# 📦 ساختار کامل پروژه میکروسرویس ردیابی GPS

1. \*\*ApiGateway\*\*

- دروازه ورودی تمام درخواست‌های خارجی به سیستم است.

- مسئول مسیریابی درخواست‌ها به سرویس‌های مناسب و مدیریت احراز هویت، لاگ‌گیری و Rate Limiting است.

- قابلیت پیاده‌سازی Load Balancing، Caching و Security Policies روی APIها را داراست.

### 2. \*\*DeviceManagementService\*\*

- مدیریت چرخه حیات دستگاه‌ها شامل ثبت (Register)، به‌روزرسانی، حذف و بازیابی اطلاعات دستگاه.

- اعتبارسنجی شماره سریال دستگاه‌ها.

- اختصاص یا تغییر مالکیت دستگاه‌ها به کاربران یا حساب‌های کاربری.

### 3. \*\*LocationTrackingService\*\*

- ردیابی موقعیت جغرافیایی دستگاه‌ها بر اساس داده‌های GPS یا سایر منابع.

- ثبت و ذخیره داده‌های مکانی برای تحلیل‌های بعدی.

- ارائه API برای دریافت آخرین موقعیت یا تاریخچه حرکتی هر دستگاه.

### 4. \*\*CommandService\*\*

- ارسال دستورات به دستگاه‌ها (مثل ریست، آپدیت Firmware، تنظیمات مجدد).

- مدیریت پاسخ‌های دریافتی از دستگاه‌ها.

- ثبت لاگ‌های مربوط به ارسال/دریافت دستورات.

### 5. \*\*NotificationService\*\*

- مدیریت ارسال اعلان‌ها به کاربران از طریق SMS، Email، Push Notification یا SignalR.

- تعریف قوانین (Rules) برای تریگر کردن اعلان‌ها بر اساس رویدادهای سیستم یا دستگاه‌ها.

- سیستم اولویت‌بندی و ارسال مجدد اعلان‌های ناموفق.

6. \*\*AuthService\*\*

- مدیریت فرآیند احراز هویت (Login/Logout) کاربران و سرویس‌ها.

- تولید و اعتبارسنجی توکن‌های JWT.

- پشتیبانی از احراز هویت چندمرحله‌ای (MFA) از طریق SMS یا اپلیکیشن‌های Authenticator.

- ثبت لاگ ورودها، تحلیل رفتارهای مشکوک و مدیریت IP کاربران.

### 7. \*\*BillingService\*\*

- محاسبه هزینه‌های مربوط به اشتراک‌ها، استفاده از سرویس‌ها یا مصرف منابع.

- مدیریت ایجاد فاکتورها، پیگیری پرداخت‌ها و سوابق مالی کاربران.

- پشتیبانی از تخفیف‌ها، مالیات و تمدید خودکار اشتراک.

### 8. \*\*SubscriptionService\*\*

- مدیریت انواع پلن‌های اشتراک (Basic, Premium و غیره).

- فعال‌سازی، تمدید یا لغو اشتراک کاربران.

- اتصال به BillingService برای پرداخت‌های مرتبط با اشتراک‌ها.

### 9. \*\*SharedKernel\*\*

- ماژول مشترک شامل Value Objects (مثل Email, PhoneNumber)، Enumerations (وضعیت دستگاه، نوع اشتراک)، Exceptions، Utilities و Interface‌های پایه.

- تضمین یکنواختی مفاهیم مشترک بین سرویس‌های مختلف.

### 10. \*\*Infrastructure\*\*

- پیاده‌سازی زیرساخت‌های فنی سیستم مثل دیتابیس‌ها (SQL/NoSQL)، Redis، Queues، فایل لاگ‌گیری و Health Checking سرویس‌ها.

- جداسازی کدهای مرتبط با تکنولوژی‌های اجرایی از منطق دامنه (Domain Logic).

### 11. \*\*EventBus\*\*

- مکانیزم ارتباط غیرهمگام بین سرویس‌ها با الگوی Pub/Sub.

