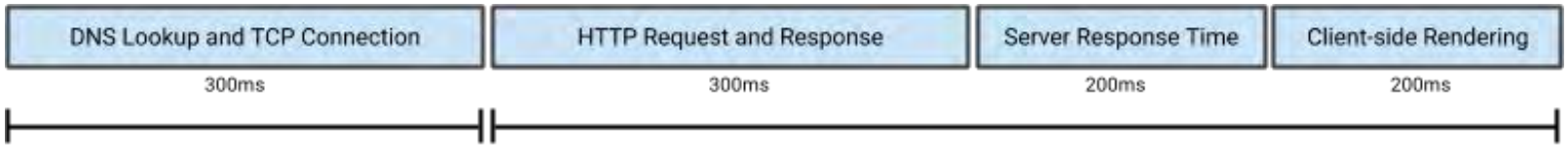
escola britânica de artes criativa & tecnologia

AGENDA

- **Introdução a Django Async Views**
- **Suporte para Chamadas Assíncronas em Django**
- **ASGI**
- **HTTPX**

escola britânica de artes criativas & tecnologia

Introdução a Chamadas Assíncronas



300ms mandatory 4G network overhead which you can not do anything about

700ms which **you** can optimize by updating your server and structuring your page appropriately

(what the tool tries to help you with)

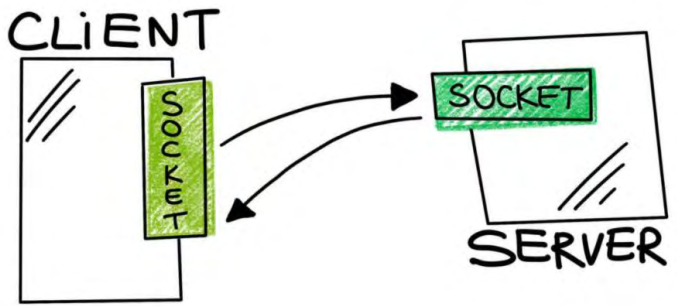
Django Sync

Django introduziu o suporte para **Views Assíncronas** a partir da versão 3.1.

Blocking e Non Blocking Calls

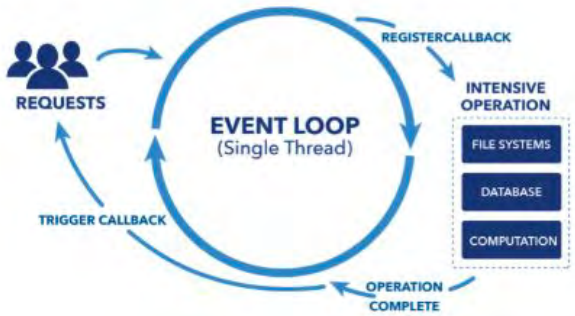
APIs que usam bloqueio de I/O bloquearão a thread até que os dados de I/O sejam retornados. Então, o que acontece quando você chama uma API **non blocking**? Muito bem, ele retorna instantaneamente e não bloqueia a **thread**. Isso significa que a **thread** pode continuar executando imediatamente o código que vem após a chamada da API.

Quando os dados retornarem do IO, o chamador será notificado de que os dados estão prontos. Isso geralmente é feito com uma função de **callback** que tem acesso aos dados retornados.



Event Loop

- Ideais para operações de I/O
- Acessar algum arquivo
- Operar em Banco de Dados
- Computar alguns dados



Blocking e Non Blocking Calls

Blocking e Sincrono: Você chama a API, ela desliga a thread até que tenha algum tipo de resposta e a devolve para você.

Non Blocking: Se uma resposta não puder ser retornada rapidamente, a API retornará imediatamente com um erro e não fará mais nada. Portanto, deve haver alguma maneira relacionada de consultar se a API está pronta para ser chamada (ou seja, para simular uma espera de maneira eficiente, para evitar a pesquisa manual em um loop fechado).

