HOLYSEARCH   
Dossier d’architecture des données

# SOMMAIRE

[I - SOMMAIRE 1](#_Toc472150828)

[II - Introduction 3](#_Toc472150829)

[III - Initialisation de la la base de données à partir de l’API OpenStreetMap 3](#_Toc472150830)

[A - Récupération et injection en BDD des continents 3](#_Toc472150831)

[B - Récupération et injection en BDD des pays du monde 5](#_Toc472150832)

[C - Récupération et injection en BDD des villes du monde 13](#_Toc472150833)

[D - Récupération et injection en BDD des capitales du monde 18](#_Toc472150834)

[E - Récupéreration et injection en BDD des plages du monde 22](#_Toc472150835)

[F - Reverse geocoding sur les plages du monde de la base de données 23](#_Toc472150836)

[IV - Injection en base de données des stations de ski à partir du fichier de gisngeo.com 25](#_Toc472150837)

[V - Reverse geocoding sur les stations de ski du monde de la base de données 25](#_Toc472150838)

[VI - Enrichissement de la base de données par les données Wikipédia 26](#_Toc472150839)

[A - Obtenir la description de la page de résultat de Wikipédia 27](#_Toc472150840)

[B - Obtenir les images des résultats de Wikipédia 28](#_Toc472150841)

[VII - Enrichissement de la base de données : attribuer à chaque pays son code international 29](#_Toc472150842)

[VIII - Enrichissement de la base de données à partir des fichiers DATAWORLDBANK 29](#_Toc472150843)

[A - Température des pays 29](#_Toc472150844)

[B - Précipitations des pays 30](#_Toc472150845)

[C - Criminalité des pays 30](#_Toc472150846)

[IX - Enrichissement de la vue des résultats depuis le site internet 31](#_Toc472150847)

[A - place = vue carte 31](#_Toc472150848)

[B - view = vue satellite 32](#_Toc472150849)

[C - streetview = vue immersive en 3D 32](#_Toc472150850)

[D - POC HTML IFRAME 33](#_Toc472150851)

# Introduction

-format de réponse de l'API overpass : JSON (par défaut dans la doc) ou XML  
-format de réponse de l'API de reverse geocoding : JSON ou XML (par défaut dans la doc)

# Initialisation de la la base de données à partir de l’API OpenStreetMap

## Récupération et injection en BDD des continents

### Requête :

------------------------------------------------------  
http://overpass-api.de/api/interpreter?data=[out:json];node[place="continent"];out;  
------------------------------------------------------

### Exemple de réponse du webservice

------------------------------------------------------

{

"type": "node",

"id": 25871341,

"lat": 51.0000003,

"lon": 9.9999997,

"tags": {

"alt\_name:vi": "Ã‚u ChÃ¢u",

"name": "Europe",

"name:ar": "Ø£ÙˆØ±ÙˆØ¨Ø§",

"name:be": "Ð•ÑžÑ€Ð¾Ð¿Ð°",

"name:ca": "Europa",

"name:cs": "Evropa",

"name:csb": "EÃ¹ropa",

"name:da": "Europa",

"name:de": "Europa",

"name:el": "Î•Ï…ÏÏŽÏ€Î·",

"name:en": "Europe",

"name:eo": "EÅ­ropo",

"name:es": "Europa",

"name:eu": "Europa",

"name:fi": "Eurooppa",

"name:fr": "Europe",

"name:fy": "Jeropa",

"name:ga": "Eoraip",

"name:hi": "à¤¯à¥‚à¤°à¥‹à¤ª",

"name:hr": "Europa",

"name:hu": "EurÃ³pa",

"name:ia": "Europa",

"name:io": "Europa",

"name:is": "EvrÃ³pa",

"name:it": "Europa",

"name:jbo": "rontu'a",

"name:kn": "à²¯à³à²°à³‹à²ªà³",

"name:ku": "Ewropa",

"name:la": "Europa",

"name:li": "Europa",

"name:lt": "Europa",

"name:mi": "Åªropi",

"name:ms": "Eropah",

"name:nb": "Europa",

"name:nl": "Europa",

"name:nn": "Europa",

"name:no": "Europa",

"name:nov": "Europa",

"name:pl": "Europa",

"name:pt": "Europa",

"name:ro": "Europa",

"name:ru": "Ð•Ð²Ñ€Ð¾Ð¿Ð°",

"name:sk": "EurÃ³pa",

"name:sl": "Evropa",

"name:sr": "Ð•Ð²Ñ€Ð¾Ð¿Ð°",

"name:sv": "Europa",

"name:szl": "Ojropa",

"name:ta": "à®à®°à¯‹à®ªà¯à®ªà®¾",

"name:tl": "Europa",

"name:tr": "Avrupa",

"name:tzl": "Europa",

"name:uk": "Ð„Ð²Ñ€Ð¾Ð¿Ð°",

"name:vi": "ChÃ¢u Ã‚u",

"name:vo": "Yurop",

"name:zea": "Europa",

"name:zh": "æ¬§æ´²",

"name:zh-classical": "æ­æ´²",

"name:zh-simplified": "æ¬§æ´²",

"name:zh-traditional": "æ­æ´²",

"name:zh\_pinyin": "ÅŒu",

"place": "continent",

"population": "739165030",

"sqkm": "10180000",

"wikipedia": "en:Europe"

}

},

------------------------------------------------------

### Traitement :

-Parser pour chaque réponse du webservice les champs :

• lat

• lon

• tags -> "name:fr"

• tags -> "name:fr"

• tags -> "population"

• tags -> "sqkm"

-Les injecter en BDD dans la table « Continent » avec les champs :

* englishname
* frenchname
* longitude
* latitude
* population
* sizekm

## Récupération et injection en BDD des pays du monde

### Requête :

------------------------------------------------------

http://overpass-api.de/api/interpreter?data=[out:json];node["place"="country"];out body;

------------------------------------------------------

### Exemple de réponse du webservice

------------------------------------------------------

{

"type": "node",

"id": 1363947712,

"lat": 46.6033540,

"lon": 1.8883335,

"tags": {

"ISO3166-1": "FR",

"ISO3166-1:alpha2": "FR",

"ISO3166-1:alpha3": "FRA",

"ISO3166-1:numeric": "250",

"alt\_name": "France mÃ©tropolitaine",

"alt\_name:en": "Metropolitan France",

"capital\_city": "Paris",

"comment": "Please do not remove this point (used to best place and display the label for the country, but NOT its administrative capital in Paris)",

"country\_code\_fips": "FR",

"country\_code\_iso3166\_1\_alpha\_2": "FR",

"currency": "EUR",

"description": "Computed geodesic center of the circle with minimum radius, 533 km, containing all terrestrial points in continental France",

