Live Event World Map

Projet – SIG Web

Hiver 2018

VINCENT LE FALHER

LEFV2603

Table des matières

[1. Sommaire 5](#_Toc512798247)

[2. Étape 1 5](#_Toc512798248)

[2.1 Définir un thème d’étude 5](#_Toc512798249)

[2.2 Architecture 6](#_Toc512798250)

[2.2.1 Composants 6](#_Toc512798251)

[1.1.1 Processus 7](#_Toc512798252)

[2.3 Rechercher au minimum 3 couches d’information (format shapefile idéalement) pour illustrer le thème d’étude 10](#_Toc512798253)

[2.3.1 Carte des pays pour créer un fond de carte personnalisé; 10](#_Toc512798254)

[2.3.2 Couche représentant les fuseaux horaires 11](#_Toc512798255)

[2.3.3 Couche représentant les stades de soccer 11](#_Toc512798256)

[2.3.4 Couche représentant les rencontres de soccer en cours 14](#_Toc512798257)

[2.3.5 Couche représentant les rencontres de soccer terminées 17](#_Toc512798258)

[2.4 Définir la zone géographique (étendue géographique et le système de coordonnées (code EPSG)) 17](#_Toc512798259)

[2.4.1 Étendue géographique 17](#_Toc512798260)

[2.4.2 Système de coordonnées 18](#_Toc512798261)

[3. Étape 2 18](#_Toc512798262)

[3.1 Créer une base de données à référence spatiale contenant vos données thématiques 18](#_Toc512798263)

[3.2 Configurer un serveur cartographique en mode WMS et en mode WFS 20](#_Toc512798264)

[3.3 Définir un style d’affichage par défaut adapté aux couches 21](#_Toc512798265)

[3.3.1 Couche des pays – ne\_10m\_admin\_0\_countries.map 22](#_Toc512798266)

[3.3.2 Couche des fuseaux horaires –ne\_10m\_time\_zones.map 23](#_Toc512798267)

[3.3.3 Couche des stades – venue\_ext.map 23](#_Toc512798268)

[3.3.4 Couche des recontres en cours – fixture\_in\_progress.map 24](#_Toc512798269)

[3.3.5 Couche des rencontres terminées – fixture\_finished.map 25](#_Toc512798270)

[3.4 Tester l’affichage (WMS et WFS) avec un logiciel cartographique 26](#_Toc512798271)

[3.4.1 QGIS avec les shapefile ESRI et vue SQL 26](#_Toc512798272)

[3.4.2 QGIS avec les couches WMS 28](#_Toc512798273)

[3.4.3 QGIS avec les couches WFS 30](#_Toc512798274)

[4. Étape 3 34](#_Toc512798275)

[4.1 Un cadre de travail personnalisé et adaptatif 34](#_Toc512798276)

[4.2 L’accès à vos couches thématiques 36](#_Toc512798277)

[4.3 Des outils de navigation 37](#_Toc512798278)

[4.4 Une vue généralisée de votre carte; les coordonnées latitude/longitude d’un point sur la carte 40](#_Toc512798279)

[4.5 L’interrogation possible d’au moins une couche 41](#_Toc512798280)

[4.6 Une liste de couches de fond extraites de Bing Map ou OpenStreetMap 44](#_Toc512798281)

[5. Étape 4 45](#_Toc512798282)

[5.1 Développer une application mobile simplifiée pour l’affichage des données 45](#_Toc512798283)

[6. Recommandations / Limitations 47](#_Toc512798284)

Figures

[Figure 1: Éléments de la solution 6](#_Toc512794133)

[Figure 2: Phase initiale 8](#_Toc512794134)

[Figure 3: Phase continue 9](#_Toc512794135)

[Figure 4: Example avec Similarity() 13](#_Toc512794136)

[Figure 5: Flux RSS 15](#_Toc512794137)

[Figure 6: Page HTML de la rencontre 16](#_Toc512794138)

[Figure 7: Sélection des vues SQL dans QGIS 27](#_Toc512794139)

[Figure 8: Affichage des fichiers shapefile ESRI et vues SQL dans QGIS 28](#_Toc512794140)

[Figure 9: Sélection des couches du service WMS 29](#_Toc512794141)

[Figure 10: Affichage des couches WMS dans QGIS 30](#_Toc512794142)

[Figure 11: Sélection des couches WFS avec le CRS par défaut 31](#_Toc512794143)

[Figure 12: Affichage des couches WFS avec le CRS par défaut dans QGIS 32](#_Toc512794144)

[Figure 13: Sélection d'un pays et le résultat de la sélection en rouge 33](#_Toc512794145)

[Figure 14: Sélection des couches WFS avec sélection du CRS EPSG:3857 33](#_Toc512794146)

[Figure 15: Affichage des couches WFS avec le CRS EPSG:3857 dans QGIS 34](#_Toc512794147)

[Figure 16: index.html de la carte 35](#_Toc512794148)

[Figure 17: Barre d'outils de l'interface Web 38](#_Toc512794149)

[Figure 18: Détails de la rencontre en cours avec vue OpenStreetMap des continents 41](#_Toc512794150)

[Figure 19: Détails de la rencontre terminée avec vue satellite ESRI 43](#_Toc512794151)

[Figure 20: Informations reliées au stade avec vue satellite ESRI 44](#_Toc512794152)

[Figure 21: Informations reliées au stade avec fond de carte OpenStreetMap 44](#_Toc512794153)

Tableaux

[Tableau 1: Spécification des serveurs 6](#_Toc512794154)

[Tableau 2: Couche des pays 11](#_Toc512794155)

[Tableau 3: Couche des fuseaux horaires 11](#_Toc512794156)

[Tableau 4: requête SPARQL pour wikidata.org 12](#_Toc512794157)

[Tableau 5: "INSERT INTO venue" query 12](#_Toc512794158)

[Tableau 6: Couche des stades 14](#_Toc512794159)

[Tableau 7: Couche des rencontres en cours 17](#_Toc512794160)

[Tableau 8: Couche des rencontres terminées 17](#_Toc512794161)

[Tableau 9: Description de la base de données Vincent 18](#_Toc512794162)

[Tableau 10: styles de l'accordeon de jquery-ui 36](#_Toc512794163)

# 

# Sommaire

* Voici le lien de l'application Web cartographique :

<http://igeomedia.com/~vincent/live/projet/index.html>

Il est préférable d'y aller durant les fins de semaines, entre 10 heures du matin et 16 heures (heure de l'EST).

Les matins, soirs et nuits de semaine il n'y a pas beaucoup de rencontres dont les stades (et leurs coordonnées) sont répertoriés dans Wikidata.

* Voici le lien de l'application mobile « LiveSoccerWorldMapApp-debug.apk »:

<https://www.dropbox.com/sh/ezolrjpy4xhbvj4/AADC93YYGE0fsOEmGgx6obnKa?dl=0>

* Voici l’emplacement des fichiers nécessaires au bon fonctionnement de l'applications Web sur le serveur « igeomedia.com » :
  + Les fichiers mapfile : /home/vincent/projet/mapfile;
  + Les couches shapefile : /home/vincent/projet/data;
  + Répertoire où est copiée la base de données et le script de restauration: /home/vincent/projet/restore;
  + Répertoire des fichiers Web : /home/vincent/public\_html/live/projet;
  + Répertoire des fichiers pour PhoneGap: /home/vincent/projet/phonegap.

# Étape 1

*« Définir/rechercher les couches vectorielles pour la thématique du projet. La thématique devra exclure les données de la Ville de Sherbrooke et la Ville de Montréal. »*

## Définir un thème d’étude

Le thème de ce projet est la présentation d’évènements sportifs en cours ou récemment terminés sur une carte du monde à des fins pédagogiques. L’évènement sportif qui a été sélectionné pour ce prototype est le football européen en raison de sa grande popularité, le grand nombre de données disponibles et la fréquence élevée des rencontres. Le terme « soccer » sera utilisé dans ce document car c’est ainsi qu’il est désigné en Amérique du nord.

Une rencontre de soccer peut être localisée grâce au stade dans lequel la partie se déroule. Chaque stade peut être localisé sur une carte si ses coordonnées sont connues.

Les rencontres de soccer qui seront représentées seront celles en cours et celles qui viennent de se terminer. Elles seront symbolisées par une épinglette/pastille de couleur pour les différencier : vert pour les rencontres de soccer en cours, rouge pour les rencontres de soccer terminées.

Les stades de soccer peuvent être représenté sur la carte peu importe s’il y a une rencontre en cours ou non grâce à une épinglette de couleur blanche.

## Architecture

### Composants



Figure 1: Éléments de la solution

Les composants faisant partie de la solution sont les suivant :

* Les sources de données « wikidata.org », « RSS ScoresPro.com » et « NaturalEarthData.com »;
* Le serveur de traitement des données Vincent@HOME;
* Le serveur hôte de la solution publique igeomedia.com;
* L’utilisateur du site via son ordinateur ou son appareil mobile.

Les serveurs de traitement et hôte possède les spécifications suivantes :

Tableau 1: Spécification des serveurs

|  |  |  |
| --- | --- | --- |
|  | Vincent@HOME | igeomedia.com |
| Système | Windows 2010 x64 Intel Core 2 Duo CPU [E8500@3.16Ghz](mailto:E8500@3.16Ghz) | Linux debian 2.6.32-5-amd64 x86\_64 GNU/Linux |
| Serveur Web | Apache/2.4.29 (Win32) | Apache/2.2.16 (Debian) |
| Serveur géographique | MapServer version 7.0.7 (MS4W 3.2.5)  OUTPUT=PNG  OUTPUT=JPEG  OUTPUT=KML  SUPPORTS=PROJ SUPPORTS=AGG SUPPORTS=FREETYPE SUPPORTS=CAIRO SUPPORTS=SVG\_SYMBOLS SUPPORTS=SVGCAIRO SUPPORTS=ICONV SUPPORTS=FRIBIDI SUPPORTS=WMS\_SERVER SUPPORTS=WMS\_CLIENT SUPPORTS=WFS\_SERVER SUPPORTS=WFS\_CLIENT SUPPORTS=WCS\_SERVER SUPPORTS=SOS\_SERVER SUPPORTS=FASTCGI SUPPORTS=THREADS SUPPORTS=GEOS SUPPORTS=POINT\_Z\_M INPUT=JPEG  INPUT=POSTGIS  INPUT=OGR  INPUT=GDAL  INPUT=SHAPEFILE | MapServer version 6.2.0 OUTPUT=GIF  OUTPUT=PNG  OUTPUT=JPEG  SUPPORTS=PROJ  SUPPORTS=GD  SUPPORTS=AGG SUPPORTS=FREETYPE SUPPORTS=ICONV SUPPORTS=WMS\_SERVER SUPPORTS=WFS\_SERVER SUPPORTS=GEOS  INPUT=JPEG  INPUT=POSTGIS  INPUT=OGR  INPUT=GDAL  INPUT=SHAPEFILE |
| Système de base de données | PostgreSQL 9.6  Extension postgis  Extension pg\_trgm  Extension unaccent | PostgreSQL 8.4.21  Extension postgis  Extension pg\_trgm |
| Base de données | Vincent  Encoding UTF8  Collation en\_CA.1252  Character Type en\_CA.1252 | Vincent  Encoding UTF8  Collation en\_CA.UTF-8  Character Type en\_CA.UTF-8 |

### Processus

Les activités décrites ci-dessous se déroulent à partir du serveur vincent@HOME.



Figure 2: Phase initiale

Les coordonnées des stades de soccer ont été récupérées de la base de données de wikidata via un script SPARQL; +4700 stades avec leurs coordonnées géographiques respectives en degré Lat/Lon (ESPG:4326), le pays hôte (« region »), leur équipe hôte (« team »), le nom de la compétition principale de l’équipe hôte (« league »), la ville hôte (« headquarter ») ont été récupérées via un script Python. (https://query.wikidata.org/)

Ces données ont été entreposées et structurées dans une base de donnée postgreSQL 9.6 avec l’extension postGIS par ce même script Python. La source de cette information est associée dans une table à cet effet (« dataprovider=wikidata.org »).



Figure 3: Phase continue

Un autre script Python roule continuellement pour collecter le fil RSS des rencontres de soccer en cours du site ScoresPro.com. (https://www.scorespro.com/rss2/live-soccer.xml). Il s’exécute au minimum toutes les minutes, sinon plus, selon le temps de traitement des données collectées (selon le nombre de rencontres).

Ce second script est chargé d’entreposer les informations de chaque rencontre, telle que les équipes, la compétition, le pays, et surtout le nom du stade, dans le même schéma de base données mais avec une source différente (« dataprovider=RSS scorespro.com »).

La mise en relation des deux sources de données provient de la paire « nom du stade/pays ». Un test de similarité est fait grâce à l’extension « pg\_trgm » disponible dans postgreSQL (version 8.4et supérieure) à cet effet. Le test de similarité ne fonctionne bien que si les accents et le texte est en minuscule. La fonction « unaccent() » pour enlever les accents d’un texte n’est disponible qu’à partir de la version postgreSQL 9.0. Le processus d’association des stades des deux sources de données est préparé sur le serveur vincent@HOME.

Trois vues principales ont été créée afin de rendre disponible les informations facilement : (1) une vue des stades qui sont géographiquement localisable sur une carte; (2) une vue des rencontres en cours; (3) une vue des rencontres terminées récemment.

