

# Cartographie - Atlas Académique

Les procédures et les informations concernant la cartographie des établissements

- Créer une carte
  - Procédures
  - Carte des détails
- Administrateurs
  - Paramétrage de la fiche détail
  - Catégorie
  - Sous-couche de carte
  - Liste des API
- GRIST
  - API DATA GOUV -> GRIST
  - GRIST -> N8N -> GOGO Carto
- API des établissements de l'académie de la Réunion

# Créer une carte



Créer une carte

# Carte des détails

<https://www.mindomo.com/mindmap/sig-acadmique-systme-dinformation-gographique-1ede8290bbc34523a72173e466472610>

# Administrateurs

# Paramétrage de la fiche détail

Paramétrer la fiche détail de l'établissement en fonction des catégories et des champs

```
<Strong>Informations générales</Strong>
```

```
{{ name|gogo_text }}
```

```
{{ email|gogo_email(label = "Courriel:") }}
```

```
{{ address|gogo_text(label = "Adresse:") }} {{ originalAddress.addressLocality }}
```

```
Téléphone: {{ telephone }}
```

```
RNE: {{ rne }}
```

```
{{ nom_commune }}
```

```
Mise à jour: {{ date_maj_ligne }}
```

```
<Strong>Autres informations</Strong>
```

```
{% for value in groups["type_etablissement"].children %}
```

```
<div>Type d'établissement: {{ value.option.name }}</div>
```

```
{% endfor %}
```

```
{% for value in groups["nom_circonscription"].children %}
```

```
<div>Circonscription: {{ value.option.name }}</div>
```

```
{% endfor %}
```

```
{% for value in groups["pial"].children %}
```

```
<div>PIAL: {{ value.option.name }}</div>
```

```
{% endfor %}
```

```
{% for value in groups["statut_public_prive"].children %}
```

```
<div>Statut: {{ value.option.name }}</div>
```

```
{% endfor %}
```

```
Nombre d'élèves: {{ nombre_d_eleves }}
```

```
{% if college_de_secteur %} Collège de secteur: {{ college_de_secteur }} {% else%} {% endif %}
```

<Strong>Informations générales</Strong>

{{ name|gogo\_text }}

{{ email|gogo\_email(label = "Courriel:") }}

{{ address|gogo\_text(label = "Adresse:") }} {{ originalAddress.addressLocality }}

{{ files|gogo\_text(label = "Toile:") }}

Téléphone: {{ telephone }}

RNE: {{ rne }}

{{ nom\_commune }}

Mise à jour: {{ date\_maj\_ligne }}

<Strong>Autres informations</Strong>

{% for value in groups["type\_etablissement"].children %}

<div>Type d'établissement: {{ value.option.name }}</div>

{% endfor %}

{% for value in groups["nom\_circonscription"].children %}

<div>Circonscription: {{ value.option.name }}</div>

{% endfor %}

{% for value in groups["pial"].children %}

<div>PIAL: {{ value.option.name }}</div>

{% endfor %}

{% for value in groups["statut\_public\_prive"].children %}

<div>Statut: {{ value.option.name }}</div>

{% endfor %}

Nombre d'élèves: {{ nombre\_d\_eleves }}

{% if college\_de\_secteur %} Collège de secteur: {{ college\_de\_secteur }} {% else%} {% endif %}

{% if groups["labellisation"].children %}

<div>Labellisation : {% for value in groups["labellisation"].children %}

{{ value.option.name }}

```

{% endfor %}
</div>
{% else%}

{% endif %}

{% if groups["projet"].children %}

<div> ??? Projets : {% for value in groups["projet"].children %}
  {{ value.option.name }}
  <details> <summary>Les objectifs???? </summary> {{ objectif|gogo_text}} </details> </a>
  <details> <summary>Les objectifs détaillés???? </summary> {{ objectifs_detaillés|gogo_text}} </details>
</a>
  <details> <summary>Plan des actions???? </summary> {{ plan_action|gogo_text}} </details> </a>
  {{ partenaires_du_projet|gogo_text(label = "Partenaires:") }}

{% endfor %}
</div>
{% else%}

{% endif %}

```

Version 3/07/2024 Intégration des données PIX avec %

```

<Strong>Informations générales</Strong>
{{ name|gogo_text }}
{{ email|gogo_email(label = "Courriel:") }}
{{ address|gogo_text(label = "Adresse:") }} {{ originalAddress.addressLocality }}

<a href="{{ files }}" target="_blank">Toile de l'établissement</a> </br>

Téléphone: {{ telephone}}
RNE: {{ rne }}
{{ nom_commune }}

{% if LUNDI_MATIN_DEBUT %}

