Où est mon train ?

Application Web – réactive – de suivi des trains.

L’application à deux modes principaux :

* La liste des trains qui passent par une gare.
* La liste des gardes desservies par un train.

A partir d’une gare, on peut lister les trains qui y passent, soit par une vue réduite n’indiquant que l’horaire, le nom et la destination du train, soit de manière détaillée indiquant l’horaire et les gares d’arrêt de chaque train. On peut avoir une première liste réduite et du moment qu’on affiche une vue détaillée, on peut aller au prochain train, toujours par rapport à la gare de référence.

La vue détaillée est la vue train. On suit un train en particulier en affichant les horaires de passage dans chaque gare.

Par-dessus le tout, on a des informations en temps réel. Si le train consulté est actuellement en cours de circulation, les horaires en temps réel sont affichés. L’affichage des horaires en temps réel est différent de celle des horaires théoriques.

Par ailleurs, le moment présent est également affiché si on visualise ce moment.

Cas d’utilisation :

1. Où est mon train ? A quelle heure passera-t-il dans la gare dans laquelle je veux le prendre ? A quelle heure arrivera-t-il à destination ? Par quelles gares passera-t-il ?

Premier objectif :

1. Visualiser les trains d’une gare vers une autre gare
2. Visualiser les détails du train sélectionné : quels sont ses arrêts, à quelle heure, où se trouve-t-il actuellement ?

--

Propriétés et états de React.

Quelles sont les données manipulées par l’application ?

* La liste des gares.
* La gare de départ.
* La gare d’arrivée.
* La liste des voyages allant de la gare de départ vers la gare d’arrivée.
* Les arrêts et les horaires des voyages.
* Les jours de services des voyages.
* La vue active : liste des voyages ou un voyage en particulier.

Parmi les données manipulées, lesquelles sont des propriétés statiques, et lesquelles sont des états de l’application ?

* La liste des gares : propriété ;
* La gare de départ : un état ;
* La gare d’arrivée : un état ;
* La liste des voyages allant de la gare de départ à la gare d’arrivée : une propriété déduite de la liste des gares et des gares de départ et d’arrivée ;
* Les arrêts et les horaires de voyages : propriété ;
* Les jours de service des voyages : propriété ;
* La vue active : un état.

Nous avons donc 3 données qui représentent l’état de l’application.

Quels sont les composants de l’application ?

* Le container main ;
* Le container voyages ;
* Le sélecteur de la gare de départ ;
* Le sélecteur de la gare d’arrivée ;
* La liste des voyages.

Le composant parent qui doit gérer l’état est clairement le composant main. Il n’y a pas photo ! S’il n’y avait qu’une seule vue, ce pourrait être le composant voyages qui gère cet état, mais il y aura aussi le composant détail d’un voyage qui recevra également les données d’état.

--

La suite. Le filtrage des gares en fonction des gares de départ et d’arrivée saisie. Deux gares. La première gare saisie filtre l’autre. Si c’est la gare de départ, on filtre les gares qui sont des gares de destinations possibles. Si c’est la gare d’arrivée, on filtre par gare de départ possible. Ce qui revient exactement au même puisque les trains circulent dans les deux sens, les gares d’arrivée possibles par rapport à une gare de départ sont les gares de départ possibles par rapport à une gare d’arrivée.

Ok. Ce qui représente un nouvel état ! Non, car c’est une liste déductible des autres états, i.e. de l’état des gares de départ et des gares d’arrivée. Du moment où une gare de départ est sélectionnée, ce sont les voyages possibles qui sont filtrées. Idem quand la gare d’arrivée est sélectionnée.

Sélection des destinations possibles en fonction de la gare de départ. On part de l’arrêt de départ. On sélectionne les voyages. Ce sont tous les voyages qui passent par la gare de départ. On sélectionne tous les arrêts de tous ces voyages à l’exception de la gare de départ. Ce sont toutes les destinations possibles à partir de la gare de départ. Comment sélectionner ces gares de manière unique ? Le résultat doit être une liste. Les arrêts doivent être uniques. On peut générer un objet avec les gares toutes les gares et transformer cet objet en liste. Oui, mais baser sur quelle entrée ? Sur l’identifiant de gare ?