

Examen National de Fin d'année
Session de Secours Report 2023

Examen de Fin de Formation (Epreuve de Synthèse)

Eléments de correction

Secteur :	Digital et Intelligence Artificiel	Niveau :	Technicien Spécialisé		
Filière :	Développement Digital Option Web Full Stack				
Variante	V1	Durée :	4h00	Barème	/100

Consignes et Conseils aux candidats :

- Toutes les réponses devront être justifiées avec le détail des calculs qui doit être indiqué sur la copie ;
- Apporter un soin particulier à la présentation de votre copie ;

Document(s) et Matériel(s) autorisés :

- Les documents ne sont pas autorisés ;
- Calculatrice simple (non programmable) autorisée.

Détail du Barème :

Question	Barème
Dossier 1	8pts
Q1	2
Q2	3
Q3	1.5
Q4	1.5
Dossier 2	14pts
Q1	2.5
Q2	2.5
Q3	5
Q4	4
Dossier 3	12pts
Q1	2.5
Q2	2.5
Q3	2
Q4	2.5
Q5	2.5
Dossier 4	6pts
Q1	6

Question	Barème
Dossier 5	12pts
Q1	3
Q2	3
Q3	3
Q4	3
Dossier 6	23
Q1	2
Q2	2
Q3	3
Q4a	2
Q4b	2
Q4c	2
Q4d	2
Q5	4
Q6	4

Question	Barème
Dossier 7	25pts
Q1	4
Q2a	3
Q2b	3
Q3a	5
Q3b	5
Q4	5

Total	100
--------------	------------

Note :

- a. La correction suivante est à titre indicatif.
- b. Le stagiaire pourrait donner des réponses avec d'autres techniques. (Voir Cours)

Partie théorique:(40 PTs)

Dossier 1 : Créer une application Cloud Native (8pts)

1. Comparez les 2 modèles **du Cloud Computing : SaaS et PaaS** de point de vue des utilisateurs du service par rapport au fournisseur. Vous pouvez expliquer avec un schéma (2pts)
 - SaaS : Les utilisateurs utilisent simplement les logiciels des applications ; ils ne sont pas responsables de la gestion ou de la maintenance de ces logiciels. Tandis que le fournisseur de cloud est responsable du provisionnement, de la gestion et de la maintenance des logiciels des applications.
 - PaaS : L'utilisateur est responsable du développement de ses propres applications. Il n'est pas responsable de la gestion de l'infrastructure ou du serveur. Tandis que Le fournisseur de cloud est responsable de la gestion des systèmes d'exploitation et de la configuration du réseau et des services. Ils fournissent une plateforme managée complète sur laquelle exécuter l'application.
2. Une application Cloud native se base sur des piliers, parmi lesquels: **Microservices** et **Livraison en continu (CD/CI)**. Donner une définition de ces deux piliers (3 pts)
 - **Microservices** : désignent à la fois une architecture et une approche de développement logiciel qui consiste à décomposer les applications, côté Back-end, en éléments les plus simples, indépendants les uns des autres.
 - **Livraison en continu (Continuous- Delivery)** : Lors d'un échange constant avec les exploitants Back-End, l'équipe de développeurs ajoute à un microservice certaines fonctionnalités **livrées automatiquement par des processus de Continuous- Delivery à l'aide de pipelines** de développement et d'infrastructure (Pipeline : Opérations automatisées Pull=>Build=>Test=>Deploy).
3. Qu'est-ce qu'une image **Docker** ? (1.5 pts)

Une image Docker est une installation pré-configurées de serveurs de données, serveurs web, serveurs NoSQL, moteurs de recherche. Cette installation est empaquetée de manière à pouvoir être placée très facilement dans un conteneur.
4. Donner la commande pour instancier une image **Docker** dans un conteneur (1.5 pts)
`docker run -name 'nom-conteneur' <options>`

Dossier 2 : Adopter l'approche Agile (14pts)