"int\_name": "France",

"is\_in:continent": "Europe",

"name": "France",

"name:ace": "Peurancih",

"name:af": "Frankryk",

"name:ak": "FrÉ›nkyeman",

"name:als": "Frankreich",

"name:am": "áˆáˆ¨áŠ•áˆ£á‹­",

"name:an": "Francia",

"name:ang": "Francland",

"name:ar": "ÙØ±Ù†Ø³Ø§",

"name:arc": "Ü¦ÜªÜ¢Ü£Ü",

"name:arz": "ÙØ±Ù†Ø³Ø§",

"name:as": "à¦«à§à§°à¦¾à¦¨à§à¦¸",

"name:ast": "Francia",

"name:ay": "Phransiya",

"name:az": "Fransa",

"name:ba": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:bar": "Frankreich",

"name:bat-smg": "PrancÅ«zÄ—jÄ—",

"name:bcl": "Pransya",

"name:be": "Ð¤Ñ€Ð°Ð½Ñ†Ñ‹Ñ",

"name:be-tarask": "Ð¤Ñ€Ð°Ð½Ñ†Ñ‹Ñ",

"name:bg": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:bi": "Franis",

"name:bjn": "Parancis",

"name:bm": "France",

"name:bn": "à¦«à§à¦°à¦¾à¦¨à§à¦¸",

"name:bo": "à½§à¾¥à¼‹à½¢à½“à¼‹à½¦à½²à¼",

"name:bpy": "à¦«à§à¦°à¦¾à¦¨à§à¦¸",

"name:br": "Bro-C'hall",

"name:bs": "Francuska",

"name:bug": "Perancis",

"name:bxr": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸",

"name:ca": "FranÃ§a",

"name:cbk-zam": "Francia",

"name:cdo": "HuÃ¡k-guÃ³k",

"name:ce": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸",

"name:ceb": "Pransiya",

"name:chr": "áŽ¦áŽ¸á¥á±",

"name:chy": "France",

"name:ckb": "ÙÛ•Ú•Û•Ù†Ø³Ø§",

"name:co": "Francia",

"name:crh": "Frenkistan",

"name:cs": "Francie",

"name:csb": "FrancÃ«jÃ´",

"name:cu": "Ð¤Ñ€Ð°Ð½ÐºÑ—ê™—",

"name:cv": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸",

"name:cy": "Ffrainc",

"name:da": "Frankrig",

"name:de": "Frankreich",

"name:diq": "Fransa",

"name:dsb": "Francojska",

"name:dv": "ÞŠÞ¦ÞƒÞ¦Þ‚Þ°ÞÞ­ÞÞ¨ÞˆÞ¨ÞÞ§ÞŒÞ°",

"name:dz": "à½•à½¢à½±à½“à½¦à½²à¼‹",

"name:ee": "France",

"name:el": "Î“Î±Î»Î»Î¯Î±",

"name:eml": "Franza",

"name:en": "France",

"name:eo": "Francio",

"name:es": "Francia",

"name:et": "Prantsusmaa",

"name:eu": "Frantzia",

"name:ext": "FrÃ¡ncia",

"name:fa": "ÙØ±Ø§Ù†Ø³Ù‡",

"name:ff": "Faransi",

"name:fi": "Ranska",

"name:fiu-vro": "Prantsusmaa",

"name:fo": "Frakland",

"name:fr": "France",

"name:frp": "France",

"name:frr": "FrÃ¥nkrik",

"name:fur": "France",

"name:fy": "Frankryk",

"name:ga": "An Fhrainc",

"name:gag": "FranÈ›iya",

"name:gan": "æ³•åœ‹",

"name:gd": "An Fhraing",

"name:gl": "Francia",

"name:gn": "HyÃ£sia",

"name:gsw": "Frankreich",

"name:gu": "àª«à«àª°àª¾àª¨à«àª¸",

"name:gv": "Yn Rank",

"name:ha": "Faransa",

"name:hak": "Fap-koet",

"name:haw": "Palani",

"name:he": "×¦×¨×¤×ª",

"name:hi": "à¤«à¤¼à¥à¤°à¤¾à¤¨à¥à¤¸",

"name:hif": "France",

"name:hr": "Francuska",

"name:hsb": "Francoska",

"name:ht": "Frans",

"name:hu": "FranciaorszÃ¡g",

"name:hy": "Õ–Ö€Õ¡Õ¶Õ½Õ«Õ¡",

"name:ia": "Francia",

"name:id": "Perancis",

"name:ie": "Francia",

"name:ii": "êƒ”ê‡©",

"name:ilo": "Pransia",

"name:io": "Francia",

"name:is": "Frakkland",

"name:it": "Francia",

"name:ja": "ãƒ•ãƒ©ãƒ³ã‚¹",

"name:jbo": "fasygu'e",

"name:jv": "Prancis",

"name:ka": "áƒ¡áƒáƒ¤áƒ áƒáƒœáƒ’áƒ”áƒ—áƒ˜",

"name:kaa": "Frantsiya",

"name:kab": "Fransa",

"name:kbd": "Ð¤Ñ€ÑÐ½Ð´Ð¶",

"name:kg": "Fwalansa",

"name:ki": "Ubaranja",

"name:kk": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:kl": "Frankrigi",

"name:km": "áž”áž¶ážšáž¶áŸ†áž„",

"name:kn": "à²«à³à²°à²¾à²¨à³à²¸à³",

"name:ko": "í”„ëž‘ìŠ¤",

"name:koi": "Ð¤Ñ€Ð°Ð½Ñ",

"name:krc": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:ks": "ÙØ±Ù›Ø§Ù†Ø³",

"name:ksh": "Frankreich",

"name:ku": "Fransa",

"name:kv": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:kw": "Pow Frynk",

"name:ky": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:la": "Francia",

"name:lad": "Fransia",

"name:lb": "FrankrÃ¤ich",

"name:lez": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:lg": "Bufalansa",

