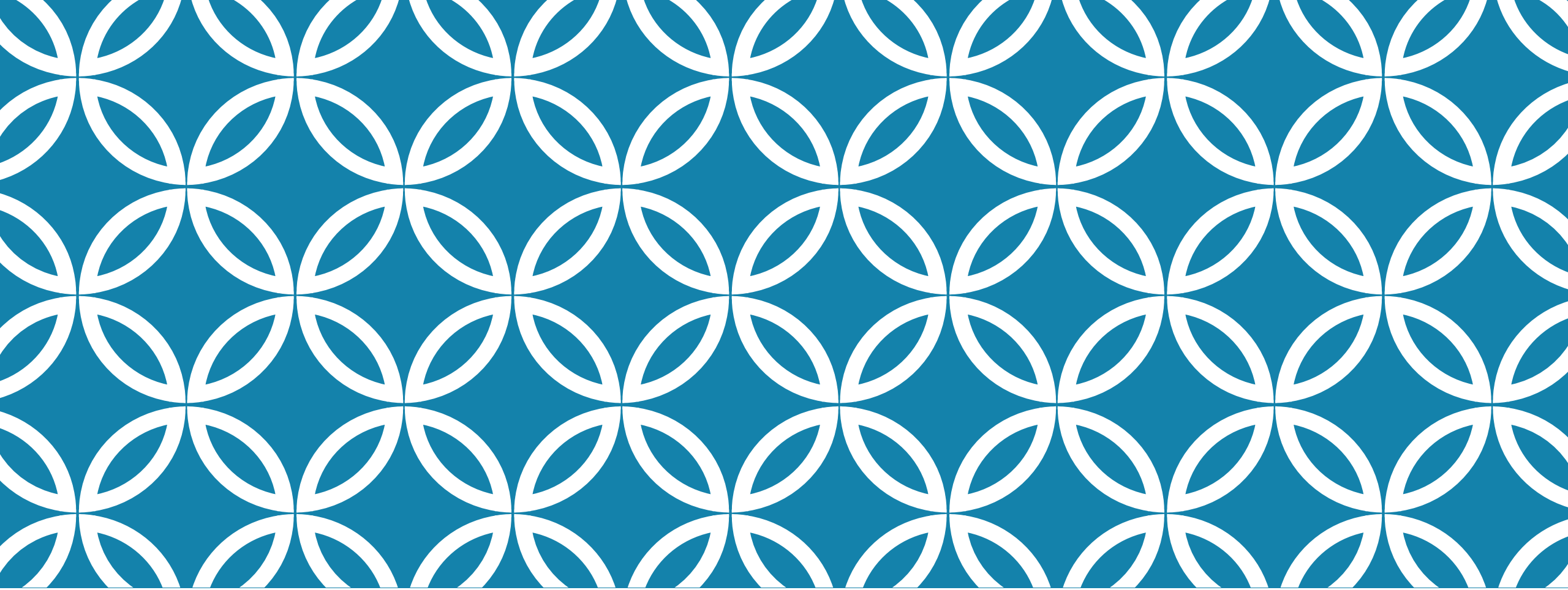
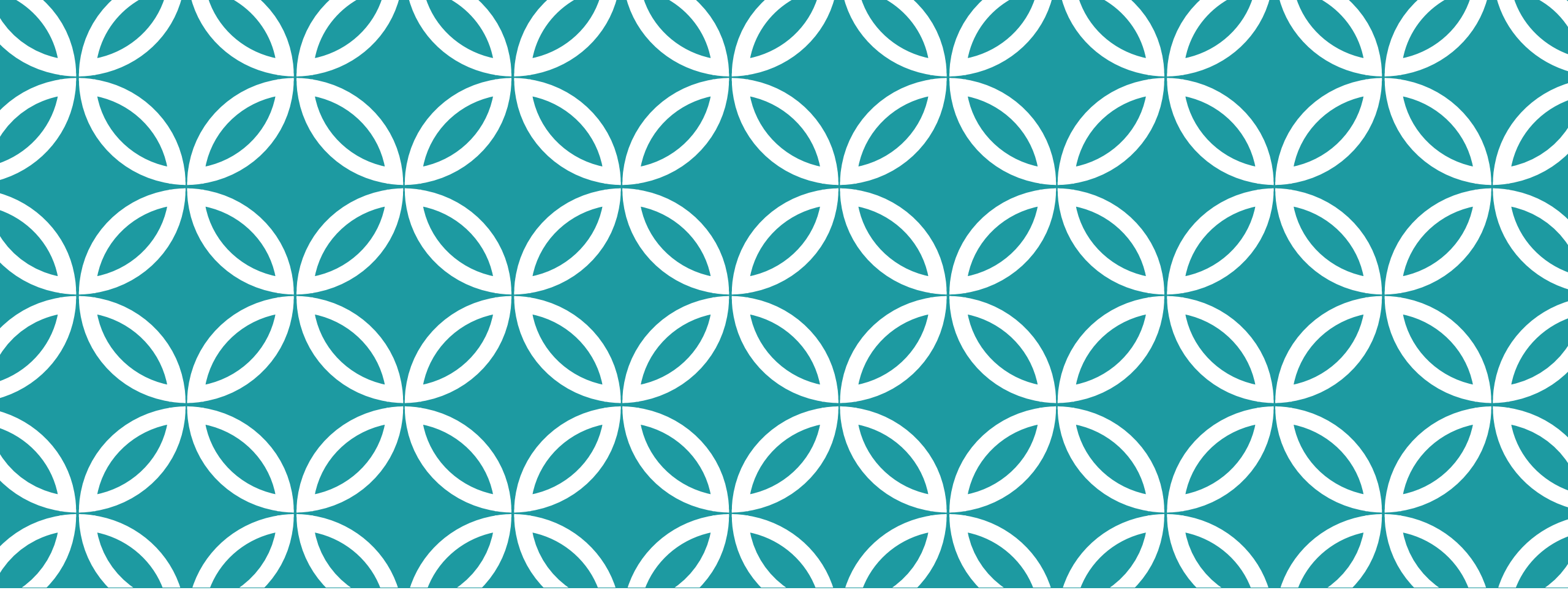


