# Restaurant Manager API documentation

## Content

# 1. Endpoints

## 1.1. Any User

### 1.1.1. Restaurant

*GET* `/rest/any/restaurants` *Get all enabled with day enabled menu*

*Table 1. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `[]` | `Array` | Restaurants |
| `[].id` | `Number` | Restaurant id |
| `[].name` | `String` | Restaurant name |
| `[].address` | `String` | Restaurant address |
| `[].phone` | `String` | Restaurant phone |
| `[].votesQuantity` | `Number` | Restaurant votes |
| `[].enabled` | `Boolean` | Restaurant activity marker |
| `[].menus` | `Array` | Restaurant menus |
| `[].menus[].id` | `Number` | Menu id |
| `[].menus[].name` | `String` | Menu name |
| `[].menus[].menuDate` | `String` | Menu date |
| `[].menus[].enabled` | `Boolean` | Menu activity marker |
| `[].menus[].menuItems` | `Array` | Menu dishes |
| `[].menus[].menuItems[].id` | `Number` | Dish id |
| `[].menus[].menuItems[].name` | `String` | Dish name |
| `[].menus[].menuItems[].price` | `Number` | Dish price |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/any/restaurants/' -i -X GET
```

*HTTP Request*

```
GET /rest/any/restaurants/ HTTP/1.1
Host: localhost:8080
```

*HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 512

[ {
  "id" : 0,
  "name" : "rest1",
  "address" : "address1",
  "phone" : "+7 (911) 111-1111",
  "enabled" : true,
  "votesQuantity" : 4,
  "menus" : [ {
    "id" : 1,
    "name" : "menu2",
    "menuDate" : "2020-01-27",
    "enabled" : true,
    "menuItems" : [ {
      "id" : 3,
      "name" : "menItm4",
      "price" : 103
    }, {
      "id" : 4,
      "name" : "menItm5",
      "price" : 104
    }, {
      "id" : 5,
      "name" : "menItm6",
      "price" : 105
    } ]
  } ]
} ]
```

### *GET* `/rest/any/restaurants/filter` *Get filtered enabled with day enabled menu*

ℹ Keep in mind:

- Empty request parameters = getAll.

*Table 2. Request parameters*

| Parameter | Optional | Description |
|---|---|---|
| name | false | Restaurant name |
| address | false | Restaurant address |

*Table 3. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `[]` | `Array` | Restaurants |
| `[].id` | `Number` | Restaurant id |
| `[].name` | `String` | Restaurant name |
| `[].address` | `String` | Restaurant address |
| `[].phone` | `String` | Restaurant phone |
| `[].votesQuantity` | `Number` | Restaurant votes |
| `[].enabled` | `Boolean` | Restaurant activity marker |
| `[].menus` | `Array` | Restaurant menus |
| `[].menus[].id` | `Number` | Menu id |
| `[].menus[].name` | `String` | Menu name |
| `[].menus[].menuDate` | `String` | Menu date |
| `[].menus[].enabled` | `Boolean` | Menu activity marker |
| `[].menus[].menuItems` | `Array` | Menu dishes |
| `[].menus[].menuItems[].id` | `Number` | Dish id |
| `[].menus[].menuItems[].name` | `String` | Dish name |
| `[].menus[].menuItems[].price` | `Number` | Dish price |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/any/restaurants/filter/?name=rest1&address=address1' -i -X GET
```

*HTTP Request*

```
GET /rest/any/restaurants/filter/?name=rest1&address=address1 HTTP/1.1
Host: localhost:8080
```

*HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 512

[ {
  "id" : 0,
  "name" : "rest1",
  "address" : "address1",
  "phone" : "+7 (911) 111-1111",
  "enabled" : true,
  "votesQuantity" : 4,
  "menus" : [ {
    "id" : 1,
    "name" : "menu2",
    "menuDate" : "2020-01-27",
    "enabled" : true,
    "menuItems" : [ {
      "id" : 3,
      "name" : "menItm4",
      "price" : 103
    }, {
      "id" : 4,
      "name" : "menItm5",
      "price" : 104
    }, {
      "id" : 5,
      "name" : "menItm6",
      "price" : 105
    } ]
  } ]
} ]
```

