

# TD5 : Initiation à Angular

Projet à rendre

## Contexte

Vous travaillez pour Air France, une compagnie aérienne qui fait voler des milliers d'avions dans le monde par an. Air France cherche à lister tous les vols qui décollent ou atterrissent dans les aéroports français. Cela dans le but d'optimiser le nombre d'avions à disponibiliser et de vérifier dans quels aéroports ils ont la plus grande concentration d'avions.

Le but de cette application sera dans un premier temps de lister tous les avions qui décollent d'un aéroport donné en paramètre entre une date de début et une date de fin. Ces 3 critères de recherche seront présents dans une section « Filtre ».

Nous utilisons un jeu de données en Open Data afin de récupérer le champ icao, un identifiant unique qui identifie le matricule d'un avion. Pour se faire, nous définissons un service pour venir faire une requête http sur ce jeu de données. Nous allons récupérer aussi l'aéroport de destination si celui-ci est connu pour l'afficher dans l'application.

Il sera possible de sélectionner un vol dans la liste, ainsi la liste des passagers de ce vol devra s'afficher dans la partie droite de l'écran. On pourra alors visualiser les passagers, la classe de leur vol (business, premium ou standard) et le nombre de bagages qu'ils emportent en soute.

Ensuite, il faudra faire la même chose mais pour l'atterrissage dans un aéroport et dans une période donnée.

Le développeur en charge de ce projet est tombé malade et est en arrêt maladie. Pour que le projet ne prenne de retard, on vous demande d'avancer sur les points indiqués dans ce document.

Le précédent développeur a quand même mis en place la structure du code et documenté son code pour qu'il soit moins compliqué pour vous de continuer.

## Consignes

**Le TD est à faire seul.** Aucun document n'est à rendre, seul le projet devra être livré avant le **14/06/2024 23h59**.

Vous devrez mettre à disposition le code du projet rendu sur un gestionnaire de versions au choix : Gitlab (ou Github). Et votre application devra être déployée et testable **via Gitlab Pages** (ou Github pages).

Pour renseigner vos informations de rendu, vous remplirez individuellement le **google form** suivant : [https://docs.google.com/forms/d/e/1FAIpQLSfqA-4hKLGNorsk7\\_siypASF1I6NpBF7Z\\_i51pV-6xhp7SVQ/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSfqA-4hKLGNorsk7_siypASF1I6NpBF7Z_i51pV-6xhp7SVQ/viewform?usp=sf_link)

Tout problème avec le code, l'application déployée ou la réponse au formulaire entraînera un 0 sans mail de relance de ma part.


Le mode d'emploi fourni dans ce TD à la section suivante ne concerne que Gitlab Pages.

Le barème est indiqué sur chaque partie. Le rendu du TD comptera pour 100% de la note du cours.

## Mode d'emploi des GitLab Pages

Gitlab met en place un service pour déployer des sites en ligne sur une url statique : les **Gitlab Pages**. Cela vous permet d'avoir une intégration et un déploiement continu sans trop d'efforts de votre part, et ainsi de pouvoir tester votre application en dehors de votre environnement de développement.

Le fichier qui porte les consignes d'intégration et de déploiement est ce fichier :

 `.gitlab-ci.yml`

Le fichier va exécuter plusieurs commandes qui vous sont maintenant familières : npm install, ng build. Le node\_modules généré est mis en cache pour être utilisé lors du lancement de l'application.

```
# The Docker image that will be used to build your app
image: node:lts

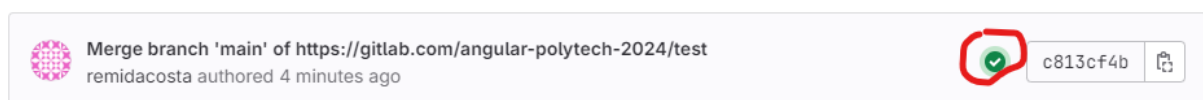
pages:
  cache:
    paths:
      - node_modules/

  stage: deploy
  script:
    - npm install -g @angular/cli@latest
    - npm install
    - ng build
    - mkdir -p public
    - mv dist/angular-polytech-td5/browser/* public/
  artifacts:
    paths:
      - public
```

