

Service Rest

Open database API

-- Connexion

Base URL : user/

URL	Méthode	Paramètres	Rôle	Réponses possibles
check	GET		Vérifie qu'un utilisateur est connecté	{token : ddd34...,connected: true} {connected: false}
connect	POST	mail, password	Effectue la connexion avec les paramètres postés	{token : ddd34...,connected: true} {connected: false}
disconnect	GET		Déconnecte (ferme la session) de l'utilisateur actif	{connected: false}
exists/:mail	GET		Vérifie que l'email existe dans la base de données	{exists: true ,mail: "admin@local.fr"} {exists: false ,mail: "jcheron@local.fr"}
add	POST	mail, password	Ajoute l'utilisateur avec les paramètres postés	{token : ddd34...,inserted: true} {inserted: false}
check/:mail/:password	GET		Pour tests : effectue une connexion avec les paramètres passé	{token : ddd34...,connected: true} {connected: false}

Exemple de connexion avec Angular :

```

data.posted={"mail":"admin@local.fr","password":"0000"};
this.connect(data,function(d){console.log("Connexion réussie, token : "+d.token)});

this.connect=function(response,callBack){
  var request = $http({
    method: "POST",
    url: restConfig.server.restServerUrl+"user/connect",
    data: $.param(response.posted),
    headers: self.headers
  });
  request.success(function(data, status, headers, config) {
    callback(data);//Connexion réussie
  }).error(function(data, status, headers, config){
    self.addMessage({type: "warning", content:"Erreur de connexion au serveur, statut de la réponse : "+status+"<br>"+data.message});
  });
}

```

Conserver la session (PHP) avec Angular :

```
module.exports=function($routeProvider,$locationProvider,$httpProvider) {  
  //$httpProvider.defaults.useXDomain = true;  
  $httpProvider.defaults.withCredentials = true;
```

-- Brasseurs

-- Liste des brasseurs

URL	Méthode	Paramètres	Rôle
/breweries	GET		Retourne tous les brasseurs

Exemple de réponse :

URL : /breweries

```
[  
  {  
    "id": "1",  
    "name": "Anchor",  
    "url": "http://www.anchorbrewing.com/",  
    "photo": "",  
    "thumbnail": "",  
    "created_at": "2015-03-06 19:11:39",  
    "updated_at": "2015-03-06 21:53:12"  
  },  
  {  
    "id": "6",  
    "name": "Dogfish Head",  
    "url": "http://www.dogfish.com/",  
    "photo": "",  
    "thumbnail": "",  
    "created_at": "2015-03-07 02:19:22",  
    "updated_at": "2015-03-07 02:19:22"  
  },  
  ...  
]
```

Erreurs possibles :

Code	Raison	Réponse
404	Aucune brasserie trouvée dans la BDD	{code:404, message: "Aucune brasserie trouvée."}

-- Brasseur correspondant à un id

URL	Méthode	Paramètres	Rôle
/breweries/:id	GET		Retourne le brasseur dont l'id est passé en paramètre

Exemple de réponse :

URL : /breweries/7

```
{
  "id": "7",
  "name": "Stone Brewing Co.",
  "url": "http:\\\\www.stonebrew.com\\/",
  "photo": "",
  "thumbnail": "",
  "created_at": "2015-03-07 02:19:22",
  "updated_at": "2015-03-07 02:19:22"
}
```

Erreurs possibles :

Code	Raison	Réponse
404	Aucune brasserie trouvée dans la BDD correspondant à l'id passé en paramètre	{"message": "Ooops! Le brasseur 7 est introuvable", "code": 404}

-- ajout d'un brasseur

URL	Méthode	Paramètres	data	Rôle
/breweries	POST	token	{"name": "Fischer", "url": "http://www.fischer.com"}	Ajoute le brasseur posté

Exemple de réponse :

URL : /breweries
Méthode : POST

```
{
  "data": {
    "id": "32",
    "name": "Fischer",
    "url": "http:\\\\www.fischer.com",
    "photo": null,
    "thumbnail": null,
    "created_at": "2015-03-08 19:47:47",
    "updated_at": "2015-03-08 19:47:47"
  },
  "message": {
    "type": "success",
    "content": "'Fischer' a \u00e9t\u00e9 correctement ajout\u00e9 dans les brasseries."
  }
}
```

Erreurs possibles :

Code	Raison	Réponse
401	Utilisateur non connecté (token invalide ou absent et force non utilisé)	{"message":"Vous n'avez pas les droits pour ajouter une brasserie","code":401}
409	Brasseur dont le nom existe déjà dans la base	{"message":"Impossible d'ajouter 'Fischer' dans la base de données","code":409}

-- Modification d'un brasseur

URL	Méthode	Paramètres	data	Rôle
/breweries/:id	PUT	token	{"name":"Fischer modifié","url":"http://www.fischer.com"}	modifie le brasseur posté

Exemple de réponse :