### *GET* `/rest/any/restaurants/{id}` *Get one with day enabled menu*

*Table 4. /rest/any/restaurants/{restId}*

| Parameter | Description |
|---|---|
| `restId` | Restaurant id |

*Table 5. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `id` | `Number` | Restaurant id |
| `name` | `String` | Restaurant name |
| `address` | `String` | Restaurant address |
| `phone` | `String` | Restaurant phone |
| `votesQuantity` | `Number` | Restaurant votes |
| `enabled` | `Boolean` | Restaurant activity marker |
| `menus` | `Array` | Restaurant menus |
| `menus[].id` | `Number` | Menu id |
| `menus[].name` | `String` | Menu name |
| `menus[].menuDate` | `String` | Menu date |
| `menus[].enabled` | `Boolean` | Menu activity marker |
| `menus[].menuItems` | `Array` | Menu dishes |
| `menus[].menuItems[].id` | `Number` | Dish id |
| `menus[].menuItems[].name` | `String` | Dish name |
| `menus[].menuItems[].price` | `Number` | Dish price |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/any/restaurants/0' -i -X GET
```

*HTTP Request*

```
GET /rest/any/restaurants/0 HTTP/1.1
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 508

{
  "id" : 0,
  "name" : "rest1",
  "address" : "address1",
  "phone" : "+7 (911) 111-1111",
  "enabled" : true,
  "votesQuantity" : 4,
  "menus" : [ {
    "id" : 1,
    "name" : "menu2",
    "menuDate" : "2020-01-27",
    "enabled" : true,
    "menuItems" : [ {
      "id" : 3,
      "name" : "menItm4",
      "price" : 103
    }, {
      "id" : 4,
      "name" : "menItm5",
      "price" : 104
    }, {
      "id" : 5,
      "name" : "menItm6",
      "price" : 105
    } ]
  } ]
}
```

# 1.2. Authenticated User

### 1.2.1. Profile

*GET `/rest/profile` Get*

*Table 6. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `id` | `Number` | User id |
| `name` | `String` | User name |
| `email` | `String` | User email |
| `registered` | `String` | User registration date/time |
| `enabled` | `Boolean` | User activity marker |
| `roles.[]` | `Array` | User roles |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/profile' -i -u 'user1@yandex.ru:password1'
-X GET
```

*HTTP Request*

```
GET /rest/profile HTTP/1.1
Authorization: Basic dXNlcjFAeWFuZGV4LnJ1OnBhc3N3b3JkMQ==
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 156

{
  "id" : 0,
  "name" : "User1",
  "email" : "user1@yandex.ru",
  "registered" : "2020-12-20T00:00:00",
  "enabled" : true,
  "roles" : [ "USER" ]
}
```

Error HTTP Responses

*Get by unauthorized user*

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm="Realm"
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

*POST* `/rest/profile/register` *Register*

*Table 7. Request fields*

| Path | Type | Optional | Description | Constraints |
|---|---|---|---|---|
| name | String | false | User name | [Must not be blank, Size must be between 2 and 100 inclusive] |
| email | String | false | User email | [Must be a well-formed email address, Must not be blank, Size must be between 0 and 100 inclusive] |
| password | String | false | User password | [Must not be blank, Size must be between 5 and 32 inclusive] |

*Table 8. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `id` | `Number` | User id |
| `name` | `String` | User name |
| `email` | `String` | User email |
| `registered` | `String` | User registration date/time |
| `enabled` | `Boolean` | User activity marker |
| `roles.[]` | `Array` | User roles |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/profile/register' -i -X POST \
    -H 'Content-Type: application/json;charset=UTF-8' \
    -d '{
  "name" : "newName",
  "email" : "newemail@ya.ru",
  "password" : "newPassword"
}'
```

*HTTP Request*

```
POST /rest/profile/register HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 88
Host: localhost:8080

{
  "name" : "newName",
  "email" : "newemail@ya.ru",
  "password" : "newPassword"
}
```

*Success HTTP Response*

```
HTTP/1.1 201 Created
Location: http://localhost:8080/rest/profile
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 161

