



AP



3-3-8 Développement et test des services web fournis

Le service GetLesParcoursDunUtilisateur

- 1- Rôle du service web
- 2- Messages retournés par le service web
- 3- Format des messages
- 4- Rappel du cas d'utilisation
- 5- scénarios de test

1- Rôle du service web

Rôle : ce service web permet à un utilisateur d'obtenir la liste de ses parcours ou la liste des parcours d'un utilisateur qui l'autorise.

Paramètres à fournir :

- **pseudo** : le pseudo de l'utilisateur
- **mdp** : le mot de passe de l'utilisateur hashé en sha1
- **pseudoConsulte** : le pseudo de l'utilisateur dont on veut consulter la liste des parcours
- **lang** : le langage utilisé pour le flux de données ("xml" ou "json")

Description du traitement :

- Vérifier que les données transmises sont complètes
- Vérifier l'authentification de l'utilisateur demandeur
- Vérifier l'existence du pseudo de l'utilisateur consulté
- Vérifier si l'utilisateur demandeur consulte ses propres traces, ou s'il est autorisé à consulter les traces de l'utilisateur consulté
- Fournir la liste des traces de l'utilisateur consulté

2- Messages retournés par le service web

Messages retournés en cas de problème :

"Erreur : données incomplètes."
"Erreur : authentification incorrecte."
"Erreur : pseudo consulté inexistant."
"Erreur : vous n'êtes pas autorisé par cet utilisateur."

Messages retournés dans le cas normal :

"Aucune trace pour l'utilisateur oxygen."
"2 trace(s) pour l'utilisateur callisto."

3- Format des messages

Le flux XML reçu aura un des formats suivants :

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Service web GetLesParcoursDunUtilisateur - BTS SIO - Lycée De La Salle - Rennes-->
<data>
  <reponse>..... (message retourné par le service web) .....</reponse>
</data>
```

Cas où on ne renvoie pas de données

Cas où on renvoie des données

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Service web GetLesParcoursDunUtilisateur - BTS SIO - Lycée De La Salle - Rennes-->
<data>
  <reponse>2 trace(s) pour l'utilisateur callisto</reponse>
  <donnees>
    <lesTraces>
      <trace>
        <id>2</id>
        <dateHeureDebut>2018-01-19 13:08:48</dateHeureDebut>
        <terminee>1</terminee>
        <dateHeureFin>2018-01-19 13:11:48</dateHeureFin>
        <distance>1.2</distance>
        <idUtilisateur>2</idUtilisateur>
      </trace>
      <trace>
        <id>1</id>
        <dateHeureDebut>2018-01-19 13:08:48</dateHeureDebut>
        <terminee>0</terminee>
        <distance>0.5</distance>
        <idUtilisateur>2</idUtilisateur>
      </trace>
    </lesTraces>
  </donnees>
</data>
```

Données de
la première
trace

La date de fin n'est indiquée
que si la trace est terminée

Ensemble
des traces

Données de
la dernière
trace

Le flux Json reçu aura un des formats suivants :

```
{
  "data": {
    "reponse": "..... (message retourné par le service web) .....",
  }
}
```

Cas où on ne renvoie pas de données

Cas où on renvoie des données

```
{
  "data": {
    "reponse": "2 trace(s) pour l'utilisateur callisto",
    "donnees": {
      "lesTraces": [
        {
          "id": "2",
          "dateHeureDebut": "2018-01-19 13:08:48",
          "terminee": "1",
          "dateHeureFin": "2018-01-19 13:11:48",
          "distance": "1.2",
          "idUtilisateur": "2"
        },
        {
          "id": "1",
          "dateHeureDebut": "2018-01-19 13:08:48",
          "terminee": "0",
          "distance": "0.5",
          "idUtilisateur": "2"
        }
      ]
    }
  }
}
```

Données de la première trace

La date de fin n'est indiquée que si la trace est terminée

Données de la dernière trace

Ensemble des traces

4- Rappel du cas d'utilisation

Ce service web sera utilisé pour l'étape n°3 du cas d'utilisation **Consulter un parcours** :

Nom cas d'utilisation : Consulter un parcours
Acteur déclencheur : Utilisateur
Pré conditions : Être authentifié comme utilisateur
Post conditions : Néant
Scénario nominal : <ul style="list-style-type: none">1- Le système affiche la liste des personnes dont il peut consulter les parcours : lui-même ou un membre qui lui a donné l'autorisation de consultation2- L'utilisateur choisit la personne dont il veut consulter un parcours : lui-même ou un membre qui lui a donné l'autorisation de consultation• 3- Le système affiche la liste des parcours de la personne choisie4- L'utilisateur choisit le parcours et valide5- Le système affiche le parcours sur une carte
Exceptions :
Contraintes : <ul style="list-style-type: none">Dans le cas d'un parcours terminé, l'application permettra de visualiser le parcours sur une carte, ainsi que les données du parcours (dates et heures de début et de fin, distance parcourue, durée, dénivelés positifs et négatifs, vitesse moyenne).Dans le cas d'un parcours non terminé, l'application permettra de visualiser le parcours sur une carte, avec une réactualisation automatique toutes les minutes pour suivre le pratiquant en temps réel. Chaque point de passage sera représenté par un marqueur. En passant la souris sur ce marqueur, l'heure de passage sera automatiquement affichée dans une info-bulle. La dernière position sera affichée de façon claire (latitude, longitude) car c'est celle qui devrait être transmise aux équipes de secours en cas de besoin.
Questions ouvertes :

5- scénarios de test



Les tests suivants seront réalisés avec un navigateur web.
On comparera le flux XML prévu avec le flux XML obtenu.
On vérifiera les envois de mails.