A chaque fois que les données du second script ont été traitées, une copie de la base de données est faite et copiée sur le serveur « igeomedia.com » (via la commande « scp ») et une restauration est initié sur le serveur igeomedia.com (via la commande « ssh »). L’authentification SSH se fait automatiquement grâce à l’échange de clé primaire/privé.

## Rechercher au minimum 3 couches d’information (format shapefile idéalement) pour illustrer le thème d’étude

Les couches d’information qui ont été collectés sont les suivantes et sont décrites ci-dessous.

1. Une couche représentant les délimitations des pays pour créer un fond de carte personnalisé;
2. Une couche représentant les délimitations des fuseaux horaires;
3. Une couche représentant les stades de soccer;
4. Une couche représentant les rencontres de soccer en cours;
5. Une couche représentant les rencontres de soccer terminées.

### Carte des pays pour créer un fond de carte personnalisé;

La partie pédagogique de ce projet veut présenter une carte du monde destinée à un jeune publique tel que des enfants dans la tranche d’âge 7-12 ans. Chaque pays sera coloré avec une couleur assez vive dont la valeur est prise aléatoirement dans la gamme de la couleur désignée pour son continent respectif : Afrique = jaune; Asie = rose; Europe = rouge-orange; Amérique du nord = vert foncé; Amérique du sud = vert clair; Océanie et reste du monde = violet.

Le style de cette couche est conçu au format SLD, sera déployé sur le serveur « igeomedia.com » et référencé par le serveur MapServer.

La couche a été téléchargée du site [www.naturalearthdata.com](http://www.naturalearthdata.com), a été validée avec QGIS. Aucune modification n’a été apportée, ni aux système de référence (EPSG :4326) ni aux attributs descriptifs.

|  |  |
| --- | --- |
| Nom | Admin 0 – Countries; ne\_10m\_admin\_0\_countries.zip |
| Type | ESRI shapefile |
| Type de géométrie (WKB type) | Polygone |
| Source | <http://www.naturalearthdata.com> |
| Échelle | 1 : 10,000,000 / 1″ = 158 miles / 1 cm = 100 km |
| Étendue | xMin,yMin -180,-90 : xMax,yMax 180,83.6341 |
| CRS | EPSG :4326, WGS 84 |
| Licence | « All versions of Natural Earth raster + vector map data found on this website are in the public domain. You may use the maps in any manner, including modifying the content and design, electronic dissemination, and offset printing. The primary authors, Tom Patterson and Nathaniel Vaughn Kelso, and all other contributors renounce all financial claim to the maps and invites you to use them for personal, educational, and commercial purposes.  No permission is needed to use Natural Earth. Crediting the authors is unnecessary.» (www.naturalearthdata.com) |

Tableau 2: Couche des pays

### Couche représentant les fuseaux horaires

L’étendue de la carte étant le monde, il est intéressant d’afficher les délimitations des fuseaux horaires. La carte permet de voir dans quelle partie du monde les rencontres se déroulent mais aussi dans quel fuseau horaire. Généralement les rencontres de soccer se déroulent en milieu d’après-midi et en soirée chez les professionnels et durant les fins de semaines, bien qu’il y a désormais régulièrement des rencontres même en semaine.

La couche a été téléchargée du site [www.naturalearthdata.com](http://www.naturalearthdata.com), a été validée avec QGIS. Aucune modification n’a été apportée, ni aux système de référence (EPSG :4326) ni aux attributs descriptifs.

|  |  |
| --- | --- |
| Nom | Timezones; ne\_10m\_time\_zones.zip |
| Type | ESRI shapefile |
| Type de géométrie (WKB type) | Polygone |
| Source | <http://www.naturalearthdata.com> |
| Échelle | 1 : 10,000,000 / 1″ = 158 miles / 1 cm = 100 km |
| Étendue | xMin,yMin -180,-90 : xMax,yMax 180,90.0002 |
| CRS | EPSG:4326, WGS 84 |
| Licence | « All versions of Natural Earth raster + vector map data found on this website are in the public domain. You may use the maps in any manner, including modifying the content and design, electronic dissemination, and offset printing. The primary authors, Tom Patterson and Nathaniel Vaughn Kelso, and all other contributors renounce all financial claim to the maps and invites you to use them for personal, educational, and commercial purposes.  No permission is needed to use Natural Earth. Crediting the authors is unnecessary.  The Natural Earth time zones are in the public domain and supplementary attribution to International Mapping or the Central Intelligence Agency is optional. » (www.naturalearthdata.com) |

Tableau 3: Couche des fuseaux horaires

### Couche représentant les stades de soccer

Les stades de soccer peuvent être représentés sur la carte par un point grâce à leurs coordonnées géographiques.

À ce jour, le site « wikidata.org » référence plus de 7400 stades (« venue » en anglais). Plus de 4300 d’entre eux ont leurs coordonnées respectives en degré Longitude et Latitude (EPSG:4326). Il est aussi possible de récupérer le pays hôte, la ville hôte, l’équipe résidente et la compétition principale de cette équipe.

Le format de la requête supporté par Wikidata pour récupérer ces informations est du SPARQL et peut être exécuté à l’adresse <https://query.wikidata.org/>.

|  |
| --- |
| SELECT ?club ?clubLabel ?league ?leagueLabel ?venue ?venueLabel ?headquarters\_locationLabel ?headquarters\_location ?located\_in\_the\_administrative\_territorial\_entity ?located\_in\_the\_administrative\_territorial\_entityLabel ?coordinates ?country ?countryLabel WHERE {  ?club wdt:P31 wd:Q476028.  ?club wdt:P115 ?venue.  ?venue wdt:P625 ?coordinates.  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }  OPTIONAL { ?club wdt:P17 ?country. }  OPTIONAL { ?club wdt:P159 ?headquarters\_location. }  OPTIONAL { ?club wdt:P131 ?located\_in\_the\_administrative\_territorial\_entity. }  OPTIONAL { ?club wdt:P118 ?league. }  } |

Tableau 4: requête SPARQL pour wikidata.org

Plusieurs formats du résultat de la requête sont disponibles. Le format sélectionné est JSON car il est plus facile à traiter via les langages Python et javascript.

Les données du fichier JSON ont pu être lues et insérées dans une base de données relationnelles PostgreSQL 9.6 locale via un script Python. Les éléments enregistrés dans la base de données sont :

* La source de données (« wikidata.org »)
* Le sport (« soccer »)
* Le pays (« region »)
* Le stade incluant son libellé (nom), ses coordonnées et sa géométrie de type Point.
* Le nom de la ville hôte (« headquarter »)
* Le nom de l’équipe résidente
* La compétition principale à laquelle participe l’équipe résidente (« league »)

Les éléments sont mis en relation dans des tables d’association grâce à leurs identifiants uniques respectifs.

Voici la requête qui insère les données du stade dans la table “venue”.

|  |
| --- |
| INSERT INTO venue(label, noaccent, xlon\_deg, ylat\_deg, geom, entityid, url) VALUES(%s, ***lower(unaccent\_text(%s))***, %s, %s, **ST\_SetSRID(ST\_MakePoint(%s, %s), 4326)**, %s, %s) RETURNING id; |

Tableau 5: "INSERT INTO venue" query

L’extension « postgis » a été installée dans la base de données PostgreSQL 9.6 locale pour pouvoir insérer une colonne de type « Geometry » dans la table des stades (« venue »). Les coordonnées provenant du site Wikidata sont utilisée telles quelles pour créer la géométrie de type Point. Le code EPSG : 4326 est associé à la géométrie, la raison est discutée dans la prochaine section « Système de coordonnées ».

Les extensions « unaccent » et « pg\_trgm » ont aussi été installées dans la base de données PostgreSQL 9.6 locale afin de pouvoir utiliser convenablement la fonction de similarité (« similarity ») et pouvoir retrouver les coordonnées du stade avec un nom de stade provenant d’une autre source de données, et donc potentiellement écrit différemment mais similaire.

Pour tenter d’optimiser les performances lors de l’utilisation fréquente de la fonction de similarité, les éléments « nom du stade », « nom du pays », « nom de la ville », « nom de l’équipe » et « nom de la compétition », sont passés à la fonction « unaccent\_text() » et converti en minuscule (« lower() »), puis conservé dans une colonne « noaccent ». C’est cette valeur (celle de la colonne « noaccent ») qui sera utilisée par la fonction de similarité directement.

Un exemple d’utilisation de la fonction de similarité est présenté ci-dessous:



Figure 4: Exemple avec Similarity()

Finalement la vue « v\_venue\_ext » est responsable de retourner la liste des stades qui peuvent être représentés sur une carte avec les informations suivantes:

* Le nom du stade
* Un identifiant unique
* La géométrie de type Point
* Le pays hôte
* L’équipe résidente
* Le nom de la compétition principale de l’équipe résidente
* Le type de sport

|  |  |
| --- | --- |
| Nom | v\_venue\_ext |
| Type | Vue SQL |
| Type de géométrie (WKB type) | Point |
| Source | <http://www.wikidata.org> |
| Échelle | - |
| Étendue | xMin,yMin -180,-90 : xMax,yMax 180,90 |
| CRS | EPSG :4326, WGS 84 |
| Licence | « Text is available under the Creative Commons Attribution-ShareAlike License.» (https://wikimediafoundation.org/wiki/Terms\_of\_Use/en) |

Tableau 6: Couche des stades

### Couche représentant les rencontres de soccer en cours

Afin de pouvoir afficher sur la carte les stades qui sont actifs, le flux RSS <https://www.scorespro.com/rss2/live-soccer.xml> du site [www.scorespro.com](http://www.scorespro.com) a été sélectionné. Le flux est gratuit et ne nécessite aucun enregistrement. Il délivre les rencontres de soccer en cours ou celles qui viennent de se terminer. Les informations qu’il délivre sont les suivantes :

* Un maximum de 20 rencontres sont délivrées à la fois par le flux RSS. Le flux est mis à jour avec différentes rencontres s’il y en a plus de 20 en cours et si les résultats évoluent;
* Il a été observé que le flux est rafraichi assez rapidement, selon l’évolution des rencontres de soccer en cours;
* Chaque élément du flux représente une rencontre et contient :
  + Une description qui décrit les équipes, le score et un état de la rencontre;
  + La date de publication;
  + Un lien vers la page HTML de la rencontre en cours et hébergé par le site scorespro.com.

Voici un exemple de flux RSS, avec la partie surlignée les éléments de la rencontre :

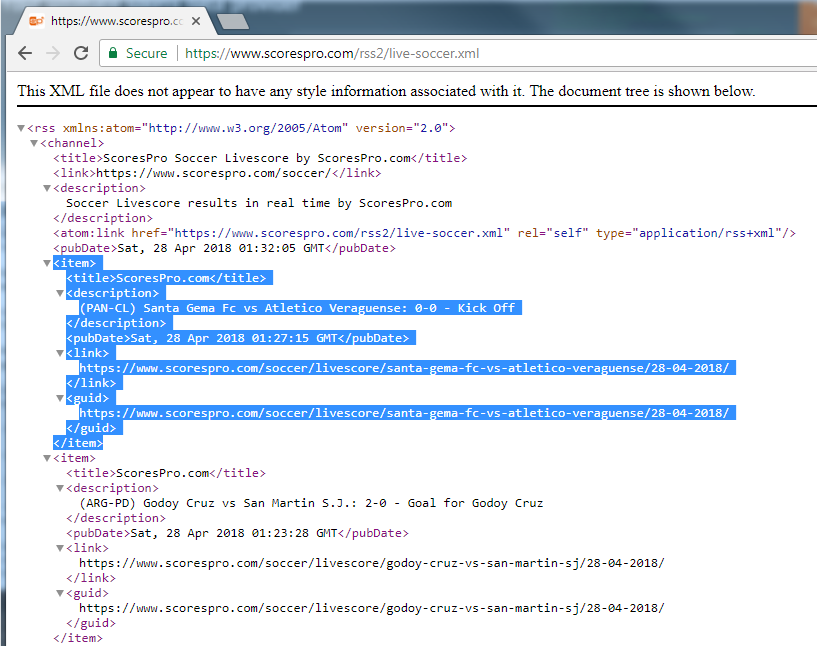


Figure 5: Flux RSS

Un script Python est exécuté toutes les minutes afin de récupérer les informations de chacune des rencontres délivrées par le flux RSS, et les insérer dans la base de donnée.

Afin de récupérer les informations minimales telles que le nom du stade, le pays, le nom des deux équipes et la compétition, le script lit ces informations directement dans la page HTML de la rencontre grâce au lien fournit dans le flux RSS de l’élément « guid ».



Figure 6: Page HTML de la rencontre

Le script insère les informations collectée dans le même schéma de la base de données, les mêmes tables, via le même module Python, mais avec comme source de données (« datasource ») « RSS scorespro.com ». La table « fixture » (rencontre) contient les détails de la rencontre tels que la date et l’heure, le statut (en cours, terminé), incluant le lien HTML de la page qui sera affichée dans la page Web au moment où une rencontre sera sélectionnée. Les équipes, la compétition, le stade et le pays sont mis en relation avec cette rencontre via les tables de relation respectives. Les coordonnées du stade n’étant pas précisées par le flux RSS, elles sont insérées avec la valeur 999.99999 dans la table « venue ».

Une fois insérée, les rencontres actives sont misent à jour dans la table « active\_fixture\_venue » selon les conditions suivantes : (1) elles peuvent être représentées avec succès sur une carte grâce à des coordonnées valident (différentes de 999.99999); (2) le pourcentage de similitude du nom du stade et du pays entre les deux sources de données « wikidata.org » et « RSS ScoresPro.com » est assez élevé (> 49%).

