Creating a REST API with Deno

侍 Pablo Magaz

https://pablomagaz.com

github.com/pmagaz twitter.com/pablo_magaz

Talk Code: github.com/pmagaz/deno-oak-rest-api

Deno is secure runtime for TypeScript that uses V8 and is built in Rust.

侍 Deno main features

- Built in Rust.
- TypeScript as default language.
- Explicit network, file or environment access.
- Standard ECMAScript module system.
- Browser compatible.
- Decentralised package manager.

Oak

Oak is a middleware framework
built over Deno's http server

侍 Application

```
import { Application } from “https://deno.land/x/oak/mod.ts";

const app = new Application();

await app.listen({ port: 8000 });
```

侍 Middleware

```
import { Application } from “https://deno.land/x/oak/mod.ts";

const app = new Application();

app.use((context) => {

  console.log("Hello World!");

});

await app.listen({ port: 8000 });
```

侍 Context

```
import { Application } from “https://deno.land/x/oak/mod.ts";

const app = new Application();

app.use((context) => {

  context.response.body = "Hello World!";

});

await app.listen({ port: 8000 });
```

侍 Routing

```
import { Application, Router } from "https://deno.land/x/oak/mod.ts";

const app = new Application();

const router = new Router();

router.get("/api/helloworld", (context) => {

  context.response.body = "Hello World!";

});

await app.listen({ port: 8000 });
```

Users CRUD REST API

侍 Project structure

- src/server.ts
- src/routes/
- src/handlers/
- src/db/
- src/middleware/
- src/types/

侍 User Model

```
export type Uuid = string;
export interface User {
  uuid: Uuid;
  name: string;
  birthDate: Date;
}
```

```
import { Router } from “https://deno.land/x/oak/mod.ts”;

import { findUser, createUser, updateUser, deleteUser } from “../handlers/users.ts”;

export const router = new Router()
  .post("/api/users", createUser)
  .get("/api/users/:userId", findUser)
  .delete("/api/users/:userId", deleteUser)
  .patch("/api/users", updateUser);
```

```
import { Context, helpers } from "https://deno.land/x/oak/mod.ts";
import type { User } from “../types/user.ts";
import * as db from “../db/index.ts";

export const findUser = async (ctx: Context) => {
  const { userId } = helpers.getQuery(ctx, { mergeParams: true });
  try {
    let user: User = await db.findUserById(userId);
    ctx.response.body = user;
   } catch (err) {
    ctx.response.status = 404;
    ctx.response.body = { msg: err.message };
  }
};
```

```
import { Context } from "https://deno.land/x/oak/mod.ts";
import type { User } from “../types/user.ts";
import * as db from “../db/index.ts";


export const createUser = async (ctx: Context) => {
  const { name, birthDate } = await ctx.request.body().value;
  try {
     let createdUser : User = await db.createUser(name, birthDate);
     ctx.response.body = createdUser;
   } catch (err) {
    ctx.response.status = 500;
    ctx.response.body = { msg: err.message };
  }
};
```

```
import type { User, Uuid } from “../types/user.ts";
import { v4 } from “https://deno.land/std@0.77.0/uuid/mod.ts”;

//Mocked db query
export const createUser = async (name: string, birthDate: Date): Promise<User> =>
  new Promise((resolve, reject) => {
      resolve({
        uuid: v4.generate(),
        name,
        birthDate,
      });
  });
```

```
import { Context, helpers } from “https://deno.land/x/oak/mod.ts";

export const logger = async (ctx: Context, next: () => any) => {
  await next();
  const body = await ctx.request.body().value;
  let params = body ? `with params ${JSON.stringify(body)}` : “";
  console.log(`${ctx.request.method} request to ${ctx.request.url} ${params}`);
};

//OUTPUT -> POST request to http://localhost:8000/api/users/ with params
{“name":"Paul","birthDate":"1980-11-22T16:56:15.667Z"}
```

```
import { Application } from “https://deno.land/x/oak/mod.ts";
import { routes } from “./routes/index.ts”;
import { logger } from “./middleware/logger.ts”;

const app = new Application();

app.use(logger);
app.use(router.routes());

console.log(`Server up on port ${PORT}`);
await app.listen({ port: 8000 });
```

Thank you.

EL BLOG ISOMÓRFICO

Slides: pablomagaz.com/#talks

Code: github.com/pmagaz/deno-oak-rest-api

twitter.com/pablo_magaz