Après avoir défini le backlog product du projet, l'équipe Scrum(*Product Owner, Scrum Master et les Développeurs*) a organisé un sprint planning lors duquel ils ont défini les tâches du premier sprint :

TÂCHE	DURÉE (en jours)	PRÉDÉCESSEURS
A	3	-
B	2	A
C	2	-
D	4	B,C

E	5	B,C
F	4	B,C
G	3	D ,E ,F

Avec :

A : Créer les maquettes	E : Développement du composant 2
B : Création des interfaces html	F : Développement du composant 3
C : Création des tables	G : Test d'intégration
D : Développement du composant 1	

Questions :

1. Définir le backlog product.(2,5pts)

Le Product Backlog est le point central de tout projet Scrum. Il est sous la responsabilité unique du Product Owner, mais doit également être accessible par l'équipe. C'est à partir de ce document que les sprints seront planifiés.

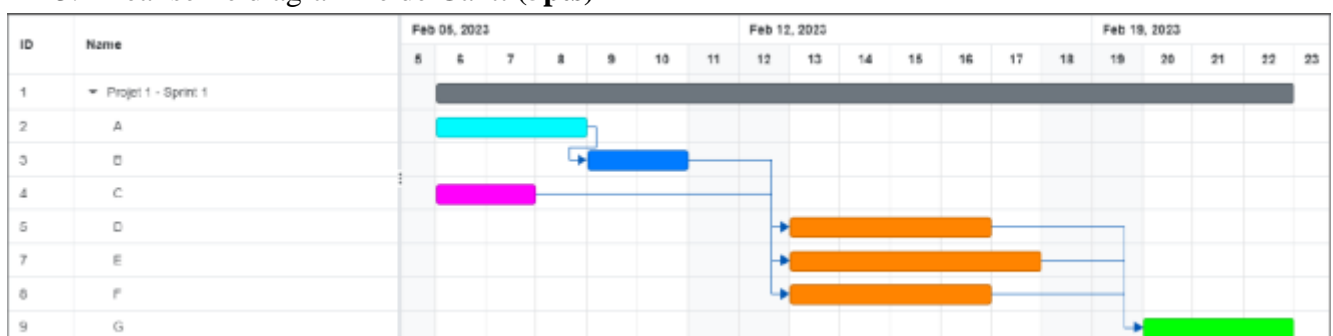
Le product backlog est défini par le guide scrum comme étant "une liste ordonnée et émergente de ce qui est nécessaire pour améliorer le produit"

2. C'est quoi le rôle du product owner ?(2,5pts)

Le Product Owner est la seule personne responsable de la gestion du Product Backlog. La gestion du backlog produit comprend :

- L'expression claire des éléments du backlog produit
- L'ordonnancement des éléments dans le backlog produit pour mieux réaliser les objectifs et les missions
- L'optimisation de la valeur du travail effectué par l'équipe de développement
- L'assurance que le backlog produit est visible, transparent et clair pour tous, et montre sur quoi l'équipe de développement travaillera prochainement
- L'assurance que l'équipe de développement comprend adéquatement les éléments du backlog produit.

3. Réaliser le diagramme de Gantt (5pts)



4. Identifier le chemin critique (4pts)

A-B-E-G

Dossier 3: Gérer les données (partie :MongoDB) (12pts)

1. afficher le nombre total des propriétaires par pays d'accueil. (2,5pts)

Filière	DEVOWFS	Variante	V1	Page	2 sur 14
Examen	Fin de Formation	Session	Report		

db.voitures.agregate([{ \$group : { _id : "proprietaire. pays_accueil" ,nbe:{\$sum,1}} }])

2. Afficher les propriétaires qui sont nés à Paris.(2,5pts)

db.voitures.find({"proprietaire.lieu_naissance" : /^paris\$/i},{_id :0,proprietaire :1})

