

# Incident management

## 1- On-Call Engineer

### شرح دقیق نقش و مسئولیت‌های On-Call Engineer در مدیریت حوادث

در سامانه‌های حساس و حیاتی نظیر کارگزاری بورس آنلاین، پایداری و دسترس‌پذیری سیستم اهمیت زیادی دارد. بخشی از الزامات تضمین این پایداری، داشتن فرآیندی شفاف برای مدیریت حوادث (Incident Management) است.

در همین راستا، نقش On-Call Engineer یا «اپراتور مسئول پاسخگویی برخط» تعریف و وظایف آن به صورت زیر شفاف‌سازی می‌شود:

### تعریف On-Call Engineer

On-Call Engineer فردی از اعضای تیم پشتیبانی یا فنی است که در هر بازه زمانی مشخص، مسئولیت نظارت، شناسایی و واکنش سریع به رخدادهای غیرمنتظره، خطاها و مشکلات مهم سامانه را بر عهده دارد. این نقش به صورت چرخشی (شیفتی) میان اعضای تیم فنی توزیع می‌شود تا همواره یک شخص آماده پاسخگویی وجود داشته باشد.

### وظایف کلیدی On-Call Engineer

**پایش مستمر سامانه:** تحت نظر داشتن سلامت کلی سرویس‌ها، بررسی سامانه‌های مانیتورینگ و دریافت هشدارها (Alerts)، چه به صورت اتوماتیک و چه دستی.

**پاسخگویی سریع به رخدادها:** دریافت و تحلیل رخدادها، (اعم از خطاهای سیستمی، کندی سرویس، افزایش تاخیر یا افت کیفیت) و اقدام فوری برای کاهش تأثیر یا رفع مشکل، بر اساس رویه‌های سازمانی و سطح اهمیت خطا.

**هماهنگی با تیم توسعه و سایر مسئولین:** در صورت رفع‌نشدن مشکل، اطلاع‌رسانی به سایر اعضای تیم، مدیریت تصاعد مشکل (escalation) و همکاری برای حل ریشه‌ای (Root Cause Analysis).

**مستندسازی فعالیت‌ها:** ثبت رخدادها، پاسخ‌ها و راهکارهای اعمال شده جهت مرور، یادگیری و بهبود فرآیند در آینده.

پس از رفع حادثه، گزارشی شامل اطلاعات زیر تهیه می‌شود:

شرح حادثه

دلیل وقوع

اقدامات انجام شده

پیشنهادهای برای جلوگیری از حادثه مشابه

ابزار	نقش
Python	زبان اصلی توسعه سرویس‌ها
FastAPI	چارچوب طراحی RESTful برای پیاده‌سازی سرویس ثبت حادثه
BlockchainAdapter	ماژولی برای اتصال به بلاک‌چین و ثبت اطلاعات حادثه
Prometheus	ابزار مانیتورینگ و تشخیص مشکلات سیستم
Alertmanager	ابزار هشداردهی برای اطلاع‌رسانی سریع به مهندس آن‌کال

**گزارش‌دهی به مدیریت:** ارائه گزارش شفاف در پایان هر شیفت یا رخداد مهم، شامل لیست مشکلات، نحوه پاسخگویی و پیشنهادات لازم برای پیشگیری‌های بعدی.

### معیارهای تعیین و شیفت‌بندی

برنامه‌ریزی شیفت‌های On-Call بر اساس تعداد اعضا، مهارت فنی، حجم تراکنش‌ها و ساعات بحرانی انجام می‌شود.

لیست اعضای مسئول و زمان‌بندی شیفت‌ها باید به صورت مستند و در دسترس کل تیم باشد. در صورت بروز تغییر در تقسیم وظایف، باید فوراً به همه اعضا اطلاع‌رسانی گردد.

