# CDA & DWWM – Brief – « Rest countries – Partie 1 »

## 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 (Imiter)

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

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

## 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.

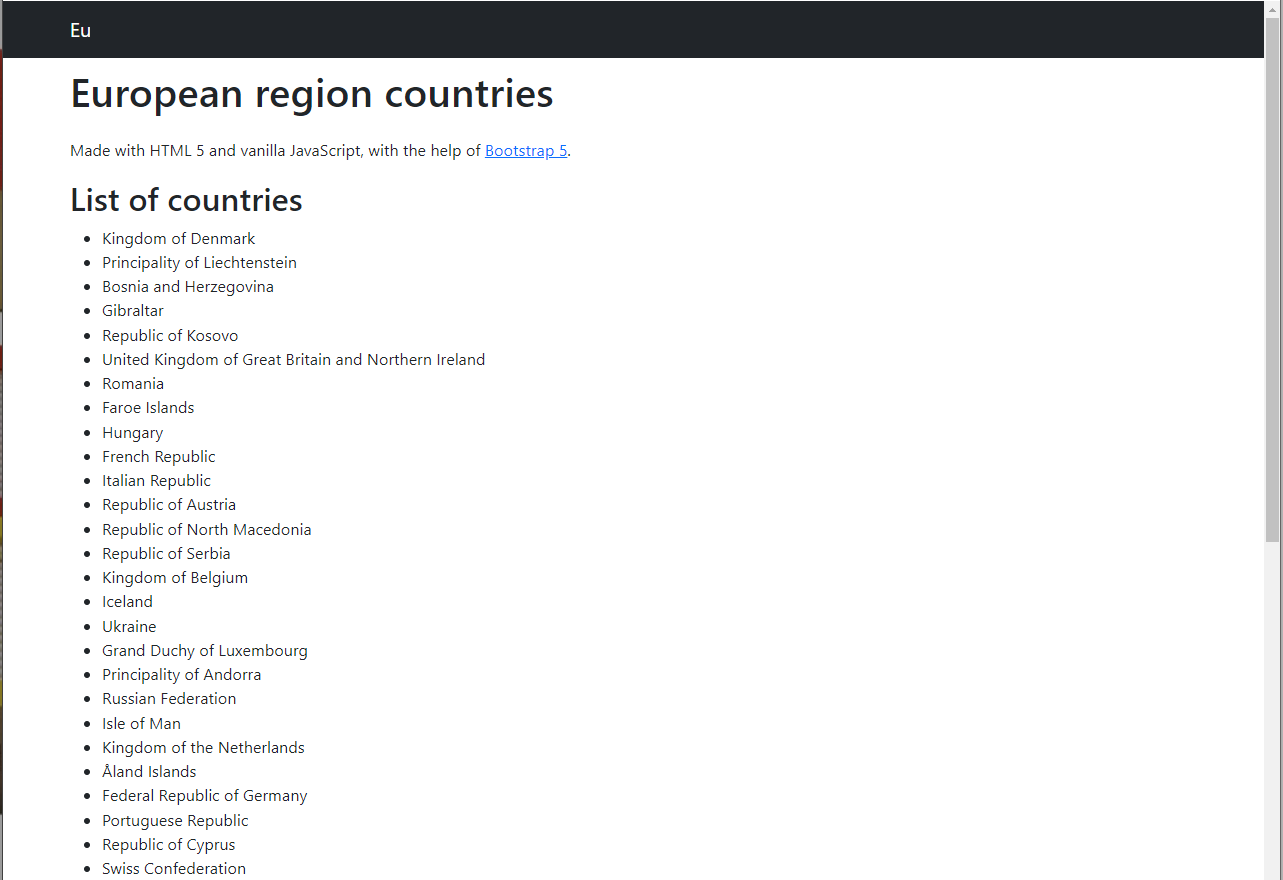
Les données sur les pays proviennent de l’API publique et gratuite Rest countries.