```

```
{# Horaire de l'établissement #}
```

```
<div>
```

```
<details> <summary><Strong>Horaire de l'établissement????</Strong> </summary>
```

```
<table style="border-collapse: collapse; border: 1px solid black" >
```

```
<tr style="border-collapse: collapse; border: 1px solid black">
```

```
<th style="border-collapse: collapse; border: 1px solid black">Lundi</th>
```

```
<th style="border-collapse: collapse; border: 1px solid black">Mardi</th>
```

```
{% if MERCREDI_MATIN_DEBUT %}
```

```
<th style="border-collapse: collapse; border: 1px solid black">Mercredi</th>
```

```
{% else%} {% endif %}
```

```
□ <th style="border-collapse: collapse; border: 1px solid black">Jeudi</th>
```

```
<th style="border-collapse: collapse; border: 1px solid black">Vendredi</th>
```

```
{% if SAMEDI_MATIN_DEBUT %}
```

```
<th style="border-collapse: collapse; border: 1px solid black">Samedi</th>
```

```
{% else%} {% endif %}
```

```
</tr>
```

```
<tr style="border-collapse: collapse; border: 1px solid black">
```

```
<td style="border-collapse: collapse; border: 1px solid black">{{ LUNDI_MATIN_DEBUT.substring(0, 5) }} -
```

```
{{ LUNDI_MATIN_FIN.substring(0, 5) }} </td>
```

```
<td style="border-collapse: collapse; border: 1px solid black">{{ MARDI_MATIN_DEBUT.substring(0, 5) }} -
```

```
{{ MARDI_MATIN_FIN.substring(0, 5) }} </td>
```

```
{% if MERCREDI_MATIN_DEBUT %}
```

```
<td style="border-collapse: collapse; border: 1px solid black">{{ MERCREDI_MATIN_DEBUT.substring(0, 5) }} -
```

```
{{ MERCREDI_MATIN_FIN.substring(0, 5) }} </td>
```

```
{% else%} {% endif %}
```

```
<td style="border-collapse: collapse; border: 1px solid black">{{ JEUDI_MATIN_DEBUT.substring(0, 5) }} - {{ JEUDI_MATIN_FIN.substring(0, 5) }} </td>
```

```
<td style="border-collapse: collapse; border: 1px solid black">{{ VENDREDI_MATIN_DEBUT.substring(0, 5) }} - {{ VENDREDI_MATIN_FIN.substring(0, 5) }} </td>
```

```
{% if SAMEDI_MATIN_DEBUT %}
```

```
<td style="border-collapse: collapse; border: 1px solid black">{{ SAMEDI_MATIN_DEBUT.substring(0, 5) }} - {{ SAMEDI_MATIN_FIN.substring(0, 5) }} </td>
```

```
{% else%} {% endif %}
```

```
</tr>
```

```
<tr style="border-collapse: collapse; border: 1px solid black">
```

```
<td style="border-collapse: collapse; border: 1px solid black">{{ LUNDI_APRES_MIDI_DEBUT.substring(0, 5) }} - {{ LUNDI_APRES_MIDI_FIN.substring(0, 5) }} </td>
```

```
<td style="border-collapse: collapse; border: 1px solid black">{{ MARDI_APRES_MIDI_DEBUT.substring(0,
```

```

5) }} - {{ MARDI_APRES_MIDI_FIN.substring(0, 5) }} </td>
    {% if MERCREDI_MATIN_DEBUT %}
    <td style="border-collapse: collapse; border: 1px solid black">{{
MERCREDI_APRES_MIDI_DEBUT.substring(0, 5) }} - {{ MERCREDI_APRES_MIDI_FIN.substring(0, 5) }} </td>
    {% else%} {% endif %}
    <td style="border-collapse: collapse; border: 1px solid black">{{ JEUDI_APRES_MIDI_DEBUT.substring(0, 5)
}} - {{ JEUDI_APRES_MIDI_FIN.substring(0, 5) }} </td>
    <td style="border-collapse: collapse; border: 1px solid black">{{
VENDREDI_APRES_MIDI_DEBUT.substring(0, 5) }} - {{ VENDREDI_APRES_MIDI_FIN.substring(0, 5) }} </td>
    {% if SAMEDI_MATIN_DEBUT %}
    <td style="border-collapse: collapse; border: 1px solid black">{{ SAMEDI_APRES_MIDI_DEBUT.substring(0,
5) }} - {{ SAMEDI_APRES_MIDI_FIN.substring(0, 5) }} </td>
    {% else%} {% endif %}
</tr>
</table>