"name:li": "Frankriek",

"name:lij": "Fransa",

"name:lmo": "FrÃ ncia",

"name:ln": "Falansia",

"name:lo": "àº›àº°à»€àº—àº”àºàº°àº¥àº±à»ˆàº‡",

"name:lt": "PrancÅ«zija",

"name:ltg": "PraÅ†ceja",

"name:lv": "Francija",

"name:mdf": "ÐšÑ€Ð°Ð½Ñ†Ð¼Ð°ÑÑ‚Ð¾Ñ€",

"name:mg": "Frantsa",

"name:mhr": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ð¹",

"name:mi": "WÄ«wÄ«",

"name:min": "Parancih",

"name:mk": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ˜Ð°",

"name:ml": "à´«àµà´°à´¾àµ»à´¸àµ",

"name:mn": "Ð¤Ñ€Ð°Ð½Ñ†",

"name:mr": "à¤«à¥à¤°à¤¾à¤¨à¥à¤¸",

"name:ms": "Perancis",

"name:mt": "Franza",

"name:mwl": "FrÃ¡ncia",

"name:my": "á€•á€¼á€„á€ºá€žá€…á€ºá€”á€­á€¯á€„á€ºá€„á€¶",

"name:myv": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ ÐœÐ°ÑÑ‚Ð¾Ñ€",

"name:mzn": "ÙØ±Ø§Ù†Ø³Ù‡",

"name:na": "Prant",

"name:nah": "Francia",

"name:nan": "Hoat-kok",

"name:nap": "Franza",

"name:nds": "Frankriek",

"name:nds-nl": "Frankriek",

"name:ne": "à¤«à¥à¤°à¤¾à¤¨à¥à¤¸",

"name:nl": "Frankrijk",

"name:nn": "Frankrike",

"name:no": "Frankrike",

"name:nov": "Fransia",

"name:nrm": "France",

"name:nso": "Fora",

"name:nv": "DÃ¡Ã¡ghahii DinÃ©Ê¼iÊ¼ BikÃ©yah",

"name:oc": "FranÃ§a",

"name:or": "à¬«à­à¬°à¬¾à¬¨à­à¬¸",

"name:os": "Ð¤Ñ€Ð°Ð½Ñ†",

"name:pa": "à¨«à©à¨°à¨¾à¨‚à¨¸",

"name:pag": "Fransia",

"name:pam": "Pransiya",

"name:pap": "Fransia",

"name:pcd": "Franche",

"name:pdc": "Frankreich",

"name:pfl": "Frongraisch",

"name:pih": "France",

"name:pl": "Francja",

"name:pms": "Fransa",

"name:pnb": "ÙØ±Ø§Ù†Ø³",

"name:pnt": "Î“Î±Î»Î»Î¯Î±",

"name:ps": "ÙØ±Ø§Ù†Ø³Ù‡",

"name:pt": "FranÃ§a",

"name:qu": "Ransiya",

"name:rm": "Frantscha",

"name:rmy": "Franchiya",

"name:rn": "Francia",

"name:ro": "FranÈ›a",

"name:roa-tara": "Frange",

"name:ru": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:rue": "Ð¤Ñ€Ð°Ð½Ñ†Ñ–Ñ",

"name:rw": "Ubufaransa",

"name:sa": "à¤«à¥à¤°à¤¾à¤¨à¥à¤¸à¤¦à¥‡à¤¶à¤ƒ",

"name:sah": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:sc": "Frantza",

"name:scn": "Francia",

"name:sco": "Fraunce",

"name:se": "Frankriika",

"name:sg": "FarÃ¢nzi",

"name:sh": "Francuska",

"name:si": "à¶´à·Šâ€à¶»à¶‚à·à¶º",

"name:simple": "France",

"name:sk": "FrancÃºzsko",

"name:sl": "Francija",

"name:sm": "Farani",

"name:so": "Faransiiska",

"name:sq": "Franca",

"name:sr": "Ð¤Ñ€Ð°Ð½Ñ†ÑƒÑÐºÐ°",

"name:srn": "Franskondre",

"name:ss": "IFulansi",

"name:st": "Fora",

"name:stq": "Frankriek",

"name:su": "Perancis",

"name:sv": "Frankrike",

"name:sw": "Ufaransa",

"name:szl": "Francyjo",

"name:ta": "à®ªà®¿à®°à®¾à®©à¯à®šà¯",

"name:te": "à°«à±à°°à°¾à°¨à±à°¸à±",

"name:tet": "Fransa",

"name:tg": "Ð¤Ð°Ñ€Ð¾Ð½ÑÐ°",

"name:th": "à¸›à¸£à¸°à¹€à¸—à¸¨à¸à¸£à¸±à¹ˆà¸‡à¹€à¸¨à¸ª",

"name:ti": "áˆáˆ¨áŠ•áˆ³á‹­",

"name:tk": "FransiÃ½a",

"name:tl": "Pransiya",

"name:to": "FalanisÄ“",

"name:tpi": "Frens",

"name:tr": "Fransa",

"name:tt": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:ty": "FarÄni",

"name:udm": "Ð¤Ñ€Ð°Ð½Ñ†Ð¸Ñ",

"name:ug": "ÙØ±Ø§Ù†Ø³Ù‰ÙŠÛ•",

"name:uk": "Ð¤Ñ€Ð°Ð½Ñ†Ñ–Ñ",

"name:ur": "ÙØ±Ø§Ù†Ø³",

"name:uz": "Fransiya",

"name:vec": "Fransa",

"name:vep": "Francii",

"name:vi": "PhÃ¡p",

"name:vls": "Vrankryk",

"name:vo": "FransÃ¤n",

"name:wa": "France",

"name:war": "Fransya",

"name:wo": "Faraas",

"name:wuu": "æ³•å›½",

"name:xal": "ÐŸÑ€Ð°Ð½Ñ†ÑÐ¸Ð½ ÐžÑ€Ð½",

"name:xmf": "áƒ¡áƒáƒ¤áƒ áƒáƒœáƒ’áƒ”áƒ—áƒ˜",

"name:yi": "×¤×¨×× ×§×¨×™×™×š",

"name:yo": "FrÃ¡nsÃ¬",

"name:yue": "æ³•åœ‹",

"name:za": "Fazgoz",

"name:zea": "Frankriek",

"name:zh": "æ³•å›½",

"name:zh-classical": "æ³•åœ‹",

"name:zh-min-nan": "Hoat-kok",

"name:zh-yue": "æ³•åœ‹",

"name:zh\_pinyin": "FÇŽguÃ³",

"name:zu": "IFulansi",

"official\_name": "RÃ©publique franÃ§aise",

"official\_name:be": "Ð ÑÑÐ¿ÑƒÐ±Ð»Ñ–ÐºÐ° Ð¤Ñ€Ð°Ð½Ñ†Ñ‹Ñ",

"official\_name:be-tarask": "Ð ÑÑÐ¿ÑƒÐ±Ð»Ñ–ÐºÐ° Ð¤Ñ€Ð°Ð½Ñ†Ñ‹Ñ",

"official\_name:el": "Î“Î±Î»Î»Î¹ÎºÎ® Î”Î·Î¼Î¿ÎºÏÎ±Ï„Î¯Î±",

"official\_name:et": "Prantsuse Vabariik",

"official\_name:fr": "RÃ©publique franÃ§aise",

"official\_name:gd": "Poblachd na Frainge",

"official\_name:id": "Republik Perancis",

"official\_name:it": "Repubblica francese",

"official\_name:ku": "Komara FransayÃª",

"official\_name:lb": "FransÃ©isch Republik",

"official\_name:lt": "PrancÅ«zijos Respublika",

"official\_name:nl": "Franse Republiek",

"official\_name:pt": "RepÃºblica Francesa",

"official\_name:sk": "FrancÃºzska republika",

"official\_name:sl": "Francoska republika",

"official\_name:sr": "Ð ÐµÐ¿ÑƒÐ±Ð»Ð¸ÐºÐ° Ð¤Ñ€Ð°Ð½Ñ†ÑƒÑÐºÐ°",

"official\_name:sv": "Republiken Frankrike",

"official\_name:uz": "Fransiya Respublikasi",

"official\_name:vi": "Cá»™ng hÃ²a PhÃ¡p",

"place": "country",

"population": "65073482",

"wikipedia": "fr:France"

}

},

------------------------------------------------------

### Traitement :

-Parser pour chaque réponse du webservice le champs :

• lat

• lon

• tags -> "ISO3166-1:alpha2"

• tags -> "ISO3166-1:alpha3"

• tags -> "name:en"

• tags -> "name:fr"

• tags -> "population"

• tags -> "currency"

**• tags -> "is\_in:continent"**

-pour chaque réponse, il faut associer un enregistrement de la table « **Continent** ». Utiliser le champ "**is\_in:continent**" pour remplir la clé étrangère sur Country : « continenid ».