Finalement, la vue « v\_in\_progress\_fixture\_venue » est responsable de retourner la liste des rencontres en cours qui peuvent être représentés sur une carte avec les informations suivantes:

* Le nom du stade
* Un identifiant unique
* La géométrie de type Point
* Le pays hôte
* L’équipe résidente
* L’équipe visiteuse
* Le nom de la compétition
* Le type de sport
* Le lien de la page HTML qui affiche les détails de la rencontre en cours

|  |  |
| --- | --- |
| Nom | v\_in\_progress\_fixture\_venue |
| Type | Vue SQL |
| Type de géométrie (WKB type) | Point |
| Source | http://www.scorespro.com |
| Échelle | - |
| Étendue | xMin,yMin -180,-90 : xMax,yMax 180,90 |
| CRS | EPSG :4326, WGS 84 |
| Licence | Les flux RSS sont généralement gratuits. Aucune mention n’indique que les informations sont dans le domaine publique ou privée. (https://www.scorespro.com/free-livescore/) |

Tableau 7: Couche des rencontres en cours

### Couche représentant les rencontres de soccer terminées

Cette couche hérite du même processus décrit dans la section pour la couche des rencontres en cours.

La vue « v\_finished\_fixture\_venue » est responsable de retourner la liste des rencontres terminées qui peuvent être représentées sur une carte avec les informations suivantes:

* Le nom du stade;
* Un identifiant unique;
* La géométrie de type Point;
* Le pays hôte;
* L’équipe résidente;
* L’équipe visiteuse;
* Le nom de la compétition;
* Le type de sport;
* Le lien de la page HTML qui affiche les détails de la rencontre terminées.

|  |  |
| --- | --- |
| Nom | v\_finished\_fixture\_venue |
| Type | Vue SQL |
| Type de géométrie (WKB type) | Point |
| Source | http://www.scorespro.com |
| Échelle | - |
| Étendue | xMin,yMin -180,-90 : xMax,yMax 180,90 |
| CRS | EPSG :4326, WGS 84 |
| Licence | Les flux RSS sont généralement gratuits. Aucune mention n’indique que les informations sont dans le domaine publique ou privée. (https://www.scorespro.com/free-livescore/) |

Tableau 8: Couche des rencontres terminées

## Définir la zone géographique (étendue géographique et le système de coordonnées (code EPSG))

### Étendue géographique

L’étendue géographique de la carte est l’étendue de la terre en degré de longitude et de latitude. Le but du projet est d’afficher la localisation des stades de soccer existant un peu partout sur la planète et référencés dans wikidata.org.

|  |  |  |  |
| --- | --- | --- | --- |
| xMin (longitude) | yMin (latitude) | xMax(longitude) | yMax (latitude) |
| -180 | -90 | 180 | 90 |

### Système de coordonnées

Les polygones des fichiers shapefile d’ESRI et les coordonnées de Wikidata sont basés sur le WGS 84, EPSG:4326[[1]](#footnote-1), un système de coordonnées **géographique**. Les coordonnées sont exprimées en **degré** de longitude et de latitude. Les outils cartographiques Web (tel que « OpenStreetMap » et « Google Map ») supportent les systèmes de coordonnée de référence (CRS) EPSG:4326 et EPSG:3857[[2]](#footnote-2). Le système EPSG:3857 est un système de coordonnées **projeté**, celui du Web Mercator. L’une des différences majeures entre les deux sont les unités : le EPSG:3857 nécessite d’avoir des coordonnées en unité de **mètre**. Comme il n’y a pas de besoin dans notre projet de mesurer des distances, l’utilisation du EPSG:4326 est correcte et nous facilite la tâche car aucune conversion n’est nécessaire.

# Étape 2

« *Configurer le serveur cartographique en mode WMS/WFS pour chaque couche avec un style d’affichage prédéfini.* »

## Créer une base de données à référence spatiale contenant vos données thématiques

La base de données à référence spatiale « vincent » a été créé sur le site « igeomedia.com » par l’administrateur de la base de donnée et contient les informations suivantes:

Tableau 9: Description de la base de données Vincent

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Nom | Description | Source des données |
| Table | Dataprovider | La source de données « wikidata.org » ou « RSS ScoresPro.com ». | « wikidata.org » et « RSS ScoresPro.com » |
| Table | Sport | Le sport « soccer ». | « wikidata.org » et « RSS ScoresPro.com » |
| Table | Region | Le pays. | « wikidata.org » et « RSS ScoresPro.com » |
| Table | Venue | Le stade incluant son libellé (nom), ses coordonnées et sa géométrie de type Point. | « wikidata.org » et « RSS ScoresPro.com » |
| Table | Headquarter | Le nom de la ville hôte. | « wikidata.org » et « RSS ScoresPro.com » |
| Table | Team | Le nom de l’équipe résidente. | « wikidata.org » et « RSS ScoresPro.com » |
| Table | Competition | La compétition principale à laquelle participe l’équipe résidente. | « wikidata.org » et « RSS ScoresPro.com » |
| Table | Fixture | Les rencontres actives. | « RSS ScoresPro.com » |
| Table | Fixture\_status | Les statuts des rencontres « en cours » et « terminée » | « RSS ScoresPro.com » |
| Table | Fixture\_archive | Les rencontres des 24 dernières heures. | « RSS ScoresPro.com » |
| Table | Active\_fixture\_venue | Les rencontres activent qui peuvent être représentées sur une carte. | « wikidata.org » |
| Vue | V\_active\_fixture\_venue\_init | Association des rencontres activent qui peuvent être représentées sur une carte avec les autres éléments (stade, pays, sport, équipe, compétition). | « wikidata.org » |
| Vue | V\_active\_fixture\_venue | Résultat final et filtré des rencontres activent avec leurs éléments respectifs, sans doublon de stade (id), potentiellement dû à des noms de stades similaires dans le même pays. Le résultat de cette vue remplacera le contenu de la table « active\_fixture\_venue » après chaque lecture du flux RSS. | « wikidata.org » |
| Vue | V\_venue | Vue simplifiée et filtrée de la table « Venue », excluant les stades sans référence spatiale (coordonnées 999.99999). |  |
| Vue | V\_venue\_ext | Association des stades qui peuvent être représentés sur une carte avec les autres éléments respectifs (sport, pays, équipe, compétition). | « wikidata.org » |
| Vue | V\_in\_progress\_fixture\_venue | Retourne les éléments de la table « active\_fixture\_venue » qui ont le statut « en cours ». | « wikidata.org » |
| Vue | V\_finished\_fixture\_venue | Retourne les éléments de la table « active\_fixture\_venue » qui ont le statut « terminé ». | « wikidata.org » |

Les tables sont mises en relation grâce à des tables d’association.

## Configurer un serveur cartographique en mode WMS et en mode WFS

Le serveur cartographique hébergé par le site igeomedia.com est utilisé pour contenir les fichiers shapefile ESRI et configurer les couches.

Les fichiers shapefile ESRI résident dans le répertoire « /home/vincent/projet/data » du serveur.

Les fichiers « .map » sont configurés dans le répertoire « /home/vincent/projet/mapfile ».

Le fichier « .map » principal « principal.map » contient :

* Le nom de la carte « Live\_Soccer\_World\_Map »;
* Le type d’image PNG;
* Les symboles référencés par les couches (les pastilles verte, rouge et blanche) :
  + Pastille verte:

« /home/vincent/public\_html/live/projet/icons/green\_light.png »;

* + Pastille rouge:

« /home/vincent/public\_html/live/projet/icons/red\_light.png »;

* + Pastille blanche:

« /home/vincent/public\_html/live/projet/icons/capital.png »;

* La projection de la carte (« EPSG :4326 »);
* L’étendue (« EXTENT -179 -89 179 89 »);
* L’URL pour les services WFS/WMS

<http://igeomedia.com/cgi-bin/mapserv?map=/home/vincent/projet/mapfile/principal.map&>;

* Les références aux couches de la carte :
  + INCLUDE "ne\_10m\_admin\_0\_countries.map";
  + INCLUDE "ne\_10m\_time\_zones.map";
  + INCLUDE "venue\_ext.map";
  + INCLUDE "fixture\_in\_progress.map";
  + INCLUDE "fixture\_finished.map".

|  |
| --- |
| MAP  NAME "carte\_principale"  CONFIG "MS\_ERRORFILE" "/tmp/vincent\_ms\_error.txt"  STATUS on  SIZE 256 256  IMAGETYPE png  OUTPUTFORMAT  NAME "png"  DRIVER AGG/PNG  MIMETYPE "image/png"  IMAGEMODE RGB  EXTENSION "png"  FORMATOPTION "GAMMA=0.75"  END  SYMBOL  NAME 'red\_light'  TYPE PIXMAP  IMAGE '/home/vincent/public\_html/live/projet/icons/red\_light.png'  TRANSPARENT 0  END  SYMBOL  NAME 'green\_light'  TYPE PIXMAP  IMAGE '/home/vincent/public\_html/live/projet/icons/green\_light.png'  TRANSPARENT 0  END  SYMBOL  NAME 'capital'  TYPE PIXMAP  IMAGE '/home/vincent/public\_html/live/projet/icons/capital.png'  TRANSPARENT 0  END  PROJECTION  "init=epsg:4326"  END  # EXTENT [minx] [miny] [maxx] [maxy]  EXTENT -179 -89 179 89  WEB  IMAGEPATH "/var/www/tmp/"  IMAGEURL "/tmp/"  METADATA  "wfs\_title" "WFS Soccer Venues"  "wfs\_onlineresource" "http://igeomedia.com/cgi-bin/mapserv?map=/home/vincent/projet/mapfile/principal.map&"  "wfs\_srs" "EPSG:4326 EPSG:3857"  "wms\_title" "WMS Soccer Venues"  "wms\_onlineresource" "http://igeomedia.com/cgi-bin/mapserv?map=/home/vincent/projet/mapfile/principal.map&"  "wms\_srs" "EPSG:4326 EPSG:3857"  "ows\_enable\_request" "\*"  "wms\_enable\_request" "\*"  END  END  INCLUDE "ne\_10m\_admin\_0\_countries.map"  INCLUDE "ne\_10m\_time\_zones.map"  INCLUDE "fixture\_in\_progress.map"  INCLUDE "fixture\_finished.map"  # INCLUDE "active\_fixture\_venue.map"  INCLUDE "venue\_ext.map"  END |

## Définir un style d’affichage par défaut adapté aux couches

Chaque couche a sa propre définition dans son propre fichier « .map ». Ils sont tous dans le répertoire « /home/vincent/projet/mapfile » sur le serveur igeomedia.com.

La projection de chacune des couches est identique : « EPSG :4326 ». Ainsi que l’étendue : « -179 -89 179 89 ».

Chaque couche peut être interrogée à l’exception de celle des fuseaux horaires. Certains paramètres sont requis pour l’interrogation WMS « GetFeatureInfo », tel que la définition des modèles « Header.html », « Template.html » et « Footer.html », ainsi que les metadonnées « gml\_include\_items », « gml\_exclude\_items », « wms\_enable\_request », « wms\_include\_items », « gml\_geometries », « gml\_geom\_type ».

### Couche des pays – ne\_10m\_admin\_0\_countries.map

La couche des pays est définie dans le fichier « ne\_10m\_admin\_0\_countries.map ».

La source de la couche est le fichier shapefile ESRI "ne\_10m\_admin\_0\_countries".

La géométrie de cette couche est de type « Polygon ».

Le style désiré de cette couche était destiné à présenter une carte du monde à un jeune publique tel que des enfants dans la tranche d’âge 7-12 ans. Chaque pays serait coloré avec une couleur assez vive dont la valeur est prise aléatoirement dans la gamme de la couleur désignée pour son continent respectif : Afrique = jaune; Asie = rose; Europe = rouge-orange; Amérique du nord = vert foncé; Amérique du sud = vert clair; Océanie et reste du monde = violet.

Le style de cette couche serait conçu au format SLD, déployé sur le serveur « igeomedia.com » et référencé par le serveur « MapServer ».

Par manque de temps, le style désiré n’a pas pu être accomplie, la fonctionnalité de base du site était la priorité. Seul un style très simple a été appliqué pour les contours des polygones : trait noir, épaisseur de 1 px, et aucune couleur de remplissage.

|  |
| --- |
| LAYER  NAME "ne\_10m\_admin\_0\_countries"  TYPE polygon  DATA "/home/vincent/projet/data/naturalearthdata/ne\_10m\_admin\_0\_countries/ne\_10m\_admin\_0\_countries"  STATUS on  DUMP TRUE  HEADER 'head.html'  TEMPLATE 'template.html'  FOOTER 'foot.html'  METADATA  "wfs\_title" "ne\_10m\_admin\_0\_countries"  "wfs\_srs" "EPSG:4326"  "wms\_title" "ne\_10m\_admin\_0\_countries"  "wms\_srs" "EPSG:4326"  "gml\_include\_items" "all"  "gml\_exclude\_items" ""  "wms\_enable\_request" "\*"  "wms\_include\_items" "all"  "gml\_geometries" "geom"  "gml\_geom\_type" "polygon"  END  PROJECTION  "init=epsg:4326"  END  EXTENT -179 -89 179 89  CLASS  NAME "ne\_10m\_admin\_0\_countries"  STYLE  WIDTH 1  OUTLINECOLOR 0 0 0  #COLOR 106 255 25  END  END  END |

### Couche des fuseaux horaires –ne\_10m\_time\_zones.map

La couche des fuseaux horaires est définie dans le fichier ne\_10m\_time\_zones.map.