Projet Big Data

Programmation Large Échelle

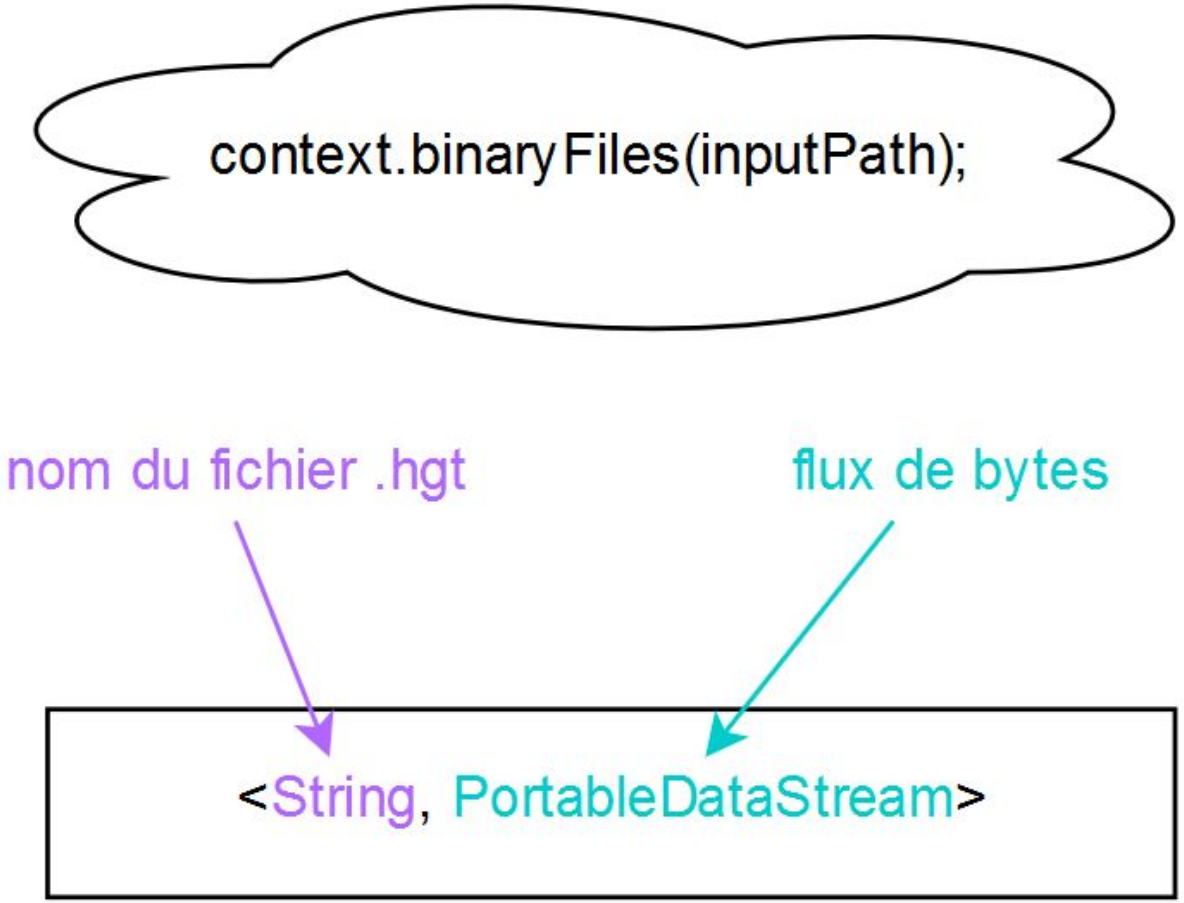
Louis Leduc
Manon Philippot

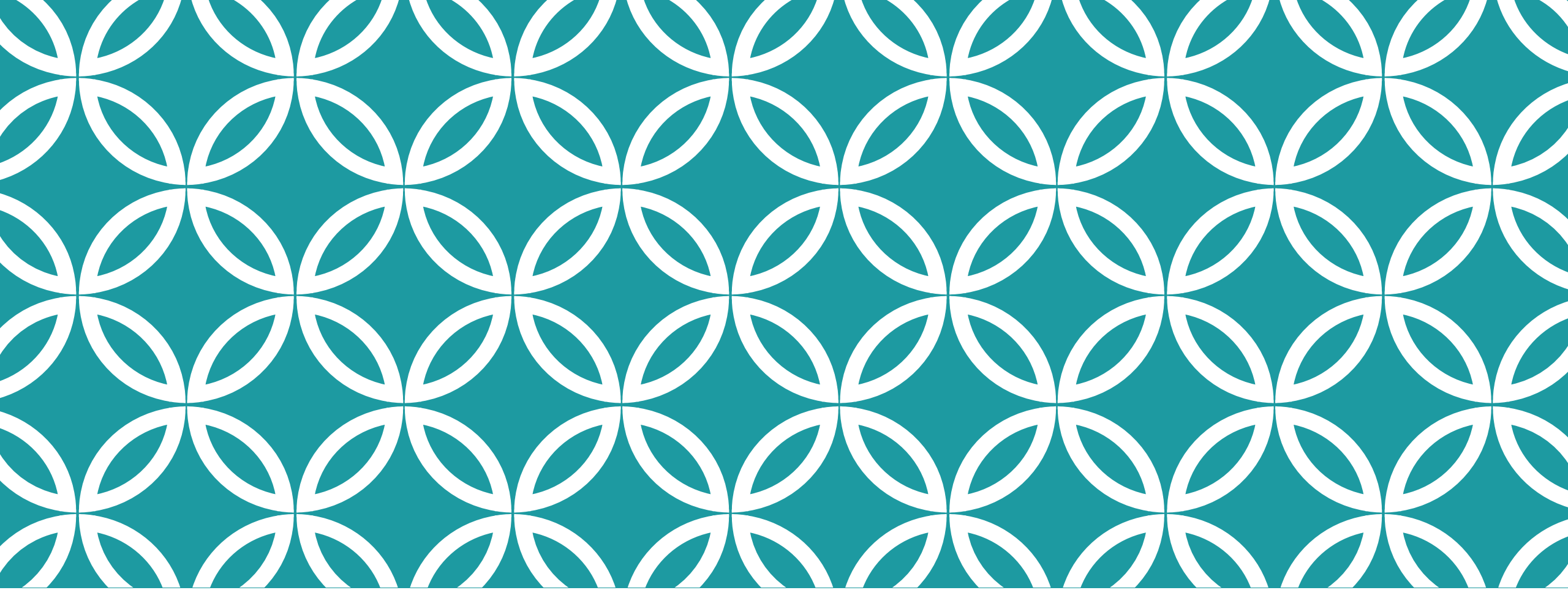
Master 2 Informatique



Solution implémentée

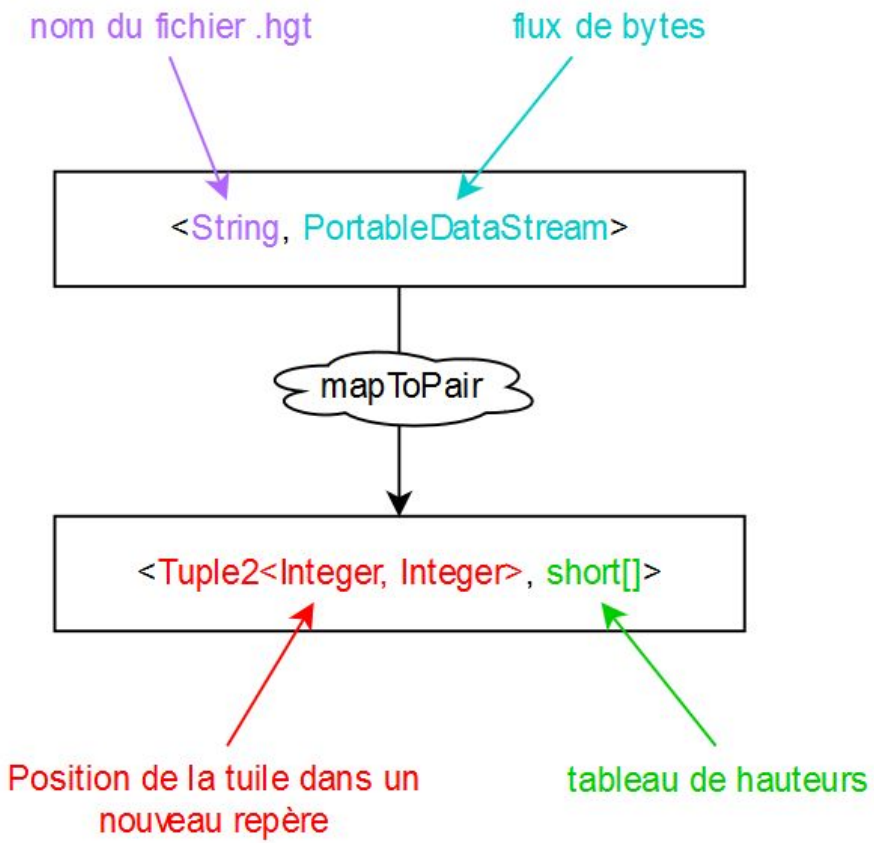
Importation des données

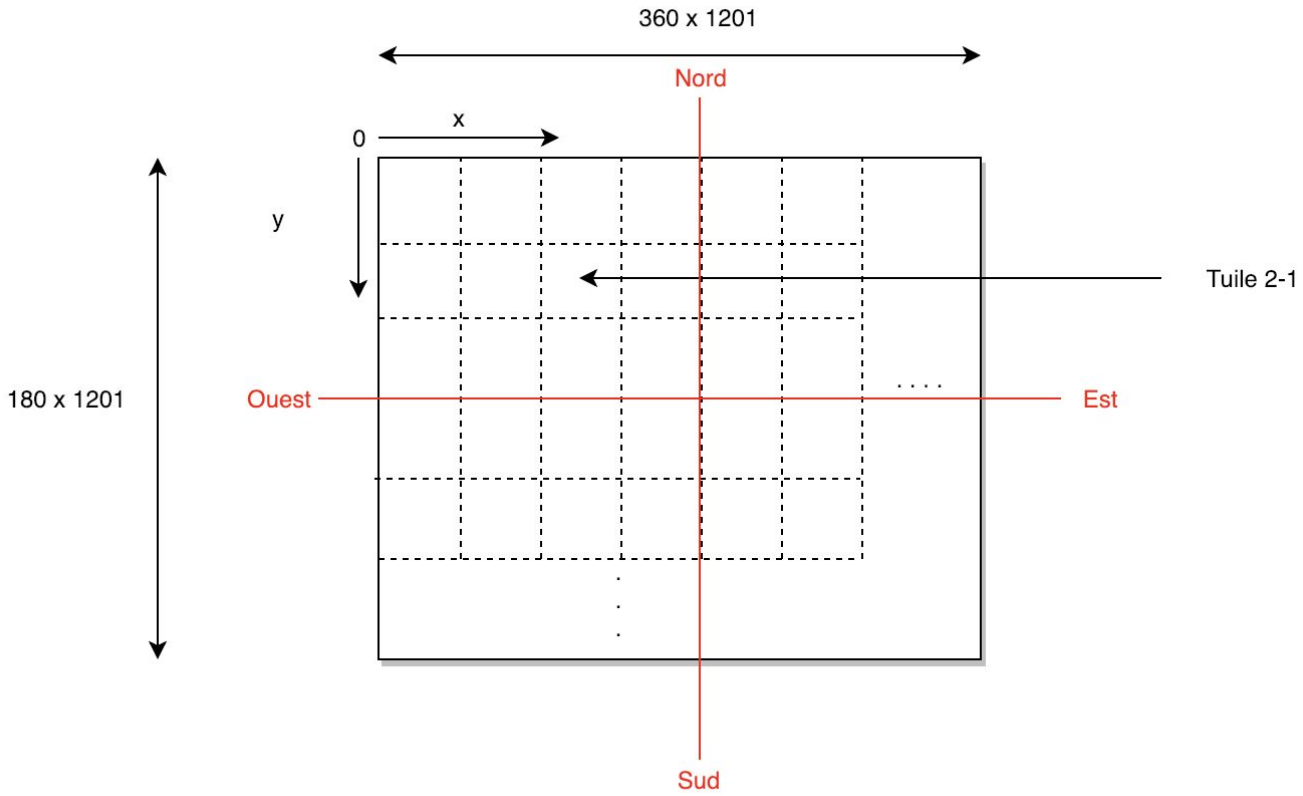




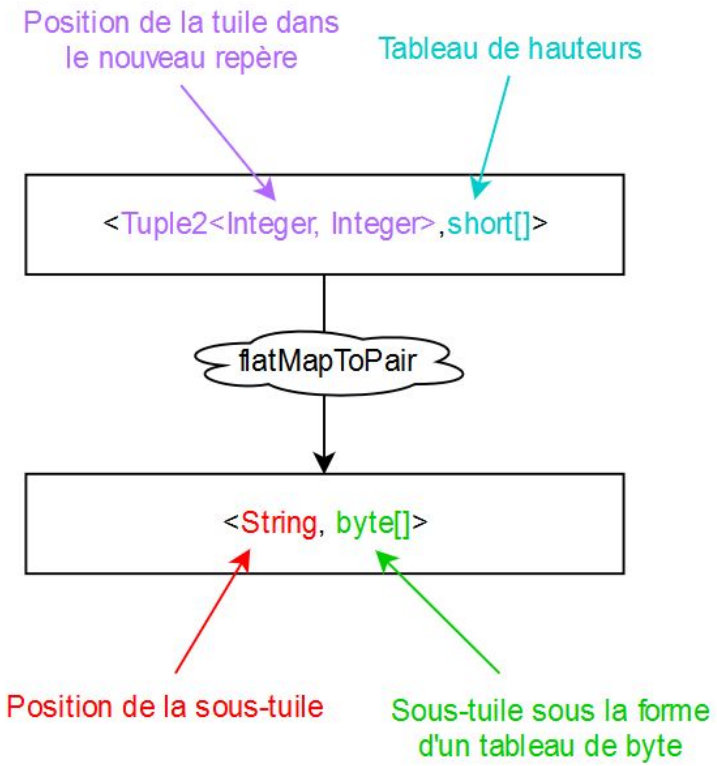
Calcul du zoom maximal

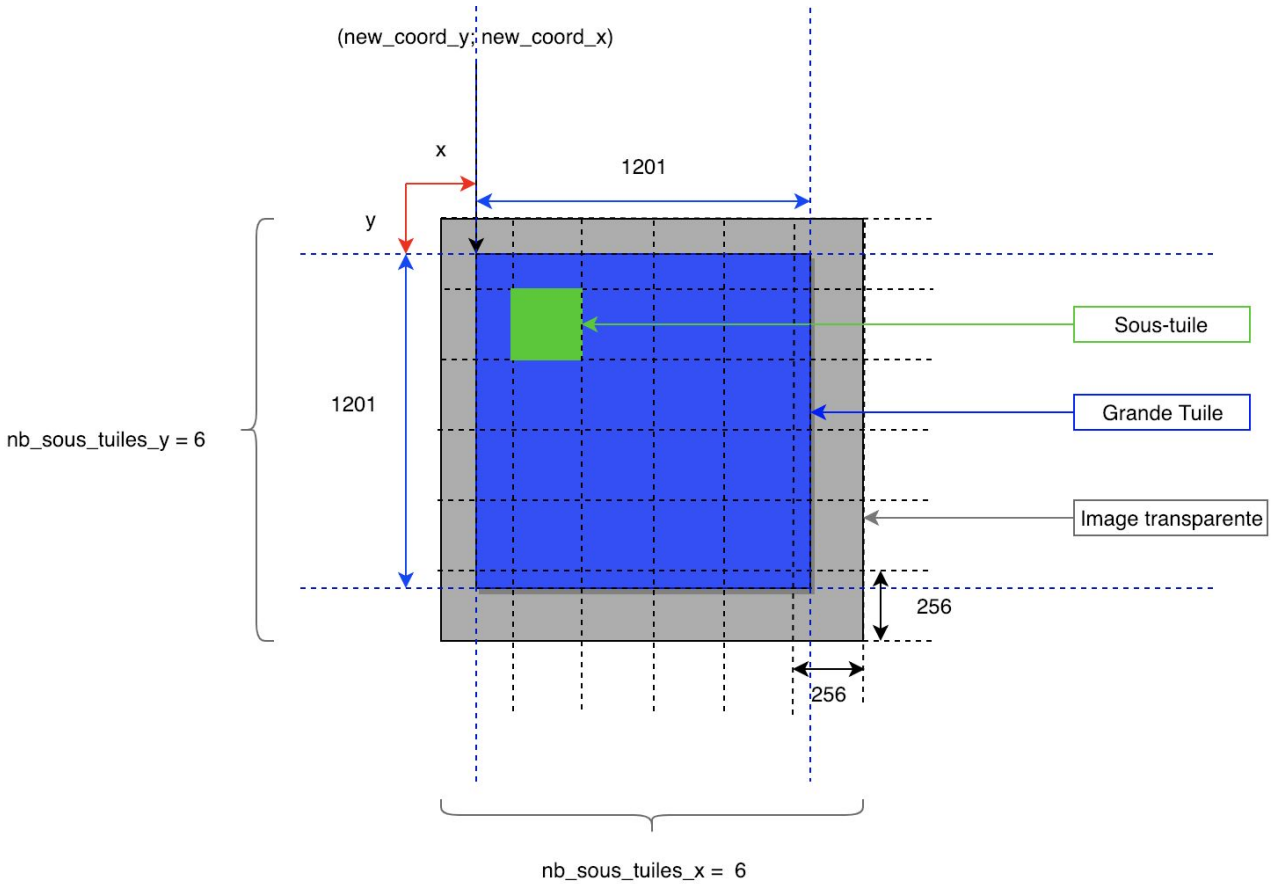
Mappeur

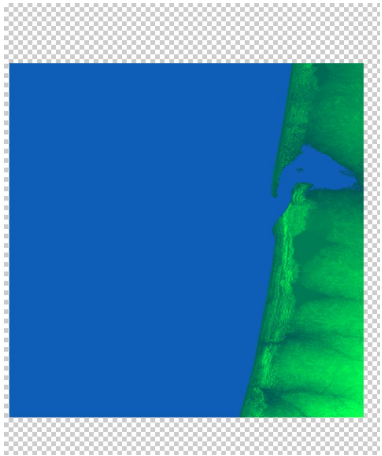

nom du fichier .hgt
flux de bytes