{
  "id" : 4,
  "name" : "newName",
  "email" : "newemail@ya.ru",
  "registered" : "2020-12-20T18:28:18.453",
  "enabled" : true,
  "roles" : [ "USER" ]
}
```

*PUT `/rest/profile` Update*

*Table 9. Request fields*

| **Path** | **Type** | **Optional** | **Description** | **Constraints** |
|---|---|---|---|---|
| name | String | false | User name | [Must not be blank, Size must be between 2 and 100 inclusive] |
| email | String | false | User email | [Must be a well-formed email address, Must not be blank, Size must be between 0 and 100 inclusive] |
| password | String | false | User password | [Must not be blank, Size must be between 5 and 32 inclusive] |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/profile' -i -u 'user1@yandex.ru:password1'
-X PUT \
    -H 'Content-Type: application/json;charset=UTF-8' \
    -d '{
  "name" : "newName",
  "email" : "newemail@ya.ru",
  "password" : "newPassword"
}'
```

*HTTP Request*

```
PUT /rest/profile HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 88
Authorization: Basic dXNlcjFAeWFuZGV4LnJ1OnBhc3N3b3JkMQ==
Host: localhost:8080

{
  "name" : "newName",
  "email" : "newemail@ya.ru",
  "password" : "newPassword"
}
```

*Success HTTP Response*

```
HTTP/1.1 204 No Content
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

*DELETE* `/rest/profile` *Delete*

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/profile' -i -u 'user1@yandex.ru:password1'
-X DELETE
```

*HTTP Request*

```
DELETE /rest/profile HTTP/1.1
Authorization: Basic dXNlcjFAeWFuZGV4LnJ1OnBhc3N3b3JkMQ==
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 204 No Content
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

### 1.2.2. Vote

### *GET* `/rest/profile/votes/{restId}` *Vote for restaurant*

If user votes again the same day:

- If it is before 11:00 we assume that he changed his mind.
- If it is after 11:00 then it is too late, vote can't be changed.

*Table 10. /rest/profile/votes/{restId}*

| **Parameter** | **Description** |
|---|---|
| `restId` | Restaurant id |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/profile/votes/0' -i -u
'user1@yandex.ru:password1' -X PATCH
```

*HTTP Request*

```
PATCH /rest/profile/votes/0 HTTP/1.1
Authorization: Basic dXNlcjFAeWFuZGV4LnJ1OnBhc3N3b3JkMQ==
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 204 No Content
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

Error HTTP Responses

*Vote unauthorized*

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm="Realm"
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

## 1.3. Admin

### 1.3.1. Profile

*GET* `/rest/admin/users` *Get all*

*Table 11. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `[]` | `Array` | Users |
| `[].id` | `Number` | User id |
| `[].name` | `String` | User name |
| `[].email` | `String` | User email |
| `[].registered` | `String` | User registration date/time |
| `[].enabled` | `Boolean` | User activity marker |
| `[].roles.[]` | `Array` | User roles |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/users/' -i -u
'admin@gmail.com:admin' -X GET
```

*HTTP Request*

```
GET /rest/admin/users/ HTTP/1.1
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 485

[ {
  "id" : 2,
  "name" : "Admin",
  "email" : "admin@gmail.com",
  "registered" : "2020-12-20T00:00:00",
  "enabled" : true,
  "roles" : [ "ADMIN", "USER" ]
}, {
  "id" : 0,
  "name" : "User1",
  "email" : "user1@yandex.ru",
  "registered" : "2020-12-20T00:00:00",
  "enabled" : true,
  "roles" : [ "USER" ]
}, {
  "id" : 1,
  "name" : "User2",
  "email" : "user2@yandex.ru",
  "registered" : "2020-12-20T00:00:00",
  "enabled" : true,
  "roles" : [ "USER" ]
} ]
```

*GET `/rest/admin/users/{id}` Get one*

*Table 12. /rest/admin/users/{id}*

| Parameter | Description |
|---|---|
| `id` | User id |

*Table 13. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `id` | `Number` | User id |
| `name` | `String` | User name |
| `email` | `String` | User email |
| `registered` | `String` | User registration date/time |
| `enabled` | `Boolean` | User activity marker |
| `roles.[]` | `Array` | User roles |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/users/2' -i -u
'admin@gmail.com:admin' -X GET
```

*HTTP Request*

```
GET /rest/admin/users/2 HTTP/1.1
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 165

{
  "id" : 2,
  "name" : "Admin",
  "email" : "admin@gmail.com",
  "registered" : "2020-12-20T00:00:00",
  "enabled" : true,
  "roles" : [ "ADMIN", "USER" ]
}
```

Error HTTP Responses

*Get not found*

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 191

{
  "url" : "http://localhost:8080/rest/admin/users/10",
  "type" : "DATA_NOT_FOUND",
  "detail" : "edu.volkov.restmanager.util.exception.NotFoundException: Not
found entity with id=10"
}
```

