

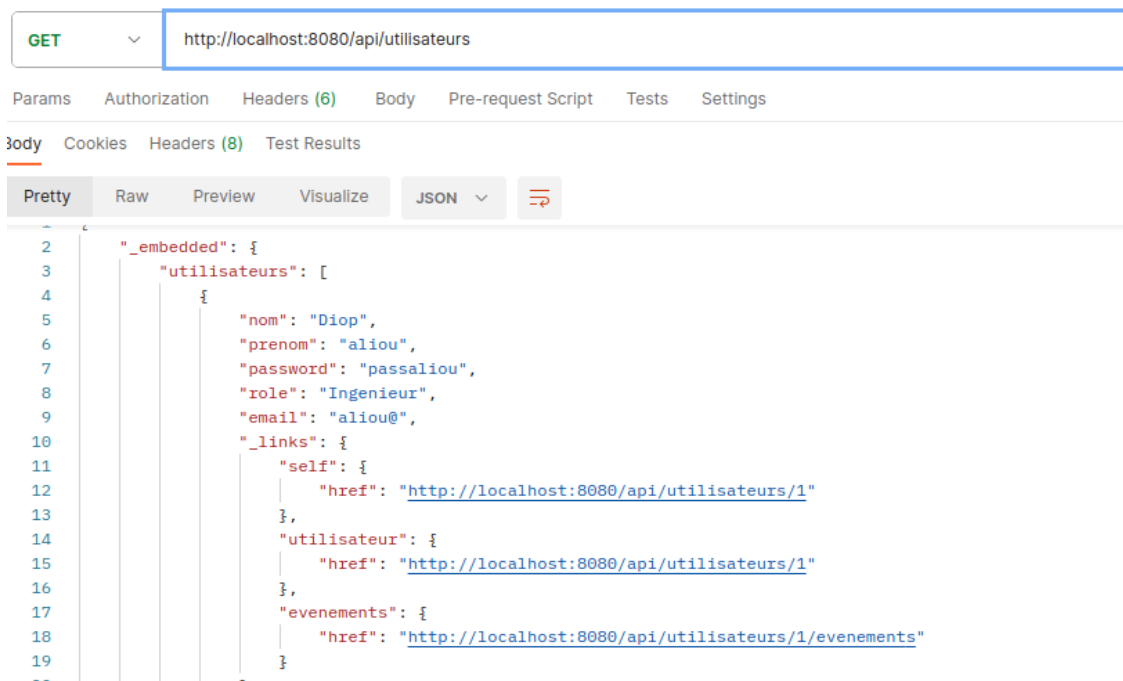
Rapport de teste du backend du projet gestion d'événement :

Suite a un requête http (GET) sur sur notre api (<http://localhost:8080/api>) on a le résultat suivant qui donne les liens vers les différentes Classes (Evenement , Prestataire , Service , Utilisateur) de notre api:

```
{
  "_links": {
    "prestataires": {
      "href": "http://localhost:8080/api/prestataires"
    },
    "utilisateurs": {
      "href": "http://localhost:8080/api/utilisateurs"
    },
    "services": {
      "href": "http://localhost:8080/api/services"
    },
    "evenements": {
      "href": "http://localhost:8080/api/evenements"
    },
    "profile": {
      "href": "http://localhost:8080/api/profile"
    }
  }
}
```

Utilisateur :

une requête (GET) sur <http://localhost:8080/api/utilisateurs> nous donne la liste des utilisateurs :



The screenshot shows a REST client interface with a GET request to <http://localhost:8080/api/utilisateurs>. The response is displayed in JSON format, showing a list of users. The first user is 'Diop', 'aliou', with password 'passaliou', role 'Ingenieur', and email 'aliou@'. The response also includes links to the user's profile, events, and other resources.

```
2  {
3    "_embedded": {
4      "utilisateurs": [
5        {
6          "nom": "Diop",
7          "prenom": "aliou",
8          "password": "passaliou",
9          "role": "Ingenieur",
10         "email": "aliou@",
11         "_links": {
12           "self": {
13             "href": "http://localhost:8080/api/utilisateurs/1"
14           },
15           "utilisateur": {
16             "href": "http://localhost:8080/api/utilisateurs/1"
17           },
18           "evenements": {
19             "href": "http://localhost:8080/api/utilisateurs/1/evenements"
20           }
21         }
22       ]
23     }
24   }
```

```

    },
    {
      "nom": "Ndiaye",
      "prenom": "Modou",
      "password": "passmodou",
      "role": "Traiteur",
      "email": "moudou@",
      "_links": {
        "self": {
          "href": "http://localhost:8080/api/utilisateurs/2"
        },
        "utilisateur": {
          "href": "http://localhost:8080/api/utilisateurs/2"
        },
        "evenements": {
          "href": "http://localhost:8080/api/utilisateurs/2/evenements"
        }
      }
    },
    {
      "nom": "Fall",
      "prenom": "Issa",
      "password": "passissa",
      "role": "Animatrice",
      "email": "issa@",
      "_links": {
        "self": {
          "href": "http://localhost:8080/api/utilisateurs/3"
        },
        "utilisateur": {
          "href": "http://localhost:8080/api/utilisateurs/3"
        },
        "evenements": {
          "href": "http://localhost:8080/api/utilisateurs/3/evenements"
        }
      }
    }
  ]
}