```

```

{% else%} {% endif %}
</details>
</div>

```

Mise à jour: {{ date\_maj\_ligne }}

<Strong>Autres informations</Strong>

```

{% for value in groups["type_etablissement"].children %}
<div>Type d'établissement: {{ value.option.name }}</div>
{% endfor %}

```

```

{% for value in groups["nom_circonscription"].children %}
<div>Circonscription: {{ value.option.name }}</div>
{% endfor %}

```

```

{% for value in groups["pial"].children %}
<div>PIAL: {{ value.option.name }}</div>
{% endfor %}

```

```
{% for value in groups["statut_public_prive"].children %}
```

```
<div>Statut: {{ value.option.name }}</div>
```

```
{% endfor %}
```

```
Nombre d'élèves: {{ nombre_d_eleves }}
```

```
{% if college_de_secteur %} Collège de secteur: {{ college_de_secteur }} {% else%} {% endif %}
```

```
{% if groups["labellisation"].children %}
```

```
<div>Labellisation-Certification : {% for value in groups["labellisation"].children %}
```

```
  {{ value.option.name }}
```

```
{% endfor %}
```

```
</div>
```

```
{% else%}
```

```
{% endif %}
```

```
{% if groups["labellisation"].children %}
```

```
{% if participants %}
```

```
<div> <strong>PIX : </strong>{{ certifies}} élèves certifiés soit {% set reussite =  
(certifies/participants*100)//1 %} {{reussite}} % </br>
```

```
{# <table style="border-collapse: collapse; border: 1px solid black">#}
```

```
{# <tr style=" border: 1px solid black">#}
```

```
  {# <th style=" border: 1px solid black">Participants:</th> #}
```

```
  {# <th style=" border: 1px solid black">Non Certifies:</th> #}
```

```
  {# <th style=" border: 1px solid black">Certifies:</th>#}
```

```
{# </tr>#}
```

```
{# <tr>#}
```

```
  {# <td style=" border: 1px solid black">{{ participants|gogo_text}}</td>#}
```

```
  {# <td style=" border: 1px solid black">{{ non_obtenues|gogo_text}}</td> #}
```

```
  {# <td style=" border: 1px solid black">{{ certifies|gogo_text}}</td>#}
```

```
{# </tr>#}
```

```
{# </table> #}
```

```
</div>
```

```
{% else%} {% endif %}
```

```
{% else%}
```

```
{% endif %}
```

```
{% if groups["projet"].children %}
```

```
<div> ??? Projets : {% for value in groups["projet"].children %}
```

```
  {{ value.option.name }}
```

```
  <details> <summary>Les objectifs???? </summary> {{ objectif|gogo_text}} </details> </a>
```

```
  <details> <summary>Les objectifs détaillés???? </summary> {{ objectifs_detaillés|gogo_text}} </details>
```

```
</a>
```

```
  <details> <summary>Plan des actions???? </summary> {{ plan_action|gogo_text}} </details> </a>
```

```
  {{ partenaires_du_projet|gogo_text(label = "Partenaires:")}}
```





```
{% endfor %}
```





```
</div>
```





```
{% else%}
```

















































```
{% endif %}
```

# Catégorie

PCAA PCAA    





Allemand PCAA-ALLEMAND    





**Sous-Catégories** 33    

































    ABIBAC <span>147</span>    
    LCE <span>164</span>    
    LLCECR <span>157</span>    
    SELO <span>158</span>    
    SI <span>SI-ALL</span>    
    LV <span>LV-ALLEMAND</span>    

Ajouter une Catégorie

Ajouter un groupe

Chinois PCAA-CHINOIS    

**Sous-Catégories** 34    

    BFI <span>149</span>    
    SI <span>160</span>    
    SELO <span>SELO-CHINOIS</span>    
    LO <span>LO</span>    