*Get by unauthorized user*

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm="Realm"
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

*Get by forbidden user*

```
HTTP/1.1 403 Forbidden
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

### *GET* `/rest/admin/users/by` *Get by email*

*Table 14. Request parameters*

| Parameter | Optional | Description |
|---|---|---|
| email | false | User email |

*Table 15. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `id` | `Number` | User id |
| `name` | `String` | User name |
| `email` | `String` | User email |
| `registered` | `String` | User registration date/time |
| `enabled` | `Boolean` | User activity marker |
| `roles.[]` | `Array` | User roles |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/users/by?email=admin@gmail.com' -i
-u 'admin@gmail.com:admin' -X GET
```

*HTTP Request*

```
GET /rest/admin/users/by?email=admin@gmail.com HTTP/1.1
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 165

{
  "id" : 2,
  "name" : "Admin",
  "email" : "admin@gmail.com",
  "registered" : "2020-12-20T00:00:00",
  "enabled" : true,
  "roles" : [ "ADMIN", "USER" ]
}
```

Error HTTP Responses

*POST* `/rest/admin/users` *Create*

*Table 16. Request fields*

| Path | Type | Optional | Description | Constraints |
|---|---|---|---|---|
| name | String | false | User name | [Must not be blank, Size must be between 2 and 100 inclusive] |
| email | String | false | User email | [Must be a well-formed email address, Size must be between 0 and 100 inclusive] |
| registered | String | false | User registration date/time | [Must not be null] |
| password | String | false | User registration date/time | [Size must be between 5 and 100 inclusive] |
| enabled | Boolean | false | User activity marker | [] |
| roles.[] | Array | false | User roles | [] |

*Table 17. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `id` | `Number` | User id |
| `name` | `String` | User name |
| `email` | `String` | User email |
| `registered` | `String` | User registration date/time |
| `enabled` | `Boolean` | User activity marker |
| `roles.[]` | `Array` | User roles |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/users/' -i -u
'admin@gmail.com:admin' -X POST \
    -H 'Content-Type: application/json;charset=UTF-8' \
    -d '{
  "name" : "New",
  "email" : "new@gmail.com",
  "registered" : "2020-12-20T19:29:15.781",
  "enabled" : false,
  "roles" : [ "USER" ],
  "password" : "newPass"
}'
```

*HTTP Request*

```
POST /rest/admin/users/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 171
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080

{
  "name" : "New",
  "email" : "new@gmail.com",
  "registered" : "2020-12-20T19:29:15.781",
  "enabled" : false,
  "roles" : [ "USER" ],
  "password" : "newPass"
}
```

*Success HTTP Response*

```
HTTP/1.1 201 Created
Location: http://localhost:8080/rest/admin/users/3
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 157

{
  "id" : 3,
  "name" : "New",
  "email" : "new@gmail.com",
  "registered" : "2020-12-20T19:29:15.793",
  "enabled" : false,
  "roles" : [ "USER" ]
}
```

*PUT* `/rest/admin/users` *Update*

*Table 18. Request fields*

| **Path** | **Type** | **Optional** | **Description** | **Constraints** |
|---|---|---|---|---|
| id | Number | false | User id | [Must not be blank, Size must be between 2 and 100 inclusive] |
| name | String | false | User name | [Must not be blank, Size must be between 2 and 100 inclusive] |
| email | String | false | User email | [Must be a well-formed email address, Size must be between 0 and 100 inclusive] |
| registered | String | false | User registration date/time | [Must not be null] |
| enabled | Boolean | false | User activity marker | [] |
| roles.[] | Array | false | User roles | [] |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/users/' -i -u
'admin@gmail.com:admin' -X PUT \
    -H 'Content-Type: application/json;charset=UTF-8' \
    -d '{
  "id" : 0,
  "name" : "UpdatedName",
  "email" : "update@gmail.com",
  "registered" : "2020-12-20T19:29:15.153",
  "enabled" : false,
  "roles" : [ "ADMIN" ]
}'
```

*HTTP Request*

```
PUT /rest/admin/users/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 169
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080

{
  "id" : 0,
  "name" : "UpdatedName",
  "email" : "update@gmail.com",
  "registered" : "2020-12-20T19:29:15.153",
  "enabled" : false,
  "roles" : [ "ADMIN" ]
}
```

