Design

Code Structure

Code is organized with clean code principals. Use Case classes are not dependent on database layer, or the web layer.

Database Schema

```
CREATE TABLE profile_views (
  id BIGINT NOT NULL,
  user_id BIGINT,
  viewer_id BIGINT,
  date_time TIMESTAMP with time zone,
  CONSTRAINT pk_profile_views PRIMARY KEY (id)
);

CREATE INDEX user_date_idx ON profile_views(user_id, date_time);
```

Analyzing our access patterns, we need an index on `user_id` and `date_time` fields together.

For optimum storage and best query performance, this table should be partitioned on `date_time` field. Partitioning on 1-day intervals and dropping partitions older than 30 days would be the best combination for storage and query performance.

Deleting large amounts of data from a table is not optimal for storage and reduces index performance. But dropping old partitions is fast and does not mess up the indexes.

Usage

Run

```
./mvnw spring-boot:run
```

Create Profile View

```
curl -X POST http://localhost:8080/api/profile/2/view -H 'Content-Type: application/json' -d '{"viewerId": 3}'
```

Get Profile Views

```
curl -X GET http://localhost:8080/api/profile/2/view
```