

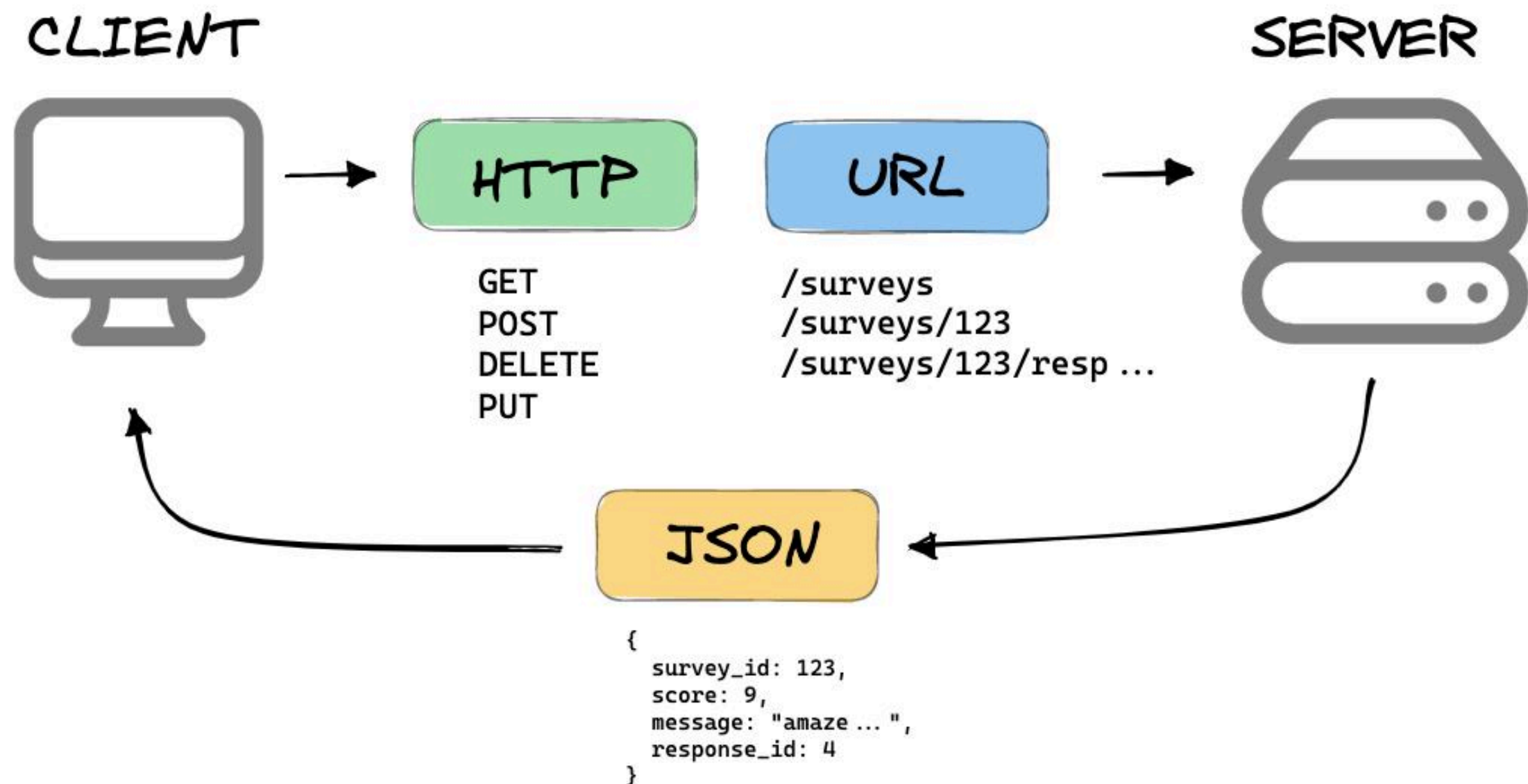


in/poyrazavsever



RESTFUL API NEDİR?

Bir **API** neden **RESTful** olur?





in/poyrazavsever



REST bir protokol değil, mimari kısıtlar setidir

Roy Fielding'in tanımıyla REST; Uniform Interface, Stateless, Cacheable, Layered System gibi kısıtlarla basit, ölçeklenebilir ve evrimleşebilir web servisleri üretme yaklaşımıdır.

