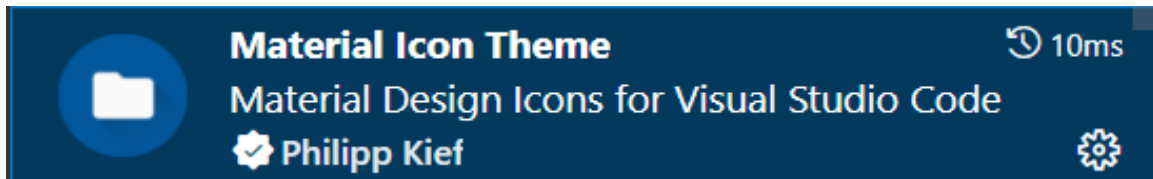


Objectifs :

- S'approprier des modules de base de Node.js
- Créer un serveur http embryonnaire
- Utiliser GitHub pour conserver une copie de sécurité
- Utiliser Glitch.com pour héberger votre service

- 
- 1) Créer un répertoire *atelier-1*
  - 2) Lancer VS Code et installer le greffon suivant :



- 3) Ouvrir le répertoire *atelier-1*
- 4) Ouvrir le panneau du terminal (Terminal/New Terminal)
- 5) Inscrire la commande suivante : `npm init`
- 6) Répondez aux questions de l'assistant de création du fichier *package.json*

```
package.json > ...
1  {
2    "name": "atelier-1",
3    "version": "1.0.0",
4    "description": "",
5    "main": "server.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "Votre nom",
10   "license": "ISC"
11 }
```

- 7) Créer le fichier *server.js* et entrer le code suivant

```
const http = require('http');
const server = http.createServer((req, res) => {
  console.log(req.url);
});
const PORT = process.env.PORT || 5000;
server.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

- 8) Ajouter un fichier de configuration du débogueur : Run/Add Configuration... et sélectionner *Node.js*. Cela ajoutera le fichier *launch.json* qui servira lors de la prochaine exécution du code se trouvant dans *server.js*
- 9) Lancer *server.js* avec le raccourci clavier F5 et regarder dans le la console *DEBUG CONSOLE*. Il devrait être inscrit : *Server running on port 5000*
- 10) Ouvrir une instance de Google Chrome, et entrer l'url suivant : [localhost:5000/api/contacts/15](http://localhost:5000/api/contacts/15)
- 11) Observer l'impact dans *DEBUG CONSOLE* de VS Code
- 12) Arrêter le serveur avec le raccourci clavier SHIFT-F5

13) Ajouter le code suivant après l'instruction `console.log(...)` :

```
let reqInfo = {url:req.url, method:req.method};
res.writeHead(200, {"Content-Type": "application/json"});
res.end(JSON.stringify(reqInfo));
```

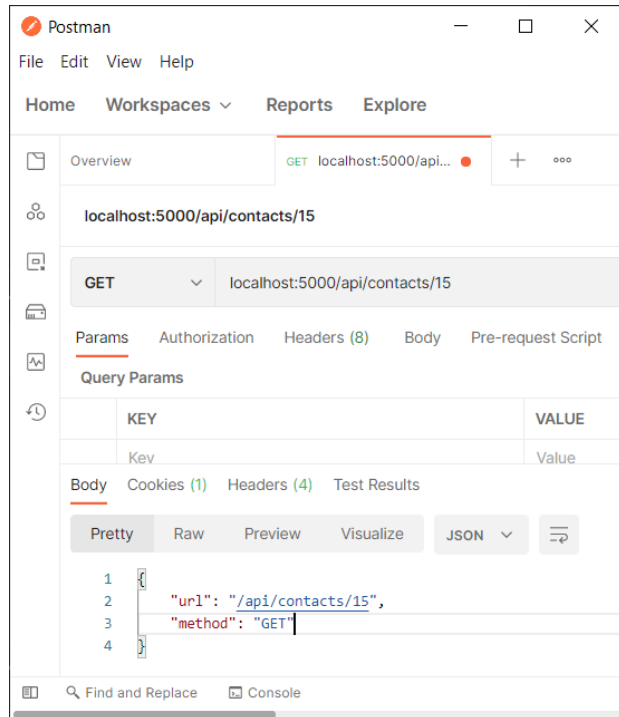
14) Tester à nouveau (penser à ajouter le greffon JSON Formatter à votre Google Chrome)

<https://chrome.google.com/webstore/detail/json-formatter/bcjindcccaagfpapjjmafapmmgkkhgoa>

15) Lancer l'application POSTMAN (télécharger et installer si pas déjà fait)

16) Ajouter une requête GET et régler l'URL à `localhost:5000/api/contacts/15`

17) Lancer la requête et observer le résultat.



18) Ajouter le module `query-string` avec la commande de console : `npm install query-string`

19) Ajouter dans `server.js` la référence à ce module :