<String, PortableDataStream>
mapToPair
<Tuple2<Integer, Integer>, short[]>
Position de la tuile dans un nouveau repère
tableau de hauteurs


360 x 1201
Nord
x
0
y
180 x 1201
Tuile 2-1
Ouest
Est
Sud

Calcul des sous-tuiles

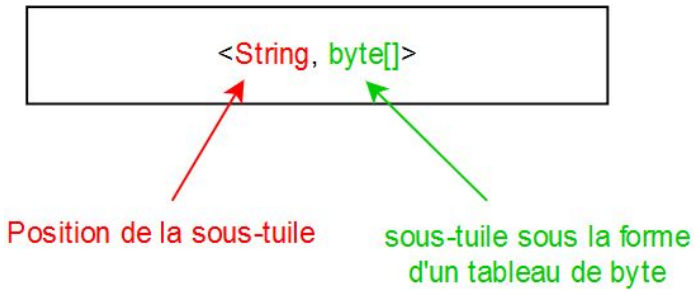


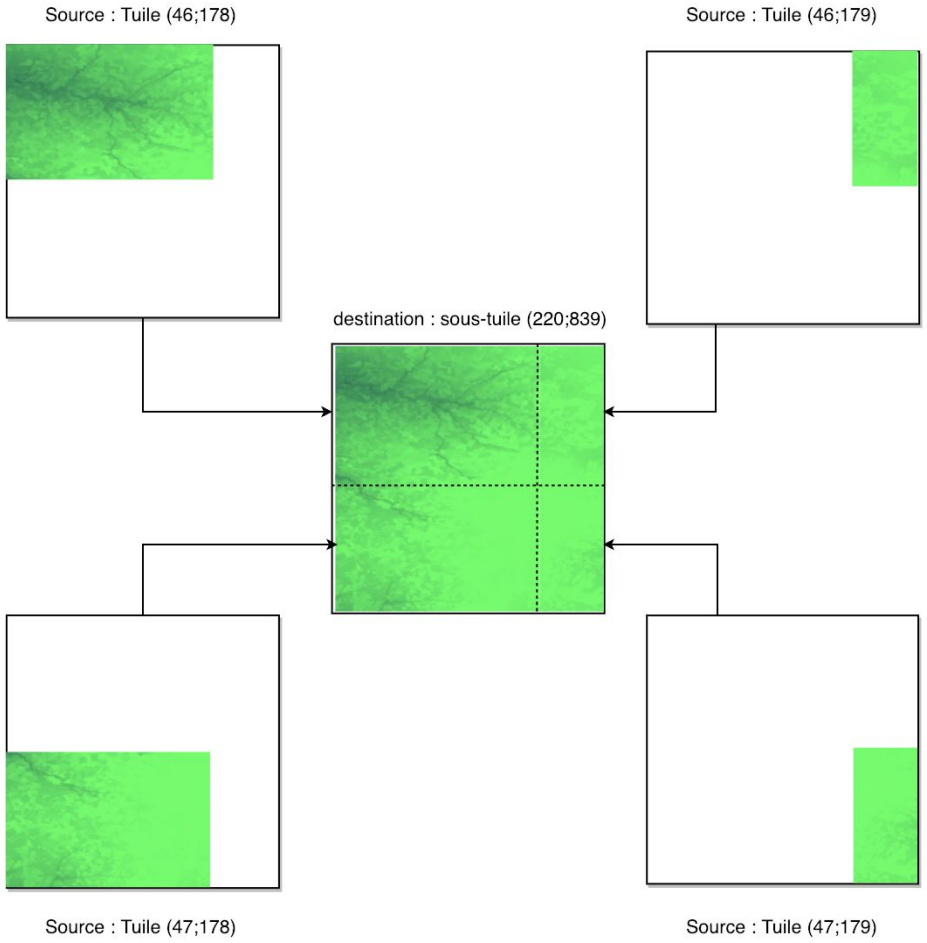


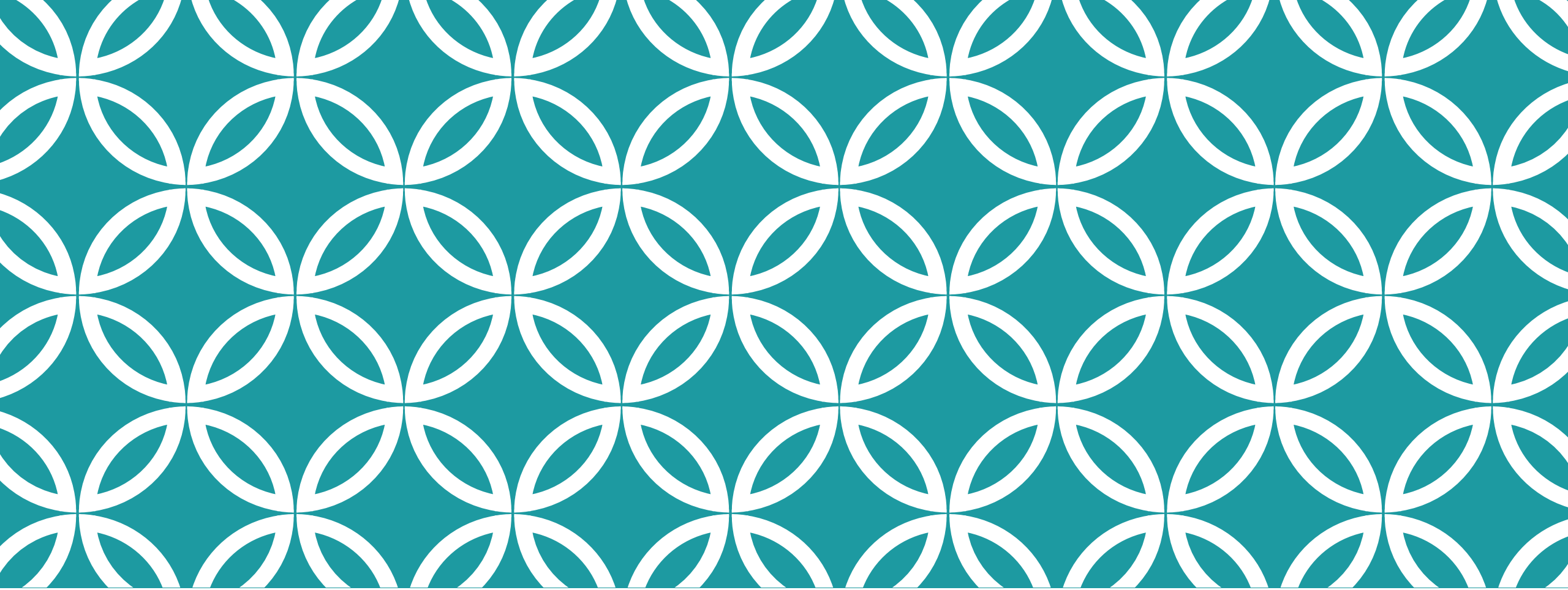


Reducer

- Les sous-tuiles chevauchant plusieurs grandes tuiles ont la même clé.
- ReduceByKey pour les réunir et former une sous-tuile complète.

