

```
package com.airport.baggage.api;
```

```
import org.springframework.web.bind.annotation.*;
```

```
import java.time.Instant;
```

```
import java.util.*;
```

```
import org.springframework.http.ResponseEntity;
```

```
@RestController
```

```
@RequestMapping("/baggage")
```

```
public class BaggageController {
```

```
    private final BaggageService baggageService;
```

```
    public BaggageController(BaggageService baggageService) {
```

```
        this.baggageService = baggageService;
```

```
    }
```

```
    // 1. Last known location of a bag
```

```
    @GetMapping("/scan/bag/{bagTagId}")
```

```
    public ResponseEntity<BagScan> getLastScan(
```

```
        @PathVariable String bagTagId,
```

```
        @RequestParam(required = false, defaultValue = "false") boolean latest) {
```

```
        Optional<BagScan> scan = latest
```

```
            ? baggageService.getLatestScanForBag(bagTagId)
```

```
            : baggageService.getFullScanHistory(bagTagId);
```

```
        return scan.map(ResponseEntity::ok)
            .orElse(ResponseEntity.notFound().build());
    }
}
```

// 2. Bags currently en route to a gate

```
@GetMapping("/active/gate/{destinationGate}")
public List<EnRouteBag> getBagsEnRoute(
    @PathVariable String destinationGate,
    @RequestParam(name = "since_minutes") int sinceMinutes) {

    return baggageService.getBagsEnRouteToGate(destinationGate, sinceMinutes);
}
}
```

// 3. Count bags per gate

```
@GetMapping("/count/gate/{gateId}")
public Map<String, Object> countBagsAtGate(@PathVariable String gateId) {
    int count = baggageService.countBagsAtGate(gateId);
    return Map.of(
        "gate_id", gateId,
        "bag_count", count,
        "last_updated", Instant.now().toString()
    );
}
}
```

// 4. Count bags per gate (for all gates) in the last N minutes

```

    @GetMapping("/states/gate-counts")
    public List<GateBagCount> getBagCountsPerGate(
        @RequestParam(name = "since_minutes") int sinceMinutes) {
        return baggageService.countBagsPerGateSince(sinceMinutes);
    }
}

public class BagScan {
    public String bagTagId;
    public String timestamp;
    public String location;
    public String status;
    public String scannerId;
}

public class EnRouteBag {
    public String bagTagId;
    public String lastScanAt;
    public String lastLocation;
}

public class GateBagCount {
    public String destinationGate;
    public int uniqueBagsCount;
}

```

✅ Service Layer (Stub Example)

java

Copy

Edit

```
import java.util.*;
```

```
public interface BaggageService {
```

```
    Optional<BagScan> getLatestScanForBag(String bagTagId);
```

```
    Optional<BagScan> getFullScanHistory(String bagTagId);
```

```
    List<EnRouteBag> getBagsEnRouteToGate(String destinationGate, int sinceMinutes);
```

```
    int countBagsAtGate(String gateId);
```

```
    List<GateBagCount> countBagsPerGateSince(int sinceMinutes);
```

```
}
```

Notes

Timestamps can use `java.time.Instant` or `ZonedDateTime` depending on how you store them.

Actual scan storage might use a relational DB with timestamp filtering for the "last N minutes" logic.

Pagination or filtering could be added later if dataset size is large.

Would you like help writing the implementation for `BaggageServiceImpl`, or generating an OpenAPI spec for this controller?

```
import java.util.*;
```

```
public interface BaggageService {
```

```
    Optional<BagScan> getLatestScanForBag(String bagTagId);
```

```
    Optional<BagScan> getFullScanHistory(String bagTagId);
```

```
    List<EnRouteBag> getBagsEnRouteToGate(String destinationGate, int sinceMinutes);
```

```
    int countBagsAtGate(String gateId);
```

```
    List<GateBagCount> countBagsPerGateSince(int sinceMinutes);
```

```
}
```