Ajouter une Catégorie

Ajouter un groupe

### Sous-Catégories

34    

    BFI	149    
    SI	160    
    SELO	SELO-CHINOIS    
    LO	LO    



Ajouter une Catégorie

### Ajouter un groupe

✓    Espagnol PCAA-ESPAGNOL    

### Sous-Catégories

35    

    BACHIBAC	151    
    LCE	161    
    LLCECR	162    
    SELO	SELO-ESPAGNOL    
    LV	LV-ESPAGNOL    

Ajouter une Catégorie

---

**GROUPE PCAA**

ESPAGNOL → \* Espagnol

CHINOIS → \* Chinois

ALLEMAND → \* Allemand

**GROUPE MODALITE\_ENS**

SELO-ALL → \* Allemand → SELO

SI-CHINOIS → \* Chinois → SI

SI-ALL → \* Allemand → SI

SELO-ESPAGNOL → \* Espagnol → SELO

SELO-CHINOIS → \* Chinois → SELO

Abibac → \* Allemand → ABIBAC

BFI-CHINOIS → \* Chinois → BFI

LV-ALLEMAND → \* Allemand → LV

LO → \* Chinois → LO

LLCECR-ESPAGNOL → \* Espagnol → LLCECR

LLCECR-ALL → \* Allemand → LLCECR

LCE-ESPAGNOL → \* Espagnol → LCE

LCE-ALL → \* Allemand → LCE

Bachibac → \* Espagnol → BACHIBAC

LV-ESPAGNOL → \* Espagnol → LV

Créer toutes les catégories manquantes

---

Administrateurs

# Sous-couche de carte

[SAEI : Les 3 antennes 01-cole\\_inclusive\\_2024.geojson](#)

[SAEI : Les 3 antennes 02-ecole\\_inclusive\\_couleur.geojson](#)

Communes de la Réunion :

<https://geo.api.gouv.fr/departements/974/communes?format=geojson&geometry=contour>

# Liste des API

## API dans l'ordre d'importance

1) Liste des établissements - Données locales <https://portail.ac-reunion.fr/swami/public/api/etablisements/>

Importation du fichier vierge avec les têtes de colonne dans GRIST : [Annuaire-ETAB-ACA-ALL.csv](#)

**Ce code permet d'ajouter, de corriger les établissements mis à jour.**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Import Lycées La Réunion</title>
  <script src="https://docs.getgrist.com/grist-plugin-api.js"></script>
  <style>
    #importEtabBtn {
      margin: 10px;
      padding: 10px 20px;
      background-color: #4CAF50;
      color: white;
      border: none;
      border-radius: 4px;
      cursor: pointer;
      font-family: sans-serif;
      font-size: 14px;
      transition: background-color 0.3s;
    }
    #importEtabBtn:hover:not(:disabled) {
      background-color: #45a049;
    }
    #importEtabBtn:disabled {
      background-color: #cccccc;
      cursor: not-allowed;
    }
  </style>
```

```
</head>
<body>
  <div style="font-family: sans-serif; padding: 1em;">
    <h2>Import des établissements</h2>
    <p>Cliquez sur le bouton ci-dessous pour mettre à jour la table Annuaire_ETAB_ACA_ALL.</p>
    <button id="importEtabBtn">Mettre à jour l'annuaire</button>
    <p id="status" style="margin-top: 10px; color: #666;"></p>
  </div>

  <script>
    // Demande un accès complet pour lire et écrire
    grist.ready({ requiredAccess: 'full' });

    // Liste des noms de colonnes pour la cohérence
    const COLUMN_NAMES = [
      "uaj", "nom", "denominationOfficielle", "uajDenominationOfficielle",
      "type", "rue", "bassin", "code_postal", "ville", "tel", "fax", "mail", "lat", "lon"
    ];

    /**
     * Récupère les données d'établissements via l'API externe.
     */
    async function fetchEtab() {
      const apiUrl = "https://portail.ac-reunion.fr/swami/public/api/etablissements/";
      try {
        document.getElementById("status").textContent = "Récupération des données en cours...";
        const response = await fetch(apiUrl);
        if (!response.ok) {
          throw new Error(`HTTP ${response.status}`);
        }
        const data = await response.json();
        const results = data.data?.values || [];
        document.getElementById("status").textContent = `${results.length} établissements trouvés.`;
        return results;
      } catch (error) {
        document.getElementById("status").textContent = `Erreur : ${error.message}`;
        console.error("Erreur API :", error);
        return [];
      }
    }
  </script>

```

```

/**
 * Met à jour (UpdateRecord) ou insère (AddRecord) les enregistrements.
 */
async function upsertIntoGristOptimized(etabs) {
  document.getElementById("status").textContent = "Préparation des mises à jour en cours...";
  const tableId = "Annuaire_ETAB_ACA_ALL";
  const uajColumn = "uaj";

  // 1. LECTURE : Récupération des enregistrements existants (docApi.fetchTable est fiable)
  const allRecords = await grist.docApi.fetchTable(tableId, ['id', uajColumn]);

  // Création de la Map de recherche rapide
  const uajToldMap = new Map();
  if (allRecords[uajColumn]) {
    allRecords[uajColumn].forEach((uaj, index) => {
      uajToldMap.set(uaj, allRecords.id[index]);
    });
  }

  const actions = []; // Liste des actions à envoyer
  let createdCount = 0;
  let updatedCount = 0;

  for (const etab of etabs) {
    const recordId = uajToldMap.get(etab.uaj);