```
const queryString = require('query-string');
```

20) Remplacer le corps du callback (`req, res`) => `{...}` par le suivant :

```
console.log(req.url);
let reqInfo = { url: req.url, method: req.method, contentType: req.headers['content-type'] };

res.writeHead(200, { "Content-Type": "application/json" });
if (req.method === 'GET') {
  res.end(JSON.stringify(reqInfo));
} else {
  if (req.method === 'POST') {
    let body = [];
    req.on('data', chunk => {
      body.push(chunk);
    }).on('end', () => {
      try {
        if (req.headers['content-type'] === "application/json")
          reqInfo.body = JSON.parse(body);
        else
          if (req.headers['content-type'] === "application/x-www-form-urlencoded")
            reqInfo.body = queryString.parse(body.toString());
          else
            reqInfo.body = body.toString();
        res.end(JSON.stringify(reqInfo));
      } catch (error) {
        console.log(error);
      }
    });
  }
}
```

Note : Vous pouvez formater le code en faisant bouton de droite et appeler « Format Document » ou appliquer l'équivalence clavier SHIT+ALT+F

21) Ajouter une nouvelle requête dans *POSTMAN* :

Méthode : POST,

URL : localhost:5000/api/contacts,

CONTENT-TYPE : application/json

BODY : { "FirstName": "Kyle", "LastName": "Ross", "Email": [Kyle.Ross@clg.qc.ca](mailto:Kyle.Ross@clg.qc.ca) }

22) Lancer la requête et observer le résultat

23) Ajouter un fichier *formData.html* et coller le contenu suivant :

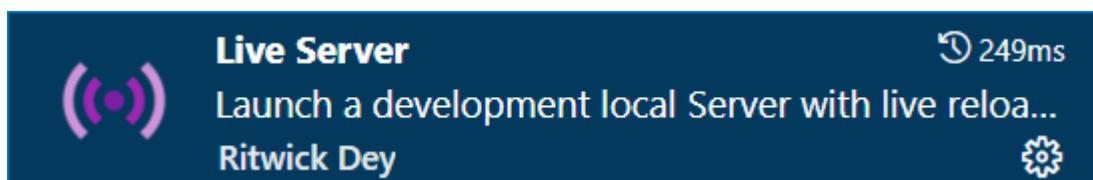
```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    input {
      margin: 10px;
      height: 28px;
    }
  </style>
</head>

<body>
  <form action="http://localhost:5000/api/contacts" method="POST">
    <input type="text" placeholder="Firstname" name="FirstName" required> <br>
    <input type="text" placeholder="Lastname" name="LastName" required><br>
    <input type="email" placeholder="Email" name="Email" required><br>
    <input type="submit" value="Soumettre">
  </form>
</body>

