



EXAMEN SERVICES WEB

Semestre : 1 ☐ 2

Session : Principale

Rattrapage ☐

Module : Services WEB

Enseignant(s) : UP_WEB

Classe(s) : 4 Arctic/4 DS/4 INFINI/4 NIDS

Documents autorisés : OUI ☐ NON ☐

Nombre de pages : 6

Calculatrice autorisée : OUI ☐ NON ☐

Internet autorisée : OUI ☐ NON ☐

Date : 31-05-2022

Heure : 08h30

Durée : 1h30min

Exercice 3 : Etude de cas : Gestion des Randonnées (12 pts) Corrigé

On souhaite mettre en place une application web RESTFUL permettant la gestion des Randonnées.

L'entité randonnée possède les informations suivantes : id, destination, prix, date_debut, date_fin, heure_depart, description.

Les ressources REST qui constituent le service web sont décrites ci-dessous :

- Ajout d'une randonnée

Requête	POST L post wl put => Entrée de données => On utilise l'@consumes	/randonnee
	Header	Content-Type : application/json
	Body	{ "id": 1, "Destination": "L'Anneau du Kerry" "prix": "1500 \$" "date_debut": "31-05-2022" "date_fin": "05-06-2022" "heure_depart": "10h:00" "description": "Connue pour ses plages immaculées de sable blanc" }
Réponse	Header	Content-type : text/plain bch nhotouha fl @Produces khaterha type de sortie de la réponse
	Status	201 CREATED / 404 NOT FOUND
	Body	Randonnée a été ajouté avec succès / Problème d'ajout I msg hedha yethat fl body mtaa I fonction fl .entity

- Affichage de la liste des randonnées

Requête	GET	/randonnee
---------	-----	------------

Réponse	Header	Content-Type : application/json
	Status	200 OK / 404 NOT FOUND (fl body mtaa l fct bch nahkiw alihom)

	Body	<pre>[{ "id": 1, "Destination": "L'Anneau du Kerry" "prix": "1500" "date_debut": "31-05-2022" "date_fin": "05-06-2022" "heure_depart": "10h:00" "description": "Connue pour ses plages immaculées de sable blanc" }, { "id": 2, "Destination": "La traversée des monts Rodna" "prix": "2000" "date_debut": "10-06-2022" "date_fin": "15-06-2022" "heure_depart": "12h:00" "description": "Situés au nord de la Roumanie," }]</pre>
--	------	--

- Récupération d'une randonnée pour une date choisie

Requête	GET	/randonnee?date_debut=10-06-2022 madem feha ? => QueryParam wl variable heya "date_debut" eli tji baaed ? bedhabet
Réponse	Header	Content-Type : application/json
	Status	200 OK
	Body	<pre>{ "id": 2, "Destination": "La traversée des monts Rodna" "prix": "2000" "date_debut": "10-06-2022" "date_fin": "15-06-2022" "heure_depart": "12h:00" "description": "Situés au nord de la Roumanie," }</pre>

- a) Développez le service web RESTful décrit ci-dessus en complétant les parties manquantes :

Remarque : IMPORTANT :

```
public Response  
=> tahki al return Response.status 200/404..  
ken mandikch response fl return rod belek tahki alihom : kima houni  
mithel : public Randonnee getRandoByDateDebut
```

La classe **RestActivator**

```
[1] .....  
public class RandonneeActivator [2] {  
}
```

Réponse :

[1] `@ApplicationPath("")` (khater il n'a pas spécifié)

[2] `extends Application`

La classe **RandonneeRessources** dans laquelle les sous-ressources seront développées.

1. Ajout d'une randonnée

```
[3] .....  
[4] .....  
[5] .....  
public Response addRando (Randonnee randonnee) {  
    if (lstrandonnee.add(randonnee))  
        [6] .....  
        [7] .....  
}
```

Réponse :

[3] `@Produces(Media Type.TEXT_PLAIN)`

[4] `@POST`

[5] `@Consumes(MediaType.APPLICATION_JSON)`

// maa post naamlou consumes

// fl header howa 9al json

[6] `return Response.status(Response.Status.CREATED).entity("Randonnee a ete ajoute avec succès").build();`

} (howa yheb yra 200)

```
else {
[7] return Response.status(Response.Status.NOT_FOUND).entity("Problème
kid'ajout").build(); } } (howa yheb yra 404)
```

2. Affichage de la liste des randonnées

```
[8] . . . . .
[9] . . . . .
public Response getAllRandonnee()
{
    if(!lstrandonee.isEmpty())
        return Response.[10].    ;
    return Response.[11].    ;
}
```

Réponse :

[8] @GET

[9] @Produces(MediaType.APPLICATION_JSON)

[10] status(Response.Status.CREATED).entity(randonnees).build(); (howa yheb yra 200)

[11].status(Response.Status.NOT_FOUND).build(); } (howa yheb yra 404)

3. Récupération d'une randonnée pour une date choisie

```
[12] . . . . .
[13] . . . . .
public Randonne getRandoByDateDebut([14]... . . . .String d)
{
    for(Randonne r : lstrandonee) {
        if(r.getDateDebut().equals(d))
            return [15].    ;
    }
    return [16].    ;
}
```

Réponse :

[12] @GET

[13] @Produces(MediaType.APPLICATION_JSON)

[14] @QueryParam("date_debut") (khater ? baedha date_debut)

[15] r

[16] null

b) Supposons qu'on va ajouter une ressource permettant de chercher la randonnée la

moins chère de toutes les randonnées :

```
@GET
@Path("/{prix}")
@Produces(MediaType.APPLICATION_JSON)
public Response
findRandomneByPrix(@PathParam("prix") String price)
{
    /*
     La fonction permet de retourner la randonne ayant
     ce prix avec status OK,
     Sinon elle va retourner Not Found.
    */
}
```

Donner la description de la requête HTTP permettant de consommer le service ainsi que la réponse HTTP retournée avec un exemple concret.

SÉCURITÉ JWT

- c) On souhaite intégrer l'aspect de sécurité sur le RESTfull de Gestion de randonnées. Que proposez- vous comme solution ?

=> JWT json web token

- d) Expliquez le principe de cette technique de sécurité.
- e) D'après la figure ci-dessous, quelles sont les informations utilisées pour la définition de la deuxième partie du jeton ?

2eme partie => PAYLOAD

sub : sujet : SVignello

iss: Origine du token: http://...

iat: date de création du token : 16255722619

nbf: A ne pas utiliser avant cette date 163347..

Encoded

eyJhbGciOiJIUzI1NiIsInR5cCI6IiJ9.eyJzdW
IiOiJTVmInbmVsbG8iLCJpc3MiOiJodHRwOi8vb
G9jYWxob3N0jGw0TAvU1ZpZ25lbGxvL2F1dGhl
bnRpZiIsImh0cCI6MTYyNTU3MjYxOStSwibmJmIjo
xNjMzNDcyMTE4fQ.LoQA9z5a2N6SAnYn6peSIj5
h2_8KcwcKiHELvOZttwU

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
"alg": "HS256",
"typ": ""
}
```

PAYLOAD: DATA

```
{
  "sub": "SVignello",
  "iss": "http://localhost:8090/SVignello/authentif",
  "iat": 1625572619,
  "nbf": 1633472118
}
```


Bon Travail ☺