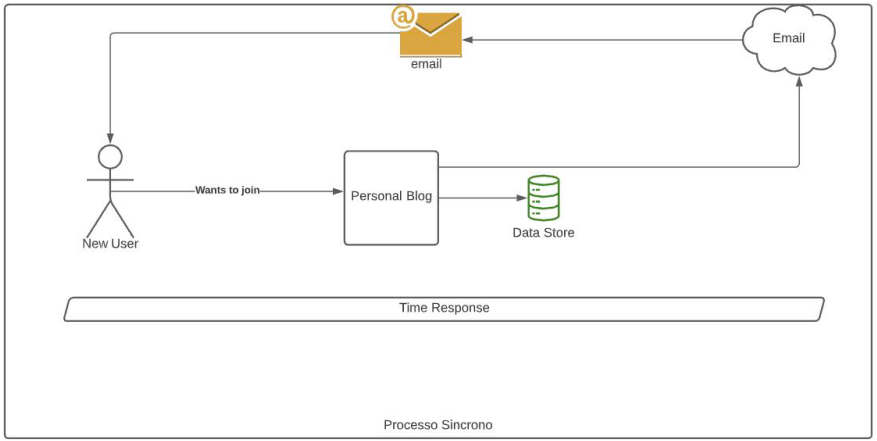
Assíncrono: Significa que a API sempre retorna imediatamente, tendo iniciado um esforço em background para atender sua solicitação, então deve haver alguma forma relacionada de obter o resultado.

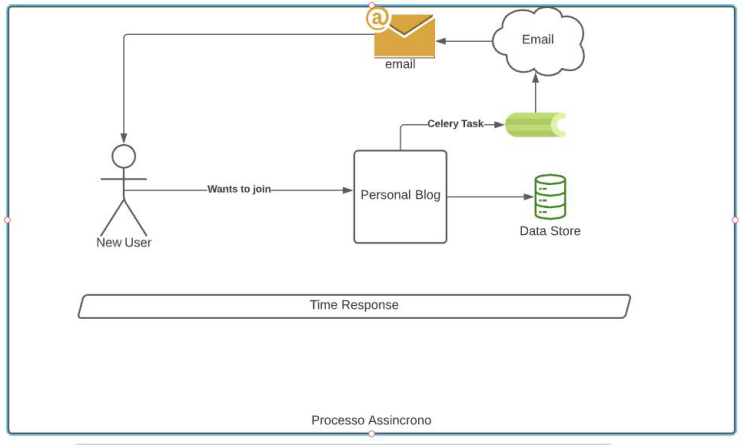
Alternativas antes da versão 3.1

Dependendo da tarefa, podemos utilizar o Celery que possui suporte para mensageria assíncrona:

https://docs.celeryproject.org/en/stable/

Celery / Python Async





ASGI

ASGI, ou Asynchronous Server Gateway Interface, é a especificação que suporta **channels** e outros protocolos de comunicação.

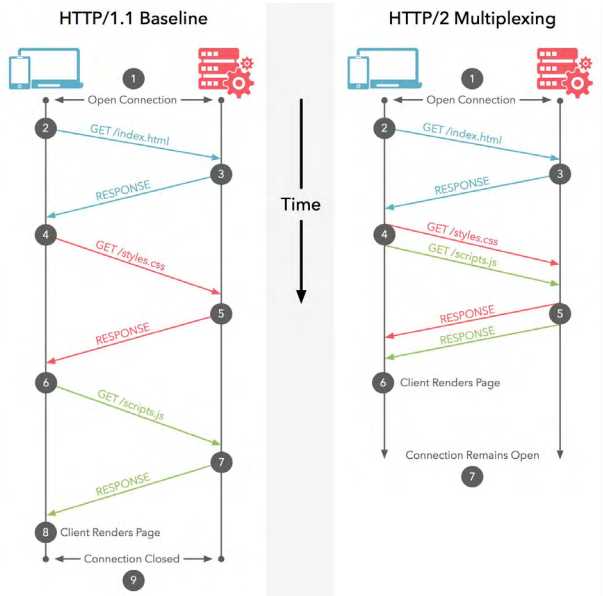
Podemos dizer que é o sucessor do WSGI.

https://asgi.readthedocs.io/en/latest/

HTTPX

HTTPX foi construído para atender requisições HTTP para Python, e provendo funcionalidades para chamadas assincrono e sincrono, e também possuí suporte para HTTP/1.1 and HTTP/2.

https://www.python-httpx.org/



Resumo

- Assincrono vs Sincrono
- Blocking calls
- Celery
- HTTPX
- Django Async Views

Exercício

escola britânica de artes criativa & tecnologia

Crie um novo projeto utilizando usando como base o projeto que criamos na aula de Django Async Views, crie uma nova view assíncrona com um contador de tempo (similar ao exemplo da aula anterior).

Crie um Pull Request e envie o link para a plataforma da EBAC