-Injecter en BDD dans la table « Country » en remplissant les autres champs :

* latitude
* longitude
* isoa2
* isoa3
* englishname
* frenchname
* population
* currency

--------------------------------------------------------------------------------------------------------------------------------------

## Récupération et injection en BDD des villes du monde

### Requête :

------------------------------------------------------

http://overpass-api.de/api/interpreter?data=[out:json];node["place"="city"];out body;

------------------------------------------------------

### Exemple de réponse du webservice

------------------------------------------------------

{ "type": "node", "id": 107775, "lat": 51.5073219, "lon": -0.1276474, "tags": { "admin\_level": "2", "capital": "yes", "ele": "15", "is\_capital": "country", "is\_in": "England, United Kingdom, UK, Great Britain, Europe", "is\_in:continent": "Europe", "is\_in:country": "United Kingdom", "is\_in:country\_code": "GB", "is\_in:iso\_3166\_2": "GB-ENG", "name": "London", "name:ab": "Лондан", "name:af": "Londen", "name:am": "ለንደን", "name:an": "Londres", "name:ang": "Lunden", "name:ar": "لندن", "name:arc": "ܠܘܢܕܘܢ", "name:arz": "لندن", "name:ast": "Londres", "name:ba": "Лондон", "name:bat-smg": "Londons", "name:bcl": "Londres", "name:be": "Лондан", "name:be-tarask": "Лёндан", "name:bg": "Лондон", "name:bn": "লন্ডন", "name:bo": "ལོན་ཊོན།", "name:br": "Londrez", "name:ca": "Londres", "name:cdo": "Lùng-dŭng", "name:ckb": "لەندەن", "name:co": "Londra", "name:cs": "Londýn", "name:cu": "Лондонъ", "name:cv": "Лондон", "name:cy": "Llundain", "name:de": "London", "name:diq": "Londra", "name:el": "Λονδίνο", "name:eml": "Lòndra", "name:en": "London", "name:eo": "Londono", "name:es": "Londres", "name:eu": "Londres", "name:ext": "Londri", "name:fa": "لندن", "name:fi": "Lontoo", "name:fr": "Londres", "name:frp": "Londres", "name:fy": "Londen", "name:ga": "Londain", "name:gan": "倫敦", "name:gd": "Lunnainn", "name:gl": "Londres - London", "name:gn": "Londye", "name:gu": "લંડન", "name:gv": "Lunnin", "name:haw": "Lākana", "name:he": "לונדון", "name:hi": "लंदन", "name:ht": "Lonn", "name:hu": "London", "name:hy": "Լոնդոն", "name:ilo": "Londres", "name:is": "Lundúnir", "name:it": "Londra", "name:ja": "ロンドン", "name:jbo": "london", "name:ka": "ლონდონი", "name:kk": "Лондон", "name:kn": "ಲಂಡನ್", "name:ko": "런던", "name:koi": "Лондон", "name:krc": "Лондон", "name:ku": "London", "name:kv": "Лондон", "name:kw": "Loundres", "name:ky": "Лондон", "name:la": "Londinium", "name:lad": "Londra", "name:lbe": "Лондон", "name:lez": "Лондон", "name:li": "Londe", "name:lij": "Londra", "name:lmo": "Lundra", "name:ln": "Londoni", "name:lo": "ລອນດອນ", "name:lt": "Londonas", "name:lv": "Londona", "name:mhr": "Лондон", "name:mi": "Rānana", "name:mk": "Лондон", "name:ml": "ലണ്ടൻ", "name:mn": "Лондон", "name:mr": "लंडन", "name:mrj": "Лондон", "name:mt": "Londra", "name:mwl": "Londres", "name:my": "လန်ဒန်မြို့", "name:mzn": "لندن", "name:nah": "Londres", "name:nap": "Londra", "name:nds-nl": "Londen", "name:ne": "लण्डन", "name:new": "लण्डन", "name:nl": "Londen", "name:no": "London", "name:nrm": "Londres", "name:oc": "Londres", "name:or": "ଲଣ୍ଡନ", "name:os": "Лондон", "name:pcd": "Londe", "name:pl": "Londyn", "name:pms": "Londra", "name:pnb": "لندن", "name:pnt": "Λονδίνο", "name:ps": "لندن", "name:pt": "Londres", "name:rm": "Londra", "name:ro": "Londra", "name:roa-rup": "Londra", "name:roa-tara": "Londre", "name:ru": "Лондон", "name:rue": "Лондон", "name:sa": "लन्डन्", "name:sah": "Лондон", "name:sc": "Londra", "name:scn": "Londra", "name:sco": "Lunnon", "name:si": "ලන්ඩන්", "name:sk": "Londýn", "name:sq": "Londra", "name:sr": "Лондон", "name:sv": "London", "name:szl": "Lůndůn", "name:ta": "இலண்டன்", "name:te": "లండన్", "name:tet": "Londres", "name:tg": "Лондон", "name:th": "ลอนดอน", "name:tl": "Londres", "name:tpi": "Landen", "name:tr": "Londra", "name:tt": "Лондон", "name:tzl": "Londra", "name:udm": "Лондон", "name:uk": "Лондон", "name:ur": "لندن", "name:vec": "Łondra", "name:vi": "Luân Đôn", "name:vls": "Londn", "name:wo": "Londar", "name:wuu": "伦敦", "name:xmf": "ლონდონი", "name:yi": "לאנדאן", "name:yo": "Lọndọnu", "name:zea": "Londen", "name:zh": "倫敦", "name:zh-classical": "倫敦", "name:zh-simplified": "伦敦", "name:zh-traditional": "倫敦", "name:zh-yue": "倫敦", "name:zh\_pinyin": "Lúndūn", "name:zu": "ILondon", "note": "Centre of London officially at the former location of the Charing Cross, now the Charles I statue, near Trafalgar Square.", "place": "city", "population": "8416535", "website": "http://www.london.gov.uk", "wikidata": "Q84", "wikimedia\_commons": "Category:London", "wikipedia": "en:London" }},

------------------------------------------------------

### Traitement :

-Parser pour chaque réponse du webservice le champs :

• lat

• long

• tags -> "name:en"

• tags -> "name:fr"

• tags -> "population"

**• tags -> "is\_in:country"**

-pour chaque réponse, il faut associer un enregistrement de la table « **Country** ». Utiliser le champ " **is\_in:country** " pour remplir la clé étrangère sur City : « countryid ».