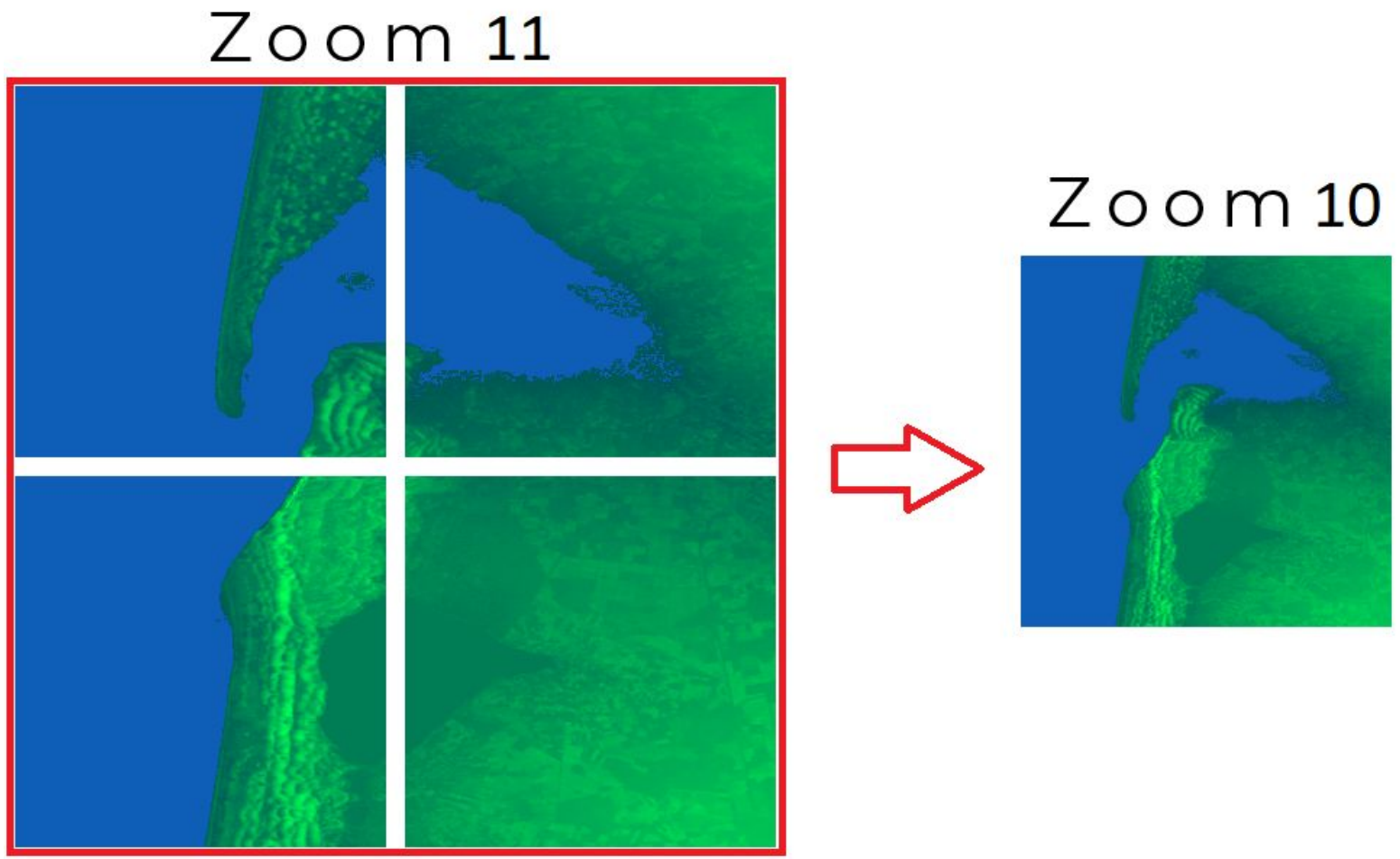




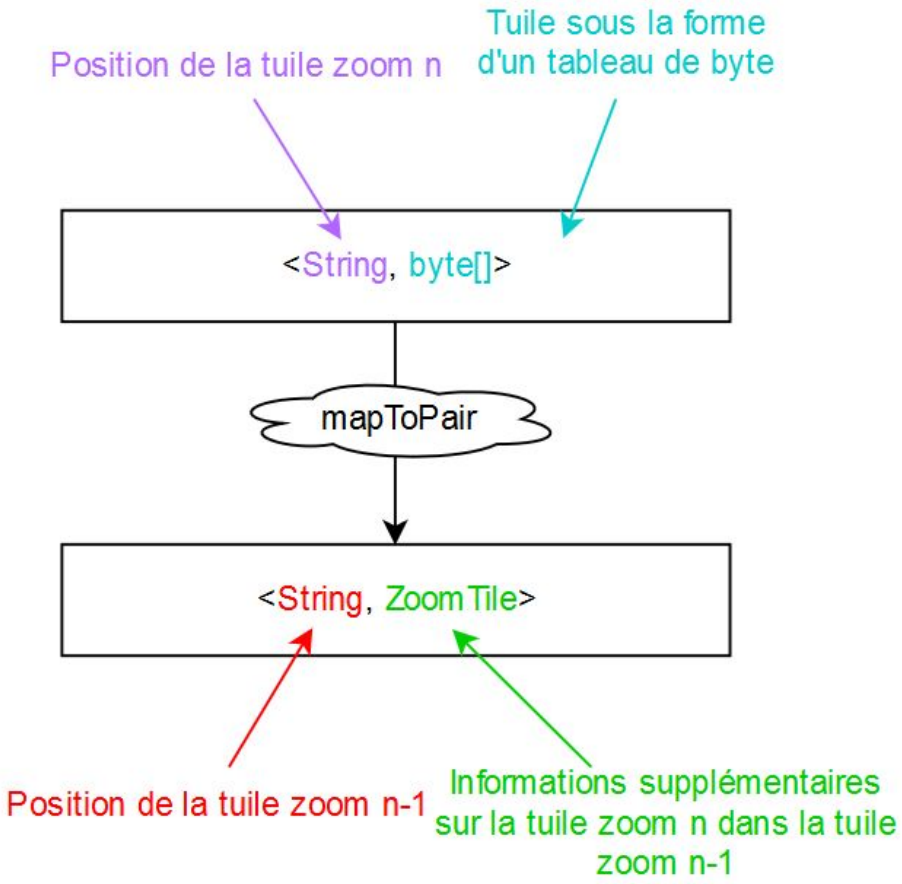


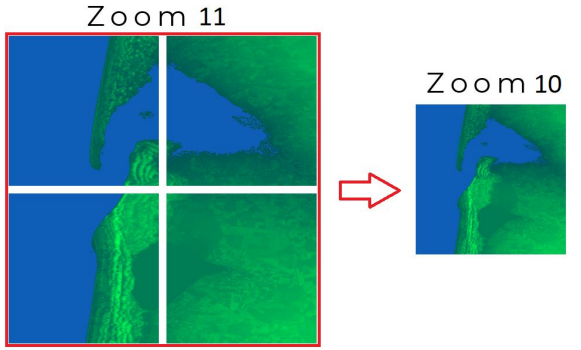
Calcul des zooms inférieurs

Principe



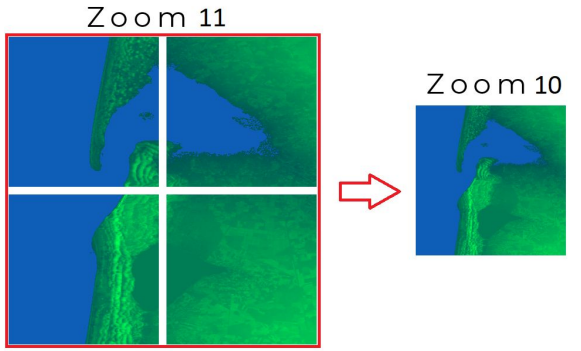
Mappeur





- Calcul de la position de la tuile au zoom n-1
- Calcul des coordonnées du pixel à partir de laquelle la tuile au zoom n devra être à l'intérieur de la tuile zoom n-1
  - (0; 0)
  - (0; 256)
  - (256; 0)
  - (256; 256)
- On stocke les informations supplémentaires de la tuile dans un ZoomTile

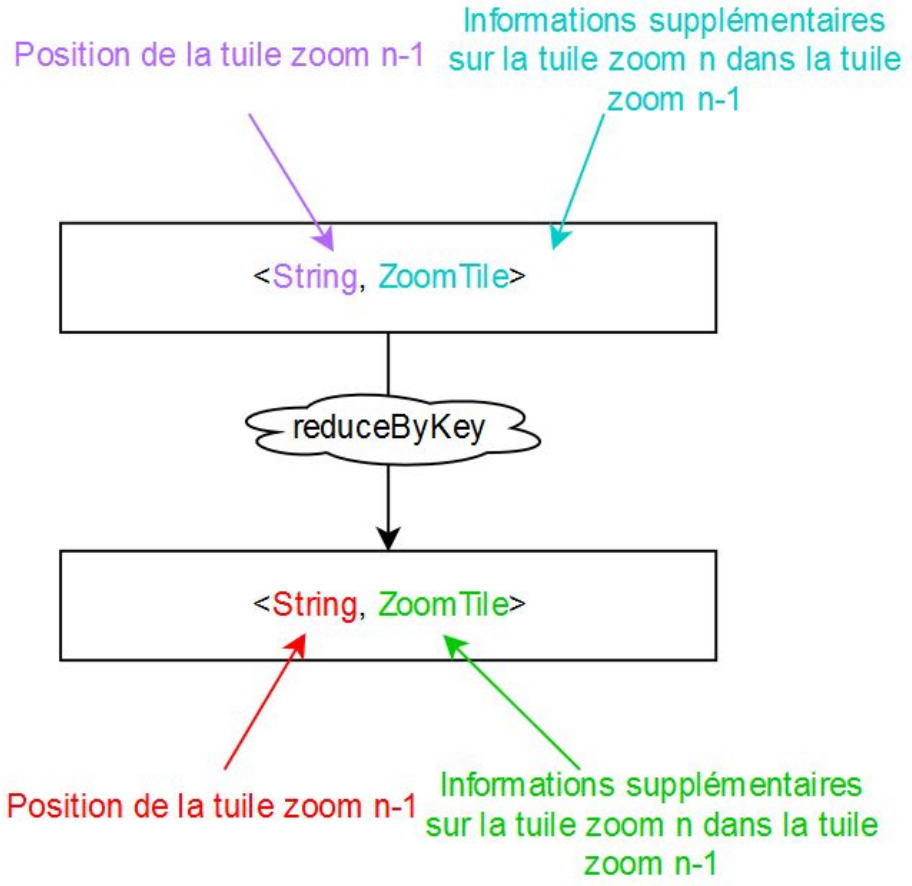
Classe zoomTile

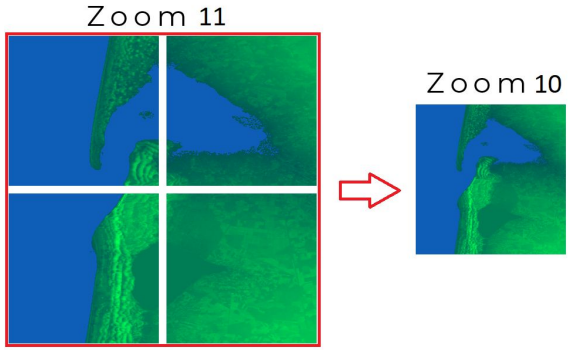


- Tuile sous la forme d'un tableau de bytes
- Position de la tuile à laquelle elle va appartenir
- Coordonnées à l'intérieure de la tuile à laquelle elle va appartenir
- Booléen : tuile traité dans le reducer ?
- Sérialisable

```
public class ZoomTile implements Serializable {

    private static final long serialVersionUID = 2113177953332563490L;
    private byte[] image;
    private int xTile;
    private int yTile;
    private int xPos;
    private int yPos;
    private boolean processedInReducer;

    public ZoomTile(byte[] image, int xTile, int yTile, int xPos, int yPos, boolean processedInReducer) {
        this.image = image;
        this.xTile = xTile;
        this.yTile = yTile;
        this.xPos = xPos;
        this.yPos = yPos;
        this.processedInReducer = processedInReducer;
    }

    ...
}
```

