# CDA & DWWM – Brief – « Rest countries – Partie 4 »

## Référentiels de compétences

Les compétences professionnelles du référentiel « CDA » couvertes par ce projet sont :

* AT1 / CP4 : Développer la partie frontend d’une interface utilisateur web (Adapter)

Les compétences professionnelles du référentiel « DWWM » couvertes par ce projet sont :

* AT1 / CP2 : Réaliser une interface utilisateur web statique et adaptable (Adapter)
* AT1 / CP3 : Développer une interface utilisateur web dynamique (Adapter)

## Modalités pédagogiques

Travail et rendu individuel mais il n’est pas interdit de s’entraider, bien au contraire !

## Projet

Réaliser un prototype Web *responsive* en anglais qui permet de consulter des informations sur les pays de la région Europe. Nous souhaitons à terme pouvoir consulter ces informations sous forme de liste, de tableau, de *cards* puis disposer d’une géolocalisation de chaque pays.

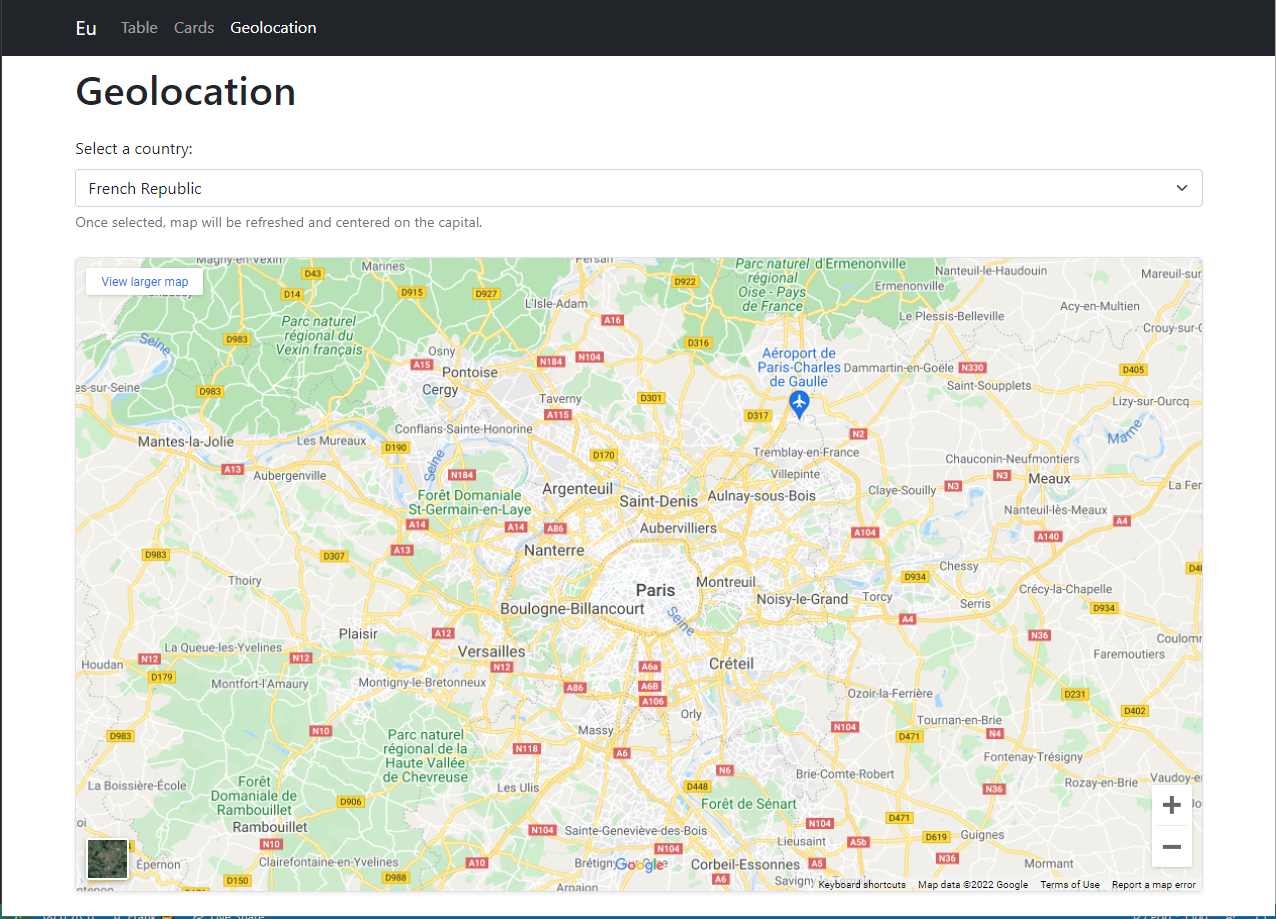
Les données sur les pays proviennent de l’API publique et gratuite Rest countries.