3. afficher les propriétaires qui ont une voiture mise en circulation avant le '2023/02/11'(2pts)

db.voitures.find({}, {Date_circulation : {\$lte : ISODate("2023-02-11")}})

4. diminuer de 2% la valeur actuelle de toutes les voitures.(2,5pts)

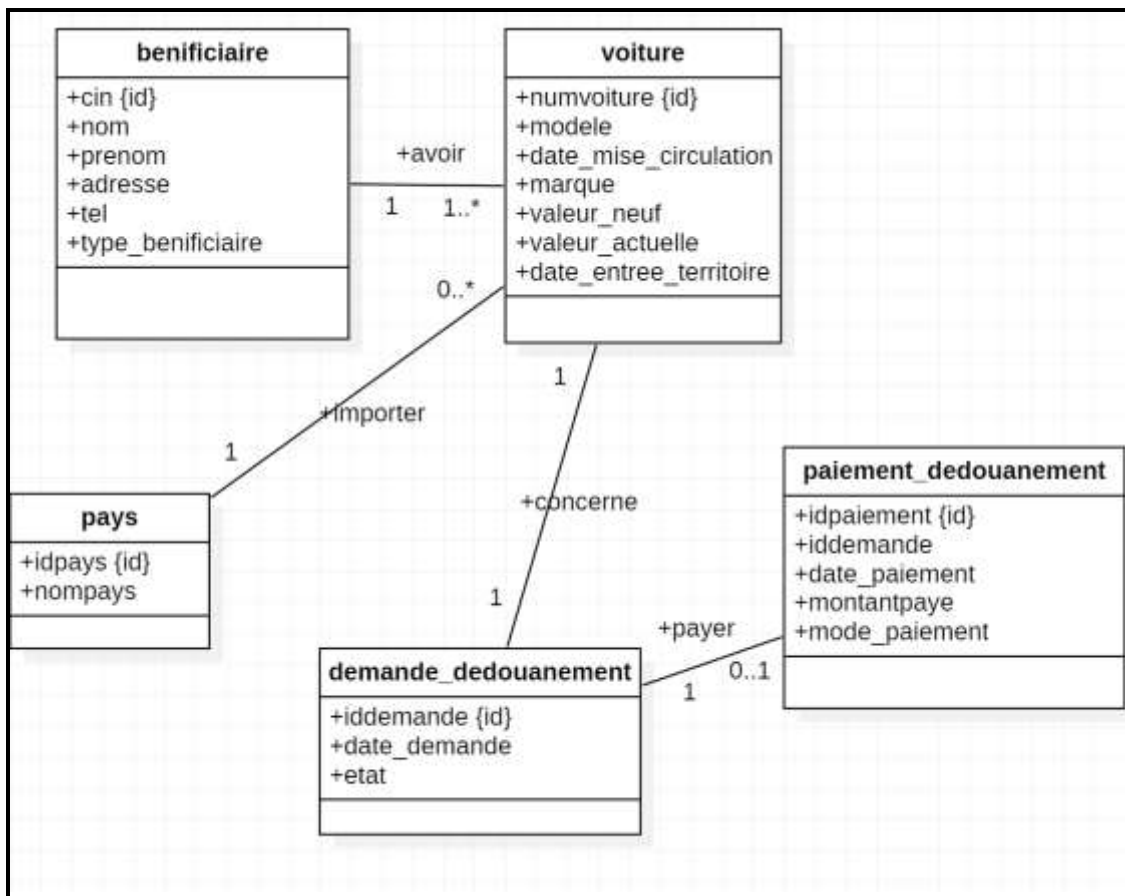
db.voitures.update({}, {\$set : { \$inc: { valeur_actuelle:- {\$multiply : { valeur_actuelle ,0.02}} } } })

5. Supprimer toutes les voitures du proprietaire num_pass=P1(2,5pts)

db.voitures.deleteMany({ "proprietaire. num_pass": "P1" })

Dossier4: préparer un projet web(6pts)