*Success HTTP Response*

```
HTTP/1.1 204 No Content
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

*PATCH* `/rest/admin/users/{id}` *Set enable*

*Table 19. /rest/admin/users/{id}*

| Parameter | Description |
|---|---|
| `id` | User id |

*Table 20. Request parameters*

| Parameter | Optional | Description |
|---|---|---|
| enabled | false | User activity marker |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/users/0' -i -u
'admin@gmail.com:admin' -X PATCH \
    -H 'Content-Type: application/json;charset=UTF-8'
```

*HTTP Request*

```
PATCH /rest/admin/users/0 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 204 No Content
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

Error HTTP Responses

*Enable not found*

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 191

{
  "url" : "http://localhost:8080/rest/admin/users/10",
  "type" : "DATA_NOT_FOUND",
  "detail" : "edu.volkov.restmanager.util.exception.NotFoundException: Not
found entity with id=10"
}
```

*DELETE* `/rest/admin/users/{id}` *Delete*

*Table 21. /rest/admin/users/{id}*

| Parameter | Description |
|---|---|
| `id` | User id |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/users/0' -i -u
'admin@gmail.com:admin' -X DELETE
```

*HTTP Request*

```
DELETE /rest/admin/users/0 HTTP/1.1
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 204 No Content
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

Error HTTP Responses

*Delete not found*

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 191

{
  "url" : "http://localhost:8080/rest/admin/users/10",
  "type" : "DATA_NOT_FOUND",
  "detail" : "edu.volkov.restmanager.util.exception.NotFoundException: Not
found entity with id=10"
}
```

### 1.3.2. Restaurant

*GET* `/rest/admin/restaurants` *Get all with day enabled menu*

*GET* `/rest/admin/restaurants/{id}` *Get one without menu*

*GET* `/rest/admin/restaurants/filter` *Get filtered with day enabled menu*

*POST* `/rest/admin/restaurants` *Create*

*PUT* `/rest/admin/restaurants/{id}` *Update*

*PATCH* `/rest/admin/restaurants/{id}` *Set enable*

*DELETE* `/rest/admin/restaurants/{id}` *Delete*

### 1.3.3. Menu

*GET* `/rest/admin/menus/{restId}` *Get all for one restaurant*

*Table 22. /rest/admin/menus/{restId}*

| Parameter | Description |
|---|---|
| `restId` | Restaurant id |

*Table 23. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `[]` | `Array` | Menus |
| `[].id` | `Number` | Menu id |
| `[].name` | `String` | Menu name |
| `[].menuDate` | `String` | Menu date |
| `[].enabled` | `Boolean` | Menu activity marker |
| `[].menuItems` | `Array` | Menu dishes |
| `[].menuItems[].id` | `Number` | Dish id |
| `[].menuItems[].name` | `String` | Dish name |
| `[].menuItems[].price` | `Number` | Dish price |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/menus/0' -i -u
'admin@gmail.com:admin' -X GET
```

*HTTP Request*

```
GET /rest/admin/menus/0 HTTP/1.1
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 939

[ {
  "id" : 2,
  "name" : "menu3",
  "menuDate" : "2020-01-28",
  "enabled" : true,
  "menuItems" : [ {
    "id" : 6,
    "name" : "menItm7",
    "price" : 106
  }, {
    "id" : 7,
    "name" : "menItm8",
    "price" : 107
  }, {
    "id" : 8,
    "name" : "menItm9",
    "price" : 108
  } ]
}, {
  "id" : 0,
  "name" : "menu1",
  "menuDate" : "2020-01-27",
  "enabled" : false,
  "menuItems" : [ {
    "id" : 0,
    "name" : "menItm1",
    "price" : 100
  }, {
    "id" : 1,
    "name" : "menItm2",
    "price" : 101
  }, {
    "id" : 2,
    "name" : "menItm3",
    "price" : 102
  } ]
}, {
  "id" : 1,
  "name" : "menu2",
  "menuDate" : "2020-01-27",
  "enabled" : true,
  "menuItems" : [ {
    "id" : 3,
    "name" : "menItm4",
    "price" : 103
  }, {
    "id" : 4,
    "name" : "menItm5",
    "price" : 104
  }, {
    "id" : 5,
    "name" : "menItm6",
    "price" : 105
  } ]
} ]
```

*GET* `/rest/admin/menus/filter/{restId}` *Get filtered for one restaurant*

Keep in mind:

- Filtering by dates includes left and right borders.
- Empty request parameters = getAll.