    // Construction de l'objet d'enregistrement (format clé-valeur)
    const recordObject = {};
    COLUMN_NAMES.forEach(col => {
      const value = etab[col];
      // Assurer que les valeurs manquantes sont correctement traitées
      if (value !== undefined && value !== null) {
        recordObject[col] = value;
      } else if (["rue", "bassin", "fax", "mail"].includes(col)) {
        recordObject[col] = ""; // Les champs texte optionnels doivent être des chaînes vides
      } else {
        recordObject[col] = null; // Autres champs (nombre/liens) sont null
      }
    });
  }
};

```

```

if (recordId) {
  // MISE À JOUR (Action Ligne par Ligne)
  // Format: ['UpdateRecord', TableID, RowID, RecordObject]
  actions.push(
    ['UpdateRecord', tableId, recordId, recordObject]
  );
  updatedCount++;
} else {
  // CRÉATION (Action Ligne par Ligne)
  // Format: ['AddRecord', TableID, RowID (null), RecordObject]
  actions.push(
    ['AddRecord', tableId, null, recordObject]
  );
  createdCount++;
}
}

// 2. ENVOI DES ACTIONS
if (actions.length > 0) {
  document.getElementById("status").textContent = `Envoi de ${actions.length} opérations à Grist...
(peut prendre du temps)`;
  // Envoi de toutes les actions (créations + mises à jour ligne par ligne)
  await grist.docApi.applyUserActions(actions);
}

return { created: createdCount, updated: updatedCount };
}

/**
 * Fonction principale déclenchée par le bouton.
 */
async function importEtab() {
  const btn = document.getElementById("importEtabBtn");
  btn.disabled = true;
  btn.textContent = "Importation en cours...";

  try {
    const etabs = await fetchEtab();
    if (etabs.length > 0) {

```

```
document.getElementById("status").textContent = "Mise à jour de la base en cours...";
const result = await upsertIntoGristOptimized(etabs);
document.getElementById("status").textContent =
  `❑ **${result.created}** établissements ajoutés, **${result.updated}** établissements mis à
jour.`;
  } else {
    document.getElementById("status").textContent = "Aucun établissement trouvé à traiter.";
  }
} catch (error) {
  document.getElementById("status").textContent = "❑ Une erreur critique est survenue. Vérifiez la
console (F12) et l'accès.";
  console.error("Erreur critique d'importation :", error);
} finally {
  btn.disabled = false;
  btn.textContent = "Mettre à jour l'annuaire";
}
}