1. En se basant sur le MLD déjà établi et points précisés, réaliser le digramme de classe équivalent.(6pts)



Dossier 5 :Gérer les données (partie : MySql) (12pts)

1. Ecrire une requête SQL qui affiche les voitures de marque « peugeot » importées de la France et qu'ont été mise en circulation cette année. (3pts)

Select voiture. from voiture join pays on voiture. id_pays_importation=pays.idpays
Where lower(marque)= 'peugeot' and year(date_mise_circulation)= year(CURDATE())) ;*

2. Ecrire une fonction qui retourne, pour un bénéficiaire passé comme paramètre, le nombre de ses demandes de dédouanement rejetées et le nombre de ses demandes validées.(3pts)

*DELIMITER \$\$
CREATE function Q2(IN pcin VARCHAR(50), out nbdA int)
RETURNS INT
READS SQL DATA
BEGIN
DECLARE nbdR INT;
SELECT count(*) into nbdR FROM demande_dedouanement join voiture on
demande_dedouanement.numvoiture=voiture.numvoiture WHERE upper(cin)= upper(pcin)
and etat='rejeté';
SELECT count(*) into nbdA FROM demande_dedouanement join voiture on
demande_dedouanement.numvoiture=voiture.numvoiture WHERE upper(cin)= upper(pcin)
and etat=' validée';
Return nbdA;
END \$\$
DELIMITER ;*

3. Ecrire une procédure qui affiche l'année où on a rejeté le plus de demandes. (3pts).

*DELIMITER \$\$
CREATE procedure Q3()
BEGIN
DECLARE
Select year(date_demande)from demande_dedouanement
Where etat='rejeté'
groub by year(date_demande)
having count(*)=*

Filière	DEVOWFS	Variante	V1	Page	4 sur 14
Examen	Fin de Formation	Session	Report		

(Select max(count()) from demande_dedouanement*

Where etat='rejeté'

group by year(date_demande))

END \$\$

DELIMITER;

4. Ecrire un déclencheur qui empêche le paiement des droits de dédouanement d'une voiture si sa demande de dédouanement est rejetée.(3pts)

DELIMITER //

CREATE TRIGGER Q4 BEFORE INSERT on paiement_dedouanement

FOR EACH ROW

BEGIN

DECLARE

vetat varchar(50) ;

select etat into vetat from demande_dedouanement where iddemande=new.iddemande;

if vetat='rejete'

THEN SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'on paie pas une demande rejeté' ; END IF ;

END // DELIMITER ;

Dossier 6: développer en Back end(19pts)

- 1.Écrire le code de la méthode up() et de la méthode down() de fichier migration de la table demande_dedouanement.(2pts).

```
Public function up(): void
{
    Schema::create('demande_dedouanements', function (Blueprint$table) {
        $table->string('iddemande')->primary();
        $table->date('date_demande');
        $table->enum('etat', ['encours', 'rejetee', 'validee']);
        $table->string('numvoiture');
        $table->foreign('numvoiture')->references('numvoiture')->on('voitures');

        $table->timestamps();

    });
}
```

Filière	DEVOWFS	Variante	V1	Page	5 sur 14
Examen	Fin de Formation	Session	Report		

```
Public function down()

{
    Schema::dropIfExists('demande_dedouanements');
}
```

2.Écrire le code qui permet de peupler la table **demande_dedouanement** par des jeux d'enregistrement **en utilisant Seeder**.(2pts).

```
class Demande_dedouanementsTableSeeder extends Seeder
{
    public function run()
    {
        DB::table('demande_dedouanements')->insert([
            [
                'iddemande'=> "D1",
                'date_demande'=> "02/12/2020",
                'etat'=> 'encours',
                'numvoiture' => '666H77777'
            ]
        ]);
    }
}
```

DatabaseSeeder.php

```
$this->call([Demande_dedouanementsTableSeeder::class]);
```

3.Écrire le code des deux modèles associés à la table **demande_dedouanement** et **voiture**. Et définir les relations entre eux.(3pts).

