



PAAS APPLICATION

فهرست

4	مقدمه
5	فرایندها
7	درخواست ها و پاسخ ها
7	• ثبت نام
8	• لاگین
9	• افزودن سایت به حساب کاربری
10	• دریافت رمز دوم
11	• ارسال لینک بازیابی کاربر
12	• اطلاعات کاربر
13	• حذف اکانت
14	• افزودن ایمیل
15	• تایید ایمیل
16	• نمایش QRcode (برای تغییر دیوایس)
17	• دریافت QRcode (برای تغییر دیوایس)
18	• دریافت لینک تغییر شماره
19	• تغییر شماره
20	• فعالسازی اکانت
21	• ریکاوری اکانت
22	• ثبت نام سایت
23	• لاگین سایت ها
24	• اضافه کردن کاربر توسط سایت
25	• تایید رمز دوم
26	• ارسال لینک بازیابی سایت
27	• افزودن کاربران قبلی
28	• نمایش کد فعالسازی کاربران سایت ها
29	• ارسال لینک حذف از راه دور اکانت
30	• حذف اکانت از راه دور
31	صفحات
31	• صفحه اصلی سایت

32	• صفحه داشبورد کاربر
33	• ثبت نام کاربران
33	• ثبت نام سایت
33	• بازیابی و تغییر رمز
34	• صفحه فراموشی رمز

هدف از نرم افزار پاس، افزایش امنیت و جلوگیری از نفوذ به حساب های کاربری افراد در سایت های مختلف است. با کمک ایده های امنیتی به کار رفته، امکان نفوذ در حساب های کاربری سایت هایی که از این نرم افزار استفاده می کنند تقریباً به صفر می رسد.

شیوه عملکرد این نرم افزار مشابه برنامه های رمز دوم پویا بانک ها برای انتقال های کارت به کارت است اما تفاوت های مهمی نیز دارد که باعث بیشتر شدن امنیت و کارایی این نرم افزار شده است.

زمان استفاده از این نرم افزار هنگام لاگین در سایت های مختلف است که بعد از وارد کردن یوزرنیم و پسورد به جای کامل شدن فرایند ورود، وارد صفحه ای می شود که درخواست یک رمز دوم شش رقمی می کند که باید از داخل حساب اپلیکیشن پاس کاربر دریافت شود و بعد از وارد کردن آن رمز، آن را به سرور نرم افزار پاس می فرستد و در صورت تایید، فرایند لاگین کردن کامل می شود. سایت هایی که از این نرم افزار استفاده می کنند، برای ورود کاربرانشان حتی می توانند از پسورد های عادی استفاده نکنند.

هر کاربر می تواند هر تعداد سایت که از این قابلیت استفاده می کنند را به اپلیکیشن خود بیافزاید و برای لاگین به هر کدام از آن سایت ها از طریق اپلیکیشن یک رمز دوم بگیرد. بدیهی است که رمز هر کدام از سایت های افزوده شده به اپلیکیشن کاربر با هم متفاوت است و همچنین در حالت عادی هیچ رمزی برای هیچ کدام از سایت های افزوده شده وجود ندارد و کاربر باید برای سایتی که قصد ورود به آن را دارد درخواست رمز دوم دهد. سپس رمز یکبار مصرف مخصوص آن سایت با مدت اعتبار یک دقیقه برایش فعال می شود. دلیل این قابلیت این است که برای دریافت رمز باید ابتدا درخواست توسط کاربر ثبت شود تا افراد دیگر با وارد کردن رمز های اتفاقی شانس ورود به حساب کاربری را نداشته باشند.

همچنین حساب کاربری پاس هر شخص فقط در یک دستگاه و از طریق یک شماره قابل ورود است و امکان اینکه همزمان در دو دستگاه وارد شود وجود ندارد. حتی امکان اینکه کاربر حسابش را از یکی از دستگاه ها خارج کند و در دستگاه دیگری وارد کند وجود ندارد و همیشه تنها از طریق یک دستگاه خاص امکان وارد شدن به حساب پاس وجود دارد. تنها راه تغییر دستگاه وارد شدن به حساب پاس این است که از طریق دستگاه دوم، QRcode نمایش داده شده توسط دستگاه اول برای تغییر دیوایس اسکن شود سپس حساب از دستگاه اول حذف شده و به دستگاه دوم منتقل می شود.

یکی دیگر از قابلیت های پاس امکان حذف از راه دور اکانت از طریق ایمیل است. در صورت ثبت ایمیل توسط کاربر، این امکان وجود دارد تا در صورت گم شدن دستگاهی که حساب پاس بر رویش نصب شده، از راه دور حساب را حذف کرد تا کاربر از طریق ساختن یک حساب دیگر و افزودن سایت ها به حساب جدید بتواند به آن سایت ها وارد شود.