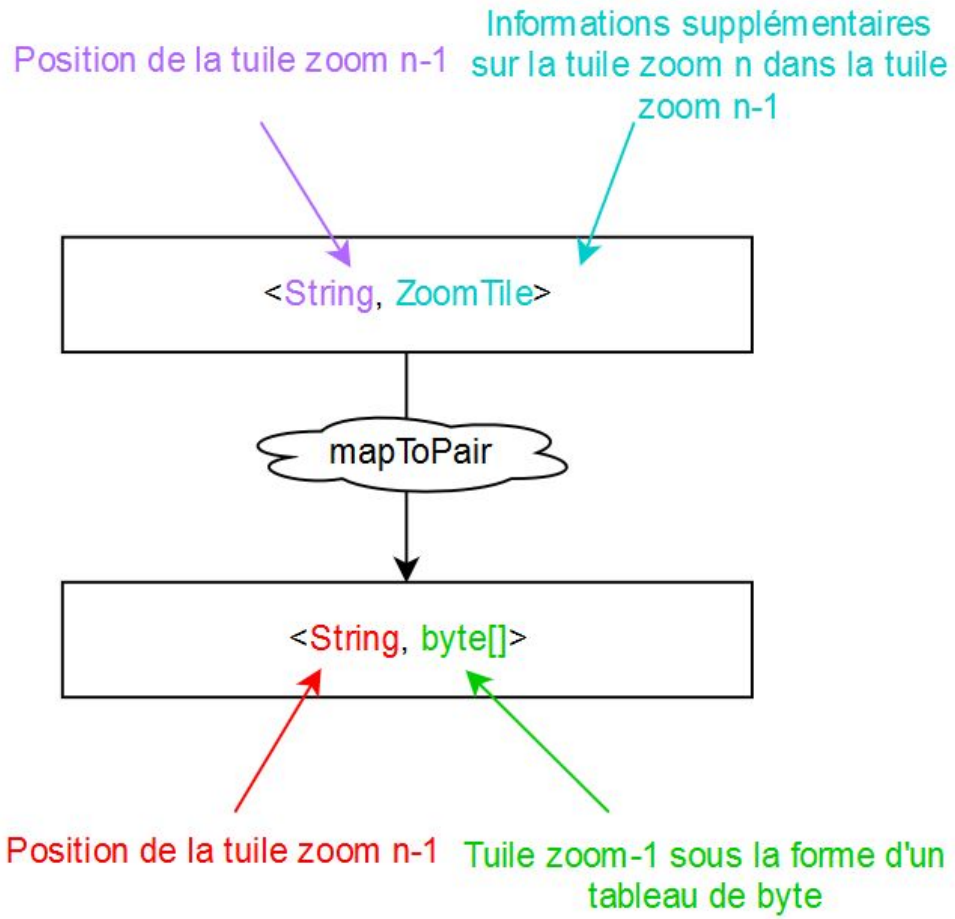
Reducer



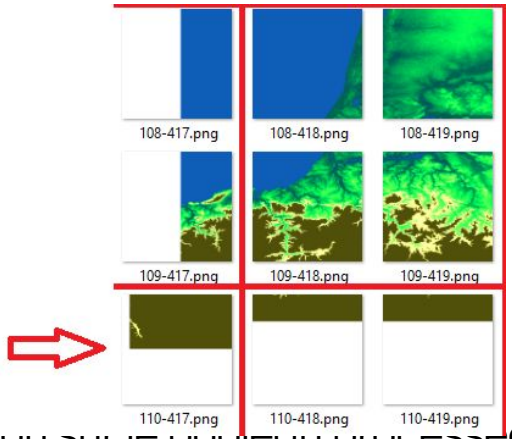


- Récupère toutes les tuiles ayant la même clé (4 généralement, mais des fois moins)
- Créé une image transparente de 512 *512
- Copie les 4 tuiles à l'intérieur selon les coordonnées contenues dans le ZoomTile

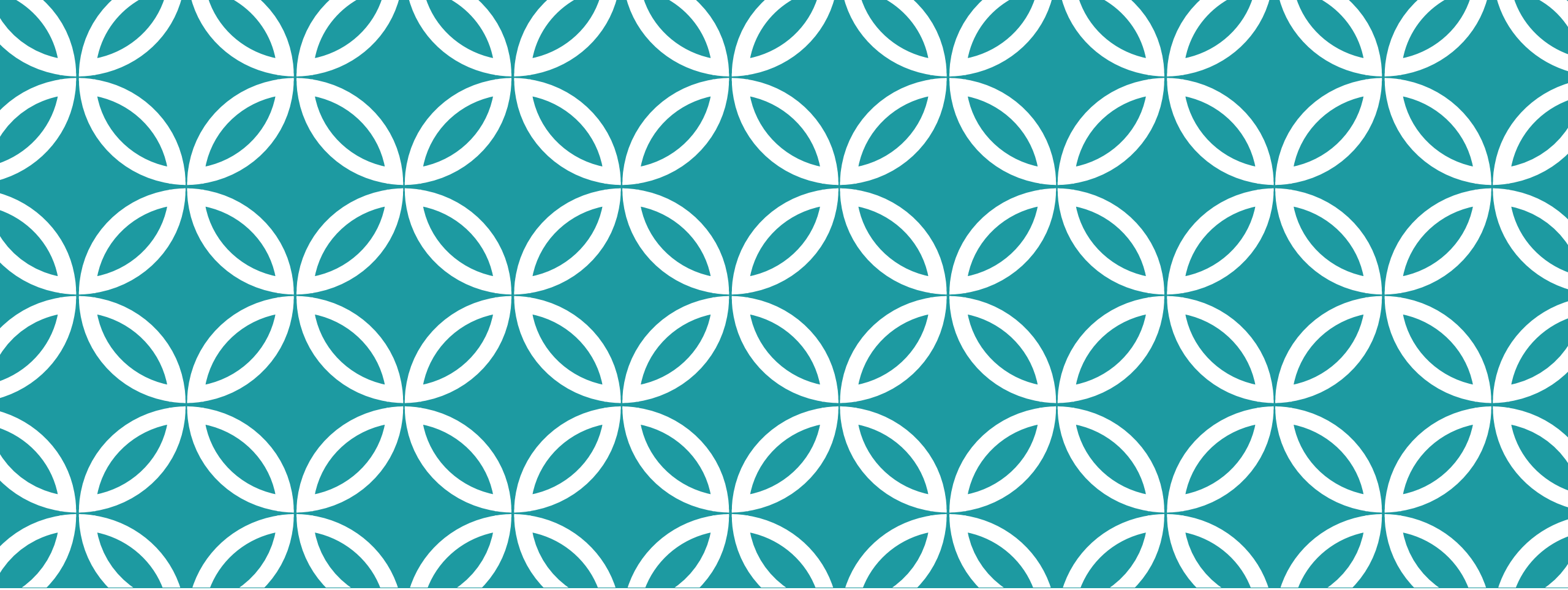
Redimensionner l'image



- Redimensionner la tuile de 512 * 512 en une tuile de 256 * 256
- Le reducer ne traite pas clés qui n'ont qu'une seule tuile associée : des tuiles ne sont pas traitées !



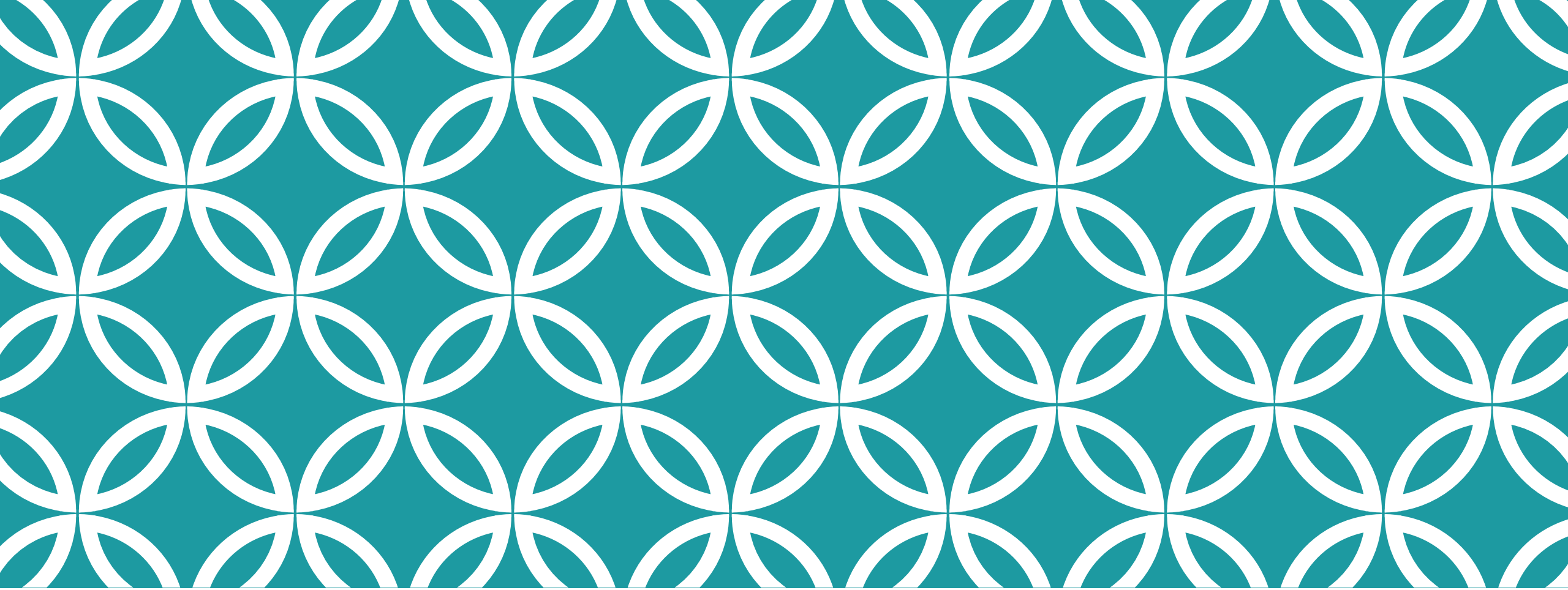
- Condition sur le booléen processedInReducer du ZoomTile pour rectifier cet oubli, puis redimensionnement