```

Remarque :

On pouvait faire une requête Get sur un des utilisateurs en utilisant son id sur le requête pour l'identifier : <http://localhost:8080/api/utilisateurs/id>

on peut aussi appliquer les méthodes comme DELETE , POST , PATCH ...

exemple :

on peut supprimer l'utilisateur (Aliou Diop) dont le id = 1 avec la requete (DELETE)
<http://localhost:8080/api/utilisateurs/1>

```
DELETE http://localhost:8080/api/utilisateurs/1

Body 200 OK 15 ms 656 B Save as example

Pretty Raw Preview Visualize JSON

1  {
2    "nom": "Diop",
3    "prenom": "aliou",
4    "password": "passaliou",
5    "role": "Ingenieur",
6    "email": "aliou@",
7    "_links": {
8      "self": {
9        "href": "http://localhost:8080/api/utilisateurs/1"
10      },
11      "utilisateur": {
12        "href": "http://localhost:8080/api/utilisateurs/1"
13      },
14      "evenements": {
15        "href": "http://localhost:8080/api/utilisateurs/1/evenements"
16      }
17    }
18  }
```

après l'exécution du requête on aura :

```
GET http://localhost:8080/api/utilisateurs

Body 200 OK 23 ms 2.32 KB Save as example

Pretty Raw Preview Visualize JSON

1  {
2    "_embedded": {
3      "utilisateurs": [
4        {
5          "nom": "Ndiaye",
6          "prenom": "Modou",
7          "password": "passmodou",
8          "role": "Traiteur",
9          "email": "moudou@",
10         "_links": {
11           "self": {
12             "href": "http://localhost:8080/api/utilisateurs/2"
13           },
14           "utilisateur": {
15             "href": "http://localhost:8080/api/utilisateurs/2"
16           },
17           "evenements": {
18             "href": "http://localhost:8080/api/utilisateurs/2/evenements"
19           }
20         }
21       }
22     }
23   }
```

on peut voir que si on fait get sur les utilisateurs on verra que l'utilisateur Aliou diop n'est plus la .

Post sur Utilisateur : création d'un nouvel utilisateur

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/api/utilisateurs`. The 'Body' tab is selected, and the request body is a JSON object. The 'JSON' format is chosen from the dropdown menu.

```
1 {
2   "nom": "Diouf",
3   "prenom": "Baye mor",
4   "password": "passmodou",
5   "role": "Infographe",
6   "email": "bmd@"
7 }
```

et après la requête on aura :

The screenshot shows the JSON response from the POST request. It includes the user details and a set of links for the newly created user.

```
{
  "nom": "Diouf",
  "prenom": "Baye mor",
  "password": "passmodou",
  "role": "Infographe",
  "email": "bmd@",
  "_links": {
    "self": {
      "href": "http://localhost:8080/api/utilisateurs/6"
    },
    "utilisateur": {
      "href": "http://localhost:8080/api/utilisateurs/6"
    },
    "evenements": {
      "href": "http://localhost:8080/api/utilisateurs/6/evenements"
    }
  }
}
```

Patch sur un utilisateur : on va modifier l'utilisateur récemment créé (Baye mor Diouf) en remplaçant son prénom par 'Souleymane' et son nom par 'Diagne' :

The screenshot shows a REST client interface with a PATCH request to `http://localhost:8080/api/utilisateurs/6`. The 'Body' tab is selected, and the request body is a JSON object. The 'JSON' format is chosen from the dropdown menu. A 'Beautify' button is visible on the right.

```
1 {
2   "nom": "Diagne",
3   "prenom": "Souleymane"
4 }
```

et après la requête on verra que l'utilisateur est modifié :

```
1  {
2    "nom": "Diagne",
3    "prenom": "Souleymane",
4    "password": "passmodou",
5    "role": "Infographe",
6    "email": "bmd@",
7    "_links": {
8      "self": {
9        "href": "http://localhost:8080/api/
10               utilisateurs/6"
11      },
12      "utilisateur": {
13        "href": "http://localhost:8080/api/
```

Nb : Pour le reste du rapport nous allons nous concentrer sur la Classe Evenement du fait qu'il est la classe qui contient tout les différents types de liaisons entre les classes de notre api .