5-1 Test avec des paramètres incorrects ou incomplets

Conditions du test :

- appeler le service avec une URL incomplète (paramètres absents ou mal nommés)

URL testée :

`http://localhost/ws-php-xxx/tracegps/api/GetLesParcoursDunUtilisateur`

Message prévu : "**Erreur : données incomplètes.**"

Message conforme :

- ☐ oui
- ☐ non

5-2 Test du format des données XML

Conditions du test :

- appeler le service avec le paramètre "**lang=xml**" (ou sans le paramètre lang)

URL testée :

`http://localhost/ws-php-xxx/tracegps/api/GetLesParcoursDunUtilisateur?lang=xml`
`http://localhost/ws-php-xxx/tracegps/api/GetLesParcoursDunUtilisateur`

Message prévu : "**Erreur : données incomplètes.**"

Format XML correct :

- ☐ oui
- ☐ non

5-3 Test du format des données Json

Conditions du test :

- appeler le service avec le paramètre "**lang=json**"

URL testée :

`http://localhost/ws-php-xxx/tracegps/api/GetLesParcoursDunUtilisateur?lang=json`

Message prévu : "**Erreur : données incomplètes.**"

Format Json correct :

- ☐ oui
- ☐ non

5-4 Test avec une authentification incorrecte

Conditions du test :

- appeler le service avec une URL complète (paramètres présents et bien nommés)
- utiliser un compte inexistant, ou un compte existant avec un mauvais mot de passe

URL testée :

```
http://localhost/ws-php-xxx/tracegps/api/GetLesParcoursDunUtilisateur?pseudo=europa
&mdp=111111111111&pseudoConsulte=callisto&lang=xml
```

Message prévu : "**Erreur : authentification incorrecte.**"

Message conforme :

- ☐ oui
- ☐ non

5-5 Test avec un pseudo inexistant

Conditions du test :

- appeler le service avec une URL complète (paramètres présents et bien nommés)
- utiliser un compte existant, avec un mot de passe correct
- utiliser un pseudo inexistant pour lister ses parcours

URL testée :

```
http://localhost/ws-php-xxx/tracegps/api/GetLesParcoursDunUtilisateur?pseudo=europa
&mdp=13e3668bbee30b004380052b086457b014504b3e&pseudoConsulte=toto&lang=xml
```

Message prévu : "**Erreur : pseudo consulté inexistant inexistant.**"

Message conforme :

- ☐ oui
- ☐ non

5-6 Test avec un utilisateur non autorisé

Conditions du test :

- appeler le service avec une URL complète (paramètres présents et bien nommés)
- utiliser un compte existant, avec un mot de passe correct
- utiliser un pseudo existant pour lister ses parcours mais qui n'a pas donné son autorisation

URL testée :

```
http://localhost/ws-php-xxx/tracegps/api/GetLesParcoursDunUtilisateur?pseudo=europa
&mdp=13e3668bbee30b004380052b086457b014504b3e&pseudoConsulte=juno&lang=xml
```

Message prévu : "**Erreur : vous n'êtes pas autorisé par cet utilisateur.**"

Message conforme :

- ☐ oui
- ☐ non

5-7 Test avec un utilisateur autorisé et le format XML

Conditions du test :

- appeler le service avec une URL complète (paramètres présents et bien nommés)
- utiliser un compte existant, avec un mot de passe correct
- utiliser un pseudo existant pour lister ses parcours et qui a donné son autorisation
- demander les données au format **XML**

URL testée :

```
http://localhost/ws-php-xxx/tracegps/api/GetLesParcoursDunUtilisateur?pseudo=europa
&mdp=13e3668bb30b004380052b086457b014504b3e&pseudoConsulte=callisto&lang=xm
```

Message prévu : "**2 trace(s) pour l'utilisateur callisto.**"

Message conforme :

- ☐ oui
- ☐ non

Format **XML** des données conforme :

- ☐ oui
- ☐ non

5-8 Test avec un utilisateur autorisé et le format Json

Conditions du test :

- appeler le service avec une URL complète (paramètres présents et bien nommés)
- utiliser un compte existant, avec un mot de passe correct
- utiliser un pseudo existant pour lister ses parcours et qui a donné son autorisation
- demander les données au format **Json**

URL testée :

```
http://localhost/ws-php-xxx/tracegps/api/GetLesParcoursDunUtilisateur?pseudo=europa
&mdp=13e3668bb30b004380052b086457b014504b3e&pseudoConsulte=callisto&lang=json
```

Message prévu : "**2 trace(s) pour l'utilisateur callisto.**"

Message conforme :

- ☐ oui
- ☐ non

Format **Json** des données conforme :

- ☐ oui
- ☐ non