تمامی کد های ذخیره شده در پایگاه داده پاس، رمز گذاری شده توسط الگوریتم hashing و غیر قابل دیکود کردن هستند و حتی در صورت لو رفتن اطلاعات دیتابیس، امکان دسترسی به رمز های کاربران وجود ندارد.

فرایند ها

1- ثبت نام:

از طریق 2 فیلد (شماره موبایل و رمز) اکانت ثبت می شود اما غیر فعال است. یک پیامک حاوی لینک فعالسازی به کاربر ارسال می شود و بعد از تایید لینک، اکانت فعال می شود.

برای سایت ها هم همین فرایند توسط 3 فیلد (آدرس-شماره- رمز) اتفاق می افتد.

2- لاگین:

از کاربر 2 فیلد (شماره و پسورد) گرفته می شود و یک فیلد (دیوایس آیدی) هم نرم افزار از دستگاه کاربر بدست می آورد و ارسال می کند.

برای سایت ها تنها 2 فیلد (آدرس سایت و پسورد) گرفته می شود و ارسال می شود.

در صورتی که کاربر یا سایت 3 بار رمز ورود را اشتباه وارد کند اکانت بن می شود و باید ریکاوری شود.

3- افزودن کاربران توسط سایت:

سایت نام کاربری فردی که تازه در سایتش ثبت نام کرده را می فرستد. نامش ثبت می شود و برای آن کاربر رمز افزودن سایت به نرم افزار ایجاد می شود.

4- افزودن کاربران قبلی

یک آرایه شامل نام های کاربری کاربرانش ارسال می کند. برای هر کدام یک رمز مخصوص ساخته می شود و هر وقت که سایت درخواست دهد، آن رمز برای افزودن سایت به اپلیکیشن کاربر برمی گردد.

5- افزودن سایت به نرم افزار کاربر

کاربر این کار را به 2 روش می تواند انجام دهد. 1- اسکن QRcode 2- وارد کردن دستی مقادیر

بعد از اینکه سایت، کاربر را در پاس ثبت کند، یک سری اطلاعات مربوط به کاربر به آن سایت برمی گردد که باید تبدیل به QRcode شده و به کاربر نشان داده شود. همچنین مقادیر اصلی آن اطلاعات هم به کاربر نمایش دهد تا بتواند به صورت دستی سایت را ثبت کند.

کاربر یا QRcode را اسکن کند یا آن مقادیر (آدرس سایت-یوزرنیم-رمز دریافت شده 6 رقمی) را وارد کند. در صورت صحیح بودن اطلاعات، سایت افزوده می شود.

6- دریافت رمز دوم

در صورتی که سایت به نرم افزار کاربر افزوده شده باشد، آدرس سایت فرستاده می شود و رمز دوم بر می گردد.

7- بازیابی اکانت

در سایت پاس درخواست بازیابی ثبت می شود و پیامک حاوی لینک ارسال می شود. با ورود به لینک، اکانت بازیابی و فعال می شود.
از طریق این فرایند 2 کار انجام می شود: 1-خارج کردن اکانت از بن 2-بازیابی رمز فراموش شده

8-تغییر شماره

کاربر در حسابش درخواست تغییر شماره می دهد و شماره جدید را ارسال می کند. یک لینک فعال سازی به شماره قبلی ارسال می شود. در صورت تایید لینک، شماره تغییر می کند.

9-تغییر دیوایس

کاربر باید نرم افزار را در دستگاه جدید نصب کند. در صفحه اول وارد بخش اسکن کد می شود. در دستگاه قدیمی وارد حساب کاربری شده و درخواست تغییر دیوایس می دهد. یک `qr code` نمایش داده می شود. آن را توسط دستگاه جدید اسکن می کند. در صورت تایید اکانت از دستگاه قبلی پاک شده و آماده لاگین با دستگاه جدید می شود.

10-افزودن ایمیل

کاربر ایمیل را وارد می کند و یک لینک به آن ایمیل ارسال می شود. در صورت تایید لینک، ایمیل فعال می شود.

11-دلیت اکانت از راه دور

تنها کاربرانی که ایمیلشان را فعال کرده باشند می توانند از این استفاده کنند. کاربران در سایت در بخش حذف از راه دور ایمیل خود را وارد می کنند. اگر ایمیل ثبت شده بود لینک حذف اکانت برایش ایمیل می شود. در صورت تایید لینک، اکانت کاربر حذف می شود.

12-دلیت اکانت

کاربر از طریق اپلیکیشن درخواست حذف می دهد.

