

Sobre as versões do FastAPI

FastAPI já está sendo usado em produção em diversas aplicações e sistemas, a cobertura de testes é mantida em 100%, mas seu desenvolvimento está avançando rapidamente.

Novos recursos são adicionados com frequência, bugs são corrigidos regularmente e o código está sempre melhorando.

Esse é o motivo das versões atuais estarem em `0.x.x`, significando que em cada versão pode haver mudanças significativas, tudo isso seguindo as [convenções de controle de versão semântica](#).

Já é possível criar aplicativos de produção com **FastAPI** (e provavelmente você já faz isso há algum tempo), apenas precisando ter certeza de usar uma versão que funcione corretamente com o resto do seu código.

Fixe a sua versão de `fastapi`

A primeira coisa que você deve fazer é "fixar" a versão do **FastAPI** que você está utilizando na mais atual, na qual você sabe que funciona corretamente para o seu aplicativo.

Por exemplo, supondo que você está usando a versão `0.45.0` em sua aplicação.

Caso você utilize o arquivo `requirements.txt`, você poderia especificar a versão com:

```
fastapi==0.45.0
```

Isso significa que você conseguiria utilizar a versão exata `0.45.0`.

Ou, você poderia fixá-la com:

```
fastapi>=0.45.0,<0.46.0
```

isso significa que você iria usar as versões `0.45.0` ou acima, mas inferiores à `0.46.0`, por exemplo, a versão `0.45.2` ainda seria aceita.

Se você usar qualquer outra ferramenta para gerenciar suas instalações, como Poetry, Pipenv ou outras, todas elas têm uma maneira que você pode usar para definir as versões específicas dos seus pacotes.

Versões disponíveis

Você pode ver as versões disponíveis (por exemplo, para verificar qual é a versão atual) em [Release Notes](#) (internal-link target=_blank).

Sobre versões

Seguindo as convenções de controle de versão semântica, qualquer versão abaixo de `1.0.0` pode adicionar mudanças significativas.

FastAPI também segue a convenção de que qualquer alteração de versão "PATCH" é para correção de bugs e alterações não significativas.

!!! tip "Dica" O "PATCH" é o último número, por exemplo, em `0.2.3`, a versão PATCH é `3`.

Logo, você deveria conseguir fixar a versão, como:

```
fastapi>=0.45.0,<0.46.0
```

Mudanças significativas e novos recursos são adicionados em versões "MINOR".

!!! tip "Dica" O "MINOR" é o número que está no meio, por exemplo, em `0.2.3`, a versão MINOR é `2`.

Atualizando as versões do FastAPI

Você deve adicionar testes para a sua aplicação.

Com **FastAPI** isso é muito fácil (graças a Starlette), verifique a documentação: [Testing](#){internal-link target=_blank}

Após a criação dos testes, você pode atualizar a sua versão do **FastAPI** para uma mais recente, execute os testes para se certificar de que todo o seu código está funcionando corretamente.

Se tudo estiver funcionando, ou após você realizar as alterações necessárias e todos os testes estiverem passando, então você pode fixar sua versão de `FastAPI` para essa mais nova.

Sobre Starlette

Não é recomendado fixar a versão de `starlette`.

Versões diferentes de **FastAPI** utilizarão uma versão específica e mais recente de Starlette.

Então, você pode deixar **FastAPI** escolher a versão compatível e correta de Starlette.

Sobre Pydantic

Pydantic inclui os testes para **FastAPI** em seus próprios testes, então as novas versões de Pydantic (acima da `1.0.0`) sempre serão compatíveis com FastAPI.

Você pode fixar qualquer versão de Pydantic que desejar, desde que seja acima da `1.0.0` e abaixo da `2.0.0`.

Por exemplo:

```
pydantic>=1.2.0,<2.0.0
```