Cache

Cache

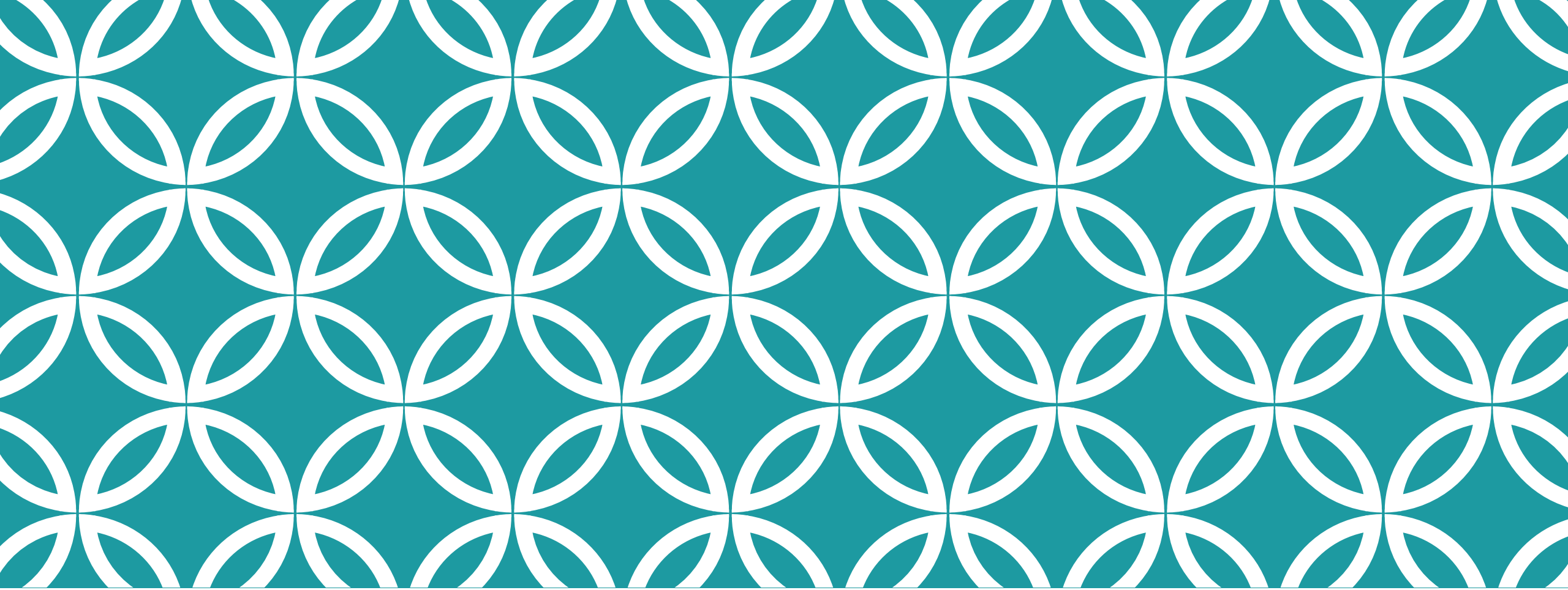
```
private static void processAndHBase(JavaPairRDD<String, byte[]> RDD0) throws IOException{
        JavaPairRDD<String, byte[]> previousRDD = RDD0;

        previousRDD.cache();

        HBase.addZoomHbase(previousRDD, 0);

        for (int i = 1; i < 12; i++){
                JavaPairRDD<String, ZoomTile> zoomMapRDD = previousRDD.mapToPair(ZoomFunction.zoomMap);
                JavaPairRDD<String, ZoomTile> zoomReducedRDD = zoomMapRDD.reduceByKey(ZoomFunction.zoomReducer);
                JavaPairRDD<String, byte[]> zoomResizedRDD = zoomReducedRDD.mapToPair(ZoomFunction.zoomResize);
                zoomResizedRDD.cache();
                int zoom = i;
                HBase.addZoomHbase(zoomResizedRDD, zoom);
                previousRDD.unpersist();
                previousRDD = zoomResizedRDD;
        }
}
```



HBase

HBase

team-rocket-ml				
row	zoom_0	zoom_1	.....	zoom_11
			.....	



API

API

GET

/Project/webapi/tile/{x}/{y}/{z}

Usage and SDK Samples

Curl | Java | Android | Obj-C | JavaScript | C# | PHP | Perl | Python

```
curl -X GET "http://localhost:8080/Project/webapi/tile/{x}/{y}/{z}"
```

Parameters

Path parameters

Name	Description
x*	Integer (int32) *la coordonnée en x de la tuile* Required
y*	Integer (int32) *la coordonnée en y de la tuile* Required
z*	Integer (int32) *le niveau de zoom de la tuile* Required