*Table 24. /rest/admin/menus/filter/{restId}*

| Parameter | Description |
|---|---|
| `restId` | Restaurant id |

*Table 25. Request parameters*

| Parameter | Optional | Description |
|---|---|---|
| startDate | false | Start date |
| endDate | false | End date |
| enabled | false | Menu activity marker |

*Table 26. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `[]` | `Array` | Menus |
| `[].id` | `Number` | Menu id |
| `[].name` | `String` | Menu name |
| `[].menuDate` | `String` | Menu date |
| `[].enabled` | `Boolean` | Menu activity marker |
| `[].menuItems` | `Array` | Menu dishes |
| `[].menuItems[].id` | `Number` | Dish id |
| `[].menuItems[].name` | `String` | Dish name |
| `[].menuItems[].price` | `Number` | Dish price |

*examples*

*Curl*

```
$ curl
'http://localhost:8080/rest/admin/menus/filter/0?startDate=&endDate=&enabled=
&startDate=&endDate=&enabled=' -i -u 'admin@gmail.com:admin' -X GET
```

*HTTP Request*

```
GET
/rest/admin/menus/filter/0?startDate=&endDate=&enabled=&startDate=&endDate=&e
nabled= HTTP/1.1
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 939

[ {
  "id" : 2,
  "name" : "menu3",
  "menuDate" : "2020-01-28",
  "enabled" : true,
  "menuItems" : [ {
    "id" : 6,
    "name" : "menItm7",
    "price" : 106
  }, {
    "id" : 7,
    "name" : "menItm8",
    "price" : 107
  }, {
    "id" : 8,
    "name" : "menItm9",
    "price" : 108
  } ]
}, {
  "id" : 0,
  "name" : "menu1",
  "menuDate" : "2020-01-27",
  "enabled" : false,
  "menuItems" : [ {
    "id" : 0,
    "name" : "menItm1",
    "price" : 100
  }, {
    "id" : 1,
    "name" : "menItm2",
    "price" : 101
  }, {
    "id" : 2,
    "name" : "menItm3",
    "price" : 102
  } ]
}, {
  "id" : 1,
  "name" : "menu2",
  "menuDate" : "2020-01-27",
  "enabled" : true,
  "menuItems" : [ {
    "id" : 3,
    "name" : "menItm4",
    "price" : 103
  }, {
    "id" : 4,
    "name" : "menItm5",
    "price" : 104
  }, {
    "id" : 5,
    "name" : "menItm6",
    "price" : 105
  } ]
} ]
```

*GET `/rest/admin/menus/{restId}/{id}` Get one*

*Table 27. /rest/admin/menus/{restId}/{id}*

| Parameter | Description |
|---|---|
| `restId` | Restaurant id |
| `id` | Menu id |

*Table 28. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `id` | `Number` | Menu id |
| `name` | `String` | Menu name |
| `menuDate` | `String` | Menu date |
| `enabled` | `Boolean` | Menu activity marker |
| `menuItems` | `Array` | Menu dishes |
| `menuItems[].id` | `Number` | Dish id |
| `menuItems[].name` | `String` | Dish name |
| `menuItems[].price` | `Number` | Dish price |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/menus/0/0' -i -u
'admin@gmail.com:admin' -X GET
```

*HTTP Request*

```
GET /rest/admin/menus/0/0 HTTP/1.1
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 311

{
  "id" : 0,
  "name" : "menu1",
  "menuDate" : "2020-01-27",
  "enabled" : false,
  "menuItems" : [ {
    "id" : 0,
    "name" : "menItm1",
    "price" : 100
  }, {
    "id" : 1,
    "name" : "menItm2",
    "price" : 101
  }, {
    "id" : 2,
    "name" : "menItm3",
    "price" : 102
  } ]
}
```

Error HTTP Responses

*Get not found*

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 230

{
  "url" : "http://localhost:8080/rest/admin/menus/0/15",
  "type" : "DATA_NOT_FOUND",
  "detail" : "edu.volkov.restmanager.util.exception.NotFoundException: Not
found entity with menu by id: 15 for restId: 0 dos not exist"
}
```

*Get by unauthorized user*

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm="Realm"
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