-Injecter en BDD dans la table « City » et remplir les autres champs :

* latitude
* longitude
* englishname
* frenchname
* population

--------------------------------------------------------------------------------------------------------------------------------------

## Récupération et injection en BDD des capitales du monde

### Requête :

------------------------------------------------------

http://overpass-api.de/api/interpreter?data=[out:json];node["place"="city"][capital=yes];out body;

------------------------------------------------------

### Exemple de réponse du webservice :

------------------------------------------------------

{

"type": "node",

"id": 17807753,

"lat": 48.8566101,

"lon": 2.3514992,

"tags": {

"addr:country": "FR",

"addr:postcode": "75004",

"admin\_level": "2",

"alt\_name:fr": "LutÃ¨ce",

"alt\_name:vi": "Ba LÃª",

"capital": "yes",

"capital\_ISO3166-1": "yes",

"contact:website": "http://www.paris.fr",

"ele": "30",

"is\_capital": "country",

"is\_in:continent": "Europe",

"is\_in:country": "France",

"is\_in:country\_code": "FR",

"is\_in:iso\_3166\_2": "FR-J",

"loc\_name:fr": "Panam",

"name": "Paris",

"name:af": "Parys",

"name:am": "á“áˆªáˆµ",

"name:an": "ParÃ­s",

"name:ar": "Ø¨Ø§Ø±ÙŠØ³",

"name:arc": "Ü¦ÜÜªÜÜ£",

"name:arz": "Ø¨Ø§Ø±ÙŠØ³",

"name:ast": "ParÃ­s",

"name:ba": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:bat-smg": "ParÄ«Å¾ios",

"name:be": "ÐŸÐ°Ñ€Ñ‹Ð¶",

"name:be-tarask": "ÐŸÐ°Ñ€Ñ‹Ð¶",

"name:bg": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:bn": "à¦ªà§à¦¯à¦¾à¦°à¦¿à¦¸",

"name:bo": "à½•à¼‹à½¢à½²à¼",

"name:br": "Pariz",

"name:bs": "Pariz",

"name:ca": "ParÃ­s",

"name:cdo": "BÄƒ-lÃ Ì¤",

"name:ckb": "Ù¾Ø§Ø±ÛŒØ³",

"name:co": "Parighji",

"name:cs": "PaÅ™Ã­Å¾",

"name:csb": "Pariz",

"name:cu": "ÐŸÐ°Ñ€Ð¸Ð¶ÑŒ",

"name:cv": "ÐŸÐ°Ñ€Ð¸Ñ",

"name:de": "Paris",

"name:el": "Î Î±ÏÎ¯ÏƒÎ¹",

"name:eml": "ParÃ¬",

"name:eo": "Parizo",

"name:es": "ParÃ­s",

"name:et": "Pariis",

"name:eu": "Paris",

"name:ext": "ParÃ­s",

"name:fa": "Ù¾Ø§Ø±ÛŒØ³",

"name:fi": "Pariisi",

"name:fiu-vro": "Pariis",

"name:fr": "Paris",

"name:fur": "ParÃ®s",

"name:fy": "Parys",

"name:ga": "PÃ¡ras",

"name:gan": "å·´é»Ž",

"name:gl": "ParÃ­s",

"name:gn": "ParÄ©",

"name:gu": "àªªà«…àª°àª¿àª¸",

"name:gv": "Paarys",

"name:ha": "Pariis",

"name:hak": "PÃ¢-lÃ¬-sá¹³",

"name:haw": "Palika",

"name:he": "×¤×¨×™×–",

"name:hi": "à¤ªà¥ˆà¤°à¤¿à¤¸",

"name:hr": "Pariz",

"name:ht": "Pari",

"name:hu": "PÃ¡rizs",

"name:hy": "Õ“Õ¡Ö€Õ«Õ¦",

"name:is": "ParÃ­s",

"name:it": "Parigi",

"name:ja": "ãƒ‘ãƒª",

"name:jbo": "Paris",

"name:ka": "áƒžáƒáƒ áƒ˜áƒ–áƒ˜",

"name:kk": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:km": "áž”áŸ‰áž¶ážšáž¸ážŸ",

"name:kn": "à²ªà³à²¯à²¾à²°à²¿à²¸à³",

"name:ko": "íŒŒë¦¬",

"name:koi": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:krc": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:ku": "ParÃ®s",

"name:kv": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:ky": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:la": "Lutetia",

"name:lb": "ParÃ¤is",

"name:lez": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:li": "Paries",

"name:lij": "Pariggi",

"name:ln": "Pari",

"name:lt": "ParyÅ¾ius",

"name:lv": "ParÄ«ze",

"name:lzh": "å·´é»Ž",

"name:mhr": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:mi": "ParÄ«",

"name:mk": "ÐŸÐ°Ñ€Ð¸Ð·",

"name:ml": "à´ªà´¾à´°à´¿à´¸àµ",

"name:mn": "ÐŸÐ°Ñ€Ð¸Ñ",

"name:mr": "à¤ªà¥…à¤°à¤¿à¤¸",

"name:mrj": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:my": "á€•á€«á€›á€®á€™á€¼á€­á€¯á€·",

"name:myv": "ÐŸÐ°Ñ€Ð¸Ð¶ Ð¾Ñˆ",

"name:mzn": "Ù¾Ø§Ø±ÛŒØ³",

"name:nan": "Pa-lÃ­",

"name:nap": "Parigge",

"name:nds": "Paris",

"name:nds-nl": "Paries",

"name:ne": "à¤ªà¥‡à¤°à¤¿à¤¸",

"name:nl": "Parijs",

"name:no": "Paris",

"name:oc": "ParÃ­s",

"name:or": "à¬ªà­à­Ÿà¬¾à¬°à¬¿à¬¸",

"name:os": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:pa": "à¨ªà©ˆà¨°à¨¿à¨¸",

"name:pl": "ParyÅ¼",

"name:pnb": "Ù¾ÛŒØ±Ø³",

"name:ps": "Ù¾Ø§Ø±ÙŠØ³",

"name:roa-rup": "Parij",

"name:roa-tara": "Parigge",

"name:ru": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:rue": "ÐŸÐ°Ñ€Ñ–Ð¶",

"name:sah": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:sc": "Parigi",

"name:scn": "Pariggi",

"name:sh": "Pariz",

"name:sk": "ParÃ­Å¾",

"name:sl": "Pariz",

"name:so": "Baariis",

"name:sq": "Parisi",

"name:sr": "ÐŸÐ°Ñ€Ð¸Ð·",

"name:sv": "Paris",

"name:szl": "ParyÅ¼",

"name:ta": "à®ªà®¾à®°à®¿à®¸à¯",

"name:te": "à°ªà°¾à°°à°¿à°¸à±",

"name:tet": "ParÃ­s",

"name:tg": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:th": "à¸›à¸²à¸£à¸µà¸ª",

"name:tk": "PariÅ¾",

"name:tl": "Paris",

"name:tt": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:tzl": "Parighi",

"name:udm": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:ug": "Ù¾Ø§Ø±Ù‰Ú˜",