Evenement :

une requête (GET) sur "<http://localhost:8080/api/evenements>" nous donne la liste des utilisateurs :

GET http://localhost:8080/api/evenements

Params Authorization Headers (6) Body Pre-request Script Tests Settings

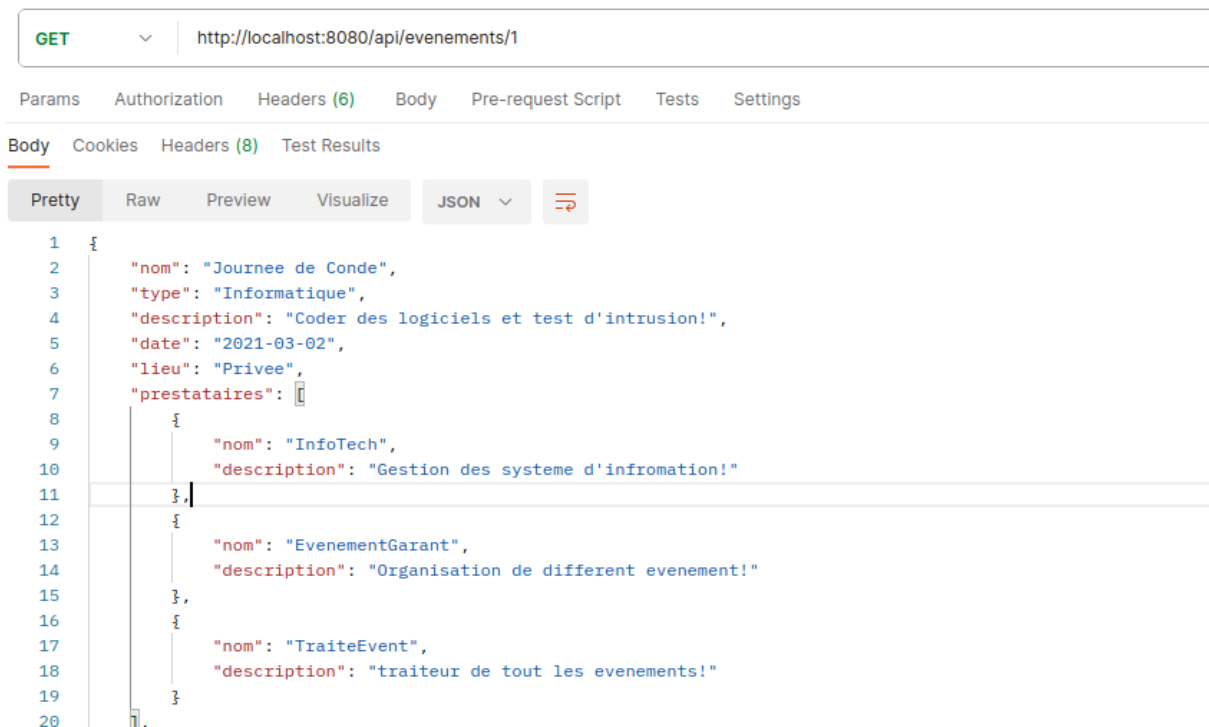
Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1  {
2    "_embedded": {
3      "evenements": [
4        {
5          "nom": "Dinee de Charite",
6          "type": "Soire",
7          "description": "Soire De ceremonie pour les employees!",
8          "date": "2021-09-01",
9          "lieu": "Sorano",
10         "prestataires": [],
11         "_links": {
12           "self": {
13             "href": "http://localhost:8080/api/evenements/2"
14           },
15           "evenement": {
16             "href": "http://localhost:8080/api/evenements/2"
17           },
18           "prestataire": {
19             "href": "http://localhost:8080/api/evenements/2/prestataire"
20           },
21           "utilisateur": {
22             "href": "http://localhost:8080/api/evenements/2/utilisateur"
23           }
24         }
25       },
26     },
27     {
28       "nom": "Dinee",
29       "type": "Soire",
```

