

TheFirstStepsTR.com - Backend Mimari ve Geliştirme Planı

Bu döküman, **Java** ekosisteminin gücünü kullanarak ölçeklenebilir, güvenli ve yüksek performanslı bir e-ticaret backend sisteminin nasıl kurgulanacağını detaylandırır.

1. Teknoloji Yığını (Tech Stack)

Sizin uzmanlık alanınız olan Java ile en modern ve endüstri standardı araçları seçtik:

- **Dil:** Java 17 veya Java 21 (LTS sürümleri)
- **Framework:** Spring Boot 3.x
- **Veritabanı (RDBMS):** PostgreSQL (İlişkisel veriler: Ürün, Sipariş, Müşteri)
- **NoSQL / Cache:** Redis (Sepet yönetimi, Session ve Önbellekleme için) & MongoDB (Ürün logları veya esnek özellikler için opsiyonel)
- **Arama Motoru:** Elasticsearch (Gelişmiş ürün arama ve filtreleme için)
- **Mesajlaşma:** Apache Kafka veya RabbitMQ (Sipariş -> Stok -> Bildirim servisleri arası asenkron iletişim)
- **Güvenlik:** Spring Security + JWT (JSON Web Token) veya OAuth2 (Keycloak entegrasyonu)
- **API Dokümantasyonu:** Swagger / OpenAPI 3
- **Build Tool:** Maven veya Gradle
- **Migration:** Liquibase veya Flyway

2. Mimari Yaklaşım: Modüler Monolitik'ten Mikroservis Geçiş

Başlangıçta karmaşıklığı yönetmek adına **Modüler Monolitik** (Modular Monolith) yapıyla başlamak, trafik arttıkça **Mikroservis** mimarisine bölünmek en sağlıklı yoldur.

Temel Modüller (Domainler)

Sistemi şu paketler/modüller altında toplayacağız:

1. **Auth Service:** Kullanıcı kayıt, giriş, rol yönetimi (Admin, User).
2. **Product Catalog Service:** Kategoriler, Ürünler, Stok takibi, Varyantlar (Renk, Beden).
3. **Order Service:** Sipariş oluşturma, Sipariş durumu (Pending, Paid, Shipped), İade süreçleri.
4. **Basket (Cart) Service:** Sepet işlemleri (Genelde Redis üzerinde tutulur, çok hızlı olmalı).
5. **Payment Service:** Ödeme entegrasyonu (Iyzico, Stripe vb. sanal pos entegrasyonları).
6. **Notification Service:** E-posta ve SMS bildirimleri (Sipariş alındı, Kargoya verildi vb.).

3. Veritabanı Tasarımı (ER Modeli Özeti)

PostgreSQL üzerinde kurgulanacak temel tablolar:

- **Users:** id, email, password_hash, full_name, role, phone
- **Addresses:** id, user_id, city, district, full_address, type
- **Products:** id, name, slug, description, price, stock_quantity, category_id, brand
- **Categories:** id, name, parent_id (Hiyerarşik yapı)
- **Orders:** id, user_id, total_price, status, payment_id, created_at
- **Order_Items:** id, order_id, product_id, quantity, price_at_purchase

4. Geliştirme Adımları (Roadmap)

Faz 1: Temel Atma ve Altyapı

1. **Proje Kurulumu:** Spring Initializr ile proje oluşturulması. Dependencies: Web, JPA, PostgreSQL, Lombok, Validation.
2. **Veritabanı Konfigürasyonu:** Docker üzerinde PostgreSQL ayağa kaldırılması.
3. **Güvenlik Katmanı:** Spring Security ile BCrypt şifreleme ve JWT token altyapısının yazılması. Interceptor/Filter katmanının kurulması.

