

Documentation REST Backend

Sandro Schweizer, Emil von Wattenwyl

www.footprintsonthemoon.net

https://bitbucket.org/schweizers/rest_backend/src

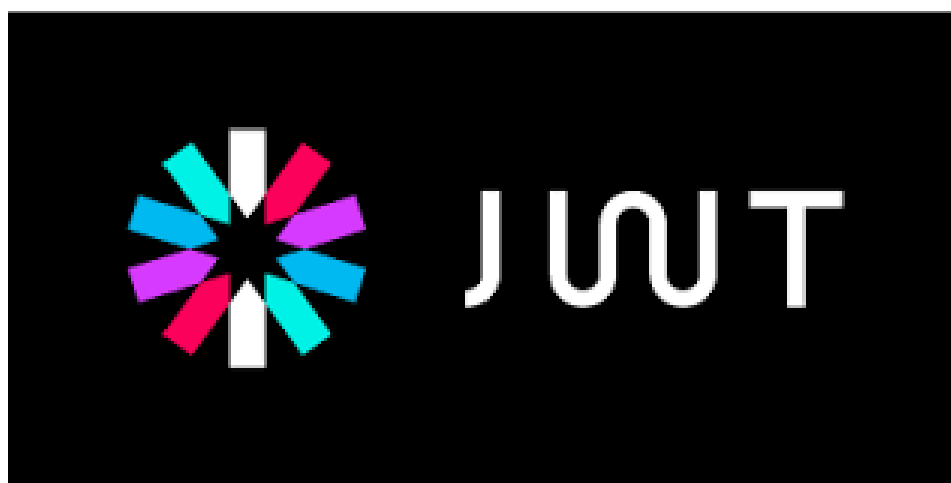


Table of Contents

| | |
|---------------------------------|----------|
| Versioning..... | 2 |
| Routes..... | 3 |
| Middleware | 3 |
| REST Backend | 4 |
| / | 4 |
| rest /login..... | 4 |
| Request:..... | 4 |
| Response: | 4 |
| rest /register..... | 5 |
| Request:..... | 5 |
| Response: | 5 |
| rest/account/create..... | 5 |
| Request:..... | 5 |
| Response: | 6 |

| | |
|--------------------------------------|----------|
| rest/account/delete/{id}..... | 6 |
| Request:..... | 6 |
| Response: | 6 |
| rest/account/others | 6 |
| Request:..... | 6 |
| Response: | 6 |
| rest/account/overview..... | 7 |
| Request:..... | 7 |
| Response: | 7 |
| rest/payment..... | 8 |
| Request:..... | 8 |
| Response: | 8 |
| rest/logout..... | 8 |
| Request:..... | 8 |
| Response: | 8 |

Versioning

| Version | Description | Author |
|--------------|-------------|------------------|
| 1.0.0 | | Sandro Schweizer |
| | | |
| | | |

Routes

| Method | URI | Action | Middleware |
|------------|--------------------------|--|---------------|
| GET HEAD | / | Closure | web |
| POST | rest /login | App\Http\Controllers\Rest\RestController@login | web |
| POST | rest /register | App\Http\Controllers\Rest\RestController@register | web |
| POST | rest/account/create | App\Http\Controllers\Rest\RestController@accountCreate | web, jwt.auth |
| DELETE | rest/account/delete/{id} | App\Http\Controllers\Rest\RestController@accountDelete | web, jwt.auth |
| GET HEAD | rest/account/others | App\Http\Controllers\Rest\RestController@accountOthers | web, jwt.auth |
| GET HEAD | rest/account/overview | App\Http\Controllers\Rest\RestController@accountOverview | web, jwt.auth |
| POST | rest /payment | App\Http\Controllers\Rest\RestController@payment | web, jwt.auth |
| GET HEAD | rest /logout | App\Http\Controllers\Rest\RestController@logout | web, jwt.auth |

Middleware

JWT-Auth for Laravel has an included Middleware:

GetUserFromToken

JWT-Auth includes some sensible default responses when, for example, the token has expired or is invalid. These responses can be overridden, by hooking into a series of events that are fired before the response is returned. To use the middlewares you will have to register them in `app/Http/Kernel.php` under the `$routeMiddleware` property (already done). More information can be found in the official wiki: <https://github.com/tymondesigns/jwt-auth/wiki>

Events:

```
// fired when the token could not be found in the request
Event::listen('tymon.jwt.absent');

// fired when the token has expired
Event::listen('tymon.jwt.expired');

// fired when the token is found to be invalid
Event::listen('tymon.jwt.invalid');

// fired if the user could not be found (shouldn't really happen)
Event::listen('tymon.jwt.user_not_found');

// fired when the token is valid (User is passed along with event)
Event::listen('tymon.jwt.valid');
```

REST Backend

/

This Closure returns the welcome page.

rest /login

Request:

Body:

```
email: ""
password: ""
```

Response:

```
{
  "token": "",
  "user": {
    "id": ,
    "name": "",
    "email": "",
    "account": [
      {
        "id": ,
        "name": "",
        "user_id": ,
        "id_number": "",
        "creditBalance": "",

```

```

    "payment": [
      {
        "id": ,
        "message": "",
        "amount": "",
        "from": ,
        "to": ,
        "created_at": "",
        "updated_at": ""
      }
    ]
  }
}

```

rest /register

Request:

Body:

```

email: ""
password: ""
name: ""

```

Response:

```

{
  "token": "",
  "user": {
    "id": ,
    "name": "",
    "email": "",
    "account": []
  }
}

```

rest/account/create

Request:

Body:

```

name: ""

```

Response:

```
{
  "user": {
    "id": ,
    "name": "",
    "email": "",
    "account": [
      {
        "id": ,
        "name": "",
        "user_id": ,
        "id_number": "",
        "creditBalance": "",
        "payment": []
      }
    ]
  }
}
```

rest/account/delete/{id}

Request:

-> GET

Response:

```
{
  "msg": "successful deleted"
}
```

rest/account/others

Request:

-> GET

Response:

```
{
  "account": [
```

```

{
  "id": ,
  "name": "",
  "user_id": ,
  "id_number": "",
  "creditBalance": "",
  "user": {
    "id": ,
    "name": "",
    "email": ""
  }
}

```

rest/account/overview

Request:

-> GET

Response:

```

{
  "account": [
    {
      "id": ,
      "name": "",
      "user_id": ,
      "id_number": "",
      "creditBalance": "",
      "user": {
        "id": ,
        "name": "",
        "email": ""
      }
    }
  ]
}

```

rest/payment

Request:

Body:

```
to: ""
from: ""
amount:
message: ""
```

Response:

```
{
  "from": {
    "id": ,
    "name": "",
    "user_id": ,
    "id_number": "",
    "creditBalance":
  },
  "to": {
    "id": ,
    "name": "",
    "user_id": ,
    "id_number": "",
    "creditBalance":
  }
}
```

rest/logout

Request:

-> GET

Response:

```
{
  "msg": "successfully logged out"
}
```