```
27     "nom": "Dinee",
28     "type": "Soire",
29     "description": "Soire pour se sentir!",
30     "date": "2021-09-01",
31     "lieu": "Palais d'or",
32     "prestataires": [],
33     "_links": {
34       "self": {
35         "href": "http://localhost:8080/api/evenements/3"
36       },
37       "evenement": {
38         "href": "http://localhost:8080/api/evenements/3"
39       },
40       "prestataire": {
41         "href": "http://localhost:8080/api/evenements/3/prestataire"
42       },
43       "utilisateur": {
44         "href": "http://localhost:8080/api/evenements/3/utilisateur"
45       }
46     }
47   },
48   {
49     "nom": "Mariage",
50     "type": "Cremonie de Famille",
51     "description": "Mariage, liaison par consentement!",
52     "date": "2021-09-01",
53     "lieu": "Medina Fall THIES",
54     "prestataires": [],
55     "_links": {
56       "self": {
57         "href": "http://localhost:8080/api/evenements/4"
58       },
59       "evenement": {
60         "href": "http://localhost:8080/api/evenements/4"
61       },
62       "prestataire": {
```

De meme on peut utiliser l'id d'un événement pour acceder a un événement donne : avec une requet (Get) sur "<http://localhost:8080/api/evenements/1>" nous permet d'accéder a l'événement dont l'id est 1 :



The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/evenements/1`. The response is displayed in JSON format, showing an event with ID 1. The event details are as follows:

```
1 {
2   "nom": "Journee de Conde",
3   "type": "Informatique",
4   "description": "Coder des logiciels et test d'intrusion!",
5   "date": "2021-03-02",
6   "lieu": "Privee",
7   "prestataires": [
8     {
9       "nom": "InfoTech",
10      "description": "Gestion des systeme d'infromation!"
11    },
12    {
13      "nom": "EvenementGarant",
14      "description": "Organisation de different evenement!"
15    },
16    {
17      "nom": "TraiteEvent",
18      "description": "traiteur de tout les evenements!"
19    }
20  ]
21 }
```

Liaison entre Evenement et Prestataire :

Entre Evenement et prestataire on a une liaison ManyToMany : un événement peut contenir plusieurs prestataire différent et de même un prestataire peut participer dans plusieurs événement .

De ce fait a partir d'un événement on peut accéder directement aux différentes prestataires de celle-ci .

Exemple : pour l'événement dont l'id = 1 on peut accéder a ces prestataires a partir d'un requête (GET) sur :

["http://localhost:8080/api/evenements/1/prestataire"](http://localhost:8080/api/evenements/1/prestataire)

```
GET http://localhost:8080/api/evenements/1/prestataire

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

1  {
2    "_embedded": {
3      "prestataires": [
4        {
5          "nom": "InfoTech",
6          "description": "Gestion des systeme d'infromation!",
7          "_links": {
8            "self": {
9              "href": "http://localhost:8080/api/prestataires/1"
10           },
11           "prestataire": {
12             "href": "http://localhost:8080/api/prestataires/1"
13           },
14           "evenements": {
15             "href": "http://localhost:8080/api/prestataires/1/evenements"
16           },
17           "services": {
18             "href": "http://localhost:8080/api/prestataires/1/services"
19           }
20         }
21       ],
22     }
23   },
24   "nom": "EvenementGarant",
25   "description": "Organisation de different evenement!",
26   "_links": {
27     "self": {
28       "href": "http://localhost:8080/api/prestataires/2"
```

De même pour un prestataire on peut lister les différent événement dans les quelles il doit participer .

Exemple : pour le prestataire dont l'id = 2 on peut accède aux différent événements dans les quelles il doit participer avec la requête (GET) sur <http://localhost:8080/api/prestataires/1/evenements> :


```
GET http://localhost:8080/api/prestataires/1/evenements

Params Authorization Headers (6) Body Pre-request Script Tests Settings
Body Cookies Headers (8) Test Results
Pretty Raw Preview Visualize JSON

1 {
2   "_embedded": {
3     "evenements": [
4       {
5         "nom": "Journee de Conde",
6         "type": "Informatique",
7         "description": "Coder des logiciels et test d'intrusion!",
8         "date": "2021-03-02",
9         "lieu": "Privee",
10        "prestataires": [
11          {
12            "nom": "InfoTech",
13            "description": "Gestion des systeme d'infomation!"
14          },
15          {
16            "nom": "EvenementGarant",
17            "description": "Organisation de different evenement!"
18          },
19          {
20            "nom": "TraiteEvent",
21            "description": "traiteur de tout les evenements!"
22          }
23        ],
24        "_links": {
25          "self": {
26            "href": "http://localhost:8080/api/evenements/1"
27          },
28          "evenement": {
```

Est ce que la suppression d'un événement entraîne la suppression de ses prestataires ou vice-versa ? :

Supprimons l'événement dont l'id = 2 pour voir