### اهداف ایجاد نقش

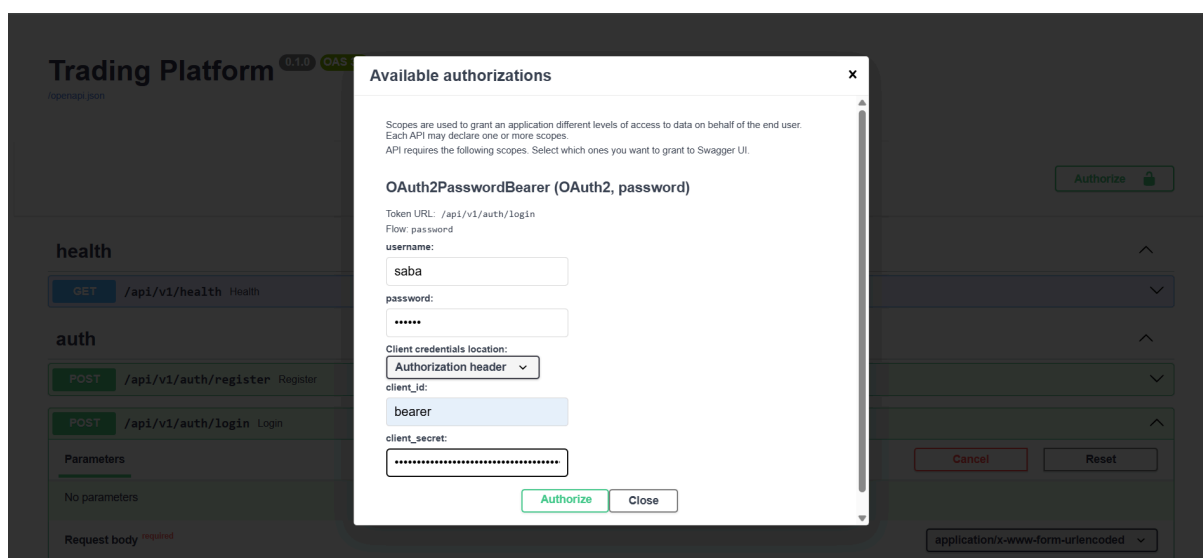
افزایش سرعت شناسایی و حل مشکلات به منظور به حداقل رساندن زمان Down بودن سرویس

تضمین ارائه سرویس پایدار و با کیفیت به کاربران نهایی  
به حداقل رساندن ریسک‌های ناشی از رخدادهای پیش‌بینی نشده و کاهش اثر آن‌ها بر کسب‌وکار و اعتبار  
سامان

تخصیص کلید owner بر blockchain

در swagger این تغییر را تست کردم:

ابتدا با یک کاربر رجیستر و لوگین میکنم و توکن داده شده را برای احراز هویت استفاده میکنم



اینجا توانستم owner قرارداد را ببینم:

Request URL		
<code>http://localhost:8000/api/v1/admin/blockchain/owner</code>		
Server response		
Code	Details	
200	<div><div>Response body</div><div><pre>{  "owner": "0x71562b71999873085b286df957af199Ec94617F7"}</pre></div><div><div>Response headers</div><div><pre>content-length: 54 content-type: application/json date: Wed, 11 Feb 2026 11:16:32 GMT server: uvicorn</pre></div></div></div>	
Responses		
Code	Description	Links

در این قسمت هم سطح دسترسی بررسی شده که درست‌ه و admin هست:

address required

string

(path)

Ox71562b71999873DB5b286dF957af199Ec5

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8000/api/v1/admin/blockchain/access-level/0x71562b71999873DB5b286dF957af199Ec94617f7' \
-H 'accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ1bWU1IiwiaWF0IjoxNjc0ODUwMzcyfQ.X4uo0UgLnSH2IDKtmIHYG_-qHTvbmAtGB-9kQ56qIQ'
```

Request URL

```
http://localhost:8000/api/v1/admin/blockchain/access-level/0x71562b71999873DB5b286dF957af199Ec94617f7
```

Server response

Code	Details
200	<div>Response body</div> <div><pre>{   "address": "0x71562b71999873DB5b286dF957af199Ec94617f7",   "level": 3,   "level_name": "ADMIN" }</pre></div> <div><div>Download</div></div>

Response headers

مثلا اگر یک ادرس بی ربط میدادیم:

address required

string

(path)

0x00

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8000/api/v1/admin/blockchain/access-level/0x0000000000000000000000000000000000000000000000000000000000000001' \
-H 'accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ1bWU1IiwiaWF0IjoxNjc0ODUwMzcyfQ.X4uo0UgLnSH2IDKtmIHYG_-qHTvbmAtGB-9kQ56qIQ'
```

Request URL

```
http://localhost:8000/api/v1/admin/blockchain/access-level/0x0000000000000000000000000000000000000000000000000000000000000001
```