La source de la couche est le fichier shapefile ESRI "ne\_10m\_time\_zones".

La géométrie de cette couche est de type « Polygon ».

Seul les lignes des fuseaux horaires sont désirées, pas de remplissage des polygones. Un style très simple a été appliqué : trait noir, épaisseur de 1 px, et aucune couleur de remplissage.

|  |
| --- |
| LAYER  NAME "ne\_10m\_time\_zones"  TYPE polygon  DATA "/home/vincent/projet/data/naturalearthdata/ne\_10m\_time\_zones/ne\_10m\_time\_zones"  STATUS on  DUMP TRUE  EXTENT -179 -89 179 89  HEADER 'head.html'  TEMPLATE 'template.html'  FOOTER 'foot.html'  METADATA  "wfs\_title" "ne\_10m\_time\_zones"  "wfs\_srs" "EPSG:4326"  "wms\_title" "ne\_10m\_time\_zones"  "wms\_srs" "EPSG:4326"  "gml\_include\_items" "all"  "gml\_exclude\_items" ""  "wms\_enable\_request" "\*"  "wms\_include\_items" "all"  "gml\_geometries" "geom"  "gml\_geom\_type" "polygon"  END  PROJECTION  "init=epsg:4326"  END  CLASS  NAME "ne\_10m\_time\_zones"  STYLE  WIDTH 1  OUTLINECOLOR 0 0 0  # COLOR 106 255 25  END  END  END |

### Couche des stades – venue\_ext.map

La couche des stades est définie dans le fichier « venue\_ext.map ».

La source de la couche est la vue « v\_venue\_ext » de la base de données « vincent » du serveur « igeomedia.com ». Il est important d’indiquer « using srid=4326 » dans la ligne DATA car la base de données postgreSQL 8.4 du serveur « igeomedia.com » ne le retourne pas (le srid est absent de la table « geometry\_columns »). Il est aussi important de préciser la colonne qui contient l’identifiant unique de la géométrie (« using unique oid ») au cas où le nom de l’identifiant de la vue n’est pas « oid ».

La géométrie de cette couche est de type « Point ».

L’épinglette des stades est représentée par le symbole « capital » référencé dans l’objet MAP du fichier « principal.map ».

|  |
| --- |
| LAYER  NAME "venue\_ext"  TYPE point  CONNECTIONTYPE POSTGIS  CONNECTION "host=igeomedia.com dbname=vincent user=vincent password=vincent$2018 port=5432"  DATA "geom from v\_venue\_ext using unique oid using srid=4326"  STATUS on  DUMP TRUE  HEADER 'head.html'  TEMPLATE 'template.html'  FOOTER 'foot.html'  METADATA  "wfs\_title" "venue\_ext"  "wfs\_srs" "EPSG:4326"  "wms\_title" "venue\_ext"  "wms\_srs" "EPSG:4326"  "gml\_include\_items" "all"  "gml\_exclude\_items" ""  "wms\_enable\_request" "\*"  "wms\_include\_items" "all"  "gml\_geometries" "geom"  "gml\_geom\_type" "polygon"  END  PROJECTION  "init=epsg:4326"  END  EXTENT -179 -89 179 89  CLASS  NAME "venue\_ext"  STYLE  SYMBOL "capital"  END  END  END |

### Couche des recontres en cours – fixture\_in\_progress.map

La couche des stades est définie dans le fichier « fixture\_in\_progress.map ».

La source de la couche est la vue « v\_in\_progress\_fixture » de la base de données « vincent » du serveur « igeomedia.com ». Il est important d’indiquer « using srid=4326 » dans la ligne DATA car la base de données postgreSQL 8.4 du serveur « igeomedia.com » ne le retourne pas (table « geometry\_columns »). Il est aussi important de préciser la colonne qui contient l’identifiant unique de la géométrie (« using unique oid ») au cas où le nom de cet identifiant n’est pas « oid ».

La géométrie de cette couche est de type « Point ».

L’épinglette des rencontres en cours est représentée par le symbole « green\_light » référencé dans l’objet MAP du fichier « principal.map ».

|  |
| --- |
| LAYER  NAME "fixture\_in\_progress"  TYPE point  CONNECTIONTYPE POSTGIS  CONNECTION "host=igeomedia.com dbname=vincent user=vincent password=vincent$2018 port=5432"  DATA "geom from v\_in\_progress\_fixture\_venue using unique oid using srid=4326"  STATUS on  DUMP TRUE  HEADER 'head.html'  TEMPLATE 'template.html'  FOOTER 'foot.html'  METADATA  "wfs\_title" "fixture\_in\_progress"  "wfs\_srs" "EPSG:4326"  "wms\_title" "fixture\_in\_progress"  "wfs\_srs" "EPSG:4326"  "gml\_include\_items" "all"  "gml\_exclude\_items" ""  "wms\_enable\_request" "\*"  "wms\_include\_items" "all"  "gml\_geometries" "geom"  "gml\_geom\_type" "polygon"  END  PROJECTION  "init=epsg:4326"  END  EXTENT -179 -89 179 89  CLASS  NAME "fixture\_in\_progress"  STYLE  SYMBOL "green\_light"  END  END  END |

### Couche des rencontres terminées – fixture\_finished.map

La couche des stades est définie dans le fichier « fixture\_finished.map ».

La source de la couche est la vue « v\_finished\_fixture » de la base de données « vincent » du serveur « igeomedia.com ». Il est important d’indiquer « using srid=4326 » dans la ligne DATA car la base de données postgreSQL 8.4 du serveur « igeomedia.com » ne le retourne pas (table « geometry\_columns »). Il est aussi important de préciser la colonne qui contient l’identifiant unique de la géométrie (« using unique oid ») au cas où le nom de cet identifiant n’est pas « oid ».

La géométrie de cette couche est de type « Point ».

L’épinglette des rencontres en cours est représentée par le symbole « red\_light »référencé dans l’objet MAP du fichier « principal.map ».

|  |
| --- |
| LAYER  NAME "fixture\_finished"  TYPE point  CONNECTIONTYPE POSTGIS  CONNECTION "host=igeomedia.com dbname=vincent user=vincent password=vincent$2018 port=5432"  DATA "geom from v\_finished\_fixture\_venue using unique oid using srid=4326"  STATUS on  DUMP TRUE  HEADER 'head.html'  TEMPLATE 'template.html'  FOOTER 'foot.html'  METADATA  "wfs\_title" "fixture\_finished"  "wfs\_srs" "EPSG:4326"  "wms\_title" "fixture\_finished"  "wfs\_srs" "EPSG:4326"  "gml\_include\_items" "all"  "gml\_exclude\_items" ""  "wms\_enable\_request" "\*"  "wms\_include\_items" "all"  "gml\_geometries" "geom"  "gml\_geom\_type" "polygon"  END  PROJECTION  "init=epsg:4326"  END  EXTENT -179 -89 179 89  CLASS  NAME "fixture\_finished"  STYLE  SYMBOL "red\_light"  END  END  END |

## Tester l’affichage (WMS et WFS) avec un logiciel cartographique

Durant la conception et l’analyse des données, des tests ont été fait avec le logiciel QGIS localement. Chaque couche a été ouverte au format shapefile ou vue SQL selon la source, et finalement en WMS et WFS lorsque déployé dans le serveur cartographique sur le serveur Vincent@HOME. Une fois le tout complété, les fichiers shapefile, mapfiles et la base de données ont été déployé sur le serveur « igeomedia.com » et de nouveaux tests de validation avec QGIS ont été effectué.

### QGIS avec les shapefile ESRI et vue SQL

L’ouverture des shapefile ESRI et des vues SQL depuis le serveur igeomedia.com s’effectue bien, sans surprise.

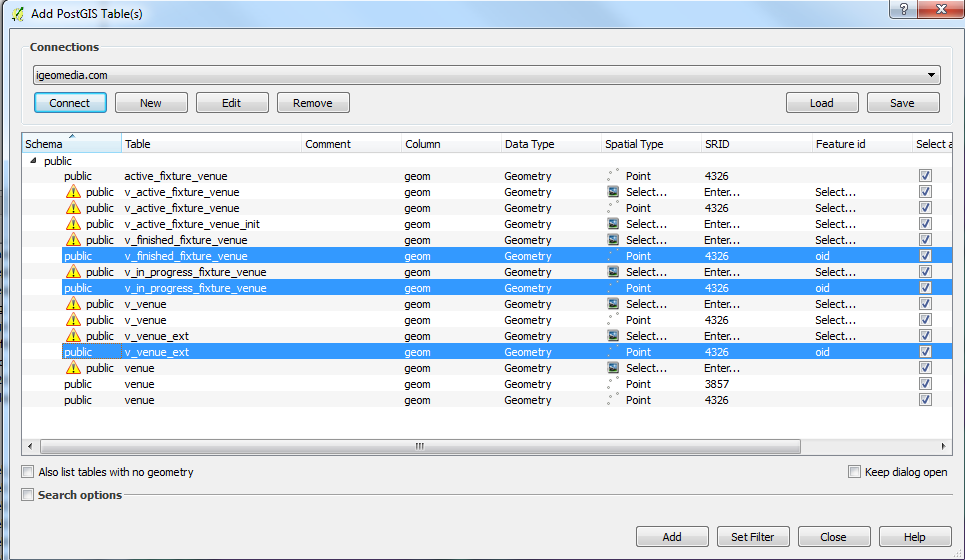


Figure 7: Sélection des vues SQL dans QGIS

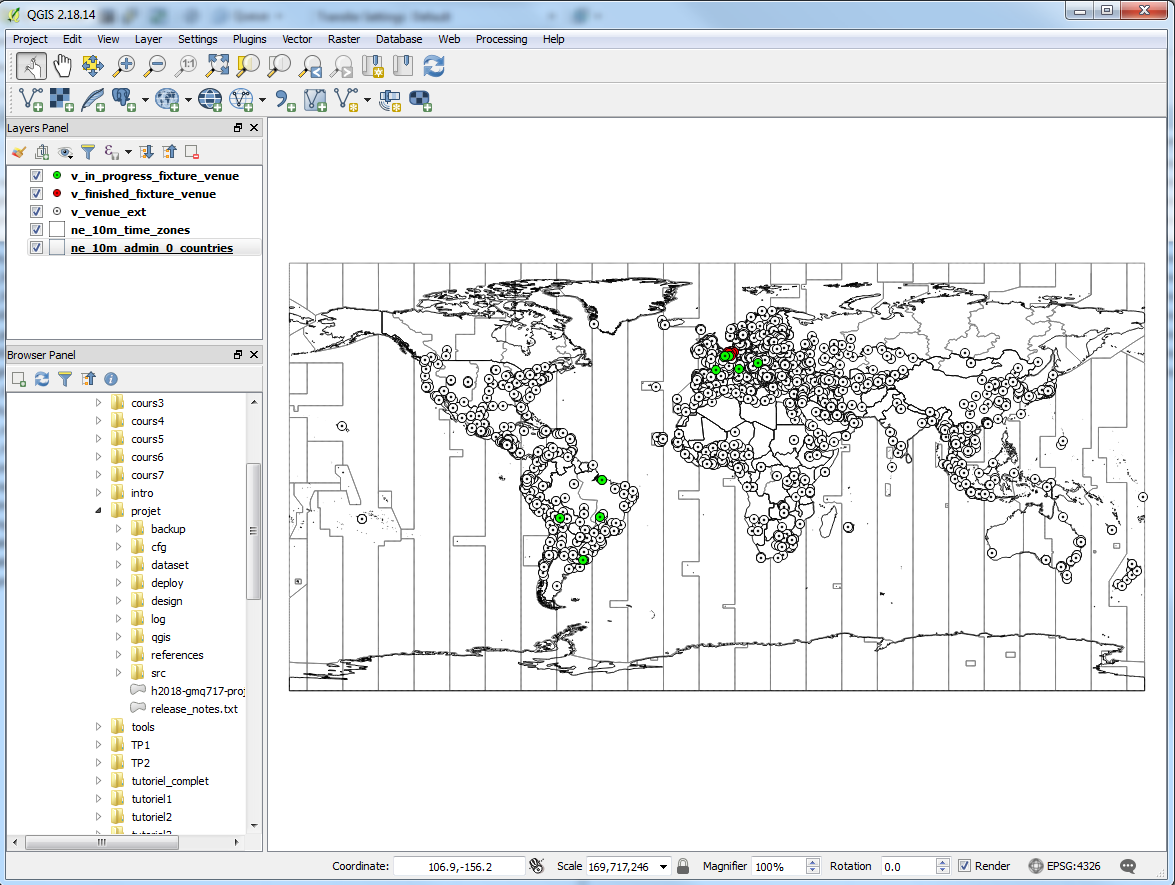


Figure 8: Affichage des fichiers shapefile ESRI et vues SQL dans QGIS

### QGIS avec les couches WMS

L’ouverture des couches via le service WMS fournie par le serveur « igeomedia.com » s’effectue bien, sans surprise.

