ASSUMPTIONS:

A-1) Dates: Dates will have no time, no offset and no timezones.
Dates will be based on API service(s) localhost dates.
Date format is yyyy-MM-dd

A-2) API Securitization is Out of Scope: The API will not have Authentication neither Authorization (anyone can do anything).
e.g.: If I had your "reservation Id" I could cancel or update your reservation)

A-3) Check-in/Check-out at 12 AM means that reservation dates are expected like this:

```
    R1 = {..., arrival="2020-10-20"  departure="2020-10-20"}   -- Stays 1 day:  Oct/20
    R2 = {..., arrival="2020-10-20"  departure="2020-10-21"}   -- Stays 2 days: Oct/20 and Oct/21
    R1 = {..., arrival="2020-10-20"  departure="2020-10-22"}   -- Stays 3 days: Oct/20, Oct/21 and Oct/22
```

A-4) RACE CONDITIONS: If N concurrent Reservation requests have date-ranges collide, only one reservation will success (others N-1 will fail with some error message like "Campsite is taken between the dates provided")

```
  e.g.:
  Campsite availability [
       2020-10-19: taken  (no vacancy)
       2020-10-20: vacant
       2020-10-21: vacant
       2020-10-22: vacant
       2020-10-23: vacant
       2020-10-24: taken
  ]
  R1 = {..., arrival="2020-10-20"  departure="2020-10-21"}
  R2 = {..., arrival="2020-10-21"  departure="2020-10-23"}
```

If R1 and R2 are received at the same time, only one of this 2 reservations will success

A-5) API usage: I assume the endpoints usage will be distributed like this (approx)

dates vacancy check:	70%
reservations:	20%
cancelations/updates:	10%

(so trade-offs could punish reservations/cancellations/updates operations in order to improve availability_check operation response times)

QUESTIONS:

Q-1) Can I implement the API using Micronaut framework (https://micronaut.io/) ?

Q-2) Is there a policy that forbids the same campers (persons) stays more than 3 running days in the island?
If the answer is "No" you can skip **Q-2.1)** & **Q-2.2)**

Q-2.1) Can the fraud-check be done offline ?
An offline job that checks if the reservations (new or updated) apply this policy should speed-up the reservation endpoint response time.

e.g.: Instead of check online this policy, the API let the “infractor” make several reservations, that added takes more than 3 running days, and then trigger an offline job that checks this policy and cancel all reservations that doesn’t apply the policy.
(could trigger one check per each reservation or one that summarize all reservations in a given time e.g. 1h, that collects all by the same email .. I have no clue the final implementation right now)

Beacuse there is no way to identify who are the campers (persons) in the campsite or the ID of the user who reserves the campsite
Q-2.2) It’s ok to assume the email as "user Id" and that a the same email can't reserve the campsite for more than 3 running days

e.g.: the same email wants to make 4 reservations of 1 day each for 4 running days but only 3 will be alloweed.
R1 = {email="a@b.com", arrival="2020-10-20" departure="2020-10-20"}
R2 = {email="a@b.com", arrival="2020-10-21" departure="2020-10-21"}
R3 = {email="a@b.com", arrival="2020-10-22" departure="2020-10-22"}
R4 = {email="a@b.com", arrival="2020-10-23" departure="2020-10-23"}

Campsite availability endpoint
I'm thinking to receive 2 optional parameters ("from" and "to")
e.g. **/availability?from=2020-11-22&to=2020-11-25**

Q-3.1) If no dates are provided (from/to params) the API should return the campsite availability from tomorrow to (today + 1 month) ?

Q-3.2) If "from" param is not provided, It's ok to set from=tomorrow ?

Q-3.3) If "to" param is not provided, It's ok to set to= today + 1 month ?

Q-3.4) If from/to params are provided Can I limit the from/to rango to 1 month ?
e.g.: It's ok to ask from 2020-01-01 to 2021-01-01 ?