Server response

Code	Details
200	<div>Response body</div> <div><pre>{   "address": "0x0001",   "level": 0,   "level_name": "NONE" }</pre></div> <div><div>Download</div></div>

Response headers

حالا قابلیت تعیین سطح دسترسی برای یک کاربر را تست میکنیم:

ExecuteClear

Responses

Curl

```
curl -X 'POST' \
'http://localhost:8000/api/v1/admin/blockchain/set-access' \
-H 'accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ1bWU1IiwiaWF0IjoxNjc0ODUwMzcyfQ.X4uo0UgLnSH2IDKtmIHYG_-qHTvbmAtGB-9kQ56qIQ' \
-d '{
  "user_address": "0x1234567890123456789012345678901234567890",
  "level": 2
}'
```

Request URL

```
http://localhost:8000/api/v1/admin/blockchain/set-access
```

Server response

Code	Details
200	<div>Response body</div> <div><pre>{   "success": true,   "tx_hash": "0xc3235d6df3c3640affc268fca7c35e7baaf124cd8aa015ca0ce13f3be2d8d41",   "message": "Access level set to 2" }</pre></div> <div><div>Download</div></div>

Response headers

```
content-length: 129
content-type: application/json
date: Wed, 11 Feb 2026 11:31:37 GMT
server: uvicorn
```

که درست کار میکند.

خلاصه ی سطوح دسترسی

سطح	نقش	توانایی ها
OWNER	مالک Contract	تخصیص/حذف همه نقش ها توقف/راه اندازی مجدد سیستم مجوزدهی به recorders حل کردن incidents انتقال مالکیت
EMERGENCY_ADMIN	مدیر اضطراری	توقف اضطراری سیستم - حل کردن incidents
OPERATOR	اپراتور	گزارش incidents
TRADER	معامله گر	ثبت trades
Authorized Recorder	ثبت کننده مجاز	ثبت trades

روش های دیگری که میشد استفاده کرد:

### 1- استفاده از الگوی کنترل مبتنی بر چند امضا (Multi-Signature Wallet):

به جای اینکه Owner یک آدرس باشد، مالکیت و کنترل به چندین آدرس واگذار می شود. اجرای عملکردهای حساس (مثل تغییر تنظیمات، آپدیت قرارداد و ...) تنها با تایید اکثریت امضاکنندگان ممکن است (مثلاً n از m).

ابزارهای مشهور مثل Gnosis Safe این روش را بسیار رایج و عملی کرده اند.

## 2- پیاده‌سازی (DAO (Decentralized Autonomous Organization

سطح دسترسی مالک به مجموعه‌ای از سهام‌داران یا اعضای سازمان غیرمتمرکز منتقل می‌شود. تصمیمات کلیدی و تغییرات با رای‌گیری (governance) انجام می‌شود (مثلاً با توکن‌های gov یا voting power).

قراردادهای معروف DAO وجود دارند که امکان ارتقای سیستم را به صورت غیرمتمرکز فراهم می‌کنند (مانند OpenZeppelin Governor).

## 3- استفاده از قراردادهای قابل ارتقا با مدیریت جمعی (Upgradeable Contracts with Collective Control):

به جای owner، یک Proxy Admin یا Governor انتخاب می‌شود که خودش تحت مدیریت گروه یا DAO است.

دسترسی ارتقا و مدیریت به مجموعه‌ای از کاربران یا نهادها تعلق می‌گیرد.

## 4- حذف کامل مالکیت (Renounce Ownership):

پس از راه‌اندازی، مالکیت به طور کامل حذف و هیچ آدرس خاصی کنترل ندارد.

توجه شود که بدون امکان مدیریت، ارتقای سیستم سخت‌تر خواهد شد.

## 5- Timelock Controller (تاخیر زمانی بر عملیات حساس):

Timelock اجرای عملیات مهم قرارداد (مثل ارتقاء یا تغییر مالکیت) را با تاخیر زمانی (مثلاً ۲ روز) همراه می‌کند.

در این مدت همه می‌توانند تغییرات قریب‌الوقوع را ببینند و در صورت لزوم واکنش نشان دهند. معمولاً همراه با multisig/DAO استفاده می‌شود و امنیت و شفافیت را افزایش می‌دهد.