Faz 2: Ürün ve Katalog Yönetimi (Catalog Domain)

1. **CRUD İşlemleri:** Admin paneli için ürün ekleme, güncelleme, silme servisleri.
2. **Görsel Yönetimi:** Ürün resimlerinin bir Cloud Storage'a (AWS S3, MinIO veya Google Cloud Storage) yüklenmesi için servis yazımı.
3. **Arama:** Ürünlerin Elasticsearch'e indekslenmesi ve hızlı arama API'sinin yazılması.

Faz 3: Sepet ve Sipariş (Core Business)

1. **Redis Sepet:** Kullanıcı sepete ekleme yaptığında veritabanını yormamak için Redis'te tutulması.
2. **Sipariş Akışı:** Checkout işlemi. Stok kontrolü -> Fiyat doğrulama -> Ödeme provizyonu -> Sipariş kaydı.
3. **Transactional Yönetim:** @Transactional anotasyonu ile veri bütünlüğünün sağlanması (Ödeme alınırsa stok düşmeli, hata olursa hepsi geri alınmalı).

Faz 4: Ödeme Entegrasyonu (Iyzico Örneği)

1. Türkiye şartlarında Iyzico veya benzeri bir ödeme sistemi entegrasyonu.
2. 3D Secure akışının backend tarafında handle edilmesi.
3. Callback (Webhook) endpointlerinin yazılması.

5. Örnek Kod Yapısı (Controller - Service - Repository)

Java bilginize istinaden, temiz kod (Clean Code) prensiplerine uygun katmanlı mimari örneği:

ProductController.java

```
@RestController
@RequestMapping("/api/v1/products")
@RequiredArgsConstructor
public class ProductController {

    private final ProductService productService;

    @GetMapping
    public ResponseEntity<List<ProductDto>> getAllProducts(@RequestParam(defaultValue = "0") int page) {
        return ResponseEntity.ok(productService.findAll(page));
    }

    @PostMapping
    @PreAuthorize("hasRole('ADMIN')")
    public ResponseEntity<ProductDto> createProduct(@Valid @RequestBody
    CreateProductRequest request) {
        return ResponseEntity.ok(productService.create(request));
    }
}
```

OrderService.java (Transactional Örneği)

```
@Service
@RequiredArgsConstructor
public class OrderService {

    private final ProductRepository productRepo;
    private final OrderRepository orderRepo;
    private final PaymentService paymentService;

    @Transactional
    public OrderDto createOrder(CreateOrderRequest request) {
        // 1. Stok Kontrolü
        checkStock(request.getItems());

        // 2. Fiyat Hesaplama
        BigDecimal total = calculateTotal(request.getItems());

        // 3. Ödeme Alma
        paymentService.charge(request.getPaymentDetails(), total);
    }
}
```

```
// 4. Stok Düşme ve Sipariş Kaydı
decreaseStock(request.getItems());
Order savedOrder = orderRepo.save(mapToEntity(request));

return mapToDto(savedOrder);
}
}
```

6. DevOps ve Deployment

Yazdığınız bu muazzam kodu canlıya almak için:

1. **Dockerize:** Her servis için Dockerfile oluşturulması.
2. **CI/CD:** GitHub Actions veya GitLab CI ile otomatik test ve build süreçleri.
3. **Sunucu:** AWS EC2 veya DigitalOcean Droplet üzerinde Docker Compose ile (veya Kubernetes) yayına alma.
4. **SSL/Domain:** Cloudflare arkasında Nginx Reverse Proxy ile api.thefirststepstr.com yönlendirmesi.

7. Öneriler

- **Loglama:** ELK Stack (Elasticsearch, Logstash, Kibana) veya basitçe Graylog kullanarak logları merkezi bir yerde toplayin.
- **Cache Stratejisi:** Çok okunan ama az değişen verileri (Kategoriler, Ana sayfa ürünleri) Redis'te cache'leyin (@Cacheable).
- **Test:** JUnit 5 ve Mockito ile Unit testleri, Testcontainers ile entegrasyon testlerini ihmal etmeyin.

Bu plan, **thefirststepstr.com**'u milyonlarca kullanıcıyı kaldırabilecek bir yapıya kavuşturacaktır.