```
{
  "nom": "Journee de Conde",
  "type": "Informatique",
  "description": "Coder des logiciels et test d'intrusion!",
  "date": "2021-03-02",
  "lieu": "Privee",
  "prestataires": [
    {
      "nom": "InfoTech",
      "description": "Gestion des systeme d'infomation!"
    },
    {
      "nom": "EvenementGarant",
      "description": "Organisation de different evenement!"
    },
    {
      "nom": "TraiteEvent",
      "description": "traiteur de tout les evenements!"
    }
  ]
},
```

On voit qu'un événement est lié à 3 prestataires différents et on peut le supprimer avec la requête (DELETE) sur <http://localhost:8080/api/evenements/1>

Et après la suppression on peut voir que les 3 prestataires concernés n'ont pas été supprimé :

```
1 {
2   "_embedded": {
3     "prestataires": [
4       {
5         "nom": "InfoTech",
6         "description": "Gestion des systeme d'infromation!",
7         "_links": {
8           "self": {
9             "href": "http://localhost:8080/api/prestataires/1"
10          },
11          "prestataire": {
12            "href": "http://localhost:8080/api/prestataires/1"
13          },
14          "evenements": {
15            "href": "http://localhost:8080/api/prestataires/1/evenements"
16          },
17          "services": {
18            "href": "http://localhost:8080/api/prestataires/1/services"
19          }
20        }
21      },
22      {
23        "nom": "EvenementGarant",
24        "description": "Organisation de different evenement!",
25        "_links": {
26          "self": {
27            "href": "http://localhost:8080/api/prestataires/2"
28          },
29          "prestataire": {
30            "href": "http://localhost:8080/api/prestataires/2"
31          },
32          "evenements": {
33            "href": "http://localhost:8080/api/prestataires/2/evenements"
34          },