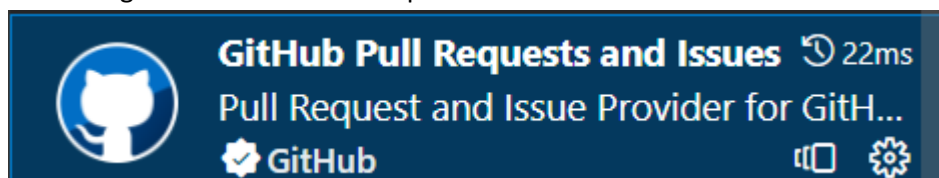
</html>
```

24) Installer les greffons suivants dans VS Code :

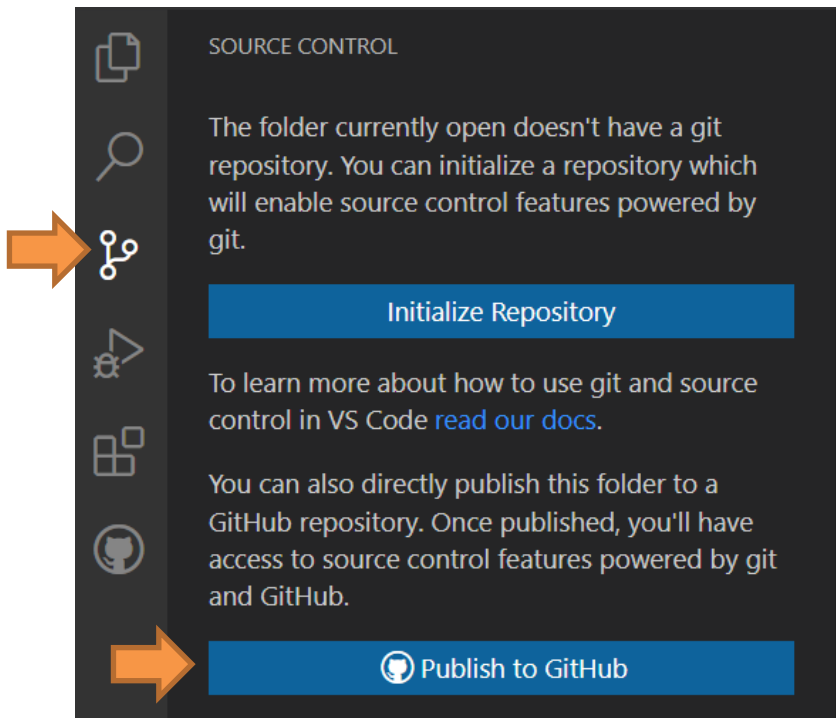


25) Lancer la page *formData.html* par l'entremise de ce dernier greffon.

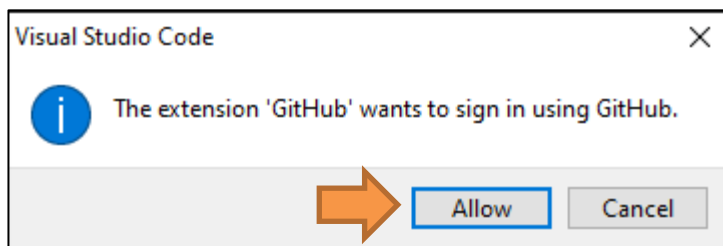
26) Installer le greffon GitHub Pull Requests and Issues dans VS Code :



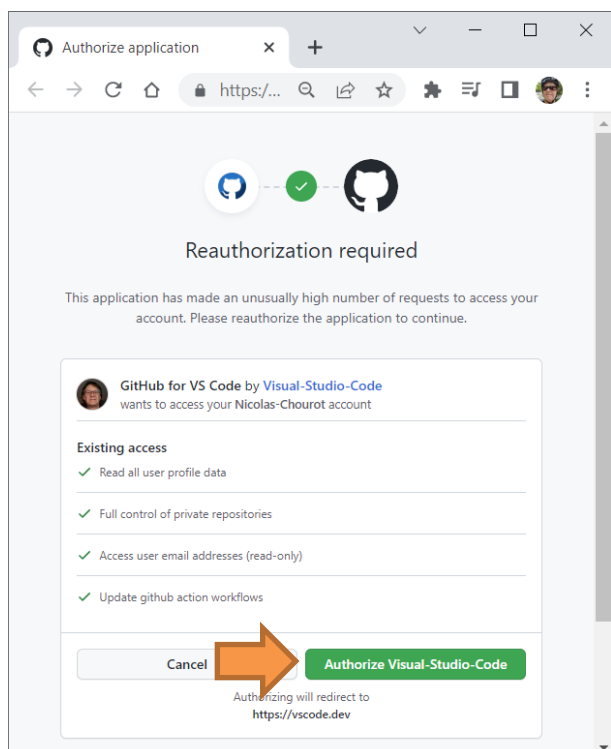
27) Publier le projet dans GitHub :



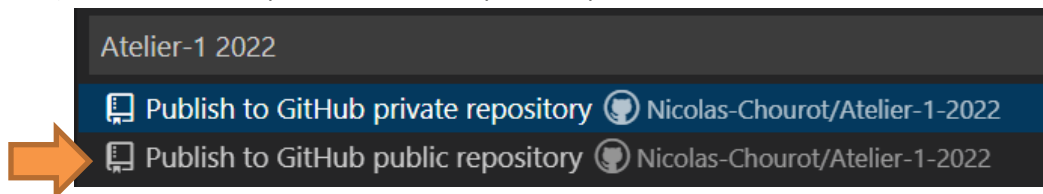
28) Accepter de se connecter via l'extension GitHub :



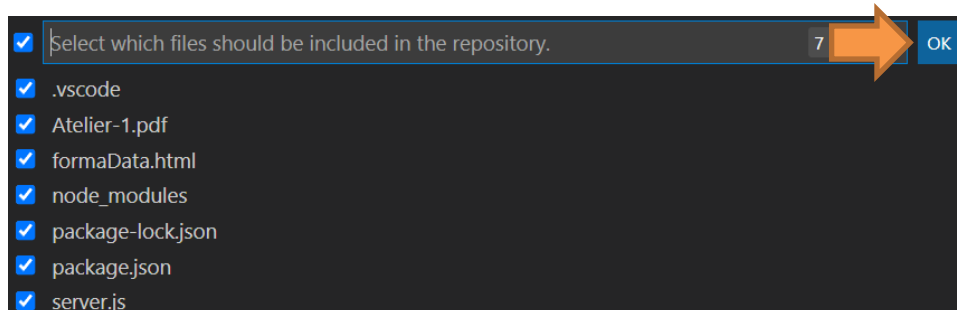