- پیاده‌سازی شده با Kafka یا RabbitMQ یا NATS.

- سرویس‌ها می‌توانند Event تولید یا به Eventهای دیگر سرویس‌ها گوش دهند.

### 12. \*\*EventStoreService\*\*

- ذخیره‌سازی تمام رویدادهای مهم سیستم برای مقاصد Auditing و بازیابی (Event Sourcing).

- قابلیت بازسازی وضعیت فعلی سیستم از طریق Replay رویدادها.

### 13. \*\*JobSchedulerService\*\*

- زمان‌بندی اجرای وظایف دوره‌ای یا زمان‌بندی‌شده مثل تمدید اشتراک‌ها، پاک‌سازی داده‌های قدیمی، ارسال ایمیل‌های دوره‌ای.

- مدیریت صف‌های اجرای وظایف و پایش وضعیت اجرای آن‌ها.

### 14. \*\*SystemMonitoringService\*\*

- نظارت بر سلامت، عملکرد و منابع (CPU، RAM، Disk I/O) سرویس‌های مختلف.

- جمع‌آوری و تحلیل متریک‌های کلیدی (KPI) سیستم.

- ارسال هشدار (Alert) در صورت بروز اختلال یا کاهش کارایی.

### 15. \*\*HealthCheckService\*\*

- بررسی وضعیت سلامت سرویس‌های داخلی و خارجی (مثل دیتابیس، کش، صف پیام).

- ارائه Endpoint‌های سلامت برای Load Balancerها یا Kubernetes.

- تشخیص سریع خرابی‌ها و کمک به خودکارسازی فرآیند بازیابی.

### 16. \*\*Tests\*\*

- شامل مجموعه‌ای از:

- تست‌های واحد (Unit Tests) برای توابع و کلاس‌ها

- تست‌های ادغام (Integration Tests) برای سرویس‌های بین میکروسرویس‌ها

- تست‌های بار (Load Testing) برای ارزیابی مقیاس‌پذیری

- تست‌های امنیتی (Security Tests) برای اطمینان از ایمن بودن نقاط ورود سیستم.

# ✨ خلاصه نهایی

این ساختار یک معماری میکروسرویس کاملاً مستقل، مقیاس‌پذیر و قابل توسعه را برای پیاده‌سازی سیستم‌هایی مانند \*\*ردیابی GPS\*\*، \*\*مدیریت ناوگان\*\*، یا هر سیستم مبتنی بر دستگاه‌های IoT فراهم می‌کند.

تمام ماژول‌ها طوری طراحی شده‌اند که هم جداگانه توسعه و تست شوند و هم بتوانند از طریق Eventها یا APIها با هم تعامل داشته باشند.

بله، این لیست به‌صورت کامل و ساختارمند تمام \*\*ماژول‌های پروژه مدیریت دستگاه‌های GPS\*\* را پوشش می‌دهد. هر ماژول با توجه به نیازهای Enterprise-Level طراحی شده است و قابلیت مقیاس‌پذیری، انعطاف‌پذیری و توسعه‌پذیری بالایی دارد. در ادامه، به توضیح دقیق‌تر هر بخش و نقش آن در سیستم می‌پردازیم:

\*\*1. ApiGateway\*\*

- \*\*نقش\*\*: دروازه ورودی سیستم.

- \*\*کاربرد\*\*:

- توزیع درخواست‌ها به میکروسرویس‌های مختلف.

- مدیریت اعتبارسنجی (Authentication) و محدودیت نرخ (Rate Limiting).

- لاگ‌گیری درخواست‌ها و مدیریت خطاهای سراسری.

- \*\*ابزارها\*\*:

- \*\*Ocelot\*\*: برای مدیریت Gateway و Routing.

- \*\*Redis\*\*: برای Rate Limiter.

- \*\*JWT Bearer\*\*: برای اعتبارسنجی توکن‌ها.