Lorsque vous créez un repo sur gitlab avec ce fichier, faites attention à la ligne «*mv dist/angular-polytech-td5/browser/\* public/*». Il faut renseigner ce qui est gras l'équivalent de ce que vous aurez dans votre fichier angular.json

```
prelink: {
  "architect": {
    "build": {
      "builder": "@angular-devkit/build-angular:application",
      "options": {
        "outputPath": "dist/angular-polytech-td5",
        "index": "src/index.html",
        "browser": "src/main.ts"
      }
    }
  }
}
```

Au niveau de votre interface Gitlab, lorsque vous aurez fait un push, vous pouvez cliquer sur le petit icône comme ci-après pour visualiser votre build :



**Project visibility**

Manage who can see the project in the public access directory. [Learn more.](#)

Private

Private  
Internal  
**Public**

☒ Require authentication to view media files

**Pages**

With GitLab Pages you can host your static websites on GitLab. GitLab Pages uses a distributed cache that is invalidated, which usually takes less than a minute. [Learn more.](#)

☒ Everyone

**Monitor**

# Pages

With GitLab Pages you can host your static website directly from your GitLab repository. [Learn more.](#)

- ☒ **Force HTTPS (requires valid certificates)**  
When enabled, all attempts to visit your website through HTTP are automatically redirected to HTTPS u  
[Learn more.](#)
- ☒ **Use unique domain**  
When enabled, a unique domain is generated to access pages.

[Save changes](#)

---

**Access pages** 📄 1

[https://angular-polytech-td5-correction-angular-\[redacted\].gitlab.io](https://angular-polytech-td5-correction-angular-[redacted].gitlab.io) 🔗

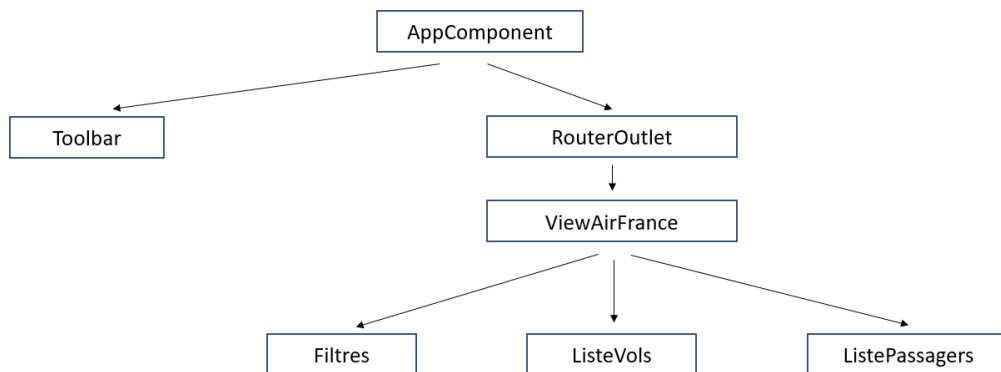
## Mise en place du projet

Allez sur gitlab et faites un `git clone` du projet du TD5 : <https://gitlab.com/angular-polytech-2024/angular-polytech-td5>