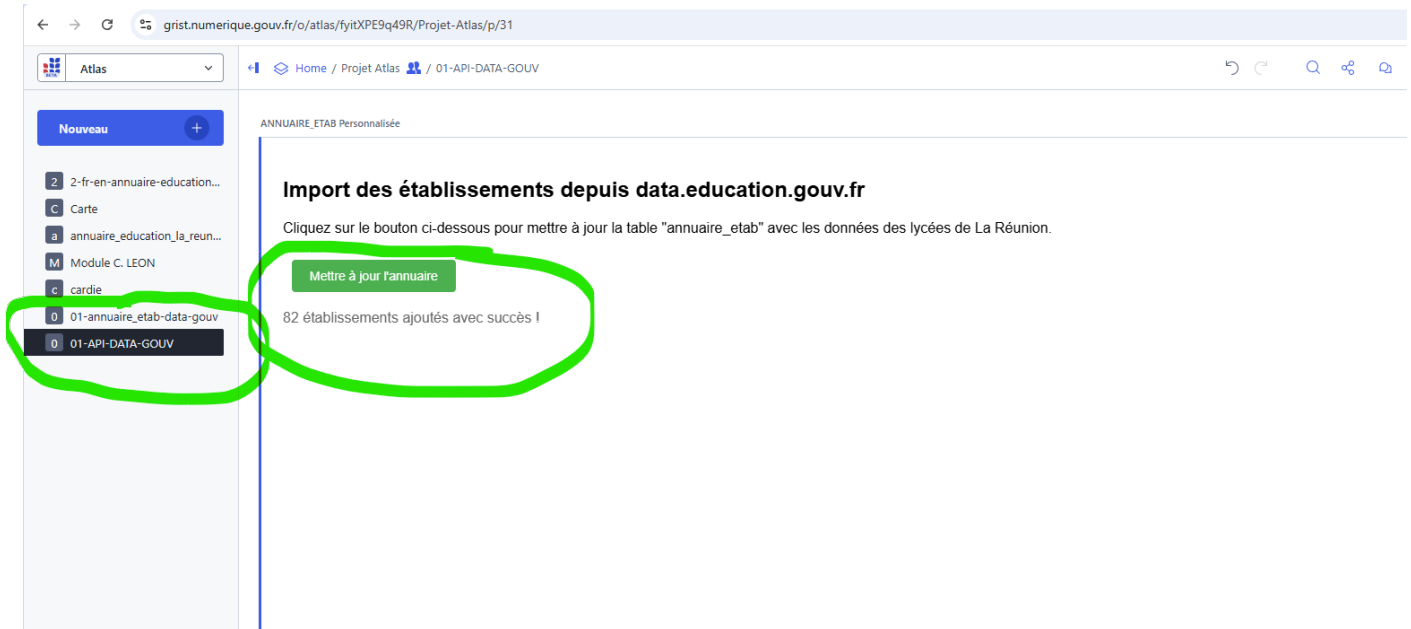
document.getElementById("importEtabBtn").addEventListener("click", importEtab);
</script>
</body>
</html>
```

# GRIST

Tous les éléments permettant l'introduction des données pour être utiliser dans GRIST

# API DATA GOUV -> GRIST

Script d'appel à synchronisation des données de DATA GOUV dans GRIST



1) Insérer le fichier CSV ci-dessous " importer un fichier"

identifiant\_de\_l\_etablissement,nom\_.csv

2) Créer une vue page ou une page avec "Custom widget"

3) Copier le code ci-dessous dans la partie HTML en faisant bien attention de faire appel à la bonne table " const table = grist.getTable("Annuaire\_etab");"

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Import Lycées La Réunion</title>
  <script src="https://docs.getgrist.com/grist-plugin-api.js"></script>
  <style>
    #importEtabBtn {
```

```
margin: 10px;
padding: 10px 20px;
background-color: #4CAF50;
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
font-family: sans-serif;
font-size: 14px;
}
#importEtabBtn:hover {
  background-color: #45a049;
}
</style>
</head>
<body>
  <div style="font-family: sans-serif; padding: 1em;">
    <h2>Import des établissements depuis data.education.gouv.fr</h2>
    <p>
      Cliquez sur le bouton ci-dessous pour mettre à jour la table "annuaire_etab"
      avec les données des lycées de La Réunion.
    </p>
    <button id="importEtabBtn">Mettre à jour l'annuaire</button>
    <p id="status" style="margin-top: 10px; color: #666;"></p>
  </div>

  <script>
    grist.ready({requiredAccess: 'full'});

    async function fetchEtab() {
      const apiUrl = "https://data.education.gouv.fr/api/explore/v2.1/catalog/datasets/fr-en-annuaire-
education/records?select=identifiant_de_l_etablissement%2Cnom_etablissement%2Ctype_etablissement%2Cstatut_public_privé%2Cadresse_1%2Cadresse_2%2Cadresse_3%2Ccode_postal%2Cnom_commune%2Cvoie_generale%2Cvoie_technologique%2Cvoie_professionnelle%2Ctelephone%2Cfax%2Cweb%2Cmail%2Crestauration%2Chebergement%2Cculis%2Capprentissage%2Csegpa%2Csection_arts%2Csection_cinema%2Csection_theatre%2Csection_sport%2Csection_internationale%2Csection_europeenne%2Clycee_agricole%2Clycee_militaire%2Clycee_des_metiers%2Cpost_bac%2Cappartenance_education_prioritaire%2Cgreta%2Cnombre_d_eleves%2Cfiche_onisep%2Ctype_contrat_privé%2Cnom_circonscription%2Clatitude%2Clongitude%2Cprecision_localisation%2Cdate_maj_ligne%2Ccode_nature%2Clibelle_nature%2Ccode_type_contrat_privé%2Cpial%2Cetablissement_mere%2Ctype_rattachement_etablissement_mere&limit=100&refine=type_etablissement%3A%22Lyc%3%A9e%22&refine=li
```

```
belle_academie%3A%22La%20R%C3%A9union%22";
```

```
try {
  document.getElementById("status").textContent = "Récupération des données en cours...";
  const response = await fetch(apiUrl);
  if (!response.ok) {
    throw new Error("HTTP " + response.status);
  }
  const data = await response.json();
  const results = data.results || [];
  document.getElementById("status").textContent =
    results.length + " établissements trouvés.";
  return results;
} catch (error) {
  document.getElementById("status").textContent =
    "Erreur lors de la récupération des données.";
  console.error("Erreur API :", error);
  return [];
}
}
```

```
async function insertIntoGrist(etabs) {
  const table = grist.getTable("Annuaire_etab");

  const records = etabs.map(etab => ({
fields: {
  identifiant_de_l_etablissement: etab.identifiant_de_l_etablissement,
  nom_etablissement: etab.nom_etablissement,
  type_etablissement: etab.type_etablissement,
  statut_public_prive: etab.statut_public_prive,
  adresse_1: etab.adresse_1 || "",
  adresse_2: etab.adresse_2 || "",
  adresse_3: etab.adresse_3 || "",
  code_postal: etab.code_postal,
  nom_commune: etab.nom_commune,
  voie_generale: etab.voie_generale,
  voie_technologique: etab.voie_technologique,
  voie_professionnelle: etab.voie_professionnelle,
  telephone: etab.telephone,
  fax: etab.fax || "",

