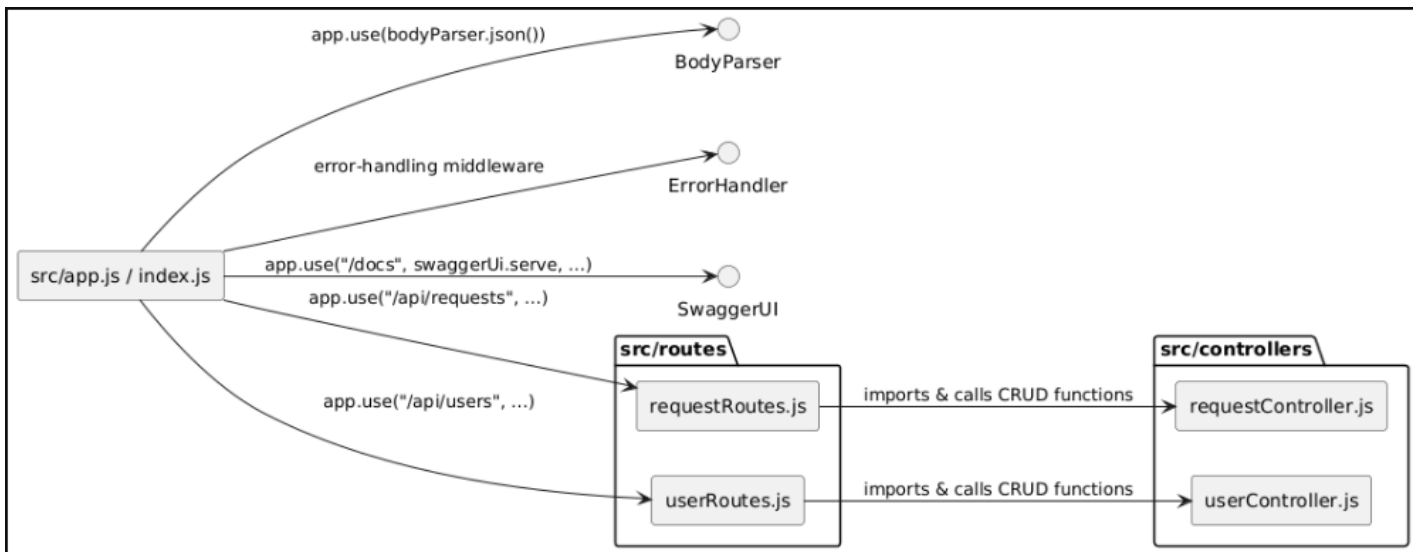


API, Route , Controller

?? ????? ?? ????? ?? ????? route , controller
????? ??? ???



- ????? ?????? ??????? Route ? Controller

- ?? ????? ?????? ?????? |src/routes/| ? |src/controllers/| ?? ??????

- ?? ??????? (????? |requests| ? |users|) ?? ????? Route ? ? ???? Controller ????
- ?????? ??????:

```

src/
├── controllers/
│   ├── requestController.js
│   └── userController.js
└── routes/
    ├── requestRoutes.js
    └── userRoutes.js
  
```

- ????? Route?? ?? ??????? ?? Express

- ?? |src/routes/requestRoutes.js| ?????? CRUD ?? ?????? ?????:

```

import { Router } from 'express';
  
```

```

import {
  createRequest,
  getAllRequests,
  getRequestById,
  updateRequest,
  deleteRequest
} from '../controllers/requestController.js';

const router = Router();

router.post('/', createRequest);
router.get('/', getAllRequests);
router.get('/:id', getRequestById);
router.put('/:id', updateRequest);
router.delete('/:id', deleteRequest);

export default router;

```

3. **????? Controller?? ?? ???? ?????**

- ?? |src/controllers/requestController.js| ????? CRUD ?? ?????? ?????:

```

import Request from '../models/request.js';
export const createRequest = async (req, res, next) => {
  try {
    const newReq = await Request.create({
      userId: req.body.userId,
      expertId: req.body.expertId,
      text: req.body.text
    });
    res.status(201).json(newReq);
  } catch (err) {
    next(err);
  }
};

export const getAllRequests = async (_req, res, next) => {
  try {
    const list = await Request.findAll();
    res.json(list);
  } catch (err) {
    next(err);
  }
};

```

```

    }
  };

  export const getRequestById = async (req, res, next) => {
    try {
      const item = await Request.findByPk(req.params.id);
      if (!item) return res.status(404).json({ message: 'Not found' });
      res.json(item);
    } catch (err) {
      next(err);
    }
  };

  export const updateRequest = async (req, res, next) => {
    try {
      const [updated] = await Request.update(req.body, {
        where: { requestId: req.params.id }
      });
      if (!updated) return res.status(404).json({ message: 'Not found' });
      res.json({ message: 'Updated' });
    } catch (err) {
      next(err);
    }
  };

  export const deleteRequest = async (req, res, next) => {
    try {
      const deleted = await Request.destroy({
        where: { requestId: req.params.id }
      });
      if (!deleted) return res.status(404).json({ message: 'Not found' });
      res.status(204).end();
    } catch (err) {
      next(err);
    }
  };

```

4. **??? Route?? ?? ????????**
 - ?? `src/app.js` ?? `index.js` ??? ?? ??:

```
import express from 'express';
import bodyParser from 'body-parser';
import requestRoutes from './routes/requestRoutes.js';
import userRoutes from './routes/userRoutes.js';

const app = express();
app.use(bodyParser.json());

app.use('/api/requests', requestRoutes);
app.use('/api/users', userRoutes);

// اطلاع دهنده
app.use((err, _req, res, _next) => {
  console.error(err);
  res.status(500).json({ error: err.message });
});

export default app;
```

Revision #2

Created 14 May 2025 07:03:41 by a.tavana

Updated 14 May 2025 07:21:31 by a.tavana