*Get by forbidden user*

```
HTTP/1.1 403 Forbidden
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

*POST* `/rest/admin/menus/{restId}` *Create*

*Table 29. /rest/admin/menus/{restId}*

| Parameter | Description |
|---|---|
| `restId` | Restaurant id |

*Table 30. Request fields*

| Path | Type | Optional | Description | Constraints |
|---|---|---|---|---|
| name | String | false | Menu name | [Must not be blank, Size must be between 2 and 100 inclusive] |
| menuDate | String | false | Menu date | [Must not be null] |
| enabled | Boolean | false | Menu activity marker | [Must not be null] |
| menuItems | Array | false | Menu dishes | [Must not be null] |
| menuItems[].id | Number | true | Dish id | [] |
| menuItems[].name | String | true | Dish name | [Must not be blank, Size must be between 2 and 100 inclusive] |
| menuItems[].price | Number | true | Dish price | [Must be at least 0, Must not be null] |

*Table 31. Success response body parameters*

| Path | Type | Description |
|---|---|---|
| `id` | `Number` | Menu id |
| `name` | `String` | Menu name |
| `menuDate` | `String` | Menu date |
| `enabled` | `Boolean` | Menu activity marker |
| `menuItems` | `Array` | Menu dishes |
| `menuItems[].id` | `Number` | Dish id |
| `menuItems[].name` | `String` | Dish name |
| `menuItems[].price` | `Number` | Dish price |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/menus/0' -i -u
'admin@gmail.com:admin' -X POST \
    -H 'Content-Type: application/json;charset=UTF-8' \
    -d '{
  "name" : "newMenu",
  "menuDate" : "2020-01-28",
  "enabled" : false,
  "menuItems" : [ {
    "id" : 0,
    "name" : "menItm1",
    "price" : 100
  }, {
    "id" : 1,
    "name" : "menItm2",
    "price" : 101
  }, {
    "id" : 2,
    "name" : "menItm3",
    "price" : 102
  } ]
}'
```

*HTTP Request*

```
POST /rest/admin/menus/0 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 300
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080

{
  "name" : "newMenu",
  "menuDate" : "2020-01-28",
  "enabled" : false,
  "menuItems" : [ {
    "id" : 0,
    "name" : "menItm1",
    "price" : 100
  }, {
    "id" : 1,
    "name" : "menItm2",
    "price" : 101
  }, {
    "id" : 2,
    "name" : "menItm3",
    "price" : 102
  } ]
}
```

*Success HTTP Response*

```
HTTP/1.1 201 Created
Location: http://localhost:8080/rest/admin/menus/6
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 313

{
  "id" : 6,
  "name" : "newMenu",
  "menuDate" : "2020-01-28",
  "enabled" : false,
  "menuItems" : [ {
    "id" : 0,
    "name" : "menItm1",
    "price" : 100
  }, {
    "id" : 1,
    "name" : "menItm2",
    "price" : 101
  }, {
    "id" : 2,
    "name" : "menItm3",
    "price" : 102
  } ]
}
```

*PUT* `/rest/admin/menus/{restId}` *Update*

*Table 32. /rest/admin/menus/{restId}/{id}*

| Parameter | Description |
|---|---|
| `restId` | Restaurant id |
| `id` | Menu id |

*Table 33. Request fields*

| Path | Type | Optional | Description | Constraints |
|---|---|---|---|---|
| id | Number | false | Menu id | [Must not be blank, Size must be between 2 and 100 inclusive] |
| name | String | false | Menu name | [Must not be blank, Size must be between 2 and 100 inclusive] |
| menuDate | String | false | Menu date | [Must not be null] |
| enabled | Boolean | false | Menu activity marker | [Must not be null] |
| menuItemTos | Array | false | Menu dishes | [Must not be null] |
| menuItemTos[].id | Number | true | Dish id | [] |
| menuItemTos[].name | String | true | Dish name | [Must not be blank, Size must be between 2 and 100 inclusive] |
| menuItemTos[].price | Number | true | Dish price | [Must be at least 0, Must not be null] |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/menus/0/0' -i -u
'admin@gmail.com:admin' -X PUT \
    -H 'Content-Type: application/json;charset=UTF-8' \
    -d '{
  "id" : 0,
  "name" : "updatedName",
  "menuDate" : "2020-01-28",
  "enabled" : false,
  "menuItemTos" : [ {
    "id" : 18,
    "name" : "updatedMenuItem1",
    "price" : 7777
  }, {
    "id" : 19,
    "name" : "updatedMenuItem2",
    "price" : 8888
  }, {
    "id" : 20,
    "name" : "updatedMenuItem3",
    "price" : 9999
  } ]
}'
```

*HTTP Request*