\*\*2. DeviceManagementService\*\*

- \*\*نقش\*\*: مدیریت ثبت، به‌روزرسانی، حذف و اختصاص دستگاه‌ها.

- \*\*کاربرد\*\*:

- اعتبارسنجی شماره سریال دستگاه‌ها.

- مدیریت وضعیت دستگاه‌ها (فعال، غیرفعال، متصل، قطع شده).

- اختصاص دستگاه‌ها به کاربران.

- \*\*ابزارها\*\*:

- \*\*AutoMapper\*\*: برای نقشه‌برداری بین مدل‌ها.

- \*\*FluentValidation\*\*: برای اعتبارسنجی داده‌ها.

- \*\*Entity Framework Core\*\*: برای تعامل با دیتابیس.

\*\*3. LocationTrackingService\*\*

- \*\*نقش\*\*: ردیابی موقعیت‌های دستگاه‌ها.

- \*\*کاربرد\*\*:

- دریافت، پردازش و ذخیره داده‌های موقعیت.

- محاسبه فواصل و مسیرها.

- ارسال داده‌های زنده به کلاینت‌ها (SignalR).

- \*\*ابزارها\*\*:

- \*\*InfluxDB / TimescaleDB\*\*: برای ذخیره داده‌های زمان‌محور.

- \*\*Redis\*\*: برای Cache کردن داده‌ها.

- \*\*Kafka\*\*: برای ارسال داده‌ها به صف‌های رویداد.

\*\*4. CommandService\*\*

- \*\*نقش\*\*: مدیریت دستورات ارسالی به دستگاه‌ها.

- \*\*کاربرد\*\*:

- ارسال دستورات به دستگاه‌ها.

- ذخیره‌سازی دستورات و پاسخ‌های دریافتی.

- مدیریت صف‌ها با Kafka.

- \*\*ابزارها\*\*:

- \*\*Kafka\*\*: برای مدیریت صف‌های رویداد.

- \*\*Outbox Pattern\*\*: برای اطمینان از ارسال پیام‌ها.

\*\*5. NotificationService\*\*

- \*\*نقش\*\*: مدیریت اعلان‌ها و هشدارها.

- \*\*کاربرد\*\*:

- ارسال اعلان‌های بلادرنگ به کاربران.

- ارسال اعلان‌ها از طریق SignalR، SMS یا ایمیل.

- \*\*ابزارها\*\*:

- \*\*SignalR\*\*: برای ارتباط بلادرنگ.

- \*\*Twilio\*\*: برای ارسال SMS.

- \*\*SendGrid\*\*: برای ارسال ایمیل.

\*\*6. AuthService\*\*

- \*\*نقش\*\*: مدیریت احراز هویت و امنیت.

- \*\*کاربرد\*\*:

- تولید و اعتبارسنجی JWT.

- پشتیبانی از MFA (Multi-Factor Authentication).

- ردیابی IP برای جلوگیری از حملات Brute-force.

- \*\*ابزارها\*\*:

- \*\*OpenIddict\*\*: برای OAuth/OpenID.

- \*\*Bcrypt\*\*: برای هش کردن رمز عبور.

\*\*7. BillingService\*\*

- \*\*نقش\*\*: مدیریت صورتحساب‌ها و پرداخت‌ها.

- \*\*کاربرد\*\*:

- ایجاد و مدیریت صورتحساب‌ها.

- پردازش پرداخت‌ها با Stripe یا PayPal.

- \*\*ابزارها\*\*:

- \*\*Stripe SDK\*\*: برای مدیریت پرداخت‌ها.

- \*\*Entity Framework Core\*\*: برای ذخیره‌سازی اطلاعات مالی.

\*\*8. SubscriptionService\*\*

- \*\*نقش\*\*: مدیریت اشتراک‌ها.

- \*\*کاربرد\*\*:

- مدیریت اشتراک‌ها و تمدید آن‌ها.

- پردازش پرداخت‌های مرتبط.