13-دریافت اطلاعات حساب

یک درخواست از اپلیکیشن کاربر ارسال می شود. شماره و لیست سایت های افزوده شده کاربر پاسخ داده می شود.

درخواست ها و پاسخ ها:

ثبت نام

آدرس: localhost:4000/signup

متد: post

فیلد های ورودی:

- Number
- DeviceId
- password

نکته: DeviceId توسط برنامه به صورت خودکار بدست می آید و کاربر وارد نمی کند

پاسخ ها:

- msg: پیام پاسخ
- link: لینک فعالسازی (باید پیامک شود که موقتا در گزارش آمده است، اگر درخواست انجام نشده باشد null است)
- success: اگر درخواست انجام شود true وگرنه false است
- error: ارور
- status: (برای پاسخ صحیح 200)

The screenshot shows a REST client interface with a POST request to `localhost:4000/signup`. The request body is a JSON object with the following fields:

```
{  "Number": "09166366715",  "password": "12345678",  "DeviceId": "randomDeviceId"}
```

The response status is 200 OK. The response body is a JSON object with the following fields:

```
{  "msg": "user created",  "link": "localhost:4000/active/users/09166366715/slubdm8qfi2wqw4mfzgf49ib8n2say6qygc43gk9a3evbb8cvkwwso15hzvu",  "success": true,  "error": [],  "status": 200}
```

لاگین

آدرس: localhost:4000/login

متد: post

فیلد های ورودی:

- Number
- password
- DeviceId

پاسخ ها:

- msg : پیام گزارش
- token : در صورت وجود
- error : ارور
- success : true یا false
- status : برای پاسخ صحیح 200
- link : لینک فعالسازی برای وقتی که اکانت بن شده باشد

The screenshot shows a REST client interface with a POST request to localhost:4000/login. The request body is a JSON object with fields: Number, password, and DeviceId. The response is a 200 OK status with a JSON body containing login details like msg, token, error, success, status, and link.

POST localhost:4000/login

Send Save

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

raw JSON Bear

```
1 {
2   "Number": "09166366715",
3   "password": "11223344",
4   "DeviceId": "randomDeviceId"
5 }
```

Body 200 OK 308 ms 514 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "login_Successful!",
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0dW1iZXIiOiIwOTE2NjM2Njc4NSIsIm1hdCI6MTYyNjAwODYxNSwiZXhwIjoxNjMDEyMjE1fQ.ayto5kfNGDjuIL_iIYnFOFs-1meD8msSRvmtYY_WFoY",
4   "error": [],
5   "success": true,
6   "status": 200,
7   "link": []
8 }
```


اضافه کردن سایت به اپلیکیشن

آدرس: localhost:4000/AddSiteToDb

متد: post

نیاز به توکن دارد

فیلد های ورودی:

- Address : آدرس سایت مورد نظر
- username : یوزرنیم کاربر در آن سایت
- code : کد 7 رقمی نمایش داده شده توسط آن سایت

پاسخ ها:

- msg
- error
- success
- status

The screenshot shows a REST client interface with a POST request to localhost:4000/AddSiteToDb. The request body is a JSON object with fields Address, username, and code. The response body is a JSON object with fields msg, error, success, and status.

POST localhost:4000/AddSiteToDb

Params Authorization Headers (9) Body

● none ● form-data ● x-www-form-urlencoded

```
1 {
2   "Address": "www.google.com",
3   "username": "user1",
4   "code": "o4cyeg8"
5 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JS

```
1 {
2   "msg": "Added",
3   "error": [],
4   "success": true,
5   "status": 200
6 }
```

دریافت رمز دوم

آدرس: localhost:4000/getcode

متد: get

نیاز به توکن دارد

فیلد های ورودی:

- Address : آدرس سایت مورد نظر

پاسخ ها:

- msg
- code : رمز دوم
- error
- success
- status

GET localhost:4000/getcode

Params Authorization Headers (9) **Body**

● none ● form-data ● x-www-form-urlencoded

```
1 {  
2   "Address": "www.google.com"  
3 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "msg": "Done",  
3   "code": "798751",  
4   "error": [],  
5   "success": true,  
6   "status": 200  
7 }
```

ارسال لینک بازیابی کاربر

دو جا کاربرد دارد:

1. فراموشی و بازیابی رمز
2. بازیابی اکانت بن شده

آدرس: localhost:4000/sendrecoverylink

متد: get

فیلد های ورودی:

• Number

پاسخ ها:

- msg
- error
- link
- success
- status

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:4000/sendrecoverylink
- Body:** JSON (raw view)

```
{
  "Number": "09166366715"
}
```
- Response:** 200 OK, 300 ms, 4 bytes
- Response Body (Pretty view):**

```
{
  "msg": "link sent",
  "error": [],
  "link": "localhost:4000/recovery/users/09166366715/fkzgj8nt2f3t03tk5dhwj2qgi3i6q402vudhoca6yz15014zewtd05c5nj",
  "success": true,
  "status": 200
}
```

اطلاعات کاربر

آدرس: localhost:4000/getuserinfo

متد: get

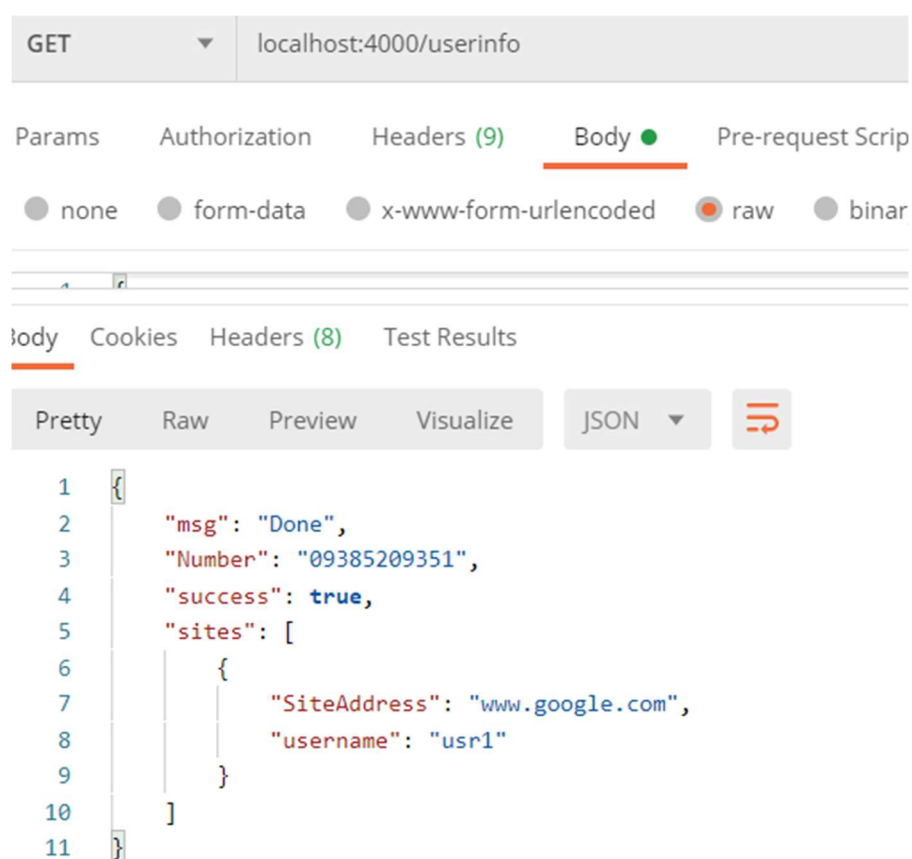
نیاز به توکن دارد

پاسخ ها:

- msg
- Number
- success
- sites

1. SiteAddress : آدرس سایت

2. نام کاربری کاربر در آن سایت



حذف اکانت

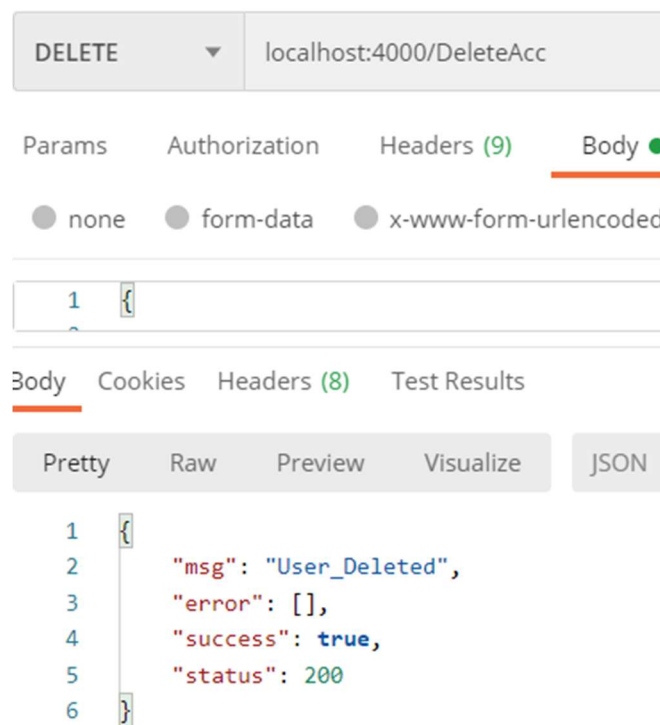
آدرس: localhost:4000/DeleteAcc

نیاز به توکن دارد

پاسخ ها:

- msg
- error
- success
- status

متد: delete



افزودن ایمیل

آدرس: localhost:4000/AddEmail

نیاز به توکن دارد

متد: post

فیلد های ورودی:

Email •

پاسخ ها:

- msg
- error
- link (موقتی)
- success
- status

POST localhost:4000/AddEmail

Params Authorization Headers (9) Body Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▼

```
1 {
2   "Email": "mohammad7979salehi@gmail.com"
3 }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 322 r

Pretty Raw Preview Visualize JSON ▼

```
1 {
2   "msg": "Email_Added",
3   "error": [],
4   "link": "localhost:4000/ActiveEmail/09166366715/m7u339tzjhmk2xqzni4cqwaote4sch7dwidgmop8cygqi42obmmcbud98fbk",
5   "success": true,
6   "status": 200
7 }
```

تایید ایمیل

آدرس: localhost:4000/ActiveEmail/(mobile Number)/(code)

متد: get

پاسخ ها:

- msg
- error
- success
- status

GET

localhost:4000/ActiveEmail/09166366715/m7u339tzjhmk2xqzni4cqwaote4sch7dwidgmop8cygqi42obmmcbud98fbk

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 354

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "msg": "Email_is_Active",
3   "error": [],
4   "success": true,
5   "status": 200
6 }
```

نمایش QRcode (برای تغییر دیوایس)

آدرس: localhost:4000/showQR

نیاز به توکن دارد

متد: get

پاسخ ها:

- msg
- Number
- code
- error
- success
- status

The screenshot shows a REST client interface with a GET request to `localhost:4000/showQR`. The request body is empty. The response is displayed in JSON format, showing a successful status (200) and various fields including a message, a number, a code, and success/error flags.

```
GET localhost:4000/showQR
```

Params Authorization Headers (9) **Body** Pre-request Script Tests Set

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● GraphQL

```
1 {
2 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "Done",
3   "Number": "09166366715",
4   "code": "xy8kt3i0r4jkecwen185x1wl4siwo4smlrvd86sd2t9a8j0y2emnyiul7b73",
5   "error": [],
6   "success": true,
7   "status": 200
8 }
```


دریافت QRcode (برای تغییر دیوایس)

آدرس: localhost:4000/getQR

متد: post

فیلد های ورودی:

- Number •
- code •
- DeviceId •

پاسخ ها:

- msg •
- error •
- success •
- status •

The screenshot displays a REST client interface. At the top, a POST request is configured for the URL `localhost:4000/getQR`. The 'Body' tab is selected, showing a JSON payload with three fields: `Number` (09166366715), `code` (a long alphanumeric string), and `DeviceId` (A21). Below the request, the 'Test Results' tab is active, showing the response in 'Pretty' JSON format. The response contains `msg` (changed), `error` (empty array), `success` (true), and `status` (200).

```
POST localhost:4000/getQR
```

Params Authorization Headers (8) **Body** Pre-request Script Tests Se

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● GraphQL

```
1 {
2   "Number": "09166366715",
3   "code": "i12xcpdkmdw271kli7i1tmi2sj0tqc6scmzb0qlt77pfiqmlsmko8nongfjt",
4   "DeviceId": "A21"
5 }
```

Body **Cookies** Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "msg": "changed",
3   "error": [],
4   "success": true,
5   "status": 200
6 }
```

دریافت لینک تغییر شماره

آدرس: localhost:4000/newNumber

متد: get

نیاز به توکن دارد

فیلد های ورودی:

newNumber •

پاسخ ها:

- msg
- error
- link (موقتی)
- success
- status

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:4000/newNumber
- Body:** A JSON object with one property: `"newNumber": "09121234567"`.
- Response:** Status 200 OK, Time: 302 ms, Size: 448 B. The response body is a JSON object: `{ "msg": "link sent", "error": [], "link": "localhost:4000/ChangeNumber/09166366715/09121234567/kxt17eszwx81ydd6nbjoznkfpdxj9i575qstffnxegydmowhx15sirxya03", "success": true, "status": 200 }`.

تغییر شماره

آدرس: localhost:4000/(Account Number)/(new Number)/(code)

متد: get

پاسخ ها:

- msg
- error
- success
- status

GET localhost:4000/ChangeNumber/09166366715/09121234567/oeudj3l1zonyeavd1vcixwv5r8yq6jv4dfwi520a2wpo4gqa3...

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "Number changed",
3   "error": [],
4   "success": true,
5   "status": 200
6 }
```