```
PUT /rest/admin/menus/0/0 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 352
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080

{
  "id" : 0,
  "name" : "updatedName",
  "menuDate" : "2020-01-28",
  "enabled" : false,
  "menuItemTos" : [ {
    "id" : 18,
    "name" : "updatedMenuItem1",
    "price" : 7777
  }, {
    "id" : 19,
    "name" : "updatedMenuItem2",
    "price" : 8888
  }, {
    "id" : 20,
    "name" : "updatedMenuItem3",
    "price" : 9999
  } ]
}
```

*Success HTTP Response*

```
HTTP/1.1 204 No Content
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

*PATCH* `/rest/admin/menus/{restId}/{id}` *Set enable*

*Table 34. /rest/admin/menus/{restId}/{id}*

| Parameter | Description |
|---|---|
| `restId` | Restaurant id |
| `id` | Menu id |

*Table 35. Request parameters*

| Parameter | Optional | Description |
|---|---|---|
| enabled | false | Menu activity marker |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/menus/0/0' -i -u
'admin@gmail.com:admin' -X PATCH \
    -H 'Content-Type: application/json;charset=UTF-8'
```

*HTTP Request*

```
PATCH /rest/admin/menus/0/0 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 204 No Content
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

Error HTTP Responses

*Enable not found*

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 230

{
  "url" : "http://localhost:8080/rest/admin/menus/0/15",
  "type" : "DATA_NOT_FOUND",
  "detail" : "edu.volkov.restmanager.util.exception.NotFoundException: Not
found entity with menu by id: 15 for restId: 0 dos not exist"
}
```

*DELETE* `/rest/admin/menus/{restId}/{id}` *Delete*

*Table 36. /rest/admin/menus/{restId}/{id}*

| Parameter | Description |
|---|---|
| `restId` | Restaurant id |
| `id` | Menu id |

*examples*

*Curl*

```
$ curl 'http://localhost:8080/rest/admin/menus/0/0' -i -u
'admin@gmail.com:admin' -X DELETE
```

*HTTP Request*

```
DELETE /rest/admin/menus/0/0 HTTP/1.1
Authorization: Basic YWRtaW5AZ21haWwuY29tOmFkbWlu
Host: localhost:8080
```

*Success HTTP Response*

```
HTTP/1.1 204 No Content
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

Error HTTP Responses

*Delete not found*

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json;charset=UTF-8
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 192

{
  "url" : "http://localhost:8080/rest/admin/menus/0/15",
  "type" : "DATA_NOT_FOUND",
  "detail" : "edu.volkov.restmanager.util.exception.NotFoundException: Not
found entity with id=0"
}
```

# 2. Convention

## 2.1. HTTP verbs

Restaurant Manager tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP verbs.

| Verb | Usage |
|---|---|
| `GET` | Used to retrieve a resource |
| `POST` | Used to create a new resource |
| `PATCH` | Used to update an existing resource, including partial updates |
| `PUT` | Used to update an existing resource, full updates only |
| `DELETE` | Used to delete an existing resource |

## 2.2. HTTP status codes

Restaurant Manager tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP status codes.

| **Status code** | **Usage** |
|---|---|
| `200 OK` | Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request, the response will contain an entity describing or containing the result of the action. |
| `201 Created` | The request has been fulfilled and resulted in a new resource being created. |
| `204 No Content` | The server successfully processed the request, but is not returning any content. |

| Status code | Usage |
|---|---|
| `400 Bad Request` | The server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing). |
| `404 Not Found` | The requested resource could not be found but may be available again in the future. Subsequent requests by the client are permissible. |
| `409 Conflict` | The request could not be completed due to a conflict with the current state of the target resource. |
| `422 Unprocessable Entity` | Validation error has happened due to processing the posted entity. |

# 3. Error processing

*Table 37. Error response body parameters*

| Path | Type | Description |
|---|---|---|
| `url` | `String` | Request url |
| `type` | `String` | Error type |
| `detail` | `String` | Error details |