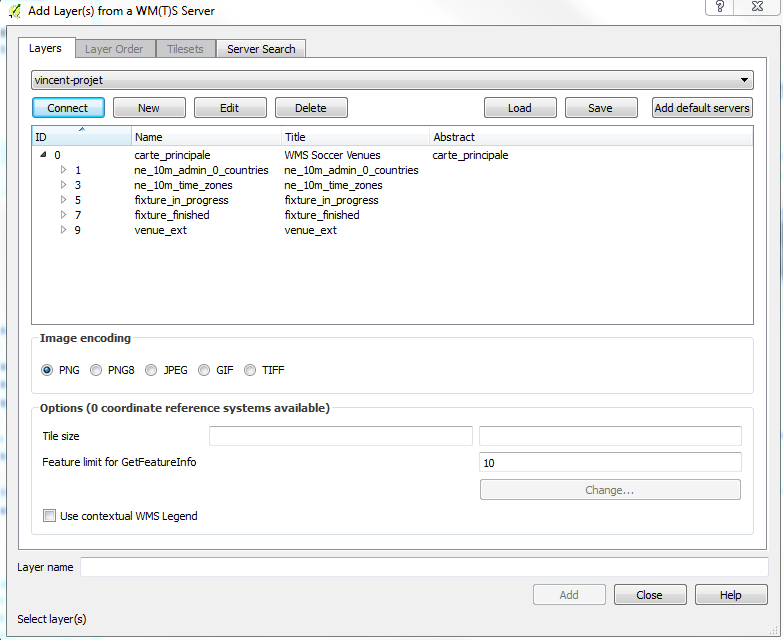


Figure 9: Sélection des couches du service WMS

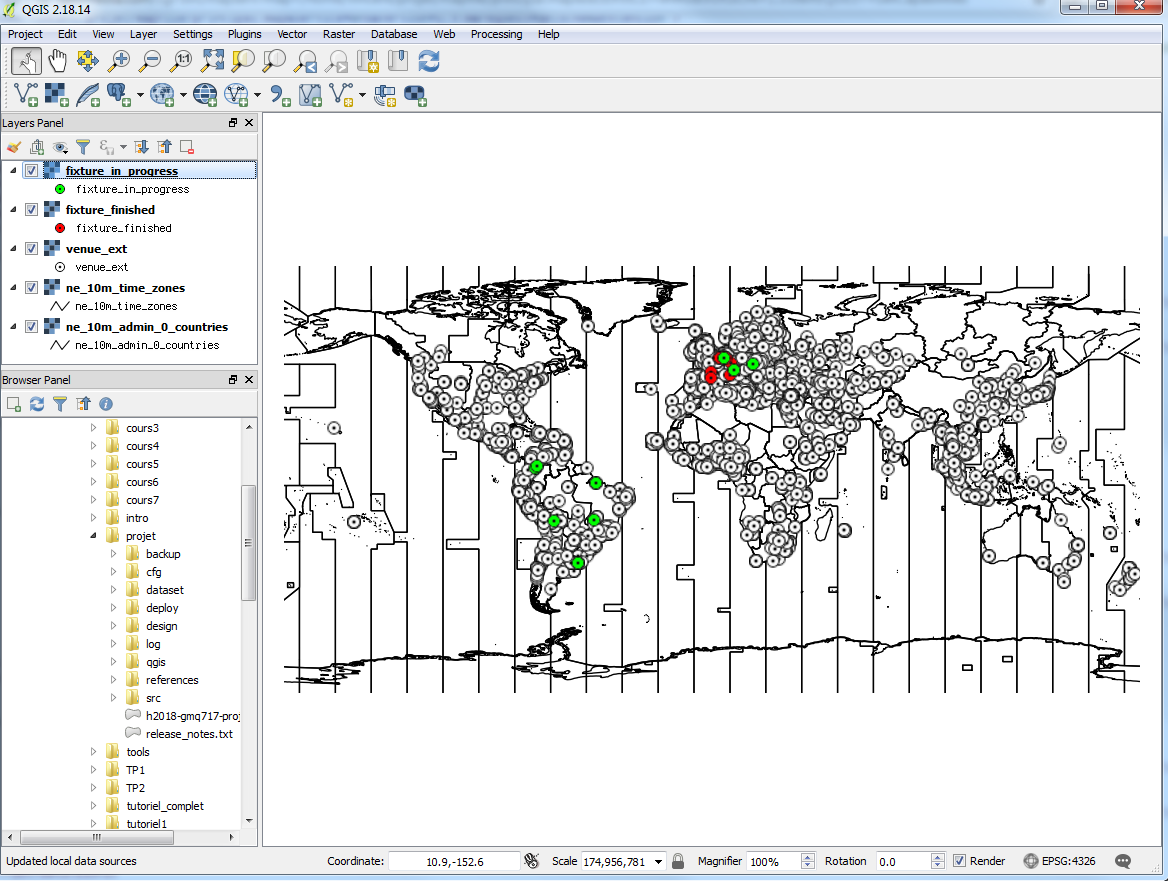


Figure 10: Affichage des couches WMS dans QGIS

### QGIS avec les couches WFS

L’ouverture des couches via le service WFS fournie par le serveur « igeomedia.com » s’effectue avec difficulté. Les couches des stades et des rencontres ne s’affichent pas, et les couches des fichiers ESRI s’affichent inversées et tournées à 90 degrés. Lorsque le CRS est modifié pour EPSG:3857, l’affichage des rencontres ne s’affichent toujours pas, mais les couches des fichiers ESRI s’affichent dans le bon sens, mais incomplète.

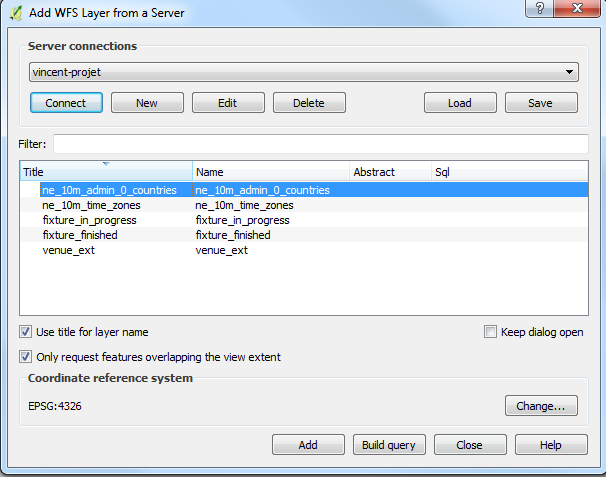


Figure 11: Sélection des couches WFS avec le CRS par défaut

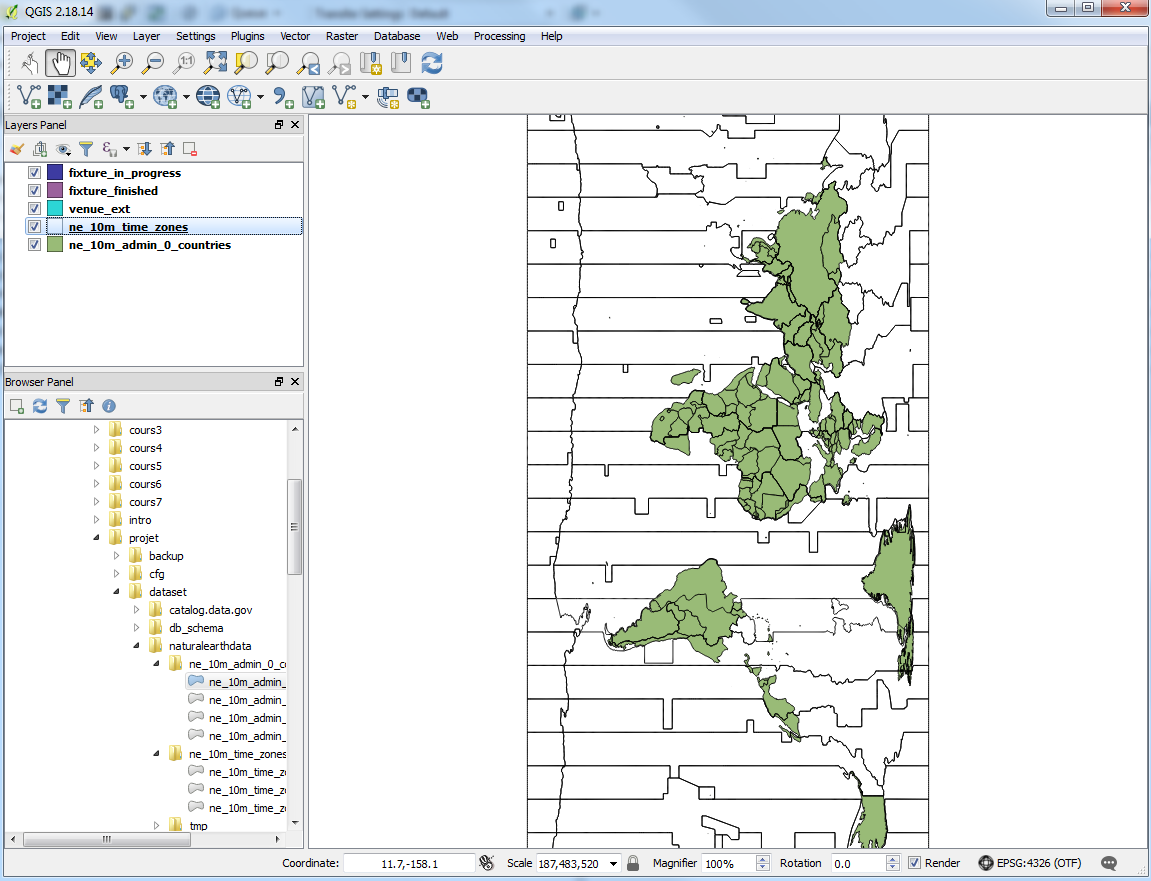


Figure 12: Affichage des couches WFS avec le CRS par défaut dans QGIS

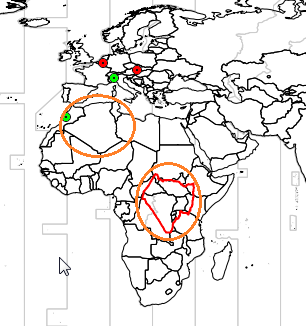


Figure 13: Sélection d'un pays et le résultat de la sélection en rouge

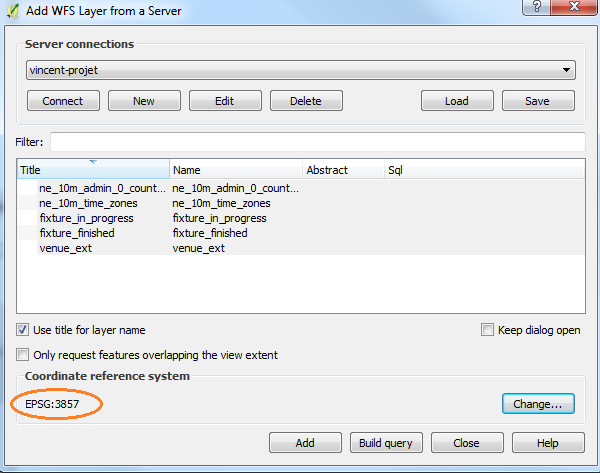


Figure 14: Sélection des couches WFS avec sélection du CRS EPSG:3857

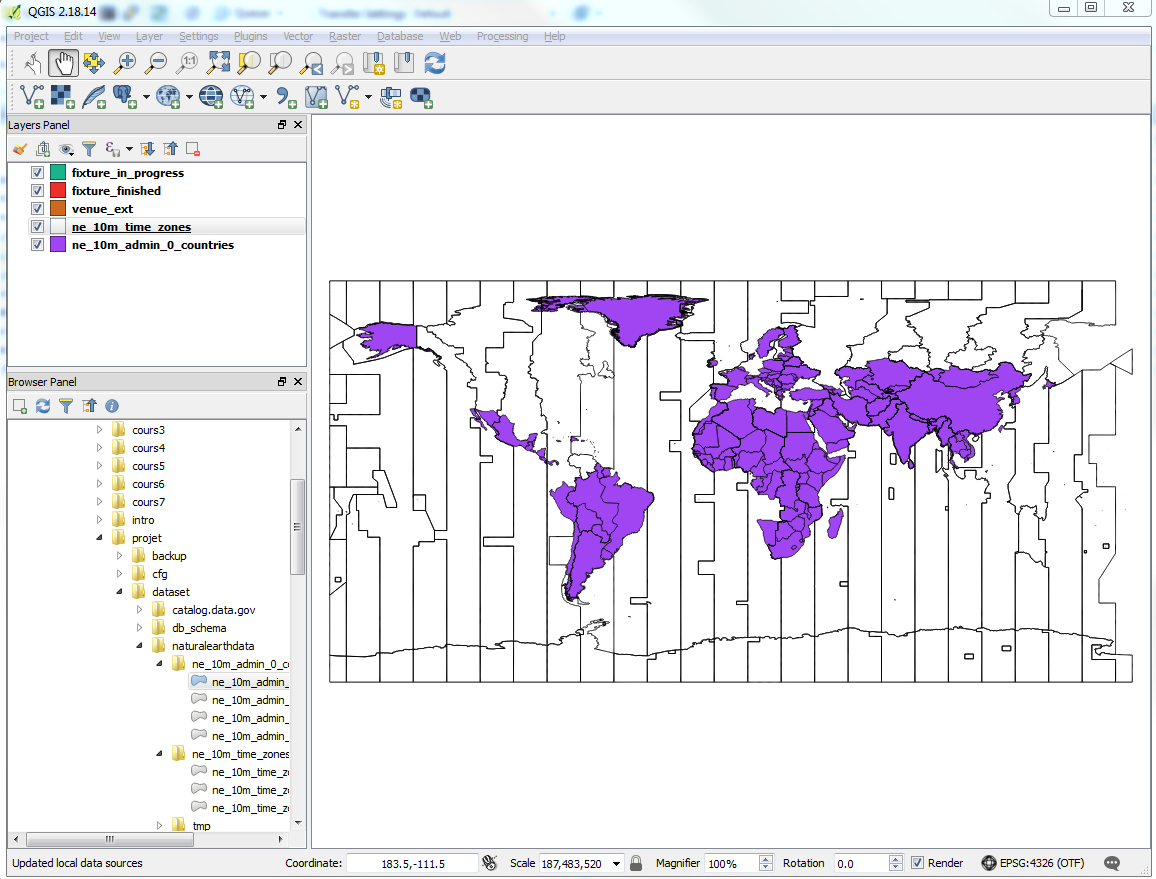


Figure 15: Affichage des couches WFS avec le CRS EPSG:3857 dans QGIS

# Étape 3

Développer une interface cartographique Web pour l’affichage des données.

