



AP



3-3-7 Développement et test des services web fournis

Le service GetUnParcoursEtSesPoints

- 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 le détail d'un de ses parcours ou d'un parcours d'un membre qui l'autorise.

Paramètres à fournir :

- **pseudo** : le pseudo de l'utilisateur
- **mdp** : le mot de passe de l'utilisateur hashé en sha1
- **idTrace** : l'id de la trace à consulter
- **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
- Vérifier l'existence de la trace demandée
- Vérifier si l'utilisateur est le propriétaire de la trace, ou si il est autorisé à le consulter
- Fournir les données complètes de la trace

2- Messages retournés par le service web

Messages retournés en cas de problème :

"Erreur : données incomplètes."

"Erreur : authentication incorrecte."

"Erreur : parcours inexistant."

"Erreur : vous n'êtes pas autorisé par le propriétaire du parcours."

Messages retournés dans le cas normal :

"Données de la trace demandée."

3- Format des messages

Le flux XML reçu aura un des formats suivants :

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Service web GetUnParcoursEtSesPoints - 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

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Service web GetUnParcoursEtSesPoints - BTS SIO - Lycée De La Salle - Rennes-->
<data>
  <reponse>Données de la trace demandée.</reponse>
  <donnees>
    <trace>
      <id>2</id>
      <dateHeureDebut>2018-01-19 13:08:48</dateHeureDebut>
      <terminee>1</terminee>
      <dateHeureFin>2018-01-19 13:11:48</dateHeureFin>
      <idUtilisateur>2</idUtilisateur>
    </trace>
    <lesPoints>
      <point>
        <id>1</id>
        <latitude>48.2109</latitude>
        <longitude>-1.5535</longitude>
        <altitude>60</altitude>
        <dateHeure>2018-01-19 13:08:48</dateHeure>
        <rythmeCardio>81</rythmeCardio>
      </point>
      .....
      <point>
        <id>10</id>
        <latitude>48.2199</latitude>
        <longitude>-1.5445</longitude>
        <altitude>150</altitude>
        <dateHeure>2018-01-19 13:11:48</dateHeure>
        <rythmeCardio>90</rythmeCardio>
      </point>
    </lesPoints>
  </donnees>
</data>
```

Cas où on renvoie des données

Données de la trace

Données du premier point

Ensemble des points

Données du dernier point

Le flux Json reçu aura un des formats suivants :

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

```
{
  "data": {
    "reponse": "Données de la trace demandée.",
    "donnees": {
      "trace": {
        "id": "2",
        "dateHeureDebut": "2018-01-19 13:08:48",
        "terminee": "1",
        "dateHeureFin": "2018-01-19 13:11:48",
        "idUtilisateur": "2"
      },
      "lesPoints": [
        {
          "id": "1",
          "latitude": "48.2109",
          "longitude": "-1.5535",
          "altitude": "60",
          "dateHeure": "2018-01-19 13:08:48",
          "rythmeCardio": "81"
        },
        .....
        {
          "id": "10",
          "latitude": "48.2199",
          "longitude": "-1.5445",
          "altitude": "150",
          "dateHeure": "2018-01-19 13:11:48",
          "rythmeCardio": "90"
        }
      ]
    }
  }
}
```

Cas où on renvoie des données

Données de la trace

Données du premier point

Données du dernier point

Ensemble des points

4- Rappel du cas d'utilisation

Ce service web sera utilisé pour l'étape n°5 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 consultation3- 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/GetUnParcoursEtSesPoints`

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

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/GetUnParcoursEtSesPoints?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/GetUnParcoursEtSesPoints?pseudo=europa
&mdp=111111111111&idTrace=2&lang=xml
```

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

Message conforme :

- ☐ oui
- ☐ non

5-5 Test avec une trace inexistante

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 numéro de trace inexistant

URL testée :

```
http://localhost/ws-php-xxx/tracegps/api/GetUnParcoursEtSesPoints?pseudo=europa
&mdp=13e3668bb30b004380052b086457b014504b3e&idTrace=1234&lang=xml
```

Message prévu : "**Erreur : parcours inexistant.**"

Message conforme :

- ☐ oui
- ☐ non

5-6 Test avec une trace non autorisée

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 une trace existante mais non autorisée

URL testée :

```
http://localhost/ws-php-xxx/tracegps/api/GetUnParcoursEtSesPoints?pseudo=europa
&mdp=13e3668bb30b004380052b086457b014504b3e&idTrace=12&lang=xml
```

Message prévu : "**Erreur : vous n'êtes pas autorisé par le propriétaire du parcours.**"

Message conforme :

- ☐ oui
- ☐ non

5-7 Test avec une trace autorisée 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 une trace existante et autorisée
- demander les données au format **XML**

URL testée :

```
http://localhost/ws-php-xxx/tracegps/api/GetUnParcoursEtSesPoints?pseudo=europa  
&mdp=13e3668bb30b004380052b086457b014504b3e&idTrace=2&lang=xml
```

Message prévu : "**Données de la trace demandée.**"

Message conforme :

- ☐ oui
- ☐ non

Format **XML** des données conforme :

- ☐ oui
- ☐ non

5-8 Test avec une trace autorisée 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 une trace existante et autorisée
- demander les données au format **Json**

URL testée :

```
http://localhost/ws-php-xxx/tracegps/api/GetUnParcoursEtSesPoints?pseudo=europa  
&mdp=13e3668bb30b004380052b086457b014504b3e&idTrace=2&lang=json
```

Message prévu : "**Données de la trace demandée.**"

Message conforme :

- ☐ oui
- ☐ non

Format **Json** des données conforme :

- ☐ oui
- ☐ non