

# Deta にデプロイ

このセクションでは、**FastAPI** アプリケーションを [Deta](#) の無料プランを利用して、簡単にデプロイする方法を学習します。📖

所要時間は約**10分**です。

!!! info "備考" [Deta](#) は **FastAPI** のスポンサーです。👏

## ベーシックな FastAPI アプリ

- アプリのためのディレクトリ (例えば `./fastapideta/`) を作成し、その中に入ってください。

### FastAPI のコード

- 以下の `main.py` ファイルを作成してください:

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
def read_root():
    return {"Hello": "World"}

@app.get("/items/{item_id}")
def read_item(item_id: int):
    return {"item_id": item_id}
```

### Requirements

では、同じディレクトリに以下の `requirements.txt` ファイルを作成してください:

```
fastapi
```

!!! tip "豆知識" アプリのローカルテストのために Uvicorn をインストールしたくなるかもしれませんが、Deta へのデプロイには不要です。

### ディレクトリ構造

以下の2つのファイルと1つの `./fastapideta/` ディレクトリがあるはずです:

```
.
├── main.py
└── requirements.txt
```

## Detaの無料アカウントの作成

それでは、[Detaの無料アカウント](#)を作成しましょう。必要なものはメールアドレスとパスワードだけです。

クレジットカードさえ必要ありません。

## CLIのインストール

アカウントを取得したら、Deta CLI をインストールしてください:

=== "Linux, macOS"

```
<div class="termy">

```console
$ curl -fsSL https://get.deta.dev/cli.sh | sh
```

</div>
```

=== "Windows PowerShell"

```
<div class="termy">

```console
$ iwr https://get.deta.dev/cli.ps1 -useb | iex
```

</div>
```

インストールしたら、インストールした CLI を有効にするために新たなターミナルを開いてください。

新たなターミナル上で、正しくインストールされたか確認します:

```
$ deta --help

Deta command line interface for managing deta micros.
Complete documentation available at https://docs.deta.sh

Usage:
  deta [flags]
  deta [command]

Available Commands:
  auth          Change auth settings for a deta micro
  ...
```

!!! tip "豆知識" CLI のインストールに問題が発生した場合は、[Deta 公式ドキュメント](#)を参照してください。

## CLIでログイン

CLI から Deta にログインしてみましょう:

```
$ deta login
```

```
Please, log in from the web page. Waiting..  
Logged in successfully.
```

自動的にウェブブラウザが開いて、認証処理が行われます。

## Deta でデプロイ

次に、アプリケーションを Deta CLI でデプロイしましょう:

```
$ deta new
```

```
Successfully created a new micro
```

```
// Notice the "endpoint" 🔍
```

```
{  
  "name": "fastapideta",  
  "runtime": "python3.7",  
  "endpoint": "https://qltncl.deta.dev",  
  "visor": "enabled",  
  "http_auth": "enabled"  
}
```

```
Adding dependencies...
```

```
---> 100%
```

```
Successfully installed fastapi-0.61.1 pydantic-1.7.2 starlette-0.13.6
```

次のようなJSONメッセージが表示されます:

```
{  
  "name": "fastapideta",  
  "runtime": "python3.7",  
  "endpoint": "https://qltncl.deta.dev",  
  "visor": "enabled",  
  "http_auth": "enabled"  
}
```

!!! tip "豆知識" あなたのデプロイでは異なる "endpoint" URL が表示されるでしょう。

## 確認

それでは、endpoint URL をブラウザで開いてみましょう。上記の例では `https://qltncl.deta.dev` ですが、あなたのURLは異なるはずです。

FastAPIアプリから返ってきたJSONレスポンスが表示されます:

```
{
  "Hello": "World"
}
```

そして `/docs` へ移動してください。上記の例では、`https://qltnci.deta.dev/docs` です。

次のようなドキュメントが表示されます:



## パブリックアクセスの有効化

デフォルトでは、Deta はクッキーを用いてアカウントの認証を行います。

しかし、準備が整えば、以下の様に公開できます:

```
$ deta auth disable

Successfully disabled http auth
```

ここで、URLを共有するとAPIにアクセスできるようになります。🚀

## HTTPS

おめでとうございます！あなたの FastAPI アプリが Deta へデプロイされました！🎉📦

また、DetaがHTTPSを正しく処理するため、その処理を行う必要がなく、クライアントは暗号化された安全な通信が利用できます。✅🔒

## Visor を確認

ドキュメントUI (`https://qltnci.deta.dev/docs` のようなURLにある) は *path operation* `/items/{item_id}` へリクエストを送ることができます。

ID `5` の例を示します。

まず、<https://web.deta.sh> へアクセスします。

左側に各アプリの `...[Micros]...` というセクションが表示されます。

また、「Details」や「Visor」タブが表示されています。「Visor」タブへ移動してください。

そこでアプリに送られた直近のリクエストが調べられます。

また、それらを編集してリプレイできます。



## さらに詳しく知る

様々な箇所で永続的にデータを保存したくなるでしょう。そのためには [Deta Base](#) を使用できます。惜しみない **無料利用枠** もあります。

詳しくは [Deta ドキュメント](#) を参照してください。