## Un cadre de travail personnalisé et adaptatif

Le fichier « normalize.css » est ajouté pour normaliser les styles de base de la page peu importe le navigateur Web. Ce fichier a été récupéré de « github.com/necolas/normalize.css » et est disponible sous la licence « MIT License ».

Le fichier « index.html » et le style « style.css » a été adapté et personnalisé pour le besoin de la carte.

|  |
| --- |
| <body>  <div class="container-fluid">  <div class="row">  <div class="col-sm-12">  <img src="icons/header.png" >  </div>  </div>  <div class="row">  <div class="col-sm-8 col-sm-push-4">  <div id="map" ></div>  <div id="toolbars">  <div id="toolbar" >  ...  </div>  <div id="basemaps" >  ...  </div>  <div id="features" >  ...  </div>  </div>  </div>  <div class="col-sm-4 col-sm-pull-8">  <div id="accordion">  <span class="accordion-title" style="color:#fff">Live Score</span>  <div id="infoPanel">  ...  </div>  <span class="accordion-title" style="color:#fff">Légende</span>  <div id="legend">  ...  </div>  <span class="accordion-title" style="color:#fff">Directives</span>  <div id="directive">  ...  </div>  <span class="accordion-title" style="color:#fff">Crédits</span>  <div id="credits">  ...  </div>  </div>  </div>  </div>  </body> |

Figure 16: index.html de la carte

Un style par défaut de l’accordéon de jquery-ui a été récupéré sur le site de jquery-ui et adapté au besoin de la carte, précisément pour les couleurs, la largeur, la hauteur et les marges.

|  |
| --- |
| .ui-layout-center , /\* has content-div \*/  .ui-layout-west , /\* has Accordion \*/  .ui-layout-east , /\* has content-div ... \*/  .ui-layout-east .ui-layout-content { /\* content-div has Accordion \*/  padding: 0;  overflow: hidden;  }  #accordion \* {  -moz-box-sizing: border-box;  -webkit-box-sizing: border-box;  box-sizing: border-box;  }  #accordion {  width: 100%;  height:90%;  margin: auto;  }  #accordion .ui-accordion-header {  background-color: #ccc;  margin: 0px;  }  #accordion .ui-accordion-header .accordion-title {  color: #fff;  line-height: 42px;  display: block;  font-size: 12pt;  width: 100%;  text-indent: 10px;  text-shadow: 1px 1px 0px rgba(0,0,0,0.2);  border-right: 1px solid rgba(0, 0, 0, .2);  border-left: 1px solid rgba(0, 0, 0, .2);  border-bottom: 1px solid rgba(0, 0, 0, .2);  border-top: 1px solid rgba(250, 250, 250, .2);  }  #accordion .ui-accordion-content {  margin: 0px;  padding: 0px;  width: 100%;  background-color: #f3f3f3;  color: #777;  font-size: 10pt;  line-height: 16pt;  box-shadow: inset 0px 1px 1px 0px rgba(0, 0, 0, .2),  inset 0px -1px 0px 0px rgba(0, 0, 0, .4);  }  #accordion .ui-accordion-content > \* {  margin: 0px;  padding: 0px;  }  #accordion .ui-accordion-content a {  color: #777;  }  #accordion .ui-accordion-header {  background-color: #fa9300; /\* original \*/  /\* background-color: #14C7AE; \*/ /\* compound \*/  /\* background-color: #14C7AE; \*/ /\* complementary \*/  background-image: -moz-linear-gradient(top, #fa9300 0%, #dc621e 100%);  background-image: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#fa9300), color-stop(100%,#dc621e));  background-image: -webkit-linear-gradient(top, #fa9300 0%,#dc621e 100%);  background-image: -o-linear-gradient(top, #fa9300 0%,#dc621e 100%);  background-image: -ms-linear-gradient(top, #fa9300 0%,#dc621e 100%);  background-image: linear-gradient(to bottom, #fa9300 0%,#dc621e 100%);  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#fa9300', endColorstr='#dc621e',GradientType=0 );  }  #accordion .ui-accordion-header a {  box-shadow: 0px -1px 0px 0px #b85c0f;  } |

Tableau 10: styles de l'accordeon de jquery-ui

## L’accès à vos couches thématiques

La carte est initialisée dans le fichier « initCarte8.js ». Chaque couche est déclarée dans ce fichier et associé avec l’objet carte via une liste. Les couches de bases sont ajoutées à la liste en premier et celle des rencontres en cours en dernier afin de respecter l’ordre de superposition.

En plus des couches personnalisées décrites précédemment, la couche de base d’OpenStreetMap et celle d’ESRI Satellite sont initialisées.

La vue de la carte est initialisée avec une projection EPSG :4326, centrée volontairement sur l’Afrique, et avec un zoom de 2 qui permet de voir dès le départ l’ensemble des continents et les rencontres actuellement actives, en cours ou récemment terminées, et leur localisation respective.

Les couches qui sont actives au départ sont la couche de base d’OpenStreetMap, celle des fuseaux horaires, celles des rencontres en cours et terminées. Les autres couches existent mais ne sont pas visibles.

|  |
| --- |
| layer\_fixture\_finished = new ol.layer.Tile({source: new ol.source.TileWMS({  url: 'http://igeomedia.com/cgi-bin/mapserv?map=/home/vincent/projet/mapfile/principal.map',  params: {LAYERS: 'fixture\_finished', VERSION: '1.3.0', FORMAT: 'image/png'}  })  });  layer\_fixture\_in\_progress = new ol.layer.Tile({source: new ol.source.TileWMS({  url: 'http://igeomedia.com/cgi-bin/mapserv?map=/home/vincent/projet/mapfile/principal.map',  params: {LAYERS: 'fixture\_in\_progress', VERSION: '1.3.0',FORMAT: 'image/png'}  })  });  layer\_venue\_ext = new ol.layer.Tile({source: new ol.source.TileWMS({  url: 'http://igeomedia.com/cgi-bin/mapserv?map=/home/vincent/projet/mapfile/principal.map',  params: {LAYERS: 'venue\_ext', VERSION: '1.3.0', FORMAT: 'image/png'}  })  });  layerCountries = new ol.layer.Tile({source: new ol.source.TileWMS({  url: 'http://igeomedia.com/cgi-bin/mapserv?map=/home/vincent/projet/mapfile/principal.map',  params: {LAYERS: 'ne\_10m\_admin\_0\_countries', VERSION: '1.3.0', FORMAT: 'image/png'}  })  });  layerTimeZones = new ol.layer.Tile({source: new ol.source.TileWMS({  url: 'http://igeomedia.com/cgi-bin/mapserv?map=/home/vincent/projet/mapfile/principal.map',  params: {LAYERS: 'ne\_10m\_time\_zones', VERSION: '1.3.0', FORMAT: 'image/png'}  })  });  ...  layerCountries.setVisible(false);  layerTimeZones.setVisible(true);  layerTimeZones.setOpacity(0.2);  layer\_venue\_ext.setVisible(false);  layer\_fixture\_finished.setVisible(true);  layer\_fixture\_in\_progress.setVisible(true);  map = new ol.Map({  target: 'map',  view: new ol.View({  projection: 'EPSG:4326',  center: [15,8],  zoom: 2  }),  layers: [layerOSM, layerSat, layerCountries, layerTimeZones, layer\_venue\_ext, layer\_fixture\_finished, layer\_fixture\_in\_progress, highlightOverlay],  ... |

## Des outils de navigation

Les outils disponibles dans l’interface Web sont les suivant : 

Figure 17: Barre d'outils de l'interface Web

* L'épinglette de la barre d'outil pour afficher les rencontres en cours sur la carte;
* L'épinglette de la barre d'outil pour afficher les rencontres terminées sur la carte;
* L'épinglette de la barre d'outil pour afficher tous les stades dont les coordonnées sont connues par Wikipédia;
* L'icône  de la barre d'outil pour revenir à la position d'origine sur la carte et le fond de carte OpenStreetMap. Le mode Zoom est toujours actif, il n'est pas possible de déplacer la carte;
* L'icône  de la barre d'outil pour pouvoir déplacer la carte. Il est possible de cliquer sur les épinglettes;
* L'icône  de la barre d'outil pour pouvoir voir le stade de la rencontre en vue satellite en cliquant sur l'épinglette. Il n'est pas possible de déplacer la carte;
* L'icône  de la barre d'outil pour afficher le fond de carte d'OpenStreetMap;
* L'icône  de la barre d'outil pour afficher le fond de carte de type satellite d'ESRI;
* L'icône  de la barre d'outil pour afficher le fond de carte épurée;
* L'icône  de la barre d'outil pour afficher les zones des fuseaux horaires;
* Les boutons de control du zoom ;
* Le bouton d’agrandissement de la carte .

Les boutons de control de base de l’API « OpenLayer » sont ajoutés, tel que l’échelle, les coordonnées en degré de longitude et de latitude selon la position de la souris, et le bouton de plein écran.

Les boutons de control du Zoom et du déplacement de la carte sont aussi ajoutés.

|  |
| --- |
| controlZoomin = new ol.interaction.DragZoom();  controlZoomin.setActive(false);  controlPan = new ol.interaction.DragPan();  ...  Controls : ol.control.defaults().extend([  new ol.control.ScaleLine(),  new ol.control.MousePosition({  coordinateFormat: ol.coordinate.createStringXY(4),  projection: 'EPSG:4326'  }),  new ol.control.FullScreen()  ]),  interactions: ol.interaction.defaults({  dragPan: false,  dragZoom: false  }).extend([  controlZoomin,controlPan  ]), |

Les barres d’outils sont créées dans le fichier « index.html ».

|  |
| --- |
| <div id="toolbars">  <div id="toolbar" >  <input type="radio" id="extent" name="toolbar" onclick="gestionControle(this);")>  <label for="extent"><span class="ui-icon extent"></span></label>  </input>  <input type="radio" id="pan" name="toolbar" checked="checked" onclick="gestionControle(this);">  <label for="pan"><span class="ui-icon pan"></span></label>  </input>  <input type="radio" id="zoomin" name="toolbar" onclick="gestionControle(this);">  <label for="zoomin" ><span class="ui-icon zoomin"></span></label>  </input>  <!-- <input type="radio" id="info" name="toolbar" onclick="gestionControle(this);">  <label for="info" ><span class="ui-icon info"></span></label>  </input> -->  </div>  <div id="basemaps" >  <input type="radio" id="osm" name="baseMapType" checked="checked">  <label for="osm"><span class="ui-icon osm"></span></label>  </input>  <input type="radio" id="esrisat" name="baseMapType">  <label for="esrisat"><span class="ui-icon esrisat"></label>  </input>  <input type="radio" id="countries" name="baseMapType">  <label for="countries"><span class="ui-icon countries"></label>  </input>  </div>  <div id="features" >  <input type="checkbox" id="inprogress" name="checkbox" checked="checked">  <label for="inprogress" ><span class="ui-icon inprogress"></label>  </input>  <input type="checkbox" id="finished" name="checkbox" checked="checked">  <label for="finished" ><span class="ui-icon finished"></label>  </input>  <input type="checkbox" id="venue\_ext" name="checkbox">  <label for="venue\_ext" ><span class="ui-icon all"></label>  </input>  <input type="checkbox" id="timezones" name="baseMapExt" checked="checked">  <label for="timezones"><span class="ui-icon tz"></label>  </input>  </div>  </div> |

Et leur style défini dans le fichier « style.css ».

|  |
| --- |
| #toolbar{  position: absolute;  left: 60px;  top: 5px;  z-index: 1000;  display: inline-block;  }  #basemaps{  position: absolute;  left: 210px;  top: 5px;  z-index: 1000;  display: inline-block;  }  #features{  position: absolute;  left: 360px;  top: 5px;  z-index: 1000;  display: inline-block;  }  .extent {background-image:url(../icons/world.png) !important;}  .zoomin {background-image:url(../icons/zoom\_in.png) !important;}  .pan {background-image:url(../icons/pan.png) !important;}  .zoomout {background-image:url(../icons/zoom\_out.png) !important;}  .ruler {background-image:url(../icons/ruler.png) !important;}  .ruler2 {background-image:url(../icons/ruler\_square.png) !important;}  .prev {background-image:url(../icons/navigation\_history\_active\_prev.png) !important;}  .next {background-image:url(../icons/navigation\_history\_active\_next.png) !important;}  .info {background-image:url(../icons/info2.png) !important;}  .osm {background-image:url(../icons/osm\_logo16x16.png) !important;}  .esrisat {background-image:url(../icons/esri16x16.png) !important;}  .countries {background-image:url(../icons/remix-world-continents16x16.png) !important;}  .tz {background-image:url(../icons/earth\_grid16x16.png) !important;}  .inprogress {background-image:url(../icons/green\_light16x16.png) !important;}  .finished {background-image:url(../icons/red\_light16x16.png) !important;}  .all {background-image:url(../icons/capital16x16.png) !important;} |

## Une vue généralisée de votre carte; les coordonnées latitude/longitude d’un point sur la carte

Voici à quoi ressemble l’interface Web de la carte dynamique des rencontres de Soccer. On peut y retrouver :

* Un en-tête
* La carte;
* La barre d’outils;
* Les coordonnées du pointeur de la souris en degré de longitude et de latitude;
* Un accès au zoom;
* Une échelle en km;
* L’accordéon avec :
  + Les détails des rencontres en cours ou terminées;
  + Une légende;
  + Des directions pour se retrouver dans les éléments de la page;
  + Les crédits.