```

```

web: etab.web || "",
mail: etab.mail || "",
restauration: etab.restauration || "",
hebergement: etab.hebergement || "",
ulis: etab.ulis || "",
apprentissage: etab.apprentissage || "",
segpa: etab.segpa || "",
section_arts: etab.section_arts || "",
section_cinema: etab.section_cinema || "",
section_theatre: etab.section_theatre || "",
section_sport: etab.section_sport || "",
section_internationale: etab.section_internationale || "",
section_europeenne: etab.section_europeenne || "",
lycee_agricole: etab.lycee_agricole || "",
lycee_militaire: etab.lycee_militaire || "",
lycee_des_metiers: etab.lycee_des_metiers || "",
post_bac: etab.post_bac || "",
appartenance_education_prioritaire: etab.appartenance_education_prioritaire || "",
greta: etab.greta || "",
nombre_d_eleves: etab.nombre_d_eleves || "",
fiche_onisep: etab.fiche_onisep || "",
type_contrat_privé: etab.type_contrat_privé || "",
nom_circonscription: etab.nom_circonscription || "",
latitude: etab.latitude,
longitude: etab.longitude,
precision_localisation: etab.precision_localisation || "",
date_maj_ligne: etab.date_maj_ligne || "",
code_nature: etab.code_nature || "",
libelle_nature: etab.libelle_nature || "",
code_type_contrat_privé: etab.code_type_contrat_privé || "",
pial: etab.pial || "",
etablissement_mere: etab.etablissement_mere || "",
type_rattachement_etablissement_mere: etab.type_rattachement_etablissement_mere || ""
}
}));

try {
  document.getElementById("status").textContent =
    "Insertion des données dans Grist...";
  await table.create(records);

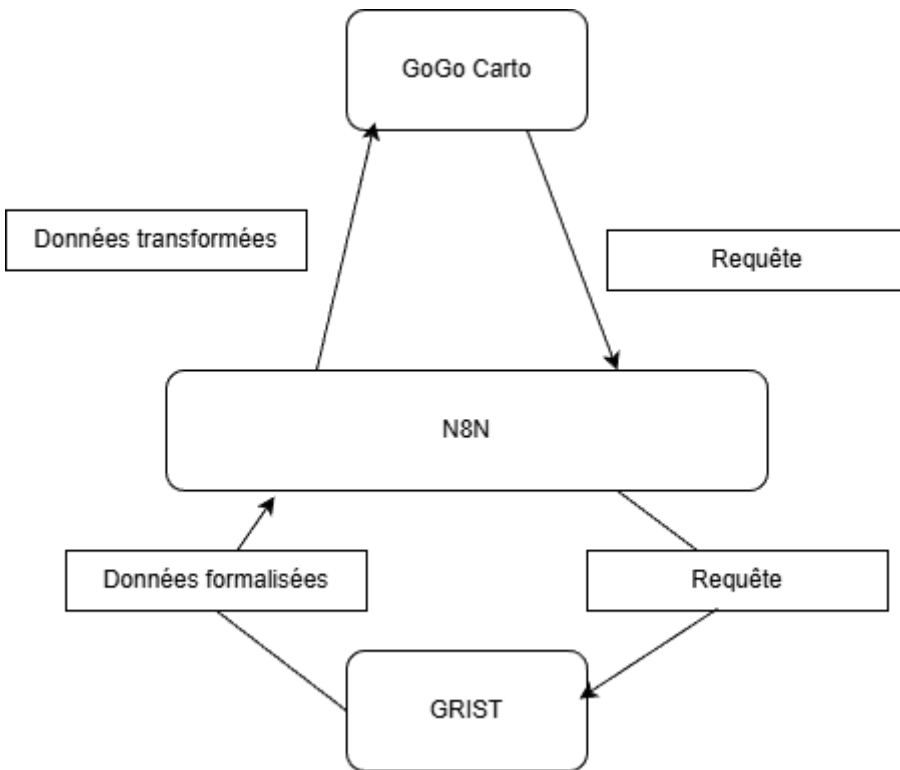
```

```
document.getElementById("status").textContent =
  records.length + " établissements ajoutés avec succès !";
} catch (error) {
  document.getElementById("status").textContent =
    "Erreur lors de l'insertion des données.";
  console.error("Erreur Grist :", error);
}
}

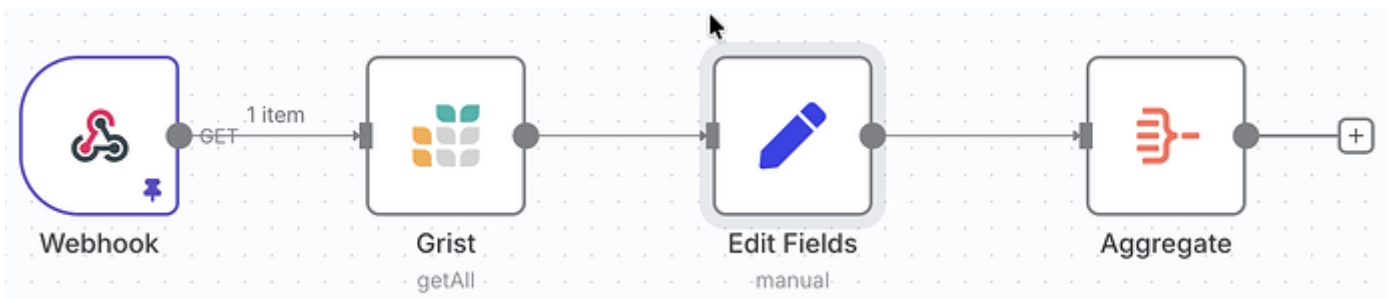
async function importEtab() {
  const etabs = await fetchEtab();
  if (etabs.length > 0) {
    await insertIntoGrist(etabs);
  }
}

document
  .getElementById("importEtabBtn")
  .addEventListener("click", importEtab);
</script>
</body>
</html>
```