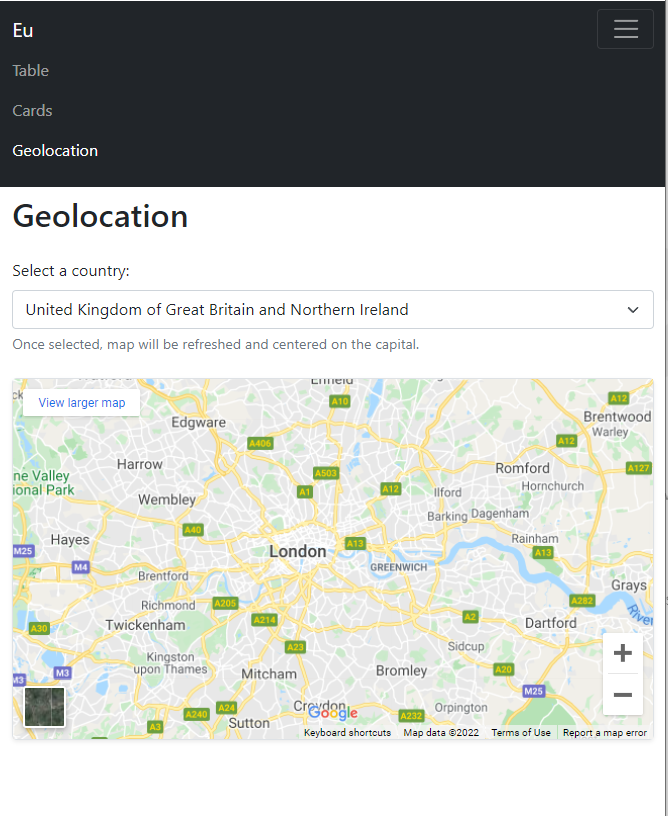
### Partie 4 du projet

* Poursuivre sur le même projet « **rest-countries** »
* Implémenter une page **geolocation.html** avec son script **geolocation.js** qui répond aux spécifications suivantes :
  + Le titre du document : *Eu - Geolocation*
  + La liste déroulante est alimentée dynamiquement au chargement de la page à partir de Rest countries avec le nom officiel des pays affichés
  + Quand on sélectionne un pays dans la liste déroulante, la carte, uniquement la carte et pas toute la page, est mise à jour pour afficher le pays concerné centré sur sa capitale grâce aux informations de géolocalisation disponibles dans Rest countries
  + Quand on arrive sur la page, le pays par défaut est la France, en conséquence présélectionnée dans la liste déroulante et la carte affichée est celle de la France centrée sur Paris (cf. 1ère capture ci-dessous)
  + Penser écoconception et performance, faire au mieux pour limiter la quantité de données récupérées sur Rest countries uniquement en fonction des besoins de la fonctionnalité en prenant soin de récupérer les données uniquement quand on en a besoin
  + S’appuyer sur le code ISO 2 caractères pour récupérer les informations nécessaires sur la capitale d’un pays
* Consigne pour la carte :
  + Utiliser Google Maps Embed API, pour ce faire il faut un compte Cloud, gratuit sans même une carte bancaire pour utiliser Google Maps Embed API, afin d’obtenir une clef d’API
  + Point de départ : <https://developers.google.com/maps/documentation/embed/get-api-key>
  + Lire la documentation pour apprendre à intégrer une carte et trouver le moyen le plus simple pour la mettre à jour quand on change de pays. Dans les exemples de code peut-être que tout n’est pas nécessaire pour répondre à notre besoi, faire en sorte que la carte soit responsive toujours en s’appuyant sur Bootstrap 5
  + Certaines options sont disponibles dans l’API, il faut en préciser deux dans cet écran, à savoir un niveau de zoom de 10 et forcer la langue anglaise pour le nom des lieux sur la carte

### Captures de l’écran

Le résultat à obtenir doit être strictement identique aux captures suivantes, il y en a plusieurs pour améliorer la lisibilité mais il s’agit bien de la même page :





### Environnement technique

* Git et GitHub
* Navigateur avec outils de développement (notamment pour vérifier l’adaptabilité de la page, accès à la console JavaScript)
* VSCode
* HTML 5, JavaScript (ES6)
* Bootstrap 5 (aucun style personnel, 100% Bootstrap)

## Livrables

* Le projet dans le même **repository** « GitHub » (public)
* Le projet doit toujours être **déployé** dans « GitHub pages »
* Donner le lien vers le repository dans le rendu du brief correspondant sur « **Simplonline** »

## Modalités d’évaluation

* Le rendu à l’écran doit **correspondre exactement** au visuel et au contenu de la maquette (captures écran)
* **Qualité** du code : organisation des dossiers et fichiers, organisation du code (ordre des importations CSS et JavaScript), asynchronie, formatage et nommage notamment
* Le code HTML (source et généré dynamiquement) doit être **valide** selon le [W3C Markup Validation Service](https://validator.w3.org/)