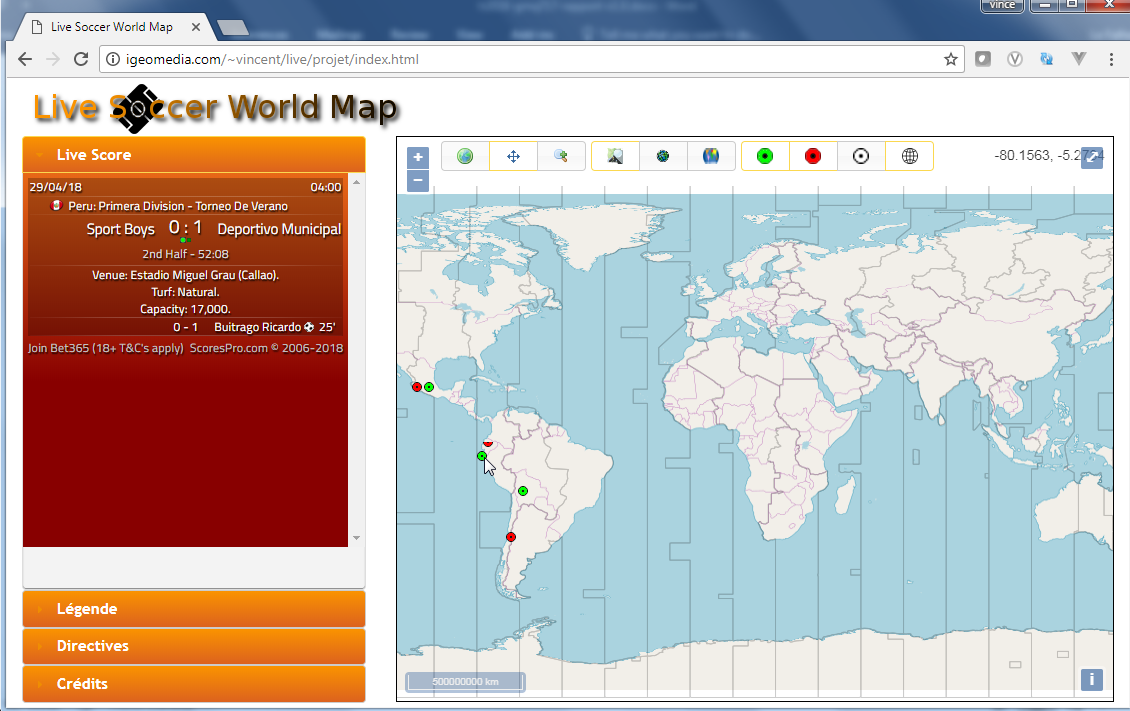


Figure 18: Détails de la rencontre en cours avec vue OpenStreetMap des continents

## L’interrogation possible d’au moins une couche

Dans le fichier « index.html », les objets « parser » et l’évènement « singleclick » sont définis en variables globales lorsque le document HTML est prêt.

|  |
| --- |
| <script>  $(document).ready(function () {  ...  var parser = new ol.format.WMSGetFeatureInfo();  var click = map.on('singleclick', function(evt) {  selectFixtureInProgress(parser, evt);  });  });  </script> |

La fonction « selectFixtureInProgress() » est appelée lors du clique sur la carte. Elle commence par activer la section de l’accordéon qui affiche le résultat de la rencontre ou les informations reliées au stade. Elle teste ensuite si la couche des rencontres en cours est active. Si c’est le cas, elle appelle l’URL de la requête WMS « GetFeatureInfo » via l’objet « $\_ajax » de jquery et valide s’il y a au moins un élément pour ces coordonnées dans la réponse via la fonction « then() » (callback). Si c’est le cas, les informations reliées à cet élément peuvent être utilisé. Dans les cas des couches des rencontres en cours ou terminées, la section de l’accordéon est remplie par un « iframe » sans bord affichant la page HTML de la rencontre fournie par le site « ScoresPro.com ». Dans le cas de la couche des stades, la section de l’accordéon est remplie avec les informations reliées au stade, c'est-à-dire son nom, le pays hôte, l’équipe résidente et le sport.

Les fonctions « selectFixtureFinished() » et « selectVenue() » sont appelées respectivement lorsque (1) la couche du dessus n’est pas active; et (2) il n’y a pas d’éléments proche des coordonnées du clique souris pour la couche du dessus.

La fonction ZoomIn() est appelée pour faire un zoom sur le stade en mode image satellite dans le cas où l’outil de zoom est sélectionné. Le niveau de zoom est assigné à une valeur de 17 car certains pays ne permettent pas plus proche (tel que le Japon, la plupart des pays d’Europe de l’Est et du Moyen-Orient).

|  |
| --- |
| function selectFixtureInProgress(parser, evt) {  var view = map.getView();  $( "#accordion" ).accordion( "option", "active", 0 );  if(layer\_fixture\_in\_progress.getVisible() === false ) {  selectFixtureFinished(parser, evt);  return;  }  var url = layer\_fixture\_in\_progress.getSource().getGetFeatureInfoUrl(evt.coordinate,  view.getResolution(), view.getProjection(),{'INFO\_FORMAT': 'application/vnd.ogc.gml'});  $.ajax(url).then(function(response) {  var features = parser.readFeatures(response);  if(features.length !== 0){  highlightOverlay.getSource().clear();  highlightOverlay.getSource().addFeatures(features);  var html = "<iframe src='" + features[0].getProperties().url + "' height='90%' width='100%' scrolling='auto' frameborder='0'></iframe>";  $("#infoPanel").html(html);  zoomIn(features[0]);  } else {  selectFixtureFinished(parser, evt);  }  });  return;  }  // ----------------------------------------------------------------------------  function zoomIn( feature ) {  if( feature !== null) {  if( controlZoomin.getActive() === true ) {  $( "#esrisat" ).prop("checked", true).change();  var point = feature.getProperties();  var coordinates = [point.boundedBy[1], point.boundedBy[0]];  map.setView(new ol.View({  projection: 'EPSG:4326',  center: coordinates,  zoom: 17 // Japan does not allow over 17 of zoom  })  );  }  }  } |

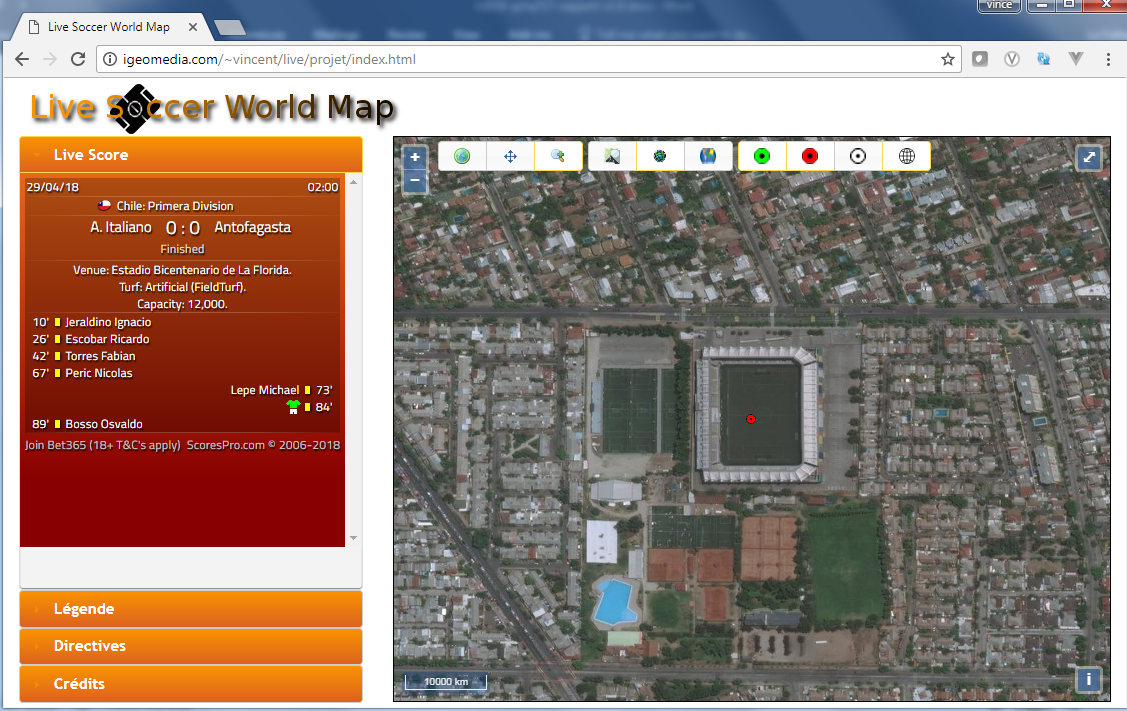


Figure 19: Détails de la rencontre terminée avec vue satellite ESRI

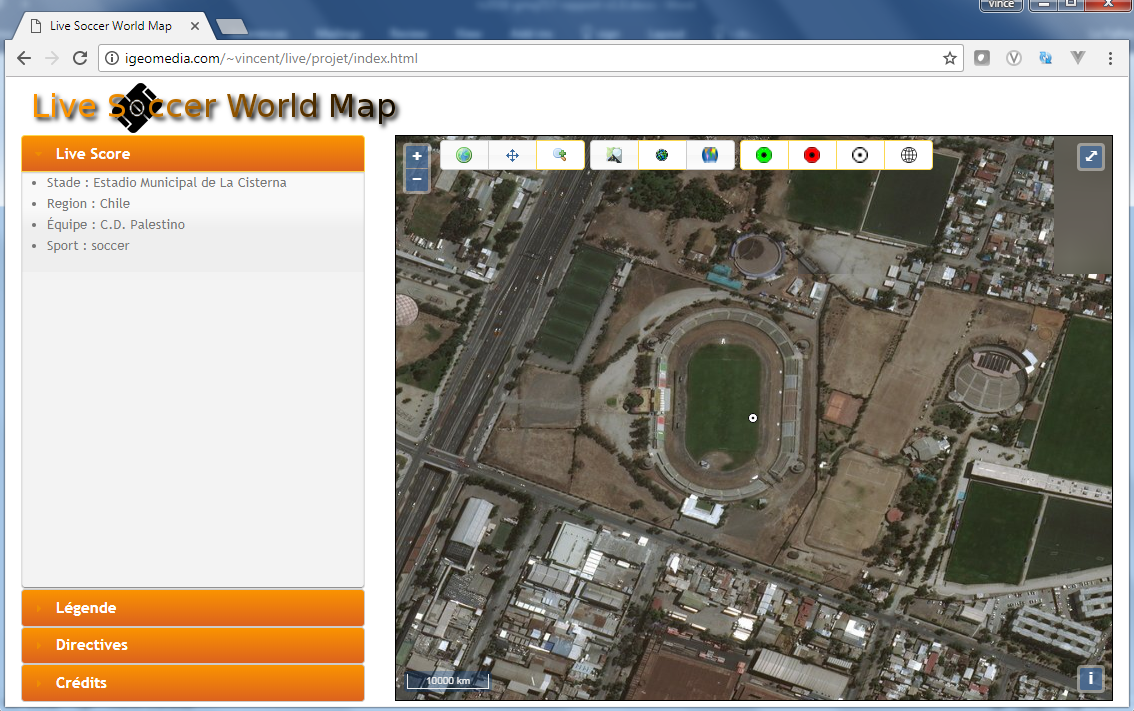


Figure 20: Informations reliées au stade avec vue satellite ESRI

## Une liste de couches de fond extraites de Bing Map ou OpenStreetMap

Deux couches de bases sont utilisées : celle d’OpenStreetMap et celle d’ESRI Satellite. La couche visible au démarrage est celle d’OpenStreetMap.

|  |
| --- |
| layerOSM = new ol.layer.Tile({source: new ol.source.OSM()});  layerSat = new ol.layer.Tile({  type: 'base',  title: 'ESRI world imagery',  visible: false,  source: new ol.source.XYZ({  attributions: [  new ol.Attribution({  html: 'Tiles &copy; Esri &mdash; Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community'  })  ],  url: 'http://server.arcgisonline.com/ArcGIS/rest/services/World\_Imagery/MapServer/tile/{z}/{y}/{x}'  })  })  ...  layerOSM.setVisible(true);  layerSat.setVisible(false);  ... |

