



Users DB

DeepDive Week 2 Day 2



Материалы и Ресурсы

Grand finale! Соберем вместе все воедино. В роли пользователей всего того, что было написано ранее мы сделаем:

- небольшую обертку над нашим CSV API, которая будет эмулировать БД на базе CSV (в нашем оригинальном CSV все же не хватает некоторых вещей, как например потокобезопасность)
- REST API для работы с записями в БД

И так, для начала модель:

User

```
package academy.kovalevskiy.javadeepdive.week2.day2;

public class User {
    public String mail;
    public String firstName;
    public String lastName;
    public String password;
}
```

UserDB

```
package academy.kovalevskiy.javadeepdive.week2.day2;

import academy.kovalevskiy.javadeepdive.week0.day2.CSV;
import academy.kovalevskiy.javadeepdive.week0.day3.InsertRequest;
import academy.kovalevskiy.javadeepdive.week0.day3.RequestException;
import academy.kovalevskiy.javadeepdive.week0.day3.SelectRequest;
import academy.kovalevskiy.javadeepdive.week0.day3.Selector;

import java.util.Arrays;
import java.util.Optional;
import java.util.stream.Collectors;

public class UserDB {
    public UserDB(CSV csv) {
        // TODO
    }
    public void addUser(User user) throws RequestException {
        // TODO
    }
    public String[] getUsersMails() throws RequestException {
        // TODO
    }
    public Optional<User> getUser(String mail) throws RequestException {
        // TODO
    }
}
```

Однако, не забывайте что эта абстракция должна быть потокобезопасной!

Ну и последние - это наш REST API

REST API

```
package academy.kovalevskiy.javadeepdive.week2.day2;

import academy.kovalevskiy.javadeepdive.week0.day2.CSV;
import academy.kovalevskiy.javadeepdive.week0.day3.RequestException;
import academy.kovalevskiy.javadeepdive.week2.day1.Controller;
import academy.kovalevskiy.javadeepdive.week2.day1.Get;
import academy.kovalevskiy.javadeepdive.week2.day1.Path;
import academy.kovalevskiy.javadeepdive.week2.day1.Post;

@Controller
public class Users {

    // TODO

    @Get
    @Path("/users")
    public String[] users() {
        // TODO
    }

    @Get
    @Path("/first")
    public User firstUser() {
        // TODO
    }

    @Post
    @Path("/users")
    public void addUser(User user) {
        // TODO
    }
}
```

Если ранее вы сделали возможность указывать более сложные пути, то само собой тут еще нужен путь, который возвращает одного студента по его почте.