- \*\*ابزارها\*\*:

- \*\*Stripe SDK\*\*: برای مدیریت اشتراک‌ها.

- \*\*Quartz.NET\*\*: برای زمان‌بندی تمدید اشتراک‌ها.

\*\*9. SharedKernel\*\*

- \*\*نقش\*\*: ارائه ابزارها و قراردادهای مشترک.

- \*\*کاربرد\*\*:

- شامل Value Objects، Enums، Utilities و Interface‌های عمومی.

- تعریف قراردادها برای ارتباط بین ماژول‌ها.

- \*\*ابزارها\*\*:

- \*\*MediatR\*\*: برای مدیریت Command و Query.

- \*\*FluentValidation\*\*: برای اعتبارسنجی.

\*\*10. Infrastructure\*\*

- \*\*نقش\*\*: مدیریت زیرساخت‌ها.

- \*\*کاربرد\*\*:

- مدیریت دیتابیس، Redis، لاگ‌گیری و Health Checks.

- \*\*ابزارها\*\*:

- \*\*Serilog\*\*: برای لاگ‌گیری.

- \*\*Prometheus + Grafana\*\*: برای مانیتورینگ.

\*\*11. EventBus\*\*

- \*\*نقش\*\*: مدیریت رویدادها.

- \*\*کاربرد\*\*:

- انتشار و مصرف رویدادها بین میکروسرویس‌ها.

- \*\*ابزارها\*\*:

- \*\*Kafka\*\*: برای مدیریت صف‌های رویداد.

- \*\*Outbox Pattern\*\*: برای اطمینان از ارسال پیام‌ها.

\*\*12. EventStoreService\*\*

- \*\*نقش\*\*: ذخیره‌سازی رویدادها.

- \*\*کاربرد\*\*:

- ردیابی تغییرات و بازیابی وضعیت سیستم.

- \*\*ابزارها\*\*:

- \*\*EventStoreDB\*\*: برای ذخیره‌سازی رویدادها.

\*\*13. JobSchedulerService\*\*

- \*\*نقش\*\*: مدیریت وظایف زمان‌بندی‌شده.

- \*\*کاربرد\*\*:

- تمدید اشتراک‌ها.

- ارسال اعلان‌ها.

- \*\*ابزارها\*\*:

- \*\*Quartz.NET\*\*: برای زمان‌بندی وظایف.

\*\*14. SystemMonitoringService\*\*

- \*\*نقش\*\*: نظارت بر عملکرد سیستم.

- \*\*کاربرد\*\*:

- جمع‌آوری متریک‌ها.

- نمایش داشبوردهای عملکرد.

- \*\*ابزارها\*\*:

- \*\*Prometheus + Grafana\*\*: برای مانیتورینگ.

\*\*15. HealthCheckService\*\*

- \*\*نقش\*\*: بررسی سلامت سیستم.

- \*\*کاربرد\*\*:

- بررسی وضعیت منابع (CPU، حافظه، دیسک، شبکه).

- \*\*ابزارها\*\*:

- \*\*ASP.NET Core Health Checks\*\*: برای بررسی سلامت.

\*\*16. Tests\*\*

- \*\*نقش\*\*: تست‌های واحد و ادغام.

- \*\*کاربرد\*\*:

- تست‌های واحد برای هر ماژول.

- تست‌های ادغام برای بررسی تعامل بین ماژول‌ها.

- \*\*ابزارها\*\*:

- \*\*xUnit\*\*: برای نوشتن تست‌ها.

- \*\*Moq\*\*: برای شبیه‌سازی وابستگی‌ها.

این ساختار کامل و مستقل طراحی شده است و می‌تواند به‌عنوان بخشی از یک سیستم ردیابی GPS یا هر سیستم دیگری استفاده شود. هر ماژول به‌صورت جداگانه قابل توسعه و تست است و از اصول Clean Architecture و Domain-Driven Design (DDD) پیروی می‌کند.