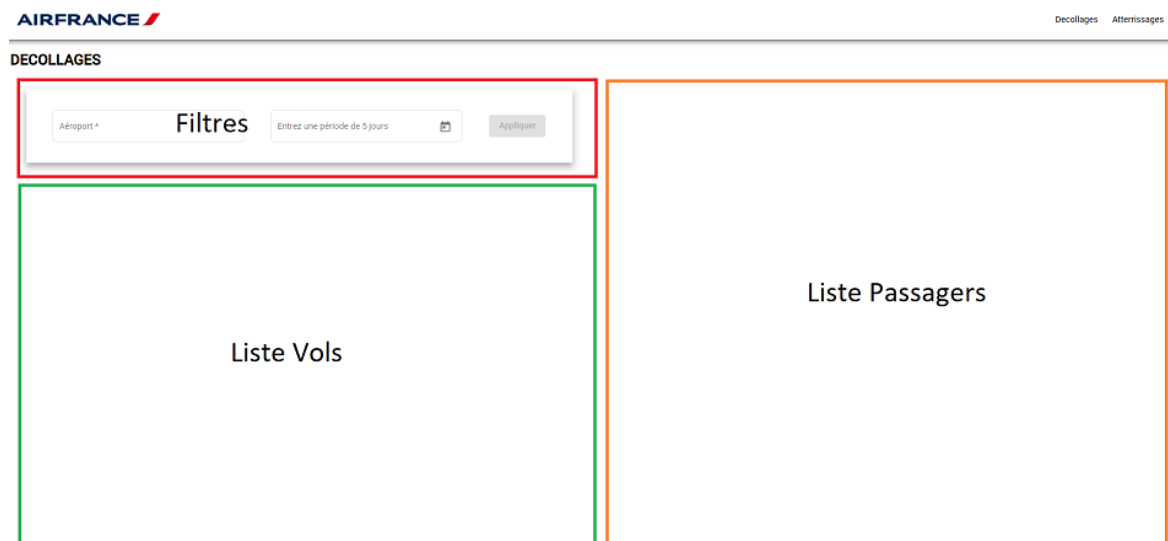
Lancez la commande `npm install` pour télécharger les dépendances nécessaires au projet.

Lancez ensuite le script `npm start` et connectez-vous sur **localhost:4200**.

Les composants disponibles sont hiérarchisés de la manière suivante :



Voici le rendu de l'application :

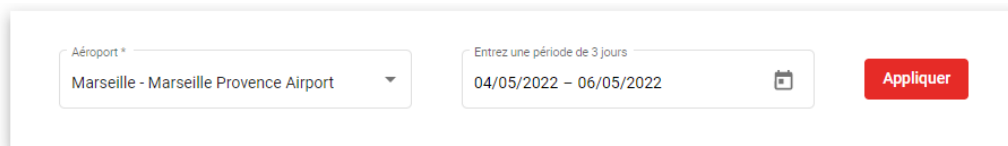


## Gérez les décollages (6 points)

Par défaut nous arrivons sur la page des décollages. Il sera possible plus tard de choisir décollage ou atterrissage depuis le menu en haut de page.

### Les filtres

La section des filtres n'est pas complètement fonctionnelle. On peut choisir un aéroport parmi une liste fixe des principaux aéroports français. On peut également saisir une période de 3 jours dans la zone de saisie.



The screenshot shows a filter bar with three main components: a dropdown menu for 'Aéroport \*' with 'Marseille - Marseille Provence Airport' selected, a date range input labeled 'Entrez une période de 3 jours' showing '04/05/2022 - 06/05/2022', and a red 'Appliquer' button.

Mais ces informations ne sont pas persistées dans le fichier .ts et ne sont pas remontées au composant ViewAirFrance.

Vous devez faire le nécessaire pour récupérer les valeurs des composants Angular Material, puis les transmettre au composant parent.

**Note :** On souhaite que le bouton « Appliquer » soit activable **uniquement** lorsque les 2 champs de saisie ont bien été renseignés.

Bien qu'il n'y ait qu'un seul champ de saisie pour la période, ce sont bien 2 variables qui doivent être récupérées : la date de début et la date de fin de la période.

```
<mat-date-range-input [rangePicker]="picker">
  <input matStartDate placeholder="Date de début">
  <input matEndDate placeholder="Date de fin">
</mat-date-range-input>
```

### Interrogez l'API open data

Nous allons pouvoir récupérer les filtres après chaque clic sur le bouton « Appliquer ». Encore faut-il récupérer le changement de filtres dans le composant parent : **ViewAirFrance**.

Nous devons ensuite appeler le service **VolService**, nous utiliserons les 3 critères récupérés pour requêter l'API d'informations des vols.

Dans le service VolService vous retrouverez la fonction getVolsDepart, utilisez-la en y renseignant les bons attributs.

**Attention :** Les attributs début et fin, sont au format Date, cependant l'API open source attend un temps exprimé en secondes.

## Affichez la liste des vols

Nous avons maintenant récupéré la liste des vols lorsque nous utilisons des filtres. Nous allons passer cette liste à afficher au composant **ListeVols**.