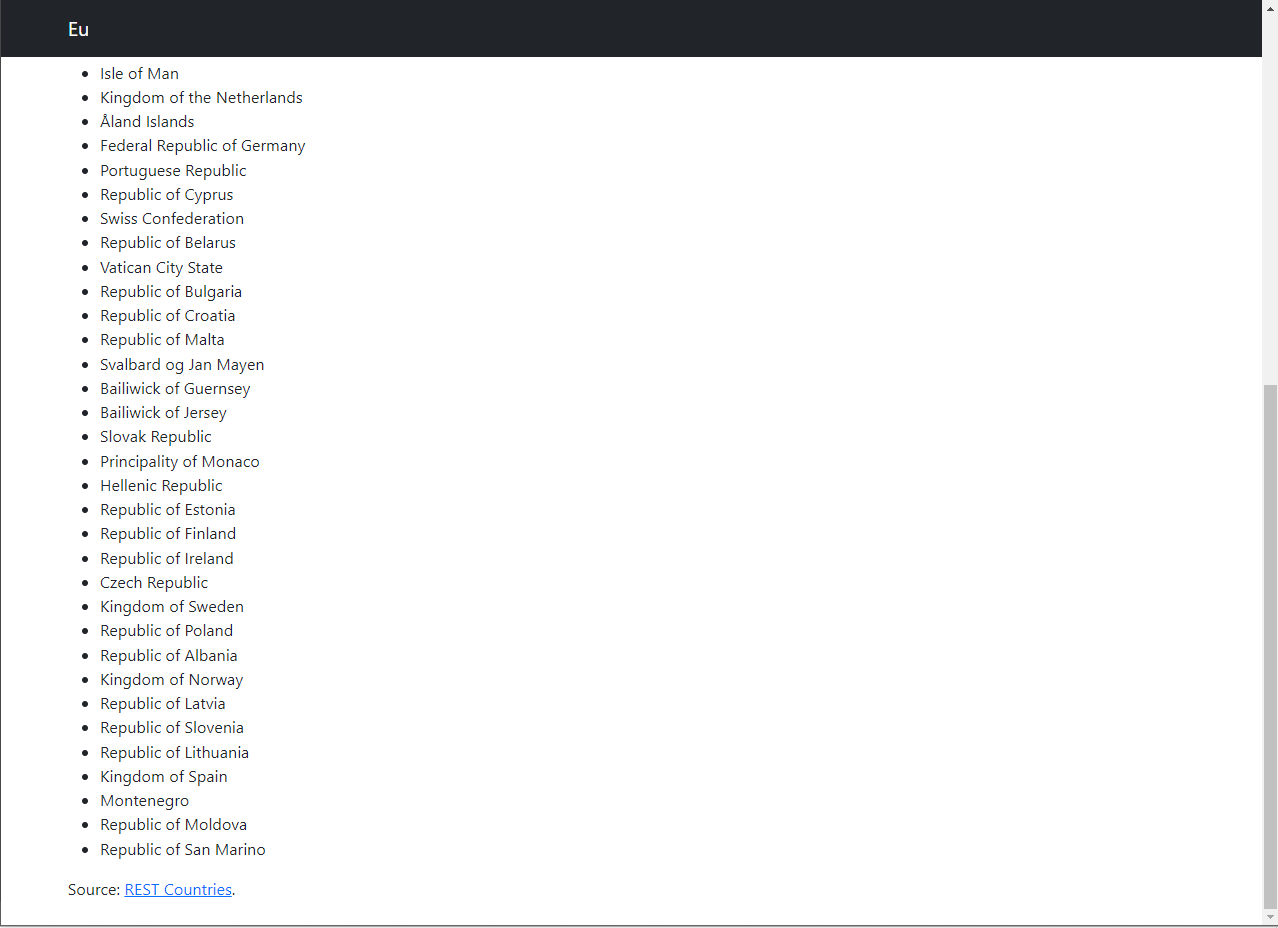
### Partie 1 du projet

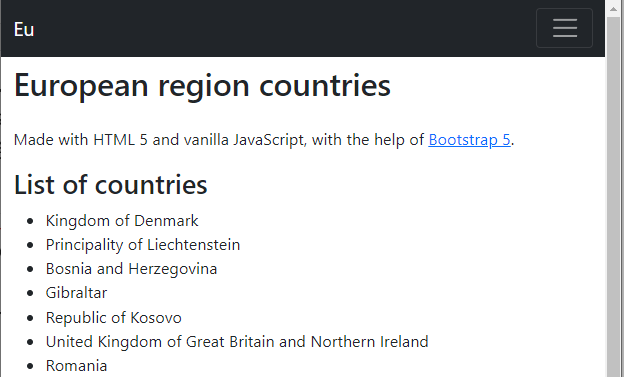
* Initialiser un nouveau projet « **rest-countries** »
* Implémenter une page **index.html** qui répond aux spécifications suivantes :
  + Titre du document : *Eu - European region countries*
  + Doit avoir une icône de favoris (*favicon*)
  + Une barre de navigation en haut toujours visible même quand on défile la page
  + La barre de navigation ne contient à ce stade que le logo « Eu » en texte avec un lien qui renvoie vers la page index.html
  + Le style de la barre de navigation est dark
  + Comportement en termes d’adaptabilité :
    - Tout le contenu de la page doit être centré pour écran xl, seul la couleur de fond de l’entête est visible sur toute la largeur
    - Quand on passe le breakpoint md le « burger menu » apparaît même s’il est vide pour le moment car aucun élément dans la barre de navigation
  + Titre principal de la page : *European region countries*
  + Un paragraphe qui contient le texte : *Made with HTML 5 and vanilla JavaScript, with the help of Bootstrap 5.*
  + Dans le paragraphe précédent il doit y avoir un lien externe (autre onglet) qui renvoie vers https://getbootstrap.com/docs/5.0/getting-started/introduction/ sur le texte *Bootstrap 5*
  + La liste des pays de la région Europe précédée du titre : *List of countries*
  + Après la liste un paragraphe cite la source des pays avec le texte : *Source: REST Countries.*
  + Dans le paragraphe précédent il doit y avoir un lien externe (autre onglet) qui renvoie vers : https://restcountries.com/
* Implémenter un script **index.js** qui répond aux spécifications suivantes :
  + Est exécuté au chargement de la page index.html
  + Au chargement une fonction asynchrone render() est appelée