29) Une instance de Google Chrome apparaîtra pour vous demander d'accepter la connexion :



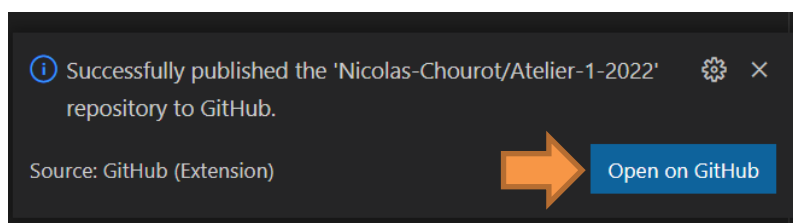
30) Sélectionner de publier dans un répertoire public :



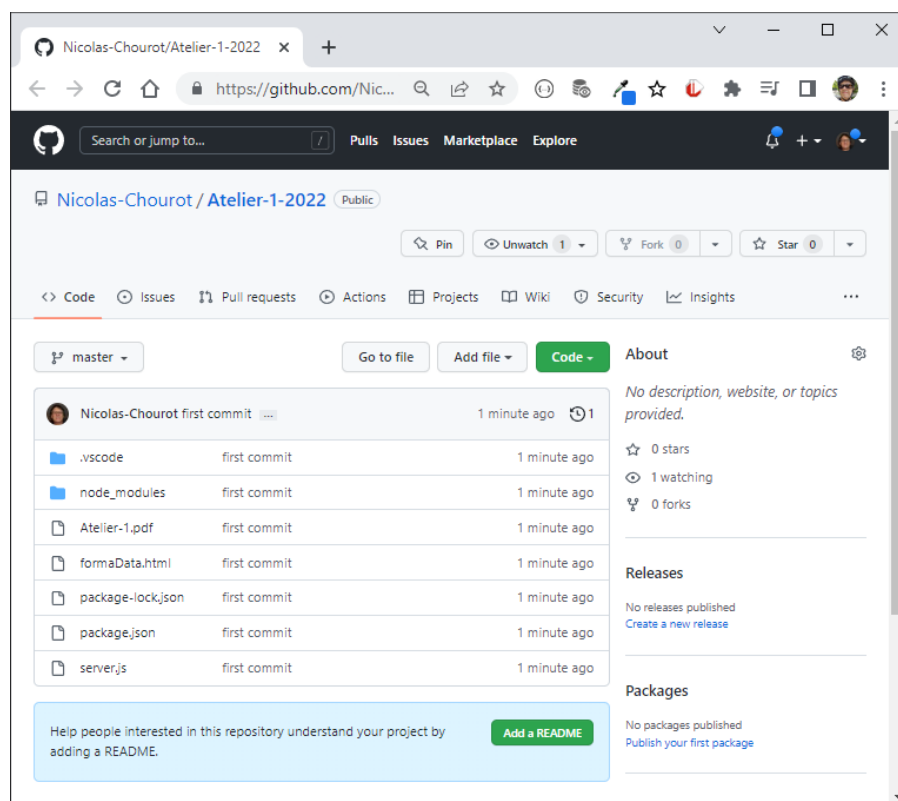
31) Accepter de publier tous les fichiers et répertoires :



32) Message de confirmation de l'opération :



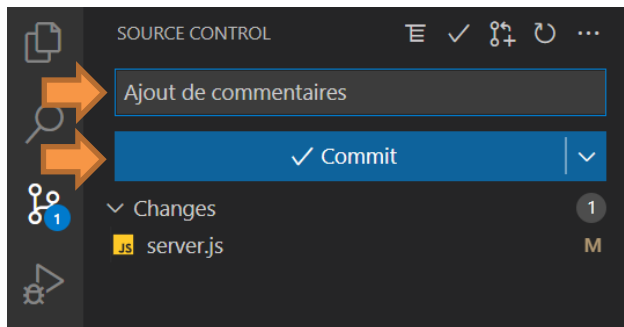
33) Votre répertoire dans GitHub :



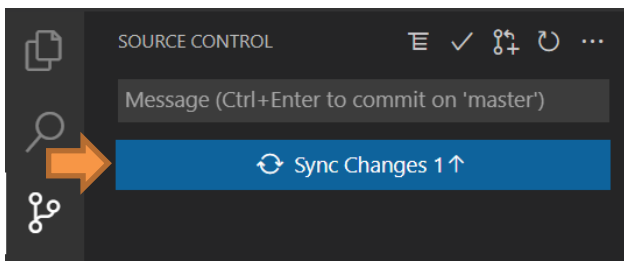
34) Ajouter le commentaire au début du fichier server.js