"name:uk": "ÐŸÐ°Ñ€Ð¸Ð¶",

"name:ur": "Ù¾ÛŒØ±Ø³",

"name:uz": "Parij",

"name:vec": "Parixe",

"name:vep": "PariÅ¾",

"name:vi": "Paris",

"name:vls": "Parys",

"name:wo": "Pari",

"name:xmf": "áƒžáƒáƒ áƒ˜áƒ–áƒ˜",

"name:yi": "×¤××¨×™×–",

"name:yo": "Parisi",

"name:yue": "å·´é»Ž",

"name:za": "Bahliz",

"name:zh": "å·´é»Ž",

"name:zh\_pinyin": "BÄlÃ­",

"name:zu": "IParisi",

"old\_name:vi": "Ba LÃª",

"place": "city",

"population": "2243833",

"rank": "0",

"ref:INSEE": "75056",

"source:population": "INSEE 2013",

"wikipedia": "fr:Paris"

}

},

------------------------------------------------------

### Traitement :

-Parser pour chaque réponse du webservice les champs :

• tags -> "name:en"

• tags -> "**name:fr**"

• tags -> "**is\_in:country**"

-pour chaque réponse, il faut retrouver les enregistrements correspondants de la table « **Country** » et « **City** ». Utiliser le champ « **name:fr** » pour retrouver l’enregistrement « City » et utiliser le champ " **is\_in:country** " pour retrouver l’enregistrement « Country ». Remplir la clé étrangère sur Country « capitalid » avec la clé primaire extraite de l’enregistrement « City ».

--------------------------------------------------------------------------------------------------------------------------------------

## Récupéreration et injection en BDD des plages du monde

### Requête :

------------------------------------------------------

http://overpass-api.de/api/interpreter?[timeout:400000];data=[out:json];node[natural="beach"];out body;

------------------------------------------------------

### Exemple de réponse du webservice

------------------------------------------------------

{ "type": "node", "id": 735086243, "lat": 48.3969774, "lon": -4.4285481, "tags": { "name": "Plage du Moulin Blanc", "natural": "beach" }},

------------------------------------------------------

### Traitement :

-Parser pour chaque réponse du webservice le champs :

• lat

• lon

• tags -> "name"

-Injecter en BDD dans la table « Beach » en remplissant les champs :

* latitude
* longitude
* frenchname

--------------------------------------------------------------------------------------------------------------------------------------

## Reverse geocoding sur les plages du monde de la base de données

### Syntaxe de la requête

------------------------------------------------------

http://nominatim.openstreetmap.org/reverse?format=xml&lat=***numeroLatitude***&lon=***numeroLongitude***&zoom=18&addressdetails=1

------------------------------------------------------

La requête permet de récupérer dans des champs séparés les informations suivantes :

* -un string "result" équivalent à un toString()
* -la ville = "county"
* -le pays = "country"

### Exemple

Exemple pour la "plage du moulin blanc :

------------------------------------------------------

http://nominatim.openstreetmap.org/reverse?format=xml&lat=48.3969774&lon=-4.4285481&zoom=18&addressdetails=1

------------------------------------------------------

Retourne :

------------------------------------------------------

<reversegeocode timestamp="Mon, 02 Jan 17 13:43:26 +0000" attribution="Data © OpenStreetMap contributors, ODbL 1.0. http://www.openstreetmap.org/copyright" querystring="format=xml&lat=48.3969774&lon=-4.4285481&zoom=18&addressdetails=1">

<result place\_id="6990559" osm\_type="node" osm\_id="735086243" ref="Plage du Moulin Blanc" lat="48.3969774" lon="-4.428548" boundingbox="48.3968774,48.3970774,-4.428648,-4.428448">

Plage du Moulin Blanc, Rue de Palaren, Guipavas, Brest, Finistère, Bretagne, 29490, France

</result>

<addressparts>

<beach>Plage du Moulin Blanc</beach>

<pedestrian>Rue de Palaren</pedestrian>

<town>Guipavas</town>

<county>Brest</county>

<state>Bretagne</state>

<postcode>29490</postcode>

<country>France</country>

<country\_code>fr</country\_code>

</addressparts>

</reversegeocode>

------------------------------------------------------

### Traitement à effectuer pour toutes les plages du monde :

-Pour chaque enregistrement de « Beach » en BDD :

-utiliser la requête de reverse geocoding avec les variables de latitude et de longitude

-Parser pour chaque réponse du webservice les champs :

• result

• **county**

• **country**

-Enrichir la table dédiée aux plages par les réponses du reverse geocoding à savoir : associer les champs « county » et « country » aux enregistrements correspondant dans les tables « **Country** » et « **City** »

-pour chaque réponse, il faut retrouver les enregistrements correspondants de la table « **Country** » et « **City** ». Utiliser le champ « **county** » pour retrouver l’enregistrement « City » et utiliser le champ " **country** " pour retrouver l’enregistrement « Country ». Remplir les clés étrangères sur Beach « cityid » et « countryid » avec les clés primaires extraites des enregistrements « Country » et « City ».

--------------------------------------------------------------------------------------------------------------------------------------

# Injection en base de données des stations de ski à partir du fichier de gisngeo.com

-Accéder au GIT et parser dans le fichier « HolySearch/donnees\_plates/ SKI/**ski.csv** » les colonnes :

* « Resort »
* « Latitude »
* « Longitude »

-Injecter dans la table « Skistation » chaque enregistrement du fichier ski.csv dans les champs :

* frenchname
* latitude
* longitude

-----------------------------------------------------------------------------------------------------------------------------------------------------------

# Reverse geocoding sur les stations de ski du monde de la base de données

### Syntaxe de la requête

------------------------------------------------------

http://nominatim.openstreetmap.org/reverse?format=xml&lat=***numeroLatitude***&lon=***numeroLongitude***&zoom=18&addressdetails=1

------------------------------------------------------

La requête permet de récupérer dans des champs séparés les informations suivantes :

* -un string "result" équivalent à un toString()
* -la ville = "county"
* -le pays = "country"

### Exemple

Exemple pour la station de ski « Chamrousse »:

------------------------------------------------------

<http://nominatim.openstreetmap.org/reverse?format=xml&lat=45.126059&lon=5.879126&zoom=18&addressdetails=887>

------------------------------------------------------

Retourne :

------------------------------------------------------

<reversegeocode timestamp="Tue, 10 Jan 17 16:09:22 +0000" attribution="Data © OpenStreetMap contributors, ODbL 1.0. http://www.openstreetmap.org/copyright" querystring="format=xml&lat=45.126059&lon=5.879126&zoom=18&addressdetails=887">

<result place\_id="80388910" osm\_type="way" osm\_id="63069027" ref="La résidence" lat="45.12605105" lon="5.87859733093222" boundingbox="45.1259557,45.1261337,5.8782872,5.8788654">

La résidence, Place de Belledonne, Chamrousse, Grenoble, Isère, Auvergne-Rhône-Alpes, 38410, France

</result>

<addressparts>

<building>La résidence</building>

<pedestrian>Place de Belledonne</pedestrian>

<village>Chamrousse</village>

<county>Grenoble</county>

<state>Auvergne-Rhône-Alpes</state>

<postcode>38410</postcode>

<country>France</country>

<country\_code>fr</country\_code>

</addressparts>

</reversegeocode>

------------------------------------------------------

### Traitement à effectuer pour toutes les stations de ski du monde :

-Pour chaque enregistrement de « Skistation » en BDD :

-utiliser la requête de reverse geocoding avec les variables de latitude et de longitude

-Parser pour chaque réponse du webservice les champs :

• result

• **country**

-pour chaque réponse, il faut retrouver les enregistrements correspondants de la table « **Country** ». Utiliser le champ « **country** » pour retrouver l’enregistrement « Country ». Remplir la clé étrangère sur Skistation « countryid » avec la clé primaire extraite des enregistrements « Country ».