URL : /breweries/32

Méthode : PUT

```
{
  "data": {
    "id": "32",
    "name": "Fischer modifié",
    "url": "http://www.fischer.com",
    "photo": null,
    "thumbnail": null,
    "created_at": "2015-03-08 19:47:47",
    "updated_at": "2015-03-08 19:47:47"
  },
  "message": {
    "type": "success",
    "content": "'Fischer' a \u00e9t\u00e9 correctement modifi\u00e9 dans les brasseries."
  }
}
```

Erreurs possibles :

Code	Raison	Réponse
401	Utilisateur non connecté (token invalide ou absent et force non utilisé)	{"message":"Vous n'avez pas les droits pour modifier une brasserie","code":401}
404	Le brasseur n'existe plus dans la base	{"message":"Suppression : La brasserie 'Fischer' n'existe plus dans la base de données","code":409}
409	Brasseur dont le nom existe déjà dans la base	{"message":"Impossible de modifier 'Fischer' dans la base de données","code":409}

-- Suppression d'un brasseur

URL	Méthode	Paramètres	data	Rôle
/breweries/:id	DELETE	token		supprime le brasseur dont l'id est passée dans l'URL

Exemple de réponse :

URL : /breweries/32

Méthode : DELETE

```

{
  "data":{
    "id":"32",
    "name":"Fischer",
    "url":"http://www.fischer.com",
    "photo":null,
    "thumbnail":null,
    "created_at":"2015-03-08 19:47:47",
    "updated_at":"2015-03-08 19:47:47"
  },
  "message":{
    "type":"success",
    "content":"'Fischer' a été correctement supprimé de l'ensemble des brasseries."
  }
}

```

Erreurs possibles :

Code	Raison	Réponse
401	Utilisateur non connecté (token invalide ou absent et force non utilisé)	{"message":"Vous n'avez pas les droits pour supprimer une brasserie","code":401}
404	Le brasseur n'existe plus dans la base	{"message":"Suppression : La brasserie 'Fischer' n'existe plus dans la base de données","code":409}
409	Brasseur ayant des bières associées	{"message":"Impossible de supprimer 'Fischer' dans la base de données","code":409}

-- Bières

-- Liste des bières

URL	Méthode	Paramètres	Rôle
/beers	GET		Retourne toutes les bières

Exemple de réponse :

URL : /beers

```

[
  {
    "id":"1",
    "name":"Strawberry Harvest",
    "description":"Strawberry Harvest Lager is a wheat beer ...",
    "abv":"4.2",
    "photo":null,
    "created_at":"2015-03-07 00:43:23",
    "updated_at":"2015-03-07 00:43:23",
  }
]

```

```
    "idBrewery": "1"
  },
  {
    "id": "2",
    "name": "Brooklyn Detonation Ale",
    "description": "For about eight years now...",
    "abv": "bda",
    "photo": null,
    "created_at": "2015-03-08 21:06:46",
    "updated_at": "2015-03-08 21:06:46",
    "idBrewery": "12"
  },
  ...
]
```

Erreurs possibles :

Code	Raison	Réponse
404	Aucune bière trouvée dans la BDD	{code:404, message: "Aucune bière trouvée."}

-- Bière correspondant à un id

URL	Méthode	Paramètres	Rôle
/beers/:id	GET		Retourne la bière dont l'id est passé en paramètre

Exemple de réponse :

URL : /beers/1

```
{
  "id": "1",
  "name": "Strawberry Harvest",
  "description": "Strawberry Harvest Lager is a wheat beer ...",
  "abv": "4.2",
  "photo": null,
  "created_at": "2015-03-07 00:43:23",
  "updated_at": "2015-03-07 00:43:23",
  "idBrewery": "1"
}
```

Erreurs possibles :

Code	Raison	Réponse
404	Aucune bière trouvée dans la BDD correspondant à l'id passé en paramètre	{"message": "Ooops! La bière d'id '1' est introuvable", "code": 404}

-- Liste des bières d'une brasserie

URL	Méthode	Paramètres	Rôle
/beers/brewery/:id	GET		Retourne les bières correspondant à la brasserie dont l'id est passé en paramètre

Exemple de réponse :

URL : /beers/brewery/12

```
[
  {
    "id": "2",
    "name": "Brooklyn Detonation Ale",
    "description": "For about eight years now, we've brewed...",
    "abv": "bda",
    "photo": null,
    "created_at": "2015-03-08 21:06:46",
    "updated_at": "2015-03-08 21:06:46",
    "idBrewery": "12"
  }, {
    "id": "3",
    "name": "Brooklyn Ama Bruna",
    "description": "Brewed by Amarcord Brewery in the small medieval town of Apecchio, Italy...",
    "abv": "bab",
    "photo": null,
    "created_at": "2015-03-08 21:06:46",
    "updated_at": "2015-03-08 21:06:46",
    "idBrewery": "12"
  }
]
```

Erreurs possibles :