فعالسازی اکانت

با کلیک روی لینکی که بعد از ثبت نام به شماره موبایل ارسال می شود فعالسازی انجام می شود

آدرس: localhost:4000/active/((mode (user or site)))/(ID (user number or site Address))/(code)

متد: get

پاسخ ها:

- msg
- error
- success
- status

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** localhost:4000/active/users/09166366715/z83nc
- Params:** Query Params (empty table)
- Headers:** 6 headers (not visible)
- Body:** Empty
- Test Results:** Pretty view of the response

Response Body (JSON):

```
1 {
2   "msg": "activated",
3   "error": [],
4   "success": true,
5   "status": 200
6 }
```

ریکاوری اکانت

در دو جا کاربرد دارد:

1. بازیابی اکانت بن شده
2. بازیابی رمز فراموش شده

آدرس: `localhost:4000/recovery/(mode (user or site))/(ID (user number or site Address))/(code)`

متد: `get`

فیلد های ورودی:

- `DeviceId`
- `newPassword` (فقط برای کاربران)

پاسخ ها:

- `msg`
- `error`
- `success`
- `status`

The image shows two side-by-side screenshots of a web browser's developer tools, specifically the Network tab. Both screenshots show a GET request to a localhost:4000 endpoint.

Left Screenshot:

- URL: `localhost:4000/recovery/sites/www.google.com/teutpdr`
- Body (raw):

```
{  "newPassword": "11223344"}
```
- Response (pretty):

```
{  "msg": "changed",  "error": [],  "success": true,  "status": 200}
```

Right Screenshot:

- URL: `localhost:4000/recovery/users/09166366715/ag8hqtxfc`
- Body (raw):

```
{  "newPassword": "11223344",  "DeviceId": "randomDeviceId"}
```
- Response (pretty):

```
{  "msg": "changed",  "error": [],  "success": true,  "status": 200}
```

ثبت نام سایت

آدرس: localhost:4000/siteregistration

متد: post

فیلد های ورودی:

- Address
- Number
- password

پاسخ ها:

- msg
- link (موقتی)
- error
- success
- status

The screenshot displays a REST client interface with a POST request to `localhost:4000/siteregistration`. The request body is a JSON object with the following fields:

```
{
  "Address": "www.google.com",
  "Number": "09385209351",
  "password": "12345678"
}
```

The response status is `200 OK` with a time of `610 ms`. The response body is a JSON object with the following fields:

```
{
  "msg": "site created",
  "link": "localhost:4000/active/sites/www.google.com/dp78xbh8bfng27oa6vgt6cq8mf4fx7js8216evmwc93tcf3oxtu94zs5du",
  "error": [],
  "success": true,
  "status": 200
}
```

لاگین سایت ها

آدرس: localhost:4000/loginForSites

متد: post

فیلد های ورودی:

- Address
- password

پاسخ ها:

- msg
- token
- error
- success
- status
- link لینک بازایی برای هنگامی که اکانت بن شده باشد

POST localhost:4000/siteLogin

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● GraphQL **JSON**

```
1 {
2   "Address": "www.google.com",
3   "password": "11223344"
4 }
```

ody Cookies Headers (8) Test Results ⊕ Status: 200 OK Time: 369 ms Size: 5

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "msg": "login_Successful!",
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb2RyZXNzIjoid3d3Lmdvb2dsZS5jb20iLCJodW1iZXIiOiIwOTM4NTIwOTM1MSIsIm1hdCI6MTYyNjAwOTcxNiwiZXhwIjoxNjU3NTY3MzE2fQ. EN4RwqtSm_35eR0qgw9-5tG6csyK6dvW1mfY2fX0usg",
4   "error": null,
5   "success": true,
6   "status": 200,
7   "link": []
8 }
```

اضافه کردن کاربر توسط سایت

متد: get

نیاز به توکن دارد

آدرس: localhost:4000/AddUserToSiteDb

فیلد های ورودی:

• username

پاسخ ها:

• msg

• username

• code

• error

• success

• status

GET localhost:4000/AddUserToSiteDb

Params Authorization Headers (9) **Body**

● none ● form-data ● x-www-form-urlencoded