  + Cette fonction récupère la liste des **noms officiels** de pays de la région Europe depuis l’API Rest countries (sauvegarder le JSON dans un fichier **index.json** et l’ajouter dans un sous-dossier json du projet)
  + Cette fonction itère sur le tableau des pays récupéré en JSON pour construire dynamiquement la liste à afficher dans la page index.html à l’aide d’un gabarit (*template literal*)
  + Le résultat de cette construction doit être inséré dans la balise ul correspondante
  + Pas de gestion des erreurs

### 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)

## Ressources

En **gras** les ressources principales pour ce brief.

* **https://restcountries.com/**
* **https://jsonformatter.curiousconcept.com/**
* **https://getbootstrap.com/docs/5.0/getting-started/introduction/**
* **https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global\_Objects/Array/forEach**
* https://developer.mozilla.org/fr/docs/Web/API/Window/load\_event
* **https://dmitripavlutin.com/javascript-fetch-async-await/**
* https://developer.mozilla.org/fr/docs/Web/API/Fetch\_API/Using\_Fetch
* https://developer.mozilla.org/fr/docs/Learn/JavaScript/Asynchronous/Async\_await#r%C3%A9%C3%A9criture\_du\_code\_des\_promesses\_avec\_asyncawait
* **https://www.growingwiththeweb.com/2014/02/async-vs-defer-attributes.html**
* https://developer.mozilla.org/fr/docs/Learn/JavaScript/Asynchronous/Introducing
* https://developer.mozilla.org/fr/docs/Web/API/Document\_Object\_Model
* https://developer.chrome.com/docs/devtools/dom/
* **https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Template\_literals**

## Livrables

* Le projet partagé dans un nouveau **repository** « GitHub » (public)
* Le projet doit 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/)