Une fois que c'est fait, nous allons appeler le composant **Vol** autant de fois qu'il y a de vols dans notre liste. Le composant Vol sera un *Dumb Component*, il se contentera de l'affichage des données qu'on lui donne.

Complétez le composant Vol pour qu'il reçoive un objet Vol du composant parent et qu'il l'affiche. Les attributs qui doivent être affichés sont : le numéro icao, le matricule de l'avion, sa compagnie aérienne, l'aéroport de départ (qui est forcément connu), et l'aéroport de destination (qui n'est pas forcément connu), si celui-ci est null, on affichera « Inconnu ».

Air France est un grand groupe il possède des filiales comme : Air France Hop et Transavia France. Il faut donc pouvoir afficher ses compagnies aussi. Les différentes valeurs de l'attribut compagnie sont : «Air France », « Air France Hop » et « Transavia France », vous pouvez retrouver ces libellés dans le fichier **compagnie.constants.ts** qui est utilisé pour formater les données.

Vous avez accès à des logos dans le dossier assets pour afficher les bons logos en fonction des différentes compagnies.

Le résultat attendu peut ressembler à cela :

Aéroport *		Entrez une période de 3 jours		Appliquer	
Marseille - Marseille Provence Airport		04/05/2022 - 06/05/2022			
✈	3950d0	AFR45KD	LFML → LFPO	AIRFRANCE	
✈	39d305	TVF68HM	LFML → Inconnu	transavia	
✈	3946e5	AFR55VY	LFML → LFPG	AIRFRANCE	

J'utilise l'icône `flight_takeoff` pour représenter un avion en train de décoller.

```
<div class="icon">
  <mat-icon>flight_takeoff</mat-icon>
</div>
```

Enfin j'utilise l'icône `arrow_right_alt` pour représenter la flèche entre les 2 aéroports

```
<div class="fleche">
  <mat-icon>arrow_right_alt</mat-icon>
</div>
```

Vous pouvez retrouver pleins d'icônes compatibles avec Angular Material ici : <https://www.angularjswiki.com/fr/angular/angular-material-icons-list-mat-icon-list/>

## Affichez la liste des passagers pour un vol (5 points)

Nous arrivons maintenant à afficher les vols qui décollent d'un aéroport. Mais on souhaite pouvoir afficher les passagers relatifs à un vol.

Nous allons tout d'abord ajouter la possibilité de sélectionner un vol. Cette nouvelle méthode va alors émettre le vol sélectionné au composant parent.

Dans le composant parent, faites le nécessaire pour récupérer le vol sélectionné.

Une fois le vol récupéré nous allons devoir passer par une étape importante avant d'afficher ses passagers. En effet le vol sélectionné ne contient aucun passager... Puisque l'API que nous avons utilisée pour récupérer les informations de vol ne possède pas ces informations.

Nous allons alors appeler un autre service dans notre composant ViewAirFrance : **PassagerService**.

Ce service n'existe pas. Nous allons créer la classe de service puis implémenter la méthode **getPassagers** selon les spécifications suivantes :

- La doc se trouve ici : <https://randomuser.me/>
- L'url de base est la suivante : <https://randomuser.me/api?>
  - o On souhaite récupérer 20 passagers pour chaque appel
  - o On souhaite récupérer les attributs « name » et « picture » pour chaque passager
  - o L'attribut seed, doit être renseigné et doit correspondre au numéro icao du vol concerné.
- Le tableau de résultats renvoyés (**response.results**) doit être convertis d'un tableau d'objets de type **IPassagerDto** en objets de type **Passager**. La classe Passage existe et possède un constructeur pour convertir automatiquement un type IPassagerDto.

L'appel à l'API de passagers nous retourne maintenant une liste de passagers que nous allons pouvoir afficher dans le bon composant.

De manière analogue à l'affichage des vols, nous allons afficher tous les attributs d'un passager : sa photo, son nom, la classe de son vol, ainsi que le nombre de bagage qu'il a en soute.