```
// Mon premier server Http
```

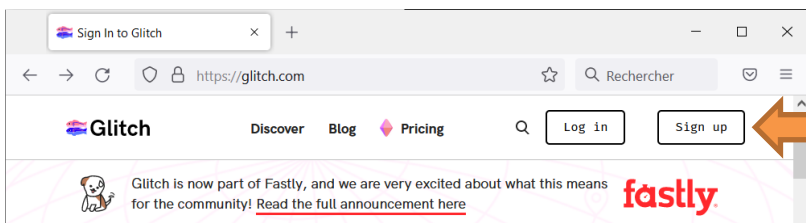
35) Commettre les changements dans Git :



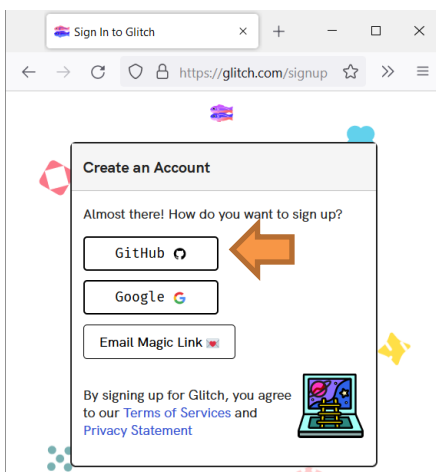
36) Synchroniser les changements dans GitHub :



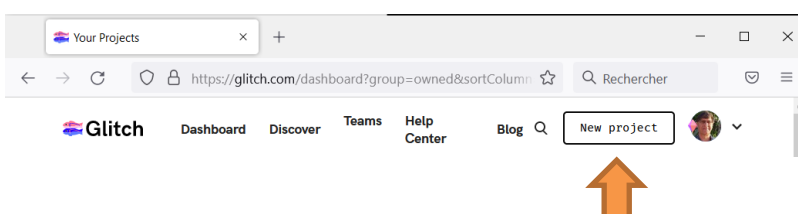
37) Créer un compte Glitch : <https://glitch.com>



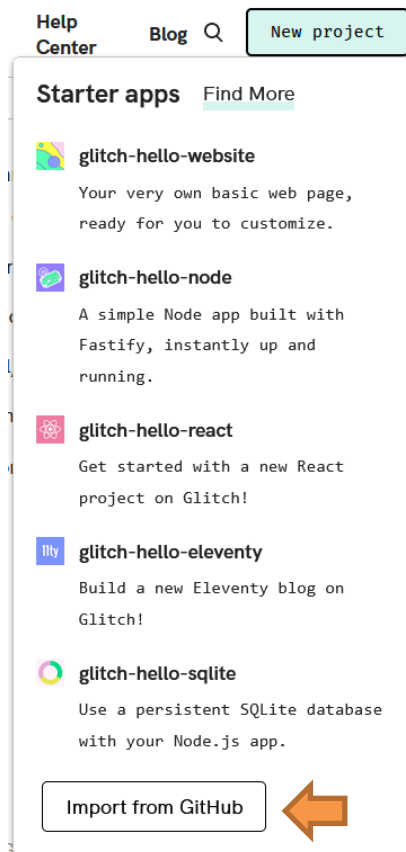
38) Sélectionnez votre compte GitHub :



39) Une fois votre compte créer, ajoutez un nouveau projet :



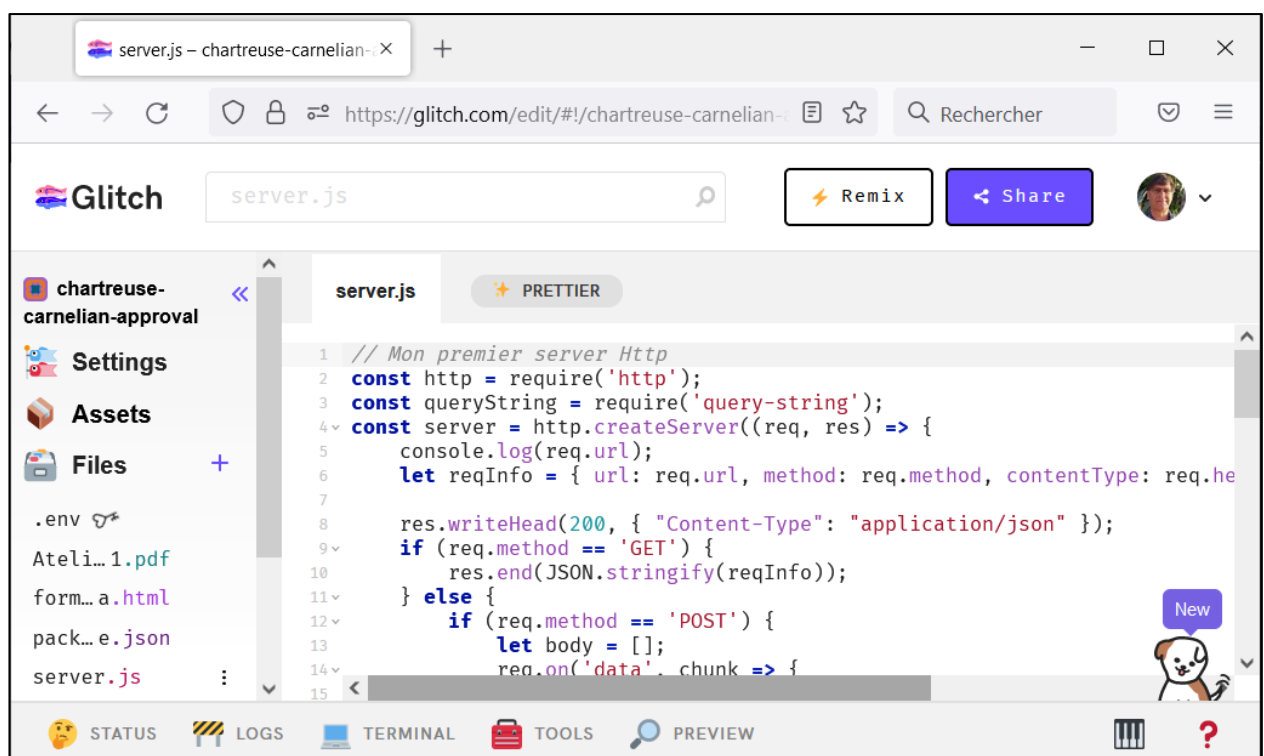
40) Importez à partir d'un répertoire GitHub :



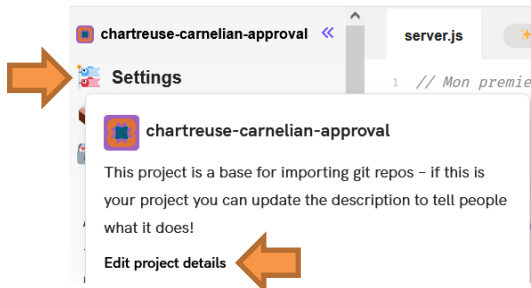
41) Copier l'url du répertoire GitHub et coller le dans le dialogue :



42) Interface de votre projet sur Glitch :



#### 43) Changer le domaine de votre site :



**Edit Project Details**

PROJECT NAME  
**chartreuse-carnelian-approval**

DESCRIPTION  
This project is a base for importing git repos – if this is your project you can update the description to tell people what it does!

**Edit Project Details**

PROJECT NAME  
**<votre numéro d'admission>-Atelier-1**

DESCRIPTION  
**Mon premier serveur Http**

#### 44) Connaître l'URL de votre site :



**Share Project** ✕

**Project permissions**  
Everyone can see your project.

**Invite project members** to edit your code.

**Project members**

	Nicolas-Chourot	Owner
--	-----------------	-------

**Project links:** Anyone with these links can view your project.

Live site	<a href="https://202212345-atelier-1.glitch.me">https://202212345-atelier-1.glitch.me</a>	<a href="#">🔗</a>
Code	<a href="https://glitch.com/edit/#!/202212345-atelier-1">https://glitch.com/edit/#!/202212345-atelier-1</a>	<a href="#">🔗</a>



45) Faire les modifications suivantes :

Dans formData.html :

