

Spring REST Docs for Tirage API

API

This is an example output for a service running at <https://tirages.herokuapp.com/tirage>

1. Create tirage

This endpoint allows a person or organisation to initialize a draw by providing the list of participant. It also let him choose whom can administration the draw

CreateTirage Request

```
POST /create-tirage HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 180
Host: localhost:8080

{"users":[{"email":"pmekeze@yahoo.fr","secureCode":"PAT","admin":true}, {"email":"pmekeze@hotmail.fr","secureCode":"PAT","admin":true}], "company":"TEST", "notificationEnabled":false}
```

CreateTirage Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 199

{"userResponses":[{"email":"pmekeze@yahoo.fr","secureCode":"PAT","notificationSend":false}, {"email":"pmekeze@hotmail.fr","secureCode":"PAT","notificationSend":false}], "company":"TEST", "message":null}
```

Table 1. Description of createTirage response content

Path	Type	Description
<code>company</code>	<code>String</code>	The company create by admin to represente this tirage
<code>userResponses</code>	<code>Array</code>	The company create by admin to represente this tirage
<code>userResponses[0].email</code>	<code>String</code>	The company create by admin to represente this tirage
<code>userResponses[0].secureCode</code>	<code>String</code>	The company create by admin to represente this tirage
<code>userResponses[0].notificationSend</code>	<code>Boolean</code>	The company create by admin to represente this tirage

Path	Type	Description
message	Null	The company create by admin to represente this tirage

2. Perform tirage

When Tirage is create, the endpoint help any participant to performe this tirage. The email and the secure code are mandatories You can contact admin the get your secureCode if you don't have it

performTirage Request

```
POST /perform-tirage HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 68
Host: localhost:8080

{"email":"pmekeze@yahoo.fr","secureCode":"TEST","company":"COMPANY"}
```

performTirage Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 64

{"email":"pmekeze@yahoo.fr","company":"COMPANY","orderNumber":1}
```

3. Verify company

Call this endpoint to check if a company with the given name already exist in database system

verifycompany Request

```
GET /verifycompany/company HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:8080
```

verifycompany Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 23

{"companyExisted":true}
```

Table 2. /verifycompany/{company}

Parameter	Description
company	unique identifier for company

4. Get results

getResults Request

```
POST /results HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 82
Host: localhost:8080

{"email":"admin@lagrangien.fr","secureCode":"securecodeAdmin","company":"COMPANY"}
```

getResults Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 179

{"tirageResponseList":[{"email":"test@lagrangien.fr","company":"COMPANY","orderNumber":2}, {"email":"test2@lagrangien.fr","company":"COMPANY","orderNumber":1}], "company":"COMPANY"}
```

5. Get user result

getResultUser Request

```
POST /result HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 70
Host: localhost:8080

{"email":"test@lagrangien.fr","secureCode":"TEST","company":"COMPANY"}
```

getResultUser Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 66

{"email":"test@lagrangien.fr","company":"COMPANY","orderNumber":2}
```

6. Get users

getUsers Request

```
POST /getusers HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 82
Host: localhost:8080

{"email":"admin@lagrangien.fr","secureCode":"securecodeAdmin","company":"COMPANY"}
```

getUsers Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 188

{"userResponseList":[{"email":"test@lagrangien.fr","secureCode":"TEST","admin":true},{
"email":"test2@lagrangien.fr","secureCode":"TEST2","admin":false}], "company":"COMPANY"
,"message":null}
```

7. Get companies

getCompanies Request

```
GET /companies HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:8080
```

getCompanies Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 21

["TIRAGE1","TIRAGE2"]
```

8. Notify user

notifyUser Request

```
POST /notifyuser HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 188
Host: localhost:8080

{"adminUser":{"email":"admin@lagrangien.fr","secureCode":"securecodeAdmin","admin":true},
"userToNotify":{"email":"test@lagrangien.fr","secureCode":"TEST","admin":true}, "company":"COMPANY"}
```

notifyUser Response

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 74

```
{"email":"test@lagrangien.fr","company":"COMPANY","notificationSend":true}
```