

# Deno KV で超速プロダクト開発

Deno の歴史がまた 1 ページ

# 自己紹介 (Twitter: @taroosg)

```
{
  "name": "Taro Ohsugi",
  "works": [
    {
      "work": "🎓 G's ACADEMY FUKUOKA 主任講師",
      "skills": ["JavaScript", "React", "PHP", "Laravel"]
    },
    {
      "work": "🎓 エンジニア",
      "skills": ["Laravel", "JavaScript", "画面設計", "DB設計"]
    }
  ],
  "like": ["💻", "📚", "✈️ 🚗 🏠 🚶", "🍹 🍷 🍺", "👤"]
}
```

# 概要

本記事では下記を述べる．

- Deno KV がすごい．
- Deno KV がすごい．
- Deno KV がすごい．

# 背景

個人開発で業務ツールとか作る。

- DB どうする？
- デプロイ先どうする？
- お財布どうする？

**Deno はすべてを解決する**

# Deno はすべてを解決する

- Fresh (Web フレームワーク)
- **Deno KV** (kv ストレージ)
- Deno Deploy (デプロイ)

# Deno KV とは

Deno で使用できる key-value ストレージ。ローカルで実行する場合は Deno KV は SQLite で動作し，アプリケーションを Deno Deploy にデプロイすると Deno KV データベースは自動的に FoundationDB によって動作する。

# 今回のお題

「X」みたいなアプリケーション「Twitter」を実装してデプロイしてみる.



# 例 (DB 接続)

```
const kv = await Deno.openKv();
```

# 例

フォームから送信されたデータを保存.

```
// データの受け取りと取り出し
const formData = await req.formData();
const name = formData.get("name")?.toString();
const tweet = formData.get("tweet")?.toString();
// 登録処理
const kv = await Deno.openKv();
await kv.set(["tweets", Date.now()], {
  tweet,
  name,
});
```

# 例（データの参照）

キー指定して 100 件取得.

```
const iter = await kv.list({ prefix: ["tweets"] }, { limit: 100 });
const tweets = [];
for await (const res of iter) {
  tweets.unshift(res.value);
}
```

# デモ

1. コードの紹介
2. ローカルでの動作確認
3. デプロイして動作確認

# まとめ

- とにかく簡単で現時点ではコストも考えなくて良い.
- 業務ツールや簡単なアプリケーションの開発 → デプロイ → 運用の流れが超速.
- ベータ版であるため API 仕様の変更や機能の追加削除などは注意.

**Enjoy!**