```
1 {  
2   "username": "user1"  
3 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "msg": "user_Added",  
3   "username": "user1",  
4   "code": "o4cyeg8",  
5   "error": [],  
6   "success": true,  
7   "status": 200  
8 }
```


تایید رمز دوم

آدرس: localhost:4000/confirm

متد: post

نیاز به توکن دارد

فیلد های ورودی:

- username
- code

پاسخ ها:

- msg
- username (یوزرنیم کاربری که پاسخش آمده)
- authentication (آیا رمز درست بوده؟)
- error
- success
- status

POST

localhost:4000/confirm

Params

Authorization

Headers (9)

☐ none

☐ form-data

☐ x-www-form-ur

1

{

2

"username": "user1",

3

"code": "335900"

4

}

body

Cookies

Headers (8)

Test Results

Pretty

Raw

Preview

Visualize

1

{

2

"msg": "login auth",

3

"username": "user1",

4

"authentication": true,

5

"error": [],

6

"success": true,

7

"status": 200

8

}

ارسال لینک بازیابی سایت

کاربرد هنگام:

1. فراموشی رمز

2. بن شدن

متد: get

آدرس: localhost:4000/sendrecoverylinkforsites

فیلد های ورودی:

• Address

پاسخ ها:

• msg

• link (موقتی)

• error

• success

• status

The screenshot shows a REST client interface. At the top, the method is set to GET and the URL is localhost:4000/sendrecoverylinkforsites. Below this, the 'Body' tab is selected, showing a raw JSON request: { "Address": "www.google.com" }. At the bottom, the 'Body' section shows the response in 'Pretty' format: { "msg": "link sent", "link": "localhost:4000/recovery/sites/www.google.com/zf3j8lw0casmeboh5f3mw16pixa8ej25if5i39cp", "error": null, "success": true, "status": 200 }. The status bar at the bottom right indicates a 200 OK response.

```
GET localhost:4000/sendrecoverylinkforsites
```

Params Auth Headers (8) **Body** Pre-req. Tests Settings

raw JSON

```
1 {
2   "Address": "www.google.com"
3 }
```

Body 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "link sent",
3   "link": "localhost:4000/recovery/sites/www.google.com/
4         zf3j8lw0casmeboh5f3mw16pixa8ej25if5i39cp",
5   "error": null,
6   "success": true,
7   "status": 200
8 }
```

افزودن کاربران قبلی

نام کاربری کاربر ها اضافه می شود اما کاربر ها باید با گرفتن کد فعالسازی، سایت را به اپلیکیشن شان بیافزایند

آدرس: localhost:4000/Addprevioususers

متد: post

نیاز به توکن دارد

فیلد های ورودی

- users (آرایه ای از نام کاربری کاربر های قبلی)

پاسخ ها:

- msg
- error
- success
- status

The screenshot shows a REST client interface. At the top, the method is set to POST and the URL is localhost:4000/Addprevioususers. The 'Body' tab is selected, showing a JSON request: { "users": ["user1", "user2", "user3"] }. Below this, the 'Body' tab of the response is selected, showing a JSON response: { "msg": "Added", "error": [], "success": true, "status": 200 }. The response is formatted in a 'Pretty' view.

```
POST localhost:4000/Addprevioususers
```

Params Authorization Headers (9) **Body** Pre-re

● none ● form-data ● x-www-form-urlencoded ● raw

```
1 {
2   "users": ["user1", "user2", "user3"]
3 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "Added",
3   "error": [],
4   "success": true,
5   "status": 200
6 }
```

نمایش کد فعالسازی کاربران سایت ها

متد: get

نیاز به توکن دارد

آدرس: localhost:4000/getusercode

فیلد های ورودی:

- username (یوزرنیم کاربری که قصد دریافت کد فعالسازی دارد)

پاسخ ها:

- msg
- username
- code
- error
- success
- status

GET localhost:4000/getusercode

Params Authorization Headers (9) Body ●

Headers 8 hidden

KEY
<input checked="" type="checkbox"/> Authorization
Key

body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "not_Added",
3   "username": "user2",
4   "code": "t4wwte",
5   "error": [],
6   "success": true,
7   "status": 200
8 }
```

ارسال لینک حذف از راه دور اکانت

متد: get

آدرس: localhost:4000/sendDeleteLink

فیلد های ورودی:

Email •

پاسخ ها:

- msg
- link (موقتی)
- error
- success
- status

The screenshot shows a REST client interface with a GET request to `localhost:4000/sendDeleteLink`. The request body is a JSON object: `{ "Email": "mohammad@gmail.com" }`. The response is also in JSON format, showing a successful status (200 OK) and a message "link sent". The response body is displayed in a "Pretty" JSON view.