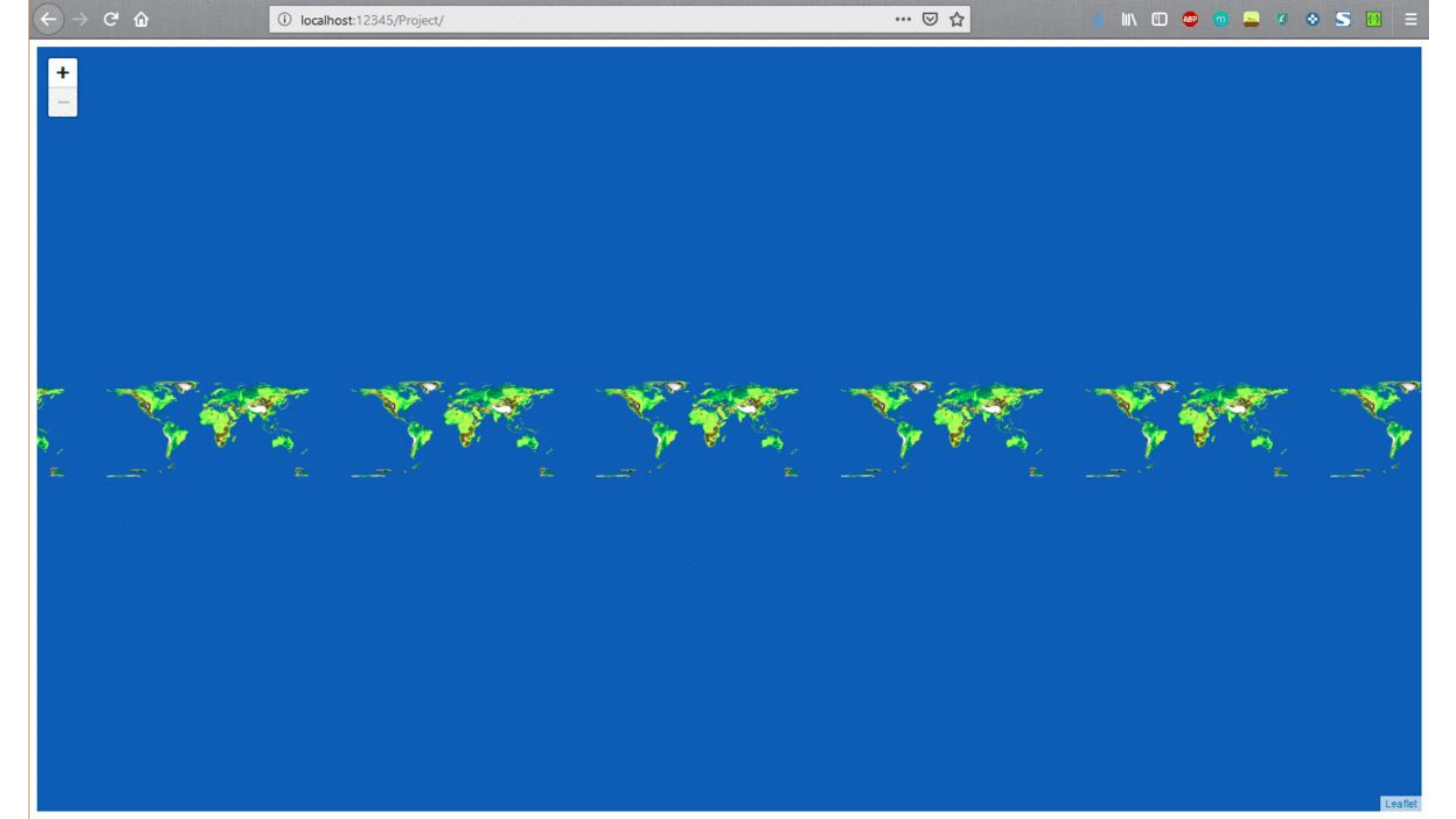
Responses

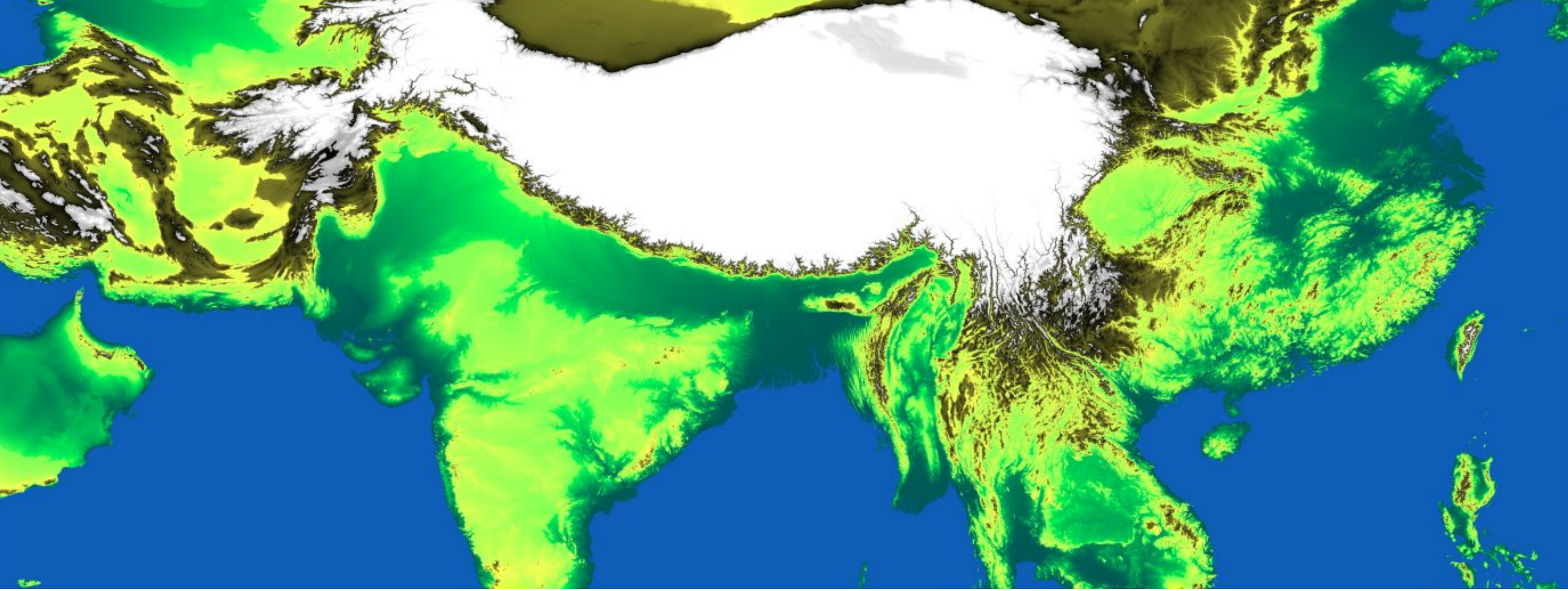
Status: 200 - la tuile a bien été récupérée et a été transmise au front

Resultats

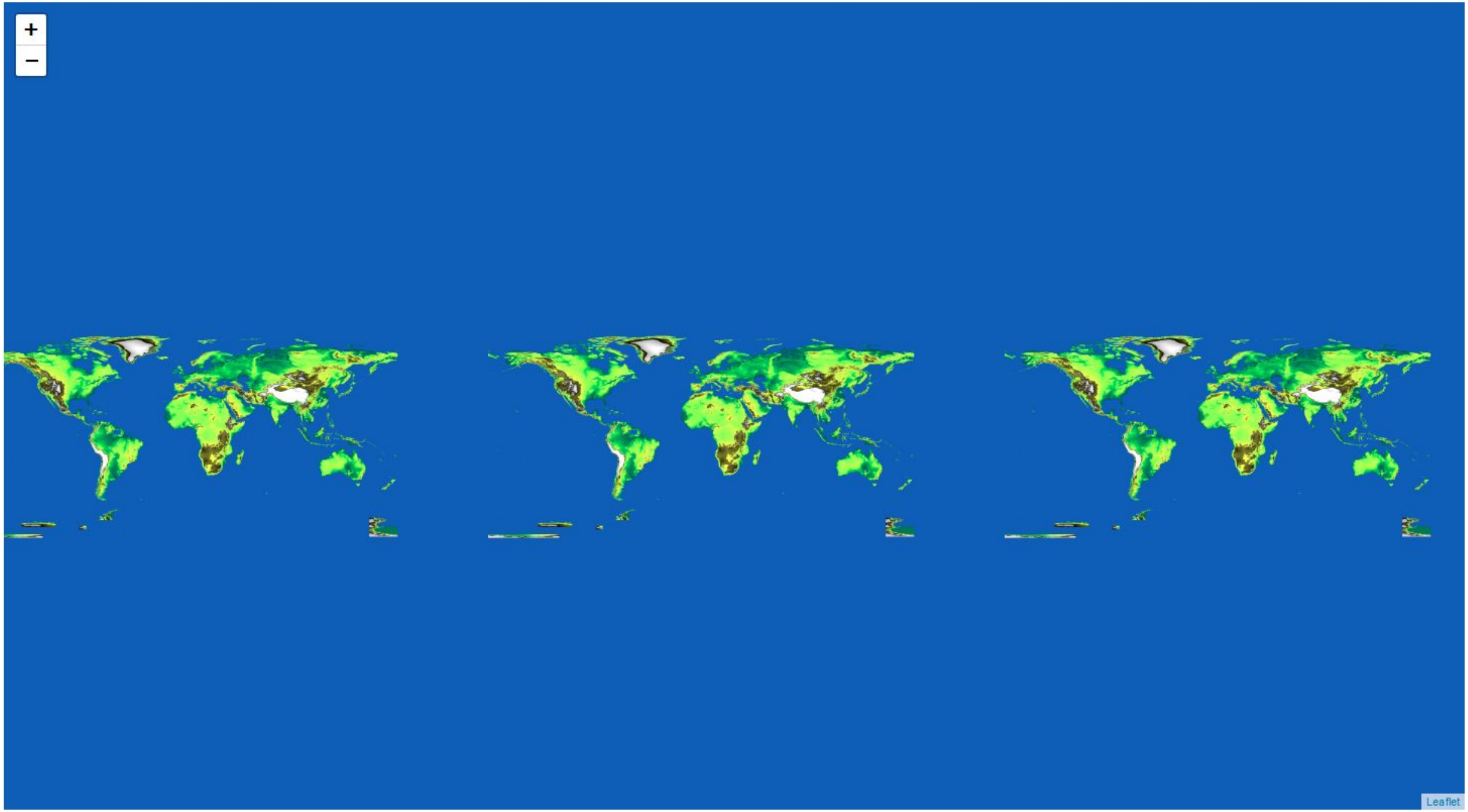
Résultats

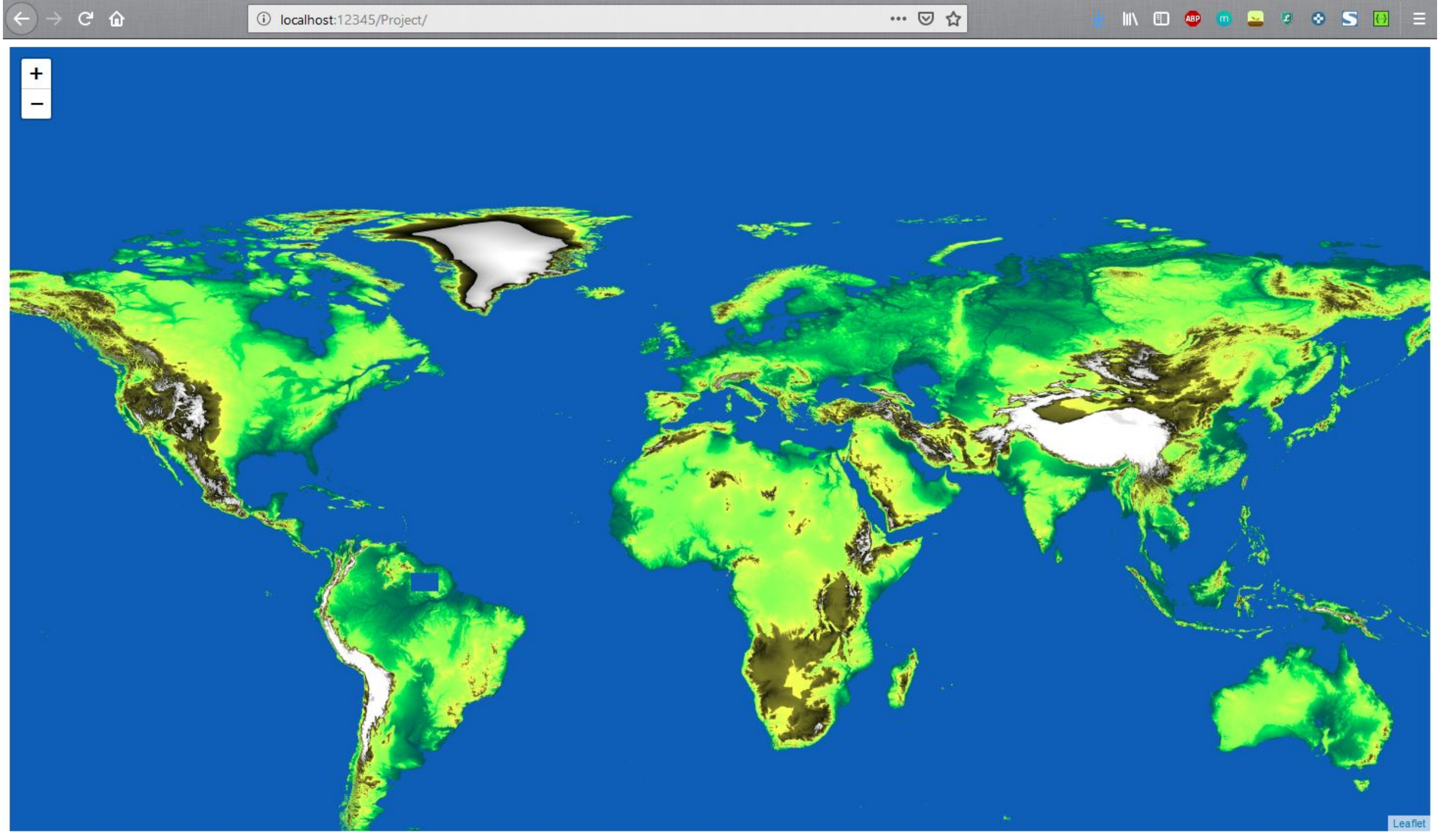
- 6 executors, 6 Go de mémoire : 1h49
- table ‘team-rocket-ml’ : 519426 lignes / 17,9 Go
- API fonctionnelle
- Map Leaflet fonctionnelle