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class voiture extends Model
{
    use HasFactory;
    public function demande_dedouanement ()
    { return $this->hasOne(demande_dedouanement::class);
```

```
}
```

```
}
```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class demande_dedouanement extends Model
{
    use HasFactory;
    protected $fillable = ['iddemande', 'date_demande', 'etat', 'numvoiture'];

    public $table = 'demande_dedouanements';
    public function voiture()
    { return $this->belongsTo(voiture::class);
    }
}
```

4.Écrire le code du contrôleur **DedouanementControleur** qui contient les méthodes :**(8pts)**.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\{ demande_dedouanement, voiture };
Class DedouanementControleur extends Controller
{
    public function recherche (Request $request){
        $demandes = demande_dedouanement ::join('voitures', ' voitures.numvoiture',
        '=', demande_dedouanements'.numvoiture') ->join(' beneficiaires', '
        beneficiaires.cin', '=', 'voitures.cin') ->where(' beneficiaires.cin', '=',
        $request['cin'])
        ->get([' demande_dedouanements.* ']);

        Return view(' vueRecherche',[ "demandes"=>$demandes]);
    }
    public function valider((request $request){
        $demande = demande_dedouanement::find($request->iddemande);
        $demande->etat = 'validee';
        $demande->save();
        Return redirect()->route('index')->with('info', 'la demande est validee');
```

Filière	DEVOWFS	Variante	V1	Page	7 sur 14
Examen	Fin de Formation	Session	Report		


```

    }

    Public function ajouter(Request$request)
    {
        $this->validate($request, [
            'numvoiture'=>required|regex:/^[0-9]+[a-zA-Z][0-9]+$',
            ' date_entree_territoire ' =>'required| date|date_format:Y-m-d|',
            ' date_mise_circulation' =>'required| date|date_format:Y-m-d| before: date_entree_territoire',
        ]);

        $voiture = voiture::create($request->post()); //$request->validated()
    }

    Public function suivre_demande(Request$request)
    {
        $ demande_dedouanement =
        demande_dedouanement::where('iddemande',$request['iddemande'])->get() ;
        if(isset($demande_dedouanement))
        {return response()->json([
            'status' => true,
            'demande' => $demande_dedouanement.etat
        ]);}
    else{{return response()->json([
        'status' => true,
        'message' => 'demande introuvable'
    ]);}
    }

}

}

```

5. Écrire les routes à ajouter dans web.php pour accéder aux 2 méthodes **Rechercher**, **Ajouter** et **Modifier** du contrôleur « **DedouanementContrôleur** » et la route à ajouter dans **api.php** pour accéder à la méthode « **Suivre_demande** » (4pts).

```

//les routes à ajouter dans web.php
Route::get('/', function(){return view('vuerecherche');});
Route::post('rech', [DedouanementContrôleur::class,'recherche'])->name('recherche');

```

Filière	DEVOWFS	Variante	V1	Page	8 sur 14
Examen	Fin de Formation	Session	Report		

```

Route::get('valider/{iddemande}', [DedouanementControleur::class,'valider'])->name('valider');
Route::post('ajouter', [DedouanementControleur::class,'ajouter'])->name('ajouter');

//la routes à ajouter dans api.php
Route::post('suivre_demande' , [DedouanementControleur::class, 'suivre_demande' ]);

```

6.Ecrire le code de la vue **Recherche.blade.php** (4pts).

