

Présentation technique 22/07/2020


Tripping

Sofia Vayas Pedersen

Plan

- 1 - Mon architecture
- 2 - Spring Boot
- 3 - Angular
- 4 - Node Js
- 5 - Heroku / Git
- 6 - Demo
- 7 - Bilan

Mon architecture

 w:1060 h:400

mon appli (localhost)

Spring Boot

Spring Initializr

Les microservices sont caractérisés
par le fait qu'il est faiblement couplé.
(Indépendant, facile à maintenir)

Spring boot embarque serveur
tomcat (standard de JEE- port par
défaut : 8080)

Modèle Microservices

 w:1060 h:450

Open Feign

Mon MS Liste ne connaît pas mon MS User, donc je dois utiliser :

- Un DTO user.
- Le framework **feign netflix** permet de faire une communication dans un architecture distribué à travers des proxies

```
@FeignClient(name="microservice.user")
public interface MicroserviceUserProxy {
    @GetMapping(path="/tripping/user/findAllUsers")
    public List<UserBean> getAllUsers() ;
}
```

Comment je récupère le proxy dans mon liste controleur?

Controleur


- `@Autowired`
- `@CrossOrigin("*")`
- `@Controller` ou `@RestController` - met en place toute la plomberie rest
- `@RequestMapping (path = "/tripping/lists")`

Miracle de logger ❤️ `Logger log = LoggerFactory.getLogger(this.getClass());`

Repo

```
Optional<ListeEntity> findByCategoryAndUser(String category, String user);
```

Postman


 w:200 h:200

Miracle de Postman ❤️

Défauts w:90 h:70

- Je n'ai pas de tests unitaires ni d'intégrations
- J'aurais voulu utilisé :
 - Zull
 - Eureka
 - Config Serveur
 - Swagger

Angular

-  w:120 h:50
- Open source
- Typescript

Commandes utiles

- `ng new my-trip-test` - création d'une nouvelle appli angular4+
- `ng g component lists` - génération d'un nouveau composant
- `ng g service lists` - génération d'un nouveau service
- `ng serve -o` - build en mémoire + démarrage serveur de test
- `ng build --prod` - construction du dossier dist (pour déploiement)
- [Angular material](#) -ex : Card ❤️

Liaison entre vue et modèle

- `<input matInput type="text" [(ngModel)]="city">`

Liaison dans les 2 sens (lecture/écriture)

- `<mat-card-content>{{f.date | date:'fullDate'}}</mat-card-content>`

Liaison dans 1 sens (affichage) ([Angular - dates](#))

- `<button (click)="onFetchforecastByCityName()"> Search</button>`

Principales directives prédéfinies

*ngFor

```
<mat-card *ngFor="let f of forecast ">
```

*ngIf

```
<div *ngIf="i.inBag ; else elseBlock">  
  <mat-checkbox type="checkbox" [(ngModel)]= "i.inBag" checked> <mat-checkbox>  
</div>  
<ng-template #elseBlock>  
  <mat-checkbox id="checkBoxInBag" type="checkbox" [(ngModel)]= "i.inBag"></mat-checkbox>  
</ng-template>
```

Communication entre angular et NodeJs / Spring

```
export class WeatherService {  
    constructor(private http: HttpClient) { }  
  
    baseUrl = 'https://tripping-nodejs.herokuapp.com'  
  
    public fetchWeatherforecastByCityname(cityname: String) :Observable<SingleWeatherforecast[]> {  
        let url = `${this.baseUrl}`+`/weather-api/public/forecastbycity/`+`${cityname}`;  
        console.log("In service, name = "+cityname+ "url= "+ url);  
        return this.http.get<SingleWeatherforecast[]>(url);  
    }  
}
```

Node.js

ex: <https://tripping-nodejs.herokuapp.com/weather-api/public/forecastbycity/paris>

```
apiRouter.route('/weather-api/public/forecastbycity/:cityname')
.get(async function(req, res, next){
  var cityname = req.params.cityname;
  const url = "http://api.openweathermap.org/data/2.5/forecast?q="+cityname+"&appid="+WEATHER_API_KEY+"&units=metric";
  const response = await fetch(url);
  const json = await response.json();
  const count = json.cnt;
  const forecastPretty = [];
  for(i=0; i<count; i+=8){
    const weather = {
      city: json.city.name,
      date: json.list[i].dt_txt,
      temp: json.list[i].main.temp,
      feels_like: json.list[i].main.feels_like,
      icon: json.list[i].weather[0].icon
    }
    forecastPretty.push(weather);
  }
  res.send(forecastPretty);
})
```

Heroku ❤️

Heroku Dashboard

Tripping App : <https://tripping-angular.herokuapp.com/>

Git


[SofiaVP/tripping-nodejs-hub](#)

[SofiaVP/tripping-MS-user-hub](#)


[SofiaVP/tripping-MS-list-hub](#)

[SofiaVP/tripping-angular-hub](#)


Evolution front 1/4

 w:1060 h:400

Evolution front 2/4

 w:1060 h:400

Evolution front 3/4

 w:1060 h:400

Evolution front 4/4

 w:1060 h:400

Petite demo ? ...

Petite demo !

(elle sera plus au point vendredi)

Tripping (localhost)

Bilan

Fiertés

- Dark mode
- J'ai appris beaucoup de choses :)

Regrets

- Pas de Docker
- Pas de Tests
- Front à revoir
- Microservice todo à faire

Difficultés rencontrés au cours du projet

- Difficultés à se lancer (API)
- Modélisation difficile pcq'il y a plusieurs niveaux (code, architecture etc...)
- Images dans assets
- Perte de spring
(D:\Project_Tripping_LeBon\WS_Eclipse_Trip.metadata.plugins\org.eclipse.core.resources.history\95)
- Gestion du temps et du stress
- Méthode de travail 25 min - partie solitaire (surtout pour le "last minute rush") me correspond super bien

Le groupe est toujours aussi soudé ce qui est super cool ! ❤️

Merci pour votre attention !