35          "services": {
36            "href": "http://localhost:8080/api/prestataires/2/services"
37          }
38        }
39      }
40    ]
41  }
42 }
```

On remarque la même chose de l'autre cote : la suppression de l'un des 3 prestataires n'entraîne pas la suppression de l'événement concerne .

Cela peut être expliquer par le fait qu'un événement peut être lié avec plusieurs prestataires et vice-versa donc la suppression d 'un événement ne peut pas provoquer la suppression de ses prestataires et vice-versa.

Liaison entre Evenement et Utilisateur:

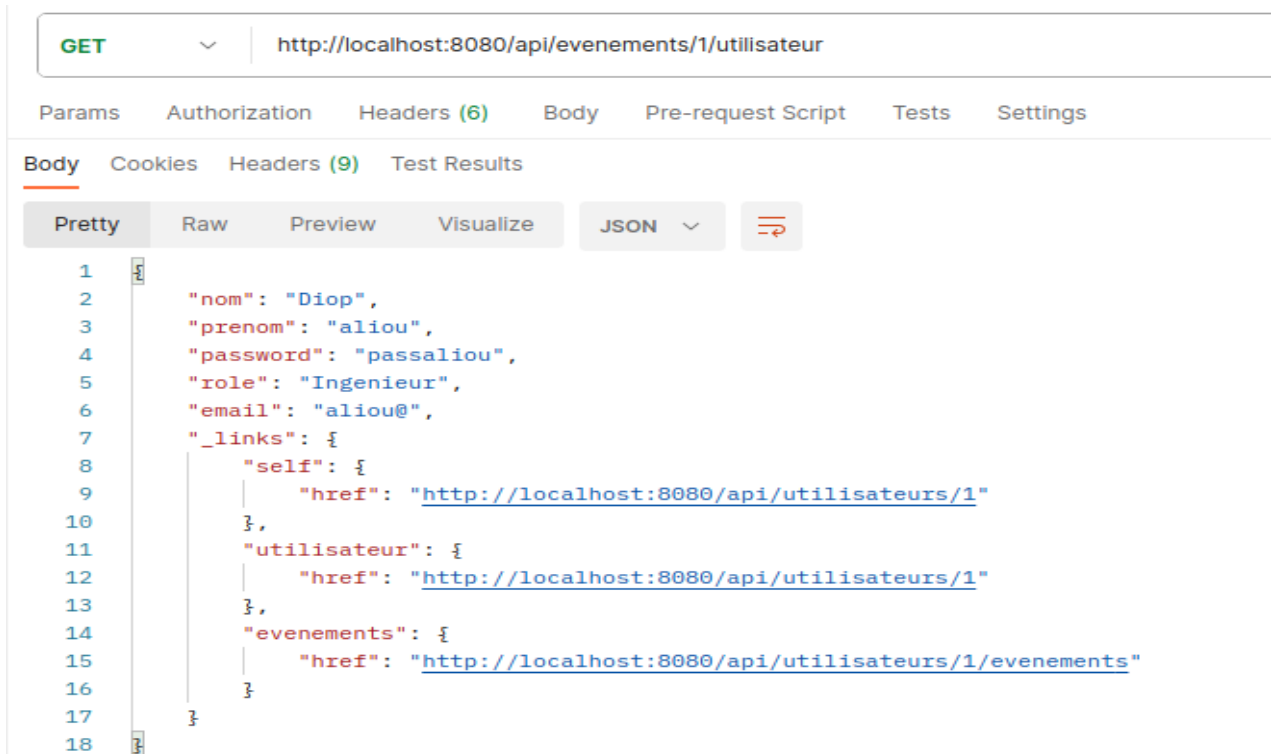
Entre Evenement et Utilisateur on a une liaison :

- ManyToOne (Evenement) : un événement ne peut contenir qu'un seul utilisateurs .
- OneToMany (Prestataire) : un utilisateur peut créer un ou plusieurs événements .

De ce fait a partir d'un événement on peut accéder directement sur l'utilisateur qui la créer .

Exemple : pour l'événement dont l'id = 1 on peut accéder a son utilisateur a partir d'un requête (GET) sur :

["http://localhost:8080/api/evenements/1/utilisateur"](http://localhost:8080/api/evenements/1/utilisateur)



GET <http://localhost:8080/api/evenements/1/utilisateur>

Params Authorization Headers (6) Body Pre-request Script Tests Settings

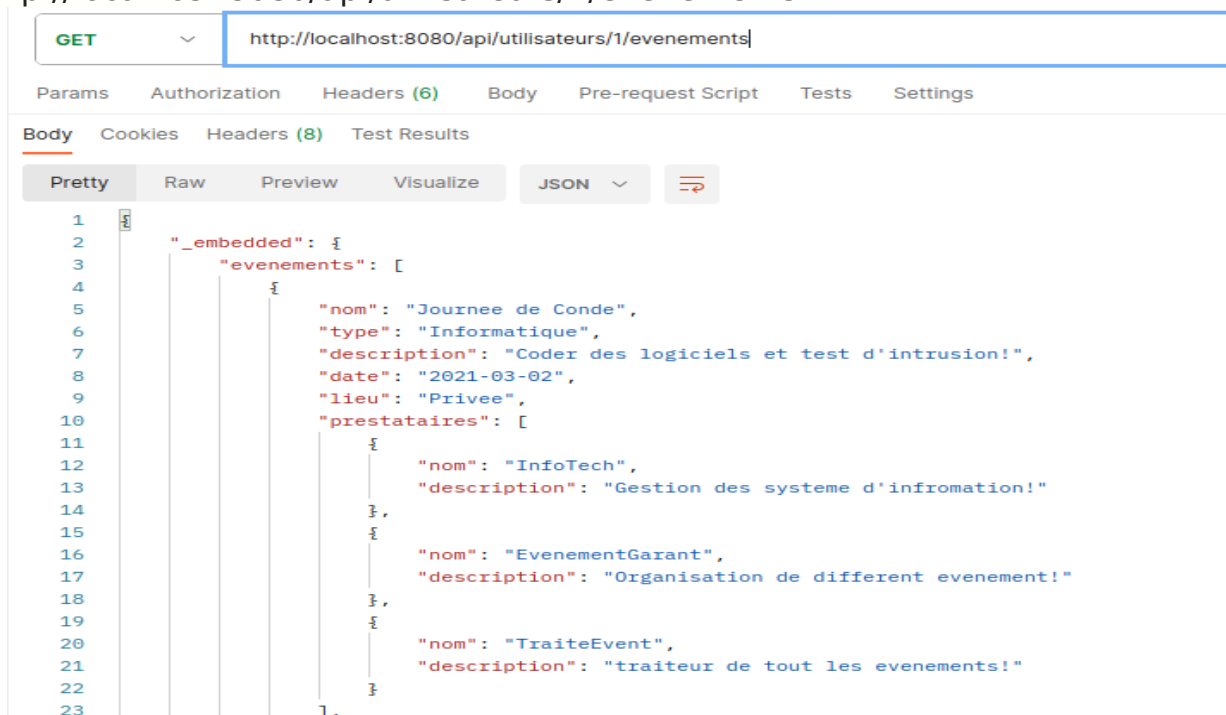
Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "nom": "Diop",
3   "prenom": "aliou",
4   "password": "passaliou",
5   "role": "Ingenieur",
6   "email": "aliou@",
7   "_links": {
8     "self": {
9       "href": "http://localhost:8080/api/utilisateurs/1"
10    },
11    "utilisateur": {
12      "href": "http://localhost:8080/api/utilisateurs/1"
13    },
14    "evenements": {
15      "href": "http://localhost:8080/api/utilisateurs/1/evenements"
16    }
17  }
18 }
```

De même pour un utilisateur on peut lister les différentes événements qu'il a créé.