```

<!DOCTYPE html>
<html>
<head>
<linkrel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<navclass="navbarnavbar-light bg-light">
  <divclass="container-fluid">
    <aclass="navbar-brand">dedouanement</a>
    <formclass="d-flex input-group w-auto" method='post'action="{{ route('recherche') }}">
      @csrf
      <input type="search"
        class="form-control rounded"
        placeholder="Saisir le cin du benificiaire"
        aria-label="Search"
        name="cin"
      />
      <spanclass="input-group-btn">
        <buttonclass="btnbtn-secondary"type="submit">rechercher</button>
      </span>
    </form>
  </div>
</nav>
<tableborder="1">
  <thead>
    <tr>
      <th>iddemande</th>
      <th>date_demande</th>
      <th>num_voiture</th>
      <th>etat</th><th>action</th>
    </tr>
  </thead>
  <tbody>
    @foreach ($demandes as $demande)
      <tr>
        <td>
          {{ $demande ->iddemande }}
        </td>
        <td>
          {{ $demande ->date_demande }}
        </td>
      </tr>
    </tbody>
  </table>

```

Filière	DEVOWFS	Variante	V1	Page	9 sur 14
Examen	Fin de Formation	Session	Report		

```

        </td>
        <td>
            {{ $ demande ->numvoiture }}
        </td><td>
            {{ $ demande ->etat }}
        </td>
        <td>
            <a href="{{ route('valider',$demande->iddemande) }}" >valider</a>
        </td>
    </tr>
</tbody>
</table>
</html>

```

Dossier 7: développer en front end: (23PTS)

2.Créer le store Redux décrit ci-dessus avec son état initial et un reducer pour les deux actions suivantes :**(6pts)**

```

import { createSlice ,configureStore} from '@reduxjs/toolkit'
const simulationSlice =createSlice(
{name:'simulation',
initialState:[
    { marque : 'peugeot', date_mise_circulation
:"01/03/2022",valeur_imposable :174420,montant:72037 }
],
reducers:{
    ajouter :(state,action) => {
        state.push({...action.payload});
        return state; },
    supprimer: (state,action) => {
        state=state.filter(item => item.marque !== action.payload);
        return state;
    }
}
});
export const {ajouter,supprimer}=simulationSlice.actions;

export const store=configureStore(
{
    reducer:{simulation:simulationSlice.reducer}
});

```

Filière	DEVOWFS	Variante	V1	Page	10 sur 14
Examen	Fin de Formation	Session	Report		

3.Écrire le code du **composant** Calcul_taxe_dédouanement.js. :

- le clic sur le bouton « **valider** » calcule et affiche la valeur imposable, Droit d'importation, TVA ,Taxe parafiscale et Montant total des droits et taxes. (5PTS)
- Le calcul doit être sauvegardé dans le store (5PTS)

```
import React, { useState } from 'react'
import { useDispatch } from "react-redux"
import { ajouter } from '../redux'

const Calcul_taxe_dédouanement = props => {
  const initialState = { marque: '', modele: '',
date_mise_circulation: '', valeur_neuf: 0 }
  const [ vehicule, setVehicule ] = useState(initialFormState);
  const [ valeur_imposable, setValeur_imposable ] = useState(0);

  const dispatch = useDispatch();
  const [ affichage, setAffichage ] = useState(false);
  const handleInputChange = event => {
    const { name, value } = event.target

    setVehicule({ ...vehicule, [name]: value })
  }

  return (
    <div>
      <form
        onSubmit={event => {
          event.preventDefault()
          let today = new Date().getTime();
          let circ_date = new
Date(vehicule.date_mise_circulation).getTime();
          let difference = Math.abs(today - circ_date);
          let age = Math.floor(difference / (1000 * 3600 * 24 * 365));
          if (age < 1) { setValeur_imposable(this.vehicule.valeur_neuf); }
          if (age < 2 & age >= 1) { setValeur_imposable(parseInt(this.vehicule
.valeur_neuf) * 0.9); }
          if (age < 3 & age >= 2) { setValeur_imposable(parseInt(this.vehicule
.valeur_neuf) * 0.8); }
          if (age >= 3) { setValeur_imposable(parseInt(this.vehicule.valeu
r_neuf) * 0.75); }
          let information = { marque : vehicule.marque,
date_mise_circulation : vehicule.date_mise_circulation, valeur_imposable
: this.valeur_imposable, montant: this.valeur_imposable * 0.377 }
          dispatch(ajouter(information));