# GRIST -> N8N -> GOGO Carto



## Création du workflow N8N



Tout d'abord, un webhook fait office de point d'entrée de la donnée. C'est là que se trouve l'URL que vous devrez renseigner dans GoGoCarto pour faire dialoguer ces outils entre eux (cf. étapes suivantes) :

HTTP Method = **GET**

Authentication = **None**

Respond = **When Last Node Finishes** (le webhook renverra automatiquement la donnée produite par le dernier nœud exécuté)

Response Data = First Entry JSON

Cliquez sur « Add Option » et ajoutez Response Content-Type = **application/json**

L'étape suivante du workflow est un nœud Action in **a app > Grist** dans lequel vous allez préciser le tableau qui contient les données à cartographier :

- Pour créer les identifiants (credentials) nécessaires à la connexion, renseignez la clé API disponible dans les paramètres de votre compte Grist
- Operation = **Get Many rows**
- vous devez renseigner le docID du document Grist contenant le tableau (disponible via Grist dans la page « Paramètres », section « API ») et le Table ID du tableau (page « Données source »)
- Activez l'interrupteur « Return All »
- Vous pouvez ajouter un filtre sur une colonne pour ne récupérer qu'une partie des données du tableau, comme vous le feriez dans Grist. Vous pourrez ainsi filtrer les données sur une colonne de validation (à créer au préalable dans le tableau Grist) et vous assurer que les données non validées ne transitent pas vers GoGoCarto.

Le nœud suivant est de type Data Transformation > Edit Fields : c'est la couche de normalisation. Grâce à lui, chaque colonne provenant de Grist sera renommée, typée, et éventuellement convertie (texte, booléen, liste, coordonnées, etc.) pour correspondre exactement à la structure attendue par GoGoCarto. Paramétrez ce nœud en fonction du modèle de données que vous voulez définir.

Le nœud final est Data Transformation > Aggregate, qui regroupe toutes les lignes transformées en un seul tableau, lequel est envoyé à GoGoCarto par le webhook :

Aggregate = All Item Data (Into a Single List)

Put Output in Field = Data

Include = All Fields

## Connexion à GOGO Carto -> N8N

Dans le volet d'administration, ouvrez la page « Import Dynamique »

Image not found or type unknown

- Dans l'onglet « Général », précisez bien " Type de la source = **API JSON**" ainsi que l'adresse du webhook (cf. étapes précédentes) dans « URL de l'API JSON »
- Sur ce même onglet, vérifiez la **fréquence de mise à jour des données**

Une fois ces réglages sauvegardés, vous n'avez plus rien à faire : GoGoCarto enverra automatiquement la requête GET via n8n pour remplir votre carte avec les données de votre Grist.

# API des établissements de l'académie de la Réunion