Exemple : pour l'utilisateur dont l'id = 1 on peut accède aux différentes événements qu'il a créé avec la requête (GET) sur <http://localhost:8080/api/utilisateurs/1/evenements> :



GET <http://localhost:8080/api/utilisateurs/1/evenements>

Params Authorization Headers (6) Body Pre-request Script Tests Settings

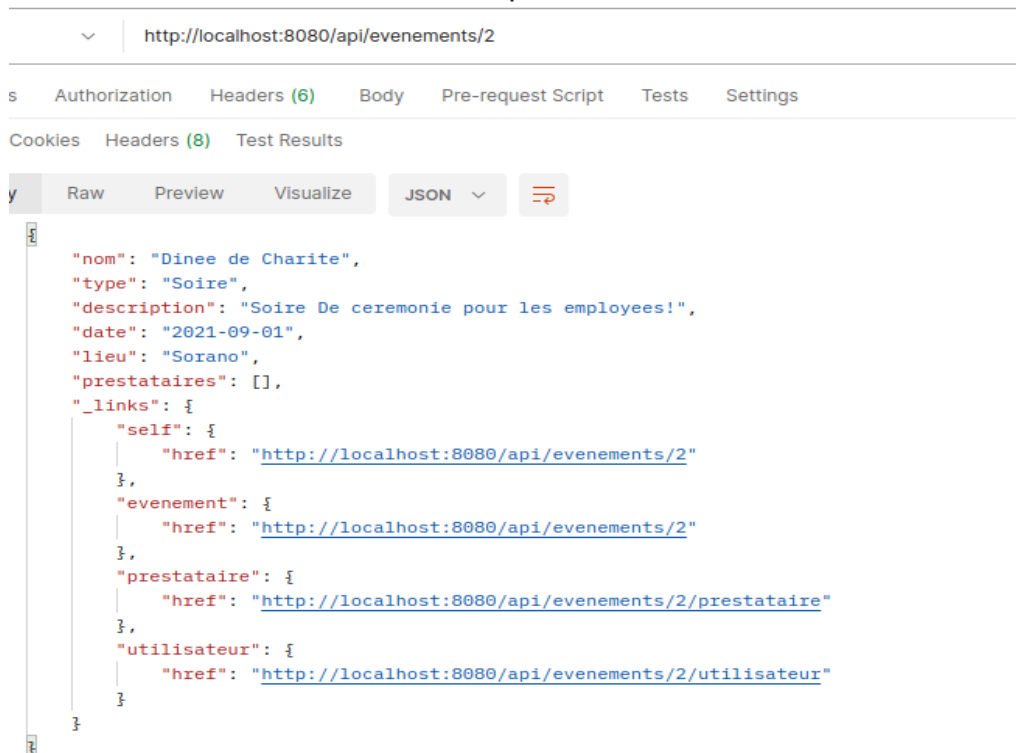
Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "_embedded": {
3     "evenements": [
4       {
5         "nom": "Journee de Conde",
6         "type": "Informatique",
7         "description": "Coder des logiciels et test d'intrusion!",
8         "date": "2021-03-02",
9         "lieu": "Privee",
10        "prestataires": [
11          {
12            "nom": "InfoTech",
13            "description": "Gestion des systeme d'infromation!"
14          },
15          {
16            "nom": "EvenementGarant",
17            "description": "Organisation de different evenement!"
18          },
19          {
20            "nom": "TraiteEvent",
21            "description": "traiteur de tout les evenements!"
22          }
23        ]
24      }
25    ]
26  }
27 }
```

Est ce que la suppression d'un événement entraîne la suppression de son utilisateur ou vice-versa ? :

Supprimons l'événement dont l'id = 2 pour voir



```
http://localhost:8080/api/evenements/2

s Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies Headers (8) Test Results
y Raw Preview Visualize JSON

{
  "nom": "Dinee de Charite",
  "type": "Soire",
  "description": "Soire De ceremonie pour les employees!",
  "date": "2021-09-01",
  "lieu": "Sorano",
  "prestataires": [],
  "_links": {
    "self": {
      "href": "http://localhost:8080/api/evenements/2"
    },
    "evenement": {
      "href": "http://localhost:8080/api/evenements/2"
    },
    "prestataire": {
      "href": "http://localhost:8080/api/evenements/2/prestataire"
    },
    "utilisateur": {
      "href": "http://localhost:8080/api/evenements/2/utilisateur"
    }
  }
}
```

cet événement est lie a l'utilisateur suivant :