```
<form action="https://202212345-atelier-1.glitch.me/api/contacts" method="POST">
```

Note : utilisez l'URL de votre site Glitch

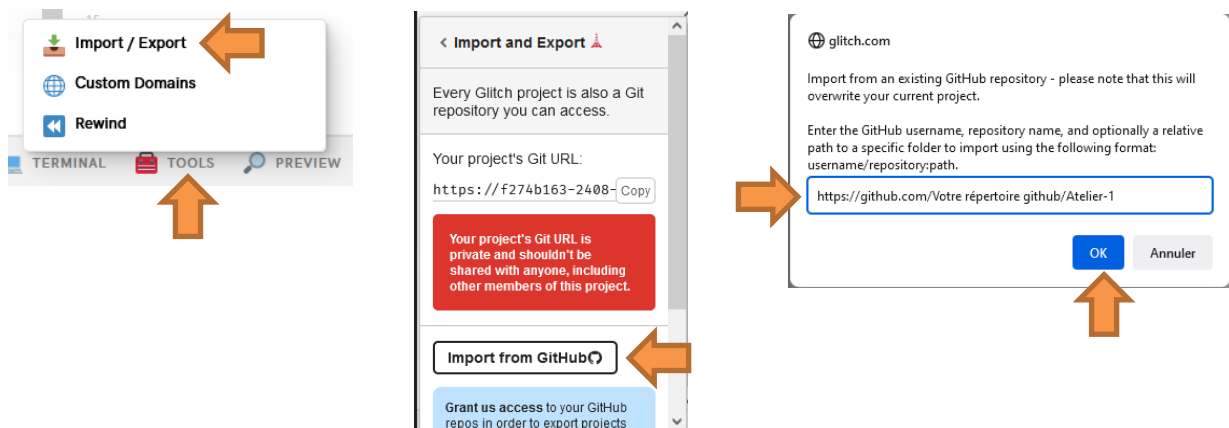
Dans package.json:

```
{
  "name": "atelier-1",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "author": "Nicolas Chourot",
  "license": "ISC",
  "dependencies": {
    "query-string": "^7.1.1"
  }
}
```

Note : n'oubliez pas la virgule après l'attribut "test":... ,

46) Commettre les changements dans GitHub

47) Mettre à jour votre site Glitch avec la version GitHub :



Note : cela peut prendre plus minutes

48) Vérifier le bon fonctionnement de votre site Glitch avec Postman et la version locale de formdata.html (utilisez l'url de votre site Glitch plutôt que <http://localhost:50000>...)

