



bad-rerquest.ts

```
// Next.js API route support: https://nextjs.org/docs/api-routes/introduction
import type { NextApiResponse, NextApiRequest } from 'next'

type Data = {
  ok: boolean;
  message: string | string[];
}

export default function handler( req: NextApiRequest, res:
NextApiResponse<Data> ) {

  const { message = 'Bad request' } = req.query;

  res.status(400).json({
    ok: false,
    message,
  })

}
```

hello.ts

```
// Next.js API route support: https://nextjs.org/docs/api-routes/introduction
import type { NextApiResponse, NextApiRequest } from 'next'

type Data = {
  name: string
}

export default function handler(

  req: NextApiRequest,
  res: NextApiResponse<Data>
) {
  res.status(200).json({ name: 'John Doe' })
}
```

seed.ts



```
// Next.js API route support: https://nextjs.org/docs/api-routes/introduction
import type { NextApiResponse, NextApiRequest } from "next";
import { db, seedData } from "../../database";
import { Entry } from "../../models";

type Data = {
  message: string;
};

export default async function handler(
  req: NextApiRequest,
  res: NextApiResponse<Data>
) {
  if (process.env.NODE_ENV === 'production') {
    res.status(401).json({ message: 'No tiene acceso a éste servicio' })
  }

  await db.connect();

  await Entry.deleteMany();

  await Entry.insertMany(seedData.entries);

  await db.disconnect();

  res.status(200).json({ message: 'Proceso realizado correctamente' });
}
```