Un passager peut être en classe Standard, Business ou Premium, ces valeurs se trouvent dans l'énumération ClasseVol.

```
export enum ClasseVol {  
    STANDARD,  
    BUSINESS,  
    PREMIUM  
}
```

Vous devriez vous retrouver avec un affichage similaire :

	Rasmus Møller	STANDARD	 x1
	Dominic Zahl	PREMIUM	 x0
	Elsa Brun	BUSINESS	 x2
	Craig Hill	STANDARD	 x1

De manière plus générale votre écran devrait ressembler à ça :

Aéroport \*

Marseille - Marseille Provence Airport

Entrez une période de 3 jours

04/05/2022 - 06/05/2022

Appliquer

3950d0

AFR45KD

LFML → LFPO

AIRFRANCE

39d305

TVF68HM

LFML → Inconnu

transavia

3946e5

AFR55VY

LFML → LFPG

AIRFRANCE

3950c9

AFR94NV

LFML → LFPO

AIRFRANCE

398572

AFR27BD

LFML → EHAM

AIRFRANCE

3950cc

AFR37DC

LFML → LFPO

AIRFRANCE

392aeb

AFR78XM

LFML → LFPG

AIRFRANCE

393320

AFR25CA

LFML → LFPG

AIRFRANCE

3944f7

AFR99XW

LFML → LFPO

AIRFRANCE

392aed

AFR41JL

LFML → LFPG

AIRFRANCE

398572

AFR43HC

LFML → EHAM

AIRFRANCE

3950cd


AFR12NZ

LFML → LFPO

AIRFRANCE


Rasmus Møller

STANDARD

 x1


Dominic Zahl

PREMIUM

 x0


Elsa Brun

BUSINESS

 x2


Craig Hill

STANDARD

 x1


Malte Kröger

STANDARD

 x1


Amelia Walker

BUSINESS

 x2


Walterus Glastra

BUSINESS

 x0

Kaïs Gautier

STANDARD

 x0

Félicitations, vous venez de gérer les décollages ! Maintenant vous allez faire tout pareil mais pour les atterrissages.



## Gérez les atterrissages (4 points)

Heureusement le plus gros du travail est fait, nous allons modifier nos composants intelligemment pour pouvoir les réutiliser.

### Réutiliser les composants existants

Faites le nécessaire pour que l'onglet « /atterrissages » soit accessible via l'url et via le bouton dans la toolbar en haut de l'écran.

Cette nouvelle route devra conduire au composant **ViewAirFrance**.

Nous allons passer un paramètre dans la route qui mène au composant ViewAirFrance. Ce paramètre permettra d'indiquer au composant s'il doit afficher les décollages ou les atterrissages.

Dans le composant ViewAirFrance, vous allez récupérer ce paramètre, grâce au service **ActivatedRoute**.

Dans le composant ViewAirFrance, définissez une variable de ce type

```
constructor(  
    private _volService: VolService,  
    private _passagerService: PassagerService,  
    private _activatedRoute: ActivatedRoute) { }
```

Dans le fichier html du composant ViewAirFrance, conditionner l'affichage du titre en fonction du type pour afficher DECOLLAGES ou ATTERISSAGES.

### Interrogez l'API open data

Cet attribut **type** qui se trouve dans le composant **ViewAirFrance**, va nous permettre de conditionner la récupération des vols. Souvenez-vous, l'API de récupération des vols, ne récupère que les vols qui décollent d'un aéroport.

Créez une fonction getVolsArrivee dans le service et adaptez le code de la fonction getVolsDepart, vous ne changerez que l'url en remplaçant **departure** par **arrival**.

Voici la documentation de l'API si besoin : <https://openskynetwork.github.io/opensky-api/rest.html#arrivals-by-airport>









Modifiez la fonction qui réagit aux changements des filtres dans le composant ViewAirFrance, pour qu'en fonction du type de recherche, elle appelle le bon service.

Si vous avez correctement implémenté le service, vous devriez avoir des vols qui s'affichent lorsque vous êtes dans l'onglet des atterrissages.