```
GET localhost:4000/sendDeleteLink
```

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼

```
1 {
2   "Email": "mohammad@gmail.com"
3 }
```

body Cookies Headers (8) Test Results 🌐 Status: 200 OK Time: 303 ms Siz

Pretty Raw Preview Visualize **JSON** ▼

```
1 {
2   "msg": "link sent",
3   "link": "localhost:4000/DelByLink/mohammad@gmail.com/pem14i36j1h4v1pyjb1qvnsjiam852d5nj3xkvm9isr2pyb5gojwd5x7nhr",
4   "error": [],
5   "success": true,
6   "status": 200
7 }
```

حذف اکانت از راه دور

متد: get

آدرس: localhost:4000/DelByLink/(Email)/(code)

پاسخ ها:

- msg
- error
- success
- status

GET

localhost:4000/DelByLink/mohammad7979salehi@gmail.com/lfue0bqz9pio1l285haxytl392umdzu4kk5zkuug26gkpw6 ...

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
-----	-------	-------------

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 335 ms

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "msg": "User_Deleted",
3   "error": [],
4   "success": true,
5   "status": 200
6 }
```

*** ورژن اصلی پاس دارای اپلیکیشن است که امکانات کامل را در اختیار دارد اما در ورژن موقت، تنها سایت وجود دارد و طبیعتاً بعضی قابلیت های پاس مانند منحصر به فرد بودن دیوایس و فرایند های مربوط به آن در این ورژن قابل اجرا شدن نیست و این ورژن موقتا برای نمایش طرز کار پاس ساخته**

شده است *

صفحات:

1-صفحه اصلی سایت

صفحه اصلی

لینک به صفحه ثبت نام سایت
لینک به صفحه ثبت نام کاربر
فیلد های شماره و رمز برای لاگین کاربر یا سایت
دکمه تعیین نوع کاربر یا سایت برای لاگین
اگر سایت لاگین کرد گزینه افزودن کاربران قبلی اضافه می شود
اگر سایت لاگین کرد گزینه خروج اضافه شود
لینک به صفحه دلیت اکانت از طریق ایمیل
لینک به صفحه بازیابی رمز عبور

در صفحه اصلی سایت فیلد های لاگین کاربران به دلیل راحت تر شدن لاگین، وجود دارد.

کاربران بعد از لاگین به صفحه داشبورد منتقل می شوند اما سایت ها نیازی به صفحه داشبورد ندارند و اگر لاگین کنند تنها دو گزینه "افزودن کاربران قبلی" و "خروج" برایشان در همان صفحه اصلی فعال می شود.

داشبورد کاربر

باکس سایت ها

آدرس سایت

نام کاربری کاربر در آن سایت

رمز

دکمه دریافت رمز

شمارنده زمان رمز

افزودن ایمیل

دلالت اکانت

افزودن سایت

تغییر شماره

هر سایت که به حساب کاربری افزوده می شود یک باکس برایش ایجاد می شود که گزینه های آدرس سایت، نام کاربری کاربر، دکمه دریافت رمز، نمایش دهنده رمز و شمارنده زمان رمز وجود دارد

یک منو وجود دارد که گزینه های افزودن ایمیل، دلالت اکانت، افزودن سایت و تغییر شماره وجود دارد

وقتی کاربر بر روی گزینه افزودن ایمیل کلیک می کند، صفحه جدیدی باز **نمی شود** و فقط یک فیلد به همراه دکمه تایید باز می شود.

با کلیک بر روی دلالت اکانت، فقط یک دکمه برای تایید نمایش داده می شود.

با کلیک بر روی افزودن سایت، سه فیلد برای آدرس، یوزرنیم و کد به همراه دکمه تایید نمایش داده می شود.

با کلیک بر روی تغییر شماره یک فیلد برای شماره با دکمه تایید نمایش داده می شود.

به صورت دیفالت رمز وجود ندارد و بعد از کلیک روی دکمه دریافت رمز، رمز دریافت شده به مدت یک دقیقه نشان داده می شود

در ورژن اصلی و دارای اپلیکیشن، افزودن سایت به صورت اسکن QRcode انجام می شود و گزینه تغییر دیوایس هم وجود دارد

3-ثبت نام کاربران:

ثبت نام کاربران
فیلد دریافت شماره موبایل فیلد دریافت پسورد دکمه تایید

4-ثبت نام سایت:

ثبت نام سایت
فیلد دریافت آدرس فیلد دریافت پسورد فیلد دریافت شماره موبایل دکمه تایید

5-بازیابی و تغییر رمز:

بازیابی پسورد
فیلد دریافت پسورد جدید دکمه تایید

صفحه ریکاوری

دکمه انتخاب سایت یا کاربر
شماره موبایل
دکمه تایید

در این صفحه یک دکمه وجود دارد که دو حالت دارد و کاربر انتخاب میکند بازیابی رمز برای اکانت کاربر انجام میشود یا سایت.

بعد شماره موبایل را وارد می کند و با دکمه تایید، آن را تایید می کند.