```
1 {
2   "nom": "Fall",
3   "prenom": "Issa",
4   "password": "passissa",
5   "role": "Animatrice",
6   "email": "issa@",
7   "_links": {
8     "self": {
9       "href": "http://localhost:8080/api/utilisateurs/3"
10    },
11    "utilisateur": {
12      "href": "http://localhost:8080/api/utilisateurs/3"
13    },
14    "evenements": {
15      "href": "http://localhost:8080/api/utilisateurs/3/evenements"
16    }
17  }
18 }
```

On peut supprimer cet événement avec la requête (DELETE) sur <http://localhost:8080/api/evenements/1>

Et après la suppression on peut voir que l'utilisateur concerné n'a pas été supprimé :

```
GET http://localhost:8080/api/utilisateurs

Params Authorization Headers (6) Body Pre-request Script Tests Settings

body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↗

1 2
2   "_embedded": {
3     "utilisateurs": [
4       {
5         "nom": "Diop",
6         "prenom": "aliou",
7         "password": "passaliou",
8         "role": "Ingenieur",
9         "email": "aliou@",
10        "_links": {
11          "self": {
12            "href": "http://localhost:8080/api/utilisateurs/1"
13          },
14          "utilisateur": {
15            "href": "http://localhost:8080/api/utilisateurs/1"
16          },
17          "evenements": {
18            "href": "http://localhost:8080/api/utilisateurs/1/evenements"
19          }
20        }
21      },
22      {
23        "nom": "Ndiaye",
24        "prenom": "Modou",
25        "password": "passmodou",
26        "role": "Traiteur",
27        "email": "moudou@",
28        "_links": {
29          "self": {
```

Pour l'utilisateur on a essayer de supprimer l'utilisateur dont l'id = 1 pour voir si les événements qu'il a créer vont eu aussi être supprimer .

Les événements de cet utilisateur sont :

```
"evenements": [
  {
    "nom": "Journee de Conde",
    "type": "Informatique",
    "description": "Coder des logiciels et test d'intrusion!",
    "date": "2021-03-02",
    "lieu": "Privee",
    "prestataires": [
      {
        "nom": "InfoTech",
        "description": "Gestion des systeme d'infromation!"
      },
      {
        "nom": "EvenementGarant",
        "description": "Organisation de different evenement!"
      },
      {
        "nom": "TraiteEvent",
        "description": "traiteur de tout les evenements!"
      }
    ]
  },
  ...
]
```

```

{
  "nom": "Journee de Conde",
  "type": "Informatique",
  "description": "Coder des logiciels et test d'intrusion!",
  "date": "2021-09-01",
  "lieu": "Privee",
  "prestataires": [],
  "_links": {
    "self": {
      "href": "http://localhost:8080/api/evenements/6"
    },
    "evenement": {
      "href": "http://localhost:8080/api/evenements/6"
    },
    "prestataire": {
      "href": "http://localhost:8080/api/evenements/6/prestataire"
    },
    "utilisateur": {
      "href": "http://localhost:8080/api/evenements/6/utilisateur"
    }
  }
}

```

Après la suppression de l'utilisateur avec la requête (DELETE) sur <http://localhost:8080/api/utilisateurs/1> .

On fait une requête (GET) sur <http://localhost:8080/api/evenements>

```
GET http://localhost:8080/api/evenements

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
{
  "_embedded": {
    "evenements": [
      {
        "nom": "Dinee de Charite",
        "type": "Soire",
        "description": "Soire De ceremonie pour les employees!",
        "date": "2021-09-01",
        "lieu": "Sorano",
        "prestataires": [],
        "_links": {
          "self": {
            "href": "http://localhost:8080/api/evenements/2"
          },
          "evenement": {
            "href": "http://localhost:8080/api/evenements/2"
          },
          "prestataire": {
            "href": "http://localhost:8080/api/evenements/2/prestataire"
          },
          "utilisateur": {
            "href": "http://localhost:8080/api/evenements/2/utilisateur"
          }
        }
      },
      {
        "nom": "Dinee",
        "type": "Soire",
        "description": "Soire pour se sentir!",
        "date": "2021-09-01",
        "lieu": "Palais d'or",
        "prestataires": [],
        "_links": {
          "self": {
            "href": "http://localhost:8080/api/evenements/3"
          },
          "evenement": {
            "href": "http://localhost:8080/api/evenements/3"
          }
        }
      }
    ]
  }
}
```

On peut voir que les deux événement concerne ont été supprimé .

Fin .