Aéroport \*  
Marseille - Marseille Provence Airport

Entrez une période de 3 jours  
04/05/2022 – 06/05/2022

Appliquer

	3950cc	AFR53SM	LFPO → LFML	
	3950cd	AFR64DL	LFPO → LFML	
	3950d0	AFR25RC	LFPO → LFML	
	39d305	TVF4281	OLBA → LFML	

On constate d'ailleurs que l'aéroport de destination est bien celui sélectionné : ici Marseille (LFML).

**A noter :** Pour pousser plus loin la personnalisation nous pourrions modifier l'icône d'avion qui décolle, en avion qui atterrit, en utilisant l'icône `flight_land` au lieu de `flight_takeoff`.

## Modifiez les passagers (5 points)

### Les différentes classes de vol

Les passagers appartiennent tous à une classe. Nous allons mettre en évidence ces classes en les affichant d'une couleur différente.

**Nous n'allons pas conditionner le css mais créer une directive** qui modifiera la couleur du texte en fonction de la valeur de l'attribut classeVol.

Nous allons ajouter un Input qui permettra de passer la valeur de la classe en paramètre.

Voici le rendu attendu :



Rasmus Møller

BUSINESS

 x0



Dominic Zahl

BUSINESS

 x0



Elsa Brun

STANDARD

 x0



Craig Hill

PREMIUM

 x0



Malte Kröger

PREMIUM

 x2

## Le poids des bagages

Le poids des bagages en soute est un sujet délicat pour les compagnies aériennes. Si l'avion est trop lourd il consommera beaucoup plus et cela va donc entraîner des coûts pour les compagnies. De plus, si tout le monde emporte 10 bagages en soute, l'avion n'aura jamais assez de place pour tout emporter, sans parler de la logistique qui serait terrible.

Air France a donc mis en place une règle de gestion en fonction des classes :

Classe	Nombre maximum en soute
Standard	1
Business	2
Premium	3

Nous allons mettre en place **une directive** qui mette le fond de la zone d'affichage du nombre de bagages en soute en rouge.



Cette directive prendra en paramètre la classe, ainsi que le nombre de bagages, et affichera en rouge le fond de la zone si la règle de gestion décrite ci-dessus n'est pas respectée.

## L'affichage des photos

A cause des problématiques RGPD, l'entreprise préfère qu'une option soit mise en place pour ne pas afficher les photos des passagers.

L'affichage des photos doit rester possible pour contrôler à posteriori la bonne identité des passagers. Mais pour une autre utilisation, les photos n'ont pas d'intérêt.

Vous allez mettre en place un slide-toggle d'Angular Material, pour conditionner l'affichage des photos dans le composant ListePassagers. **Par défaut le slide-toggle doit être désactivé et les photos ne doivent pas être affichées.**

Lien de la doc : <https://material.angular.io/components/slide-toggle/overview>

## L'affichage de l'adresse mail

L'API de récupération des passagers ne récupère pas l'attribut d'adresse mail. Pourtant l'API met à disposition cet attribut. Dans un futur proche, il sera sans doute possible d'envoyer des mails directement depuis l'application aux passagers.

Nous allons donc ajouter cet attribut.

Tout d'abord ajoutons-le dans la méthode getPassagers du service.

Suivez la documentation de la section **Including/Excluding fields** : <https://randomuser.me/documentation>

Pensez à ajouter ensuite l'attribut dans le model.

Enfin, nous allons afficher cet attribut sous forme d'infobulle lorsque l'utilisateur survole le nom du passager.

Pour ajouter une infobulle, vous allez devoir ajouter le module MatTooltipModule dans votre composant, comme spécifié dans la doc : <https://material.angular.io/components/tooltip/api>. Si nécessaire relancez votre application pour que le module soit pris en compte.

Vous trouverez dans la documentation des exemples simples pour afficher une infobulle.