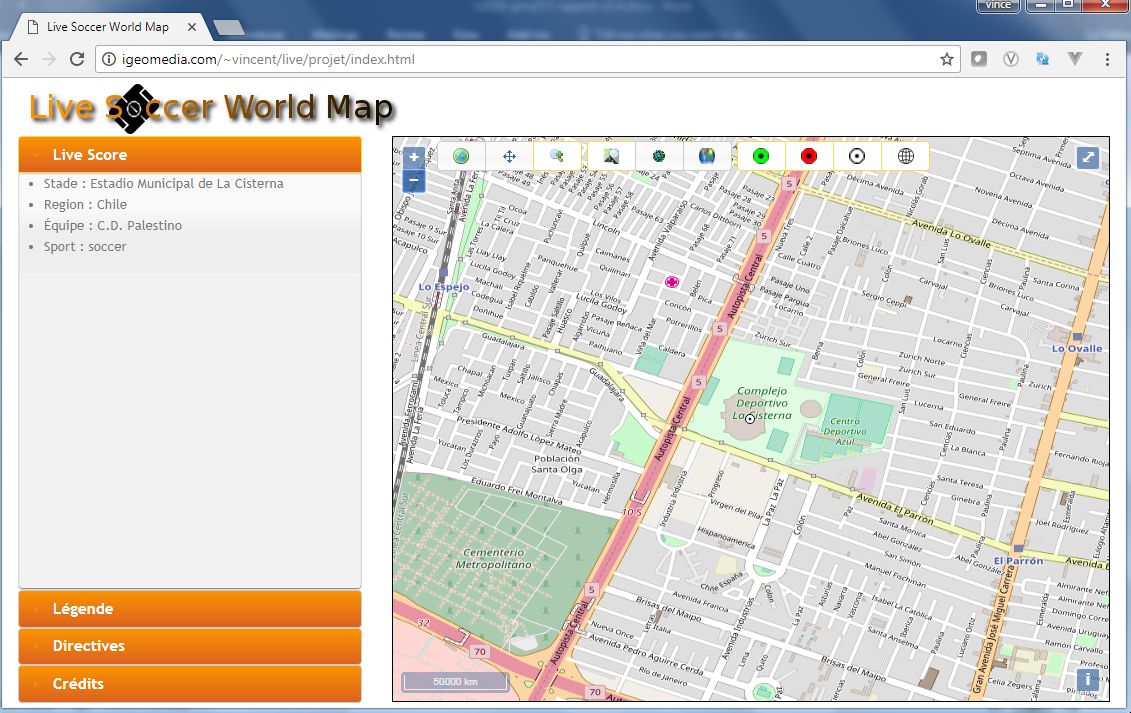


Figure 21: Informations reliées au stade avec fond de carte OpenStreetMap

# Étape 4

## Développer une application mobile simplifiée pour l’affichage des données

PhoneGap Desktop a été utilisé comme cadre de travail pour développer la version mobile de la carte. Le site <https://build.phonegap.com/> a été utilisé pour générer l’application mobile pour Android.

Un projet modèle (« HelloWorld ») a été créé au départ avec PhoneGap. Il a été compressé et converti en application afin de tester la procédure de génération d’application pour Android.

Un nouveau projet « LiveSoccerWorldMapApp » a ensuite été créé en copiant la structure de répertoire et les fichiers du modèle « HelloWorld ». Les fichiers styles « CSS », images « PNG » et scripts javascript « JS » du projet du site Web ont été copiées dans les répertoires « css », « img » et « js » respectivement du projet « LiveSoccerWorldMapApp », sans aucune modification.

Le fichier « index.html » a été adapté pour représenter la carte d’OpenLayer, le cadre de présentation de Bootstrap et les éléments graphiques de jquery-ui (accordéon, barre d’outils).

|  |
| --- |
| <html>  <head>  <meta charset="utf-8" />  <meta name="viewport" content="width=device-width, initial-scale=1">  <meta name="format-detection" content="telephone=no" />  <meta name="msapplication-tap-highlight" content="no" />  <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width" />  <!-- This is a wide open CSP declaration. To lock this down for production, see below. -->  <!-- Good default declaration:  \* gap: is required only on iOS (when using UIWebView) and is needed for JS->native communication  \* https://ssl.gstatic.com is required only on Android and is needed for TalkBack to function properly  \* Disables use of eval() and inline scripts in order to mitigate risk of XSS vulnerabilities. To change this:  \* Enable inline JS: add 'unsafe-inline' to default-src  \* Enable eval(): add 'unsafe-eval' to default-src  \* Create your own at http://cspisawesome.com  -->  <!-- <meta http-equiv="Content-Security-Policy" content="default-src 'self' data: gap: 'unsafe-inline' https://ssl.gstatic.com; style-src 'self' 'unsafe-inline'; media-src \*" /> -->  <link rel="stylesheet" type="text/css" href="css/index.css" />  <link rel="stylesheet" href="http://code.jquery.com/ui/1.11.4/themes/ui-lightness/jquery-ui.css">  <link rel="stylesheet" href="https://openlayers.org/en/v4.6.5/css/ol.css">  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">  <script src="http://code.jquery.com/jquery-1.10.2.js"></script>  <script src="http://code.jquery.com/ui/1.11.4/jquery-ui.js"></script>  <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>  <script src="https://openlayers.org/en/v4.6.5/build/ol.js"></script>  <link rel="stylesheet" href="css/normalize.css">  <link rel="stylesheet" href="css/style.css">  <script src="js/initCarte8.js"></script>  <script src="js/initOutils7.js"></script>  <script src="js/initSelectionCouche2.js"></script>  <style type="text/css">  html { height: 100% }  body { height: 100%; margin: 0px; padding: 0px }  #map { height: 100% }  </style>  <title>Live Soccer World Map</title>  </head>  <body>  <script type="text/javascript" src="cordova.js"></script>  <script type="text/javascript" src="js/index.js"></script>  <script type="text/javascript">  app.initialize();  </script>  <div class="container-fluid">  <div class="row">  <div class="col-sm-12">  <img src="img/header.png" >  </div>  </div>  <div class="row">  <div class="col-sm-8 col-sm-push-4">  <div id="map" ></div>  <div id="toolbars">  <div id="toolbar">  ...  </div>  <div id="basemaps" >  ...  </div>  <div id="features" >  ...  </div>  </div>  </div>  <div class="col-sm-4 col-sm-pull-8">  <div id="accordion">  <span class="accordion-title" style="color:#fff">Live Score</span>  <div id="infoPanel">  ...  </div>  <span class="accordion-title" style="color:#fff">Légende</span>  <div id="legend">  ...  </div>  <span class="accordion-title" style="color:#fff">Directives</span>  <div id="directive">  ...  </div>  </div>  </div>  </div>  </div>  <script type="text/javascript">  var btn = $.fn.button.noConflict() // reverts $.fn.button to jqueryui btn  $.fn.btn = btn // assigns bootstrap button functionality to $.fn.btn  $( "#accordion" ).accordion({  img: {},  heightStyle: "fill",  active: 0,  autoHeight: false,  navigation: true  });  creationCarte();  creationBarreOutils();  creationSelectionCouche();  var parser = new ol.format.WMSGetFeatureInfo();  var click = map.on('singleclick', function(evt) {  selectFixtureInProgress(parser, evt);  });  </script>  </body>  </html> |

Il est aussi intéressant de noter que l’application mobile ne sera affichée qu’en mode paysage, le mode portrait ne facilitant pas l’interaction avec l’application. Cette configuration a été apporté dans le fichier « config.xml » du projet PhoneGap.

|  |
| --- |
| <preference name="orientation" value="landscape" /> |

Avant que le répertoire ne soit compressé pour être compilé, le fichier « config.xml » doit être présent dans le répertoire « www ». Il a été nettoyé de tous les « plugins » présents. Le répertoire du projet « LiveSoccerWorldMapApp » a été compressé en fichier zip, déployé sur le site build.phonegap.com, et converti en application Android. Le fichier « apk » a été téléchargé et déposé dans le répertoire DropBox suivant :

<https://www.dropbox.com/s/nbw70ytqb6n38yp/LiveSoccerWorldMapApp-debug.apk?dl=0>

Pour installer l’application sur le téléphone mobile Android, il suffit d’aller à l’URL DropBox ci-dessus avec le navigateur de l’appareil mobile. Pour des raisons de sécurités, il n’est pas permis par défaut d’installer des applications qui ne proviennent pas du « Play Store » de Google. Il faut donc permettre dans les propriétés de configuration de sécurité l’installation d’applications de sources inconnues et confirmer l’autorisation. L’installation peut alors s’effectuer et l’application s’ouvrir.

# Recommandations / Limitations

Voici une liste des quelques recommandations et limitations qui ont été collectées.

* Mauvaise représentation de la pastille verte / rouge

Une rencontre peut être terminée mais la capsule reste verte. Le statut (« terminé ») n’est pas mis à jour car l’information n’est pas reçue du flux RSS.

Lorsqu’une rencontre vient de se terminer et qu’un zoom est appliqué, la pastille peut apparaitre verte au niveau de zoom 3 mais rouge au niveau de zoom 4 et plus. L’hypothèse (non vérifiée) est que cela est dû au (non-)rafraichissement des tuiles en mémoire du serveur cartographique mapserver. Un rafraichissement de la carte dans le navigateur (ctrl-+F5) rectifie la situation. Le désavantage est que l’utilisateur à tendance à remettre en question les statuts de chaque rencontre affichée. C’est un défaut majeur qu’il est recommandé de rectifier en priorité.

* Sélection de la pastille dans la carte

Lorsque la pastille de couleur est sélectionnée, son symbole devrait être différent des autres pastilles non-sélectionnées afin de la dissocier et localiser sur la carte les détails de la rencontre ou du stade qui sont présents dans l’accordéon.

* Décalage horaire

Il serait recommandé d’afficher l’heure locale de la rencontre et l’heure correspondant à celle de l’utilisateur.

* Liste des rencontres en cours

Il serait intéressant d’afficher la liste des rencontres en cours dans une partie de l’accordéon. Cela permettrait à l’utilisateur de pouvoir avoir accès directement aux détails des rencontres, tel que le nom des équipes, le score, le temps de jeu. Il aurait aussi accès à toutes les rencontres en cours fournies par le flux RSS ScoresPro.com et non juste à celles qui peuvent être représentée géographiquement. Il faut se souvenir que les rencontres peuvent être représentées géographiquement si :

* + Une similarité dans le nom du stade et du pays hôte du flux RSS ScoresPro.com est faite avec le nom du stade et du pays hôte provenant des données de wikidata.org;
  + Si le nom du stade existe dans le site wikidata.org;
  + Si les coordonnées du stade sont associée au stade dans le site wikidata.org.
* Liste des rencontres terminées

De même que pour la liste des rencontres en cours, la liste des rencontres terminées pourraient être affichées. Une option pourrait être offerte à l’utilisateur pour déterminer quelles rencontres terminées à afficher (dernières 15-30-60 minutes) ou un bouton permettant d’afficher les dernières rencontres par groupe de 20, sans limiter le nombre de rencontres terminée à afficher (« scrollbar »).

* Mise à jour de la liste des stades

Il serait intéressant de collecter les données de Wikidata régulièrement afin d’avoir les données les plus à jour.

* Amélioration de la détection du stade

Lorsque le nom du stade fourni par le flux RSS ScoresPro.com n’est pas assez informatif et précis, comme les mots singuliers et trop génériques (exemple : « Municipal »), la similarité va avoir un résultat assez raisonnable ( > 50% de similitude) mais le stade ne correspondra pas au lieu de la rencontre. Il est a noté que cela survient lorsque plusieurs noms de stade du même pays sont identiques et proches. A ce jour le filtre de similarité ne compare uniquement que le nom du stade et le pays hôte.

* Statut des rencontres

Actuellement il n’existe que deux statuts de rencontre supportés : « en cours » et « terminée ». Il a été observé que les rencontres peuvent aussi être « abandonnée » ou « remise » (« postponed » à une date ultérieure).

* Application mobile Android

Sur un téléphone mobile Android, il est très difficile de cliquer sur une pastille pour faire afficher les informations de la rencontre ou zommer sur le stade.

* Infrastructure

Le déploiement dans le nuage (« cloud ») de la solution serait à évaluer (Amazon AWS, IBM, Microsoft) afin de donner à la solution l’option d’automatiquement répondre à la charge (« scaling »), d’hériter des maintenances et de l’évolution des systèmes et de l’infrastructure hôte.

* Intégration de systèmes

Le serveur cartographique devrait gérer la mise à jour des données lui-même afin d’éviter la copie et la restauration de la base de données.

* Évolutions

Vue que le processus et la base de l’application est en place, il est possible d’imaginer ajouter sans trop d’effort d’autres évènements sportifs comme le hockey sur glace, le football américain, le rugby, le tennis, le basketball, le baseball.

La carte a un certain potentiel dans le contexte des évènements en cours. Voici quelques idées : une couche météorologique simple pourrait être rajouté selon le niveau de zoom (température, pluie); une couche des attractions touristiques aux alentours du stade pourrait être incluse; des liens vers les statistiques des équipes, des compétitions et des paris sportifs pourraient être proposés; l’affichage sur la carte des équipes, du scores et du temps restant; le rafraichissement et la notification automatique et en directe d’événements intra-sportif à l’évènement, comme les moment forts, les buts, le début de rencontre, mi-temps, fin de rencontre, les pénalités (carton jaune, rouge).

1. +proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs [↑](#footnote-ref-1)
2. +proj=merc +a=6378137 +b=6378137 +lat\_ts=0.0 +lon\_0=0.0 +x\_0=0.0 +y\_0=0 +k=1.0 +units=m +nadgrids=@null +wktext +no\_defs [↑](#footnote-ref-2)