-----------------------------------------------------------------------------------------------------------------------------------------------------------

# Enrichissement de la base de données par les données Wikipédia

- Plusieurs formats de sortie sont disponibles : XML, JSON, XMLMF, JSONFM

## Obtenir la description de la page de résultat de Wikipédia

### Exemple de requête pour la ville de Brest

------------------------------------------------------  
https://fr.wikipedia.org/w/api.php?format=xml&action=query&prop=extracts&exintro=&explaintext=&titles=**Brest**  
------------------------------------------------------

### Résultat

------------------------------------------------------

<api batchcomplete="">

<query>

<pages>

<page \_idx="2482305" pageid="2482305" ns="0" title="Brest">

<extract xml:space="preserve">Brest ([bʁɛst], ) est une commune française, chef-lieu d’arrondissement du département du Finistère dans la région Bretagne. C'est un port important, deuxième port militaire en France après Toulon, à l'extrémité ouest de la Bretagne. Avec ses 139 386 habitants, les Brestois et les Brestoises, et une métropole comptant 207 210 habitants (2013), Brest est la première agglomération de l'ouest breton rayonnant sur un bassin de plus d'un million d'habitants, deuxième de région Bretagne derrière Rennes et la 25e commune la plus peuplée de France. Bien que Brest, aussi appelée la cité du Ponant, soit la ville la plus peuplée du Finistère, la préfecture du département est Quimper.

En 2013, son aire urbaine comptait 315 982 habitants, ce qui faisait d'elle la trentième aire urbaine de France et la deuxième de Bretagne administrative après Rennes. Au Moyen Âge, l'histoire de Brest se confond avec l'histoire de son château. Puis Richelieu en fait un port militaire. Brest se développera autour de son arsenal, jusqu'à la seconde moitié du XXe siècle. Fortement marquée par des bombardements lors de la Seconde Guerre mondiale, Brest a vu son centre historique presque entièrement renouvelé pendant la Reconstruction. À la fin du XXe et au début du XXIe siècle, la désindustrialisation de la ville s'accompagne du développement du secteur tertiaire.

Brest est aujourd'hui une ville universitaire importante de 23 000 étudiants. Son université est pluridisciplinaire et comporte une faculté de médecine ; par ailleurs Brest et son pays compte plusieurs grandes écoles : des écoles d'ingénieurs —IMT Atlantique, ENSTA Bretagne, ENIB et ISEN—, l'école d'agronomie ESIAB, l'école de commerce BBS, EURO-Institut d'Actuariat EURIA et l'École navale de formation des officiers de la Marine nationale. Brest est aussi un pôle de recherche important, axé sur la mer, avec notamment le plus grand des centres Ifremer, le CEDRE, le SHOM ainsi que l'Institut universitaire européen de la mer et l'Institut polaire français. Le Technopôle Brest-Iroise facilite la mise en œuvre de ces compétences scientifiques et techniques dans le domaine économique et industriel.

Brest a toujours été très marquée par son passé maritime : l'Académie de marine y fut fondée en 1752, le porte-avions Charles de Gaulle y fut construit, et tous les quatre ans, en juillet, Brest accueille la grande fête internationale de la mer, des bateaux et des marins.

</extract>

</page>

</pages>

</query>

</api>

------------------------------------------------------

### Traitement

Pour chaque CONTINENT, PAYS, VILLE, PLAGE, STATION DE SKI contenu dans la BDD :

-Lancer la requête en se basant sur le champ « frenchname »  
-Extraire du résultat le contenu du champs « extract »

-L’injecter dans l’enregistrement correspondant dans le champ « wikidescription »

## Obtenir les images des résultats de Wikipédia

### Exemple de requête pour la ville de Brest

------------------------------------------------------  
[https://fr.wikipedia.org/w/api.php?format=xml&action=query&prop=pageimages&pithumbsize=9000&titles=**Brest**](https://fr.wikipedia.org/w/api.php?format=xml&action=query&prop=pageimages&pithumbsize=9000&titles=Brest)

------------------------------------------------------

### Résultat

------------------------------------------------------

<api batchcomplete="">

<query>

<pages>

<page \_idx="2482305" pageid="2482305" ns="0" title="Brest" pageimage="Brest\_-\_Le\_Château\_-\_PA00089847\_-\_011.JPG">

<thumbnail source="**https://upload.wikimedia.org/wikipedia/commons/6/65/Brest\_-\_Le\_Ch%C3%A2teau\_-\_PA00089847\_-\_011.JPG**" width="4320" height="3240"/>

</page>

</pages>

</query>

</api>

------------------------------------------------------

### Traitement

Pour chaque CONTINENT, PAYS, VILLE, PLAGE, STATION DE SKI contenu dans la BDD :

-Lancer la requête en se basant sur le champ « frenchname »  
-Extraire dans le résultat l’adresse URL du champ « thumbnail source »  
-Télécharger l’image grâce à l’URL  
-Stocker en BDD le nom du fichier image dans le champ « wikipicture »

--------------------------------------------------------------------------------------------------------------------------------------

# Enrichissement de la base de données : attribuer à chaque pays son code international

-Le code de chaque pays permettra d’associer les données DATAWORLBANK (climat, criminalité...) à notre base de données.

### Traitement

-Accéder au GIT et parser le fichier « HolySearch/donnees\_plates/ CODES\_ISO\_ALPHA\_3\_PAYS/ **country\_iso\_alpha\_3\_code\_CSV.csv** ».

-Pour chaque ligne du fichier, parser les colonnes « Country » et « ISO ALPHA 3 code ».

-Pour chaque ligne parsée, utiliser la colonne « Country » pour rechercher l’enregistrement « **Country** » correspondant en BDD en utilisant le champ « englishname ».

-Une fois le pays contenu en base de données retrouvé, y renseigner le code pays (champ « ISO ALPHA 3 code » du fichier CSV) dans le champ « isoa3 ».

# Enrichissement de la base de données à partir des fichiers DATAWORLDBANK

## Température des pays

-Moyenne des températures annuelles des pays entre 1961 et 1999.

### Traitement

-Accéder au GIT et parser les colonnes « ISO\_3DIGIT », « Annual\_temp » et « qualif\_temperature » du fichier « HolySearch/donnees\_plates/ CLIMAT/**countries\_annual\_temperatures.csv** » :

-Pour chaque ligne du fichier, utiliser la colonne « ISO\_3DIGIT » pour retrouver en base de données l’enregistrement « **Country** » correspondant en utilisant le champ « isoa3 ».