Code	Raison	Réponse
404	Aucune bière trouvée dans la BDD correspondant correspondant à la brasserie dont l'id passé en paramètre	{"message": "Ooops! La brasserie 12 ne dispose d'aucune bière", "code": 404}

-- ajout d'une bière

URL	Méthode	Paramètres	data	Rôle
/beers	POST	token	{"name": "Brooklyn Ama Bruna modif", "description": "Brewed by Amarcord Brewery in the small medieval town of Apecchio...", "abv": "bab"}	Ajoute la bière postée

Exemple de réponse :

URL : /beers
Méthode : POST

```
{
  "data": {
    "id": "3",
    "name": "Brooklyn Ama Bruna",
    "description": "Brewed by Amarcord Brewery in the small medieval town of Apecchio...",
  }
}
```

```
    "abv": "bab",
    "photo": null,
    "created_at": "2015-03-08 21:06:46",
    "updated_at": "2015-03-08 21:06:46",
    "idBrewery": "12"
  },
  "message": {
    "type": "success",
    "content": "'Brooklyn Ama Bruna' a \u00e9t\u00e9 correctement ajout\u00e9e dans les bi\u00e8res."
  }
}
```

Erreurs possibles :

Code	Raison	R\u00e9ponse
401	Utilisateur non connect\u00e9 (token invalide ou absent et force non utilis\u00e9)	{"message": "Vous n'avez pas les droits pour ajouter une bi\u00e8re", "code": 401}
409	Bi\u00e8re dont le nom existe d\u00e9j\u00e0 dans la base	{"message": "Impossible d'ajouter 'Fischer' dans la base de donn\u00e9es", "code": 409}

-- Modification d'une bi\u00e8re

URL	M\u00e9thode	Param\u00e8tres	data	R\u00f4le
/beers/:id	PUT	token	{"name": "Brooklyn Ama Bruna modif", "description": "Brewed by Amarcord Brewery in the small medieval town of Apecchio...", "abv": "bab"}	modifie la bi\u00e8re post\u00e9e

Exemple de r\u00e9ponse :

URL : /beers/3
M\u00e9thode : PUT

```
{
  "data": {
    "id": "3",
    "name": "Brooklyn Ama Bruna modif",
    "description": "Brewed by Amarcord Brewery in the small medieval town of Apecchio...",
    "abv": "bab",
    "photo": null,
    "created_at": "2015-03-08 21:06:46",
    "updated_at": "2015-03-08 21:06:46",
    "idBrewery": "12"
  },
  "message": {
    "type": "success",
    "content": "'Brooklyn Ama Bruna modif' a \u00e9t\u00e9 correctement modifi\u00e9e dans les brasseries."
  }
}
```

Erreurs possibles :

Code	Raison	Réponse
401	Utilisateur non connecté (token invalide ou absent et force non utilisé)	{"message":"Vous n'avez pas les droits pour modifier une bière","code":401}
404	La bière n'existe plus dans la base	{"message":"Suppression : La brasserie 'Brooklyn Ama Bruna' n'existe plus dans la base de données","code":409}
409	Bière dont le nom existe déjà dans la base	{"message":"Impossible de modifier 'Brooklyn Ama Bruna' dans la base de données","code":409}

-- Suppression d'une bière

URL	Méthode	Paramètres	data	Rôle
/beers/:id	DELETE	token		supprime la bière dont l'id est passée dans l'URL

Exemple de réponse :

URL : /beers/3

Méthode : DELETE

```
{
  "data":{
    "id":"3",
    "name":"Brooklyn Ama Bruna",
    "description":"Brewed by Amarcord Brewery in the small medieval town of Apecchio...",
    "abv":"bab",
    "photo":null,
    "created_at":"2015-03-08 21:06:46",
    "updated_at":"2015-03-08 21:06:46",
    "idBrewery":"12"
  },
  "message":{
    "type":"success",
    "content":"'Brooklyn Ama Bruna' a \u00e9t\u00e9 correctement supprim\u00e9 de l'ensemble des bières."
  }
}
```

Erreurs possibles :

Code	Raison	Réponse
401	Utilisateur non connecté (token invalide ou absent et force non utilisé)	{"message":"Vous n'avez pas les droits pour supprimer une bière","code":401}
404	La bière n'existe plus dans la base	{"message":"Suppression : La bière 'Brooklyn Ama Bruna' n'existe plus dans la base de données","code":409}
409	Aucune raison...	{"message":"Impossible de supprimer 'Brooklyn Ama Bruna' dans la base de données","code":409}

Last update: 2019/08/31 14:40 slam4:richclient:angularjs:project:openbeerdatabase:rest http://slamwiki2.kobject.net/slam4/richclient/angularjs/project/openbeerdatabase/rest?rev=1427223413

From:
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:
<http://slamwiki2.kobject.net/slam4/richclient/angularjs/project/openbeerdatabase/rest?rev=1427223413>

Last update: **2019/08/31 14:40**