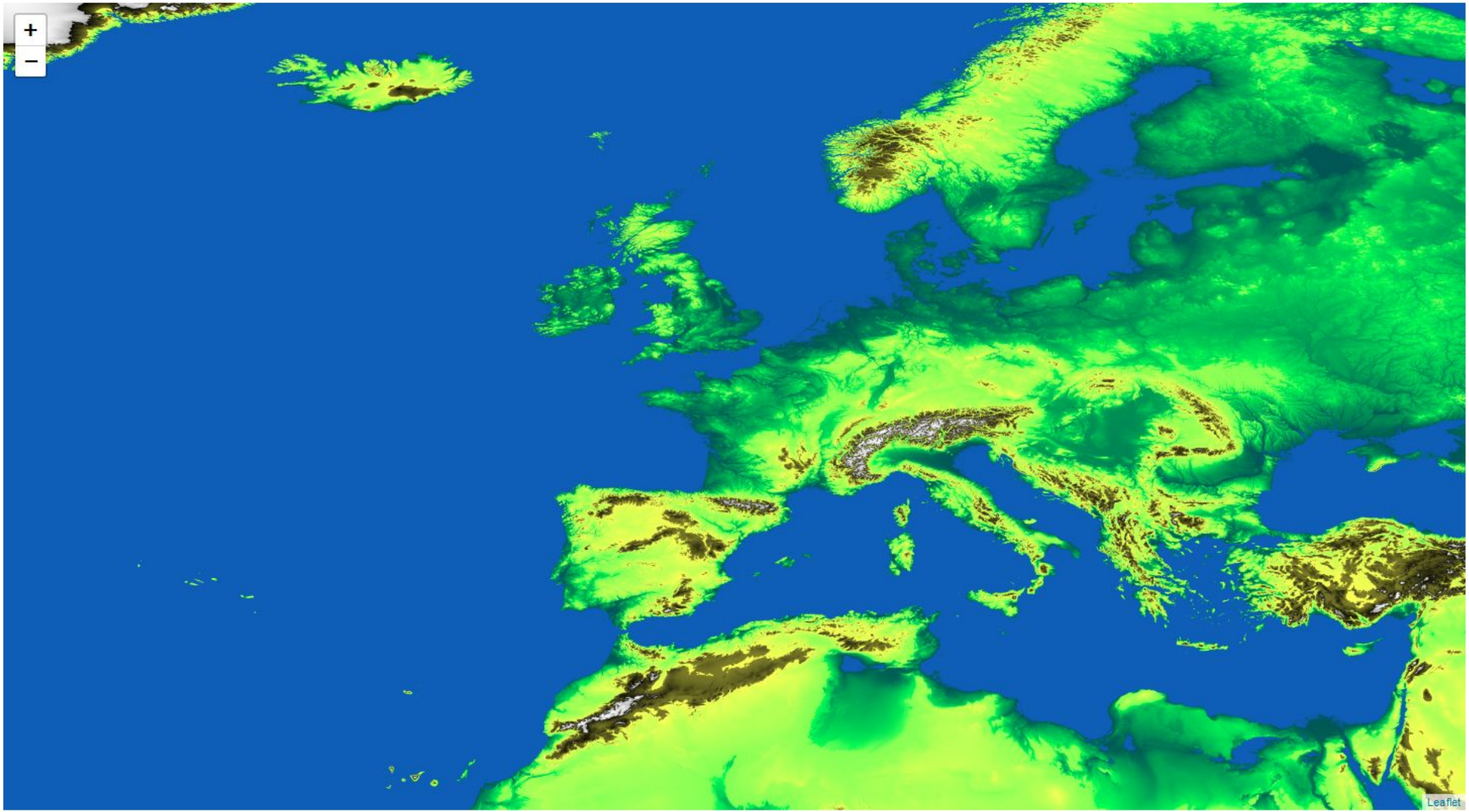

localhost:12345/Project/
+
—
Leaflet

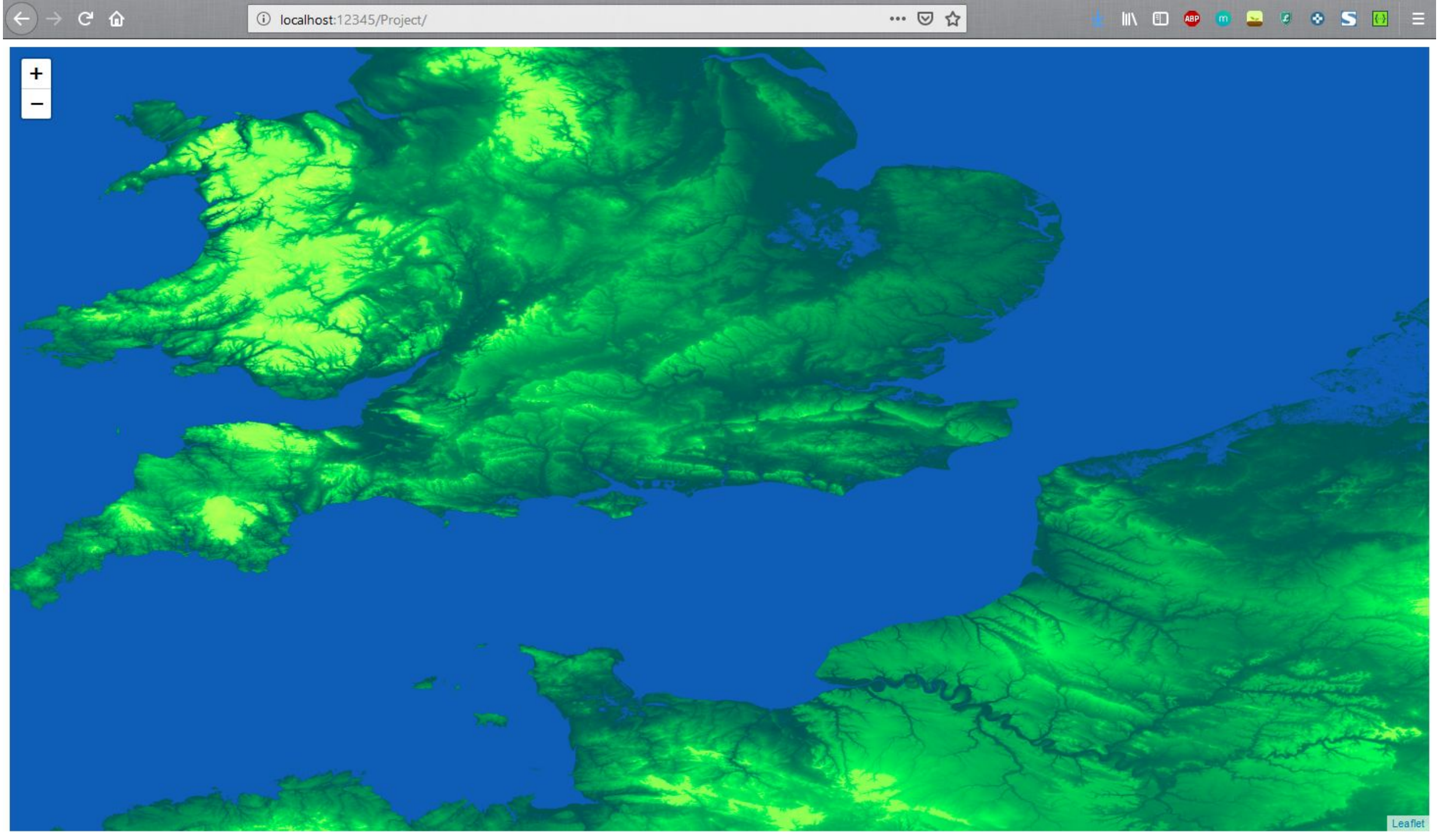


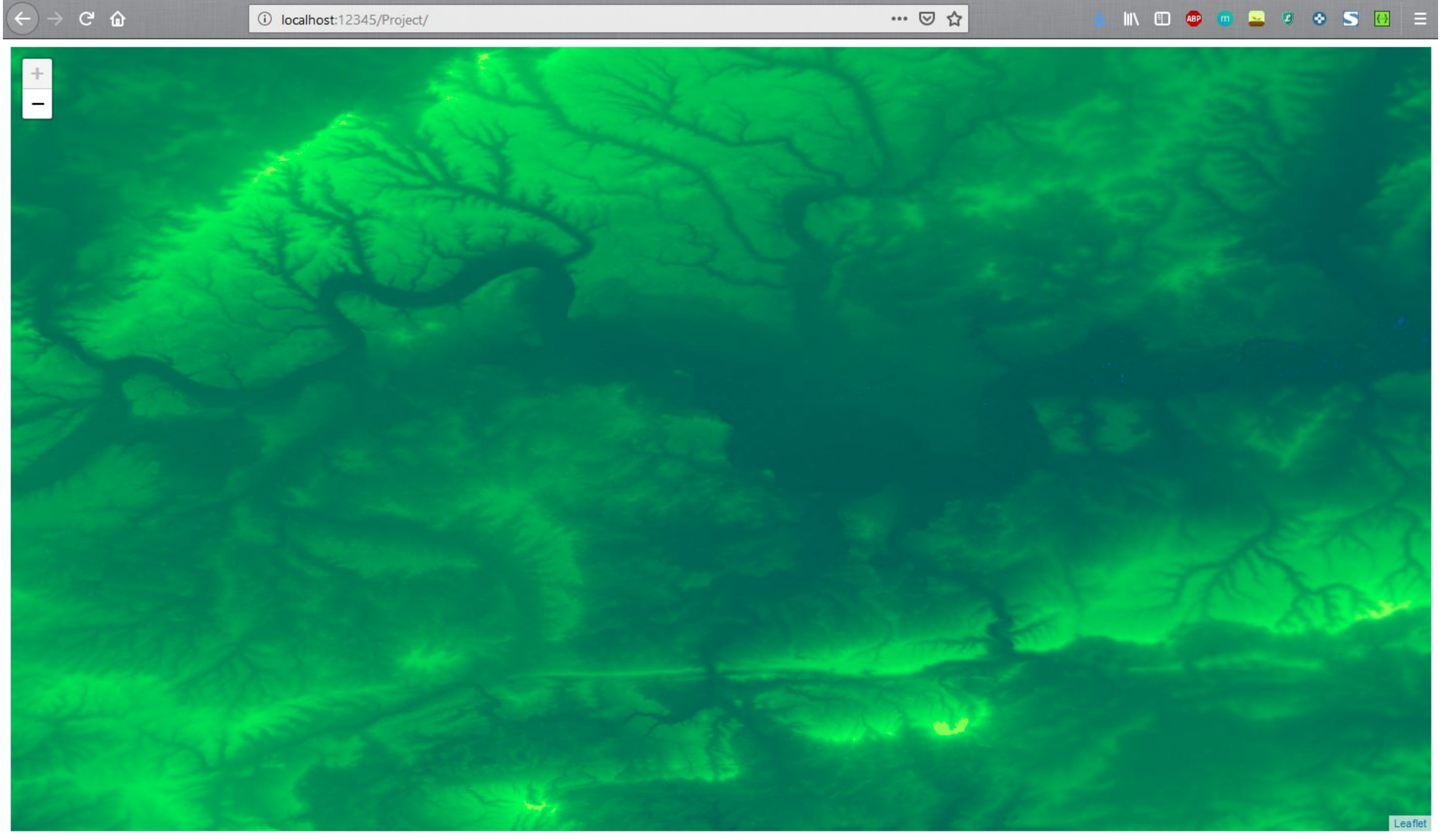
Conclusion


localhost:12345/Project/
+
−
Leaflet


localhost:12345/Project/
+
−
Leaflet


localhost:12345/Project/
+
−
Leaflet


localhost:12345/Project/
+
−
Leaflet


localhost:12345/Project/
+
−
Leaflet