-Injecter pour chaque pays la **température annuelle** («Annual\_temp ») obtenue lors du parsage du fichier **+ qualificatif de température («**qualif\_temperature »**)** dans les colonnes « temperature » et « temperaturelevel ».

Le qualificatif de température est déterminé selon les règles suivantes :

* SI température annuelle comprise entre -100 et 0,75 -> **très froid**
* SI température annuelle comprise entre 0,76 et 6,37 -> **froid**
* SI température annuelle comprise entre 6,38 et 15,22 -> **tempéré**
* SI température annuelle comprise entre 14,23 et 24,97 -> **chaud**
* SI température annuelle comprise entre 24, 98 et 100 -> **très chaud**

## Précipitations des pays

-Moyenne des précipitations annuelles des pays en mm entre 1961 et 1999.

### Traitement

-Accéder au GIT et parser les colonnes « ISO\_3DIGIT », « Annual\_precip » et « Qualificatif » du fichier « HolySearch/donnees\_plates/ CLIMAT/**countries\_annual\_precipitations.csv** ».

-Pour chaque ligne du fichier, utiliser la colonne « ISO\_3DIGIT » pour retrouver en base de données l’enregistrement « **Country** » correspondant en utilisant le champ « isoa3 ».

-Injecter pour chaque pays la **précipitation annuelle** (« Annual\_precip ») obtenue lors du parsage du fichier + **qualificatif de précipitation** (« Qualificatif  ») dans les champs « precipitation » et « precipitationlevel ».

Le qualificatif de précipitation est déterminé selon les règles suivantes :

* SI précipitation annuelle comprise entre 0 et 108,59 -> très sec
* SI précipitation annuelle comprise entre 108,60 et 436,25 -> sec
* SI précipitation annuelle comprise entre et 436,26 -> 1280,08 normal
* SI précipitation annuelle comprise entre 1280,09 et 2487,03 -> pluvieux
* SI précipitation annuelle comprise entre 2487,04 et 5000 -> très pluvieux

## Criminalité des pays

-Homicides intentionnels hors conflits armés (pour 100 000 personnes) en 2012.

### Traitement

-Accéder au GIT et parser les colonnes « Country Code », « Annee\_2012 » et « Qualificatif » du fichier « HolySearch/donnees\_plates/CRIMINALITE/countries\_annual\_criminality.csv ».

-Pour chaque ligne du fichier, utiliser la colonne « Country Code » pour retrouver en base de données l’enregistrement « **Country** » correspondant en utilisant le champ « isoa3 ».

-Injecter pour chaque pays la **criminalité annuelle** (« Annee\_2012 ») obtenue lors du parsage du fichier + **qualificatif de criminalité** (« Qualificatif ») dans les champs « criminality » et «criminalitylevel ».

Le qualificatif de précipitation est déterminé selon les règles suivantes :

* SI criminalité annuelle comprise entre 0 et 2,3 -> très faible criminalité
* SI criminalité annuelle comprise entre 2,4 et 3,9 -> faible criminalité
* SI criminalité annuelle comprise entre 4 et 11,5 -> criminalité normale
* SI criminalité annuelle comprise entre 11,6 et 23,9 -> forte criminalité
* SI criminalité annuelle comprise entre 24 et 1000 -> très forte criminalité

# Enrichissement de la vue des résultats depuis le site internet

-Documentation complète = https://developers.google.com/maps/documentation/embed/guide?hl=fr

-URL de base :

------------------------------------------------------

https://www.google.com/maps/embed/v1/**MODE**?key=**YOUR\_API\_KEY**&parameters

------------------------------------------------------

-Clé API = AIzaSyDpkNNOLMOicbOHu-pRzv6rJ8JtPAzEsJA

-Zoom = de 0 (le monde entier) à 21 (bâtiment individuel)

## place = vue carte

-Mode = « **place** »  
-Titre de l’emplacement = « **q** »  
-Remplacer les espaces par des « + »  
-Syntaxe de l’emplacement = « **emplacement\_1,ville\_emplacement+pays\_emplacement** »

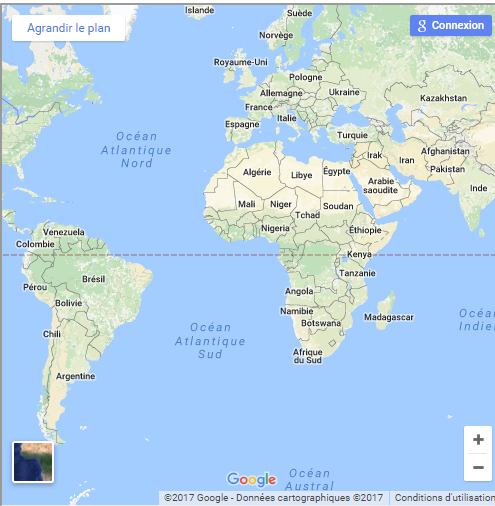
Exemple :

------------------------------------------------------

https://www.google.com/maps/embed/v1/**place**?&key=AIzaSyDpkNNOLMOicbOHu-pRzv6rJ8JtPAzEsJA&**q**=Paris,Paris+France

------------------------------------------------------

Rendu Iframe en l’absence de résultat :



## view = vue satellite

-Utiliser le mode « **view** »

-Positionnement par coordonnées géographiques = « **center** »  
-« **maptype=satellite** »

Exemple :

------------------------------------------------------

https://www.google.com/maps/embed/v1/**view**?key= AIzaSyDpkNNOLMOicbOHu-pRzv6rJ8JtPAzEsJA&**center**=48.8566101,2.3514992&**zoom**=15&**maptype=satellite**

------------------------------------------------------

## streetview = vue immersive en 3D

Exemple :

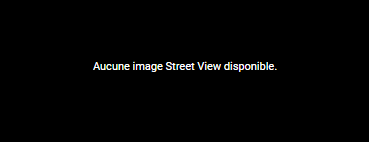
-mode = « **streetview** »  
-Coordonnées géographiques avec « **location** »

------------------------------------------------------

https://www.google.com/maps/embed/v1/**streetview**?key=AIzaSyDpkNNOLMOicbOHu-pRzv6rJ8JtPAzEsJA&**location**=48.8566101,2.3514992

------------------------------------------------------

Rendu Iframe en l’absence de résultat :



## POC HTML IFRAME

------------------------------------------------------

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Titre</title>

</head>

<body>

<iframe width="500" height="500"

src="https://www.google.com/maps/embed/v1/place?&key=AIzaSyDpkNNOLMOicbOHu-pRzv6rJ8JtPAzEsJA&q=Paris,Paris+France">

</iframe>

<iframe width="500" height="500"

src="https://www.google.com/maps/embed/v1/view?key= AIzaSyDpkNNOLMOicbOHu-pRzv6rJ8JtPAzEsJA&center=48.8566101,2.3514992&zoom=15&maptype=satellite">

</iframe>

<iframe width="500" height="500"

src="https://www.google.com/maps/embed/v1/streetview?key=AIzaSyDpkNNOLMOicbOHu-pRzv6rJ8JtPAzEsJA&location=48.8566101,2.3514992"

></iframe>

</body>

</body>

</html>

------------------------------------------------------