Voici le rendu final souhaité :

AIRFRANCE

ATTERRISSAGES

Aéroport\*

Avord - Avord (BA 702) Air Base

Dates\*

07/05/2024 - 09/05/2024

Appliquer

Entrez une période de 3 jours

✈

zyp0k7

HOPg2a

LFL → LFOA

AIRFRANCE HOP

✈

exs379

TVFxx4

LFP → LFOA

transavia

✈

e57n7u

TVFpe9

LFL → LFOA

transavia

✈

thgpiu

TVF7w9

LFSB → LFOA

transavia

✈

ee7v9l

HOPb45

LFPO → LFOA

AIRFRANCE HOP

✈

mmwirs

HOPdcf

LFPO → LFOA

AIRFRANCE HOP

✈

z0mvuo

AFRevf

LFBD → LFOA

AIRFRANCE

✈

t64v5a

AFRm4u

LFP → LFOA

AIRFRANCE

✈

peuyaa

AFR0yp

LFBD → LFOA

AIRFRANCE

✈

93ajv4

HOPmwc

LFMN → LFOA

AIRFRANCE HOP

✈

w4ip1v

AFRi6x

LFBD → LFOA

AIRFRANCE

Afficher les photos

Brad Oliver

PREMIUM

x1

Emma Hughes

STANDARD

x2

Willie Adams

STANDARD

x2

Emmi Kauppi

PREMIUM

x2

Nastasija Zeljković

PREMIUM

x2

Yones Santos

BUSINESS

x1

Gael Treviño

BUSINESS

x1

نارین نجاشی

BUSINESS

x1

Natalie Thompson

BUSINESS

x2

Benjamin Snyder

BUSINESS

x0

شهابی ناند

BUSINESS

x0

Shabari Nand

STANDARD

x2

Ricardo Diez

STANDARD

x1

Silje Sørensen

BUSINESS

x1

Anusree Mugeraya

BUSINESS

x2

AIRFRANCE

DecollagesAtterrissages

ATTERRISSAGES

Aéroport\*

Avord - Avord (BA 702) Air Base

Dates\*

07/05/2024 - 09/05/2024

Appliquer

Entrez une période de 3 jours

✈

zyp0k7

HOPg2a

LFLL → LFOA

AIRFRANCE

✈

exs379

TVFxx4

LFPG → LFOA

transavia

✈

e57n7u

TVFpe9

LFLL → LFOA

transavia

✈

thgpiu

TVF7w9

LFSB → LFOA

transavia

✈

ee7v9l

HOPb45

LFPO → LFOA

AIRFRANCE

✈

mmwirs

HOPdcf

LFPO → LFOA

AIRFRANCE

✈

z0mvuo

AFRevf

LFBD → LFOA

AIRFRANCE

✈

t64v5a

AFRm4u

LFPG → LFOA

AIRFRANCE

✈

peuyaa

AFR0yp

LFBD → LFOA

AIRFRANCE

✈

93ajv4

HOPmwc

LFMN → LFOA

AIRFRANCE

✈

w4ip1v

AFRi6x

LFBD → LFOA

AIRFRANCE

Afficher les photos

Brad Oliver

PREMIUM

x1

Emma Hughes

STANDARD

x2

Willie Adams

STANDARD

x2

Emmi Kauppi

PREMIUM

x2

Nastasija Zeljković

PREMIUM

x2

Yones Santos

BUSINESS

x1

Gael Treviño

BUSINESS

x1

AIRFRANCE

Avord - Avord (BA 702) Air Base

14/05/2024 - 16/05/2024

Appliquer

Entrez une période de 3 jours

n3jpkh

AFR251

LFOA

→

LFB0

AIRFRANCE

olwvj6

HOPzph

LFOA

→

LFPO

AIRFRANCE

gc9i75

AFRi75

LFOA

→

LFML

AIRFRANCE

5ap92j

HOP38q

LFOA

→

LFB0

AIRFRANCE

wvuyma

TVFfqs

LFOA

→

LFB0

transavia

Qqhtoh

AFRkn8

LFOA

→

LFB0

AIRFRANCE

833aug

TVF7vq

LFOA

→

LFB0

transavia

g4qv94

HOPsdv

LFOA

→

LFB0

AIRFRANCE

Afficher les photos

Jesus Montero

PREMIUM

x1

نينا زارحي

STANDARD

x2

Karla Araújo

PREMIUM

x3

Sessa Valencia

STANDARD

x2

Velibor Pejić

PREMIUM

x2

Kalenik Turchina

PREMIUM

x3

Antsje Qualm

STANDARD

x2