          setAffichage(true);
        }}
      </form>
    </div>
  )
}
```

```

    >

    <label>marque du vehicule</label>
    <input type="text" name="marque" value={vehicule.marque}
onChange={handleInputChange} />
    <label>Modele du vehicule</label>
    <input type="text" name="modele" value={vehicule.modele}
onChange={handleInputChange} />
    <label>date de 1ere circulation</label>
    <input type="date" name="date_mise_cir"
value={vehicule.date_mise_cir} onChange={handleInputChange} />
    <label>valeur à l'etat neuf</label>
    <input type="text" name="valeur_neuf"
value={vehicule.valeur_neuf} onChange={handleInputChange} />
    <button className='btn btn-success'>valider</button>
  </form>

  {this.affichage&&<div>
    <h1>details des droits et des taxes</h1>
    <h2>
      valeur taxable: {this.valeur_imposable} DH
    </h2>
    <h2>
      droit d'importation: {this.valeur_imposable*0.175} DH
    </h2>
    <h2>
      TVA: {this.valeur_imposable*0.2} DH
    </h2>
    <h2>
      Taxe parafiscale: {this.valeur_imposable*0.0025} DH
    </h2>
    <h2>
      montant total des droits et des taxes:
      {this.valeur_imposable*0.377} DH
    </h2>
  </div>
}

  </div>
)
}

export default Calcul_taxe_dedouanement

```

4.Écrire le code du composant ‘Suivre_Demande.js’ qui permet de suivre l’état d’une demande de dédouanement donnée. En utilisant l’api créé dans partie backend dont les informations suivantes (5PTS)

Filière	DEVOWFS	Variante	V1	Page	12 sur 14
Examen	Fin de Formation	Session	Report		

URL de l'api	http://127.0.0.1:8000/apis/suivre/{iddemande}
Méthode HTTP	GET
Résultat	Résultat de la fonction Suivre_demande du backend

```
import React, { useState } from "react";

import axios from 'axios'

function Suivre_Demande() {

  const [demande, setDemande] = useState({ iddemande: '', numvoiture: '',
date_demande: '',etat:'' });

  const [istrouve, setIstrouver] = useState(false);
  const [affichemessage, setAffichemessage] = useState(false);
  const [ismessage, setMessage] = useState('');
  const verifier = async (numdemande) => {

    await axios.get("https://172.0.0.1:8000/api/suivre",{params:
    {
      numdemande: numdemande
    }}
    )
    .then((response) => {
      if(response.data.message==null)
      {setDemande( response.data);
      setIstrouver(true);}
      else{setMessage( response.data); setAffichemessage(true);}
    })
    .catch((error) => console.log(error));
  };

  const handleInputChange = event => {
    setIstrouver(false);
    const { name, value } = event.target

    setDemande({ ...demande, [name]: value })

  }
  const handleverivierSubmit = (evt) => {
    evt.preventDefault();
    verifier( evt.target.iddemande.value);
  };

  return (
    <div className="App">
```

```

    <h1>suivie demande</h1>
    <form onSubmit={handleverivierSubmit}>

    <label>iddemande</label>
    <input type="text" name="iddemande" value={demande.iddemande}
onChange={handleInputChange} />
    <button className='btn btn-success'>suivre</button>
    </form>
    {this.affichemessage&& <h1>this.message</h1>}
    {this.istrouve&&<div>
    <h1>detail demande</h1>
    <h2>
        idedemande: {this.demande.iddemande}
    </h2>
    <h2>
        date demande: {this.demande.date_demande}
    </h2>
    <h2>
        etat: {this.demande.etat}
    </h2>
    <h2>
        numvoiture: {this.demande.numvoiture}
    </h2>

    </div>
    }
    </div>
    );
}

export default Suivre_Demande;

```