RESTful demek, bu kısıtlarla uyumlu demektir.



in/poyrazavsever



Basit arayüz, geniş uyumluluk, düşük sürtünme

HTTP'nin doğal kabiliyetlerini (metotlar, durum kodları, cache, bağlantısızlık) kullanır.

Öngörülebilir URL'ler ve standart yanıtlar ile istemci entegrasyonlarını hızlandırır; bakım ve dokümantasyonu kolaylaştırır.



in/poyrazavsever



Kaynak odaklı tasarım

Her şey bir **kaynak (resource)**tır:

/users, /orders/42/items.

HTTP metodları anlam taşır:

GET (okuma),

POST (oluşturma),

PUT (tam güncelleme),

PATCH (kısmi güncelleme),

DELETE (silme).

İsimlendirme çoğul ve fiilsiz olmalıdır.



in/poyrazavsever



Yan etkiler ve tekrar çağrı güvenliği

- GET ve HEAD: Güvenli ve idempotent (tekrar çağrıda aynı etki).
- PUT & DELETE: İdempotent olmalı.
- POST: Genelde idempotent değildir (aynı istekte birden fazla kayıt oluşabilir).



```
PUT /users/123 HTTP/1.1
```

```
Content-Type: application/json
```

```
{"name": "Poyraz", "email": "p@ex.com"}
```



in/poyrazavsever



Her istek tek başına anlamlı olmalı

Sunucu oturum durumu tutmaz; kimlik doğrulama ve bağlam her istekte taşınır (ör. Authorization: Bearer ...).

Bu sayede yatay ölçekleme kolaylaşır, yük dengeleme basitleşir.



in/poyrazavsever



Hiyerarşi net olmalı

- Koleksiyon: /orders
- Tekil kaynak: /orders/42
- Alt kaynak: /orders/42/items
- Filtreleme/sıralama: /orders?status=shipped&sort=-createdAt

Not: URL'lerde fiil yok, kaynak adı var.



in/poyrazavsever



Başarı ve hata sinyalleri standart olmalı

- 200 OK / 201 Created / 204 No Content
- 400 Bad Request / 401 Unauthorized / 403 Forbidden / 404 Not Found
- 409 Conflict / 422 Unprocessable Entity / 429 Too Many Requests



```
{  
  "error": "validation_error",  
  "message": "email is invalid",  
  "fields": {"email": "invalid format"}  
}
```




in/poyrazavsever



Değişim kaçınılmaz → sürümleri yönetin

- URL tabanlı: /v1/orders (basit, görünür)
- Header tabanlı: Accept: application/vnd.api.v2+json (daha esnek)
- Alan/özellik düzeyi: Geriye uyumlu değişikliklere öncelik verin (ekleme güvenlidir, kaldırma kırar).



in/poyrazavsever



Büyük veriyle zarif çalışma

- Query parametreleri ile: page, limit, sort, filter
- Yanıt meta bilgisi: total, page, pageSize, hasNext



```
GET /products?page=2&limit=20&sort=-price&category=lighting
```



```
{  
  "data": [/* ... */],  
  "meta": {"page": 2, "pageSize": 20, "total": 538,  
    "hasNext": true}  
}
```



in/poyrazavsever



Güvenlik baştan tasarımın parçası

- OAuth2/OIDC ile kimlik doğrulama
- RBAC/ABAC ile yetkilendirme
- Rate limiting & throttling (429)
- CORS politikaları
- Duyarlı alanları maskeleyme, audit log, TLS zorunlu



in/poyrazavsever



REST'i REST olmaktan çıkaran hatalar

- URL'de fiiller: /createUser, /deleteOrder
- 200 OK ama gövdede hata dönmek
- Kimlik doğrulamayı cookie/oturum durumuna aşırı bağımlı kılmak
- Gereksiz iç içe gövdeler ve tutarsız adlandırma
- Geriye uyumsuz değişiklikleri küçük sürüm gibi sunmak

Sizin REST'te en çok zorlandığınız konu ne?

Yorumlara yazın: Versiyonlama mı, hata şeması mı, performans mı?



in/poyrazavsever

