

ACTOR-BASED DESIGN PATTERNS

Amadou Sarjo Jallow
Vrije Universiteit Brussels
Software Architectures

Content

- I. General flow
- II. Adding a reservation
- III. Cancelling a reservation
- IV. Result dump

General flow

This is an actor-based design pattern to illustrate microservice concepts. I have adopted the ask pattern and the forward flow pattern to implement the online booking system. Once the system is up and running, multiple client actors are created which send search requests for different types of properties(hotels, apartments, resorts) that are available on a particular date. The [ClientActor](#) actor sends these search requests to the [SystemService](#) actor which returns a result containing the list of available properties on that date. A [Property](#) is randomly chosen and a [MakeReservation](#) message is sent to the [SystemService](#) to make a reservation. The [SystemService](#) then delegates this task to an ephemeral child which checks if a booking already exists for that property on that date or not. Based on the result of this query, a [AddReservation](#) message or [ChooseDifferentProperty](#) message is sent to the [ReservationService](#) actor or [ClientActor](#) actor accordingly. Once the [ReservationService](#) receives the [AddReservation](#) message, it checks if there exists a reservation against that [Property](#) on that date. The reason for checking here again is that our system might not be the only system using this [ReservationService](#). If the property is available on that date, a reservation is made, updating the bookings in the [SystemService](#) and sending messages to the [ClientActor](#) to log success or fail messages. A similar procedure also happens for [CancelReservation](#). The [ClientActor](#) sends the [CancelReservation](#) message to the [SystemService](#), and if the date of the reservation is 48 hours or more away, then the reservation is cancelled else the cancellation fails and an acknowledgement message is sent back to client.

Adding a reservation

```
case class Slug(cancellation: Boolean, propertyId: UUID)

class ClientActor(client: Client,
                  propertyType: Kind,
                  reservationDate: LocalDate,
                  action : ListBuffer[Slug],
                  searchActor: ActorRef) extends Actor with ActorLogging{

  if(action.isEmpty) {
    if(propertyType == Hotel){
      searchActor ! SearchHotel(reservationDate)
    }else if(propertyType == Apartment){
      searchActor ! SearchApartment(reservationDate)
    }else{ //Resort
      searchActor ! SearchResort(reservationDate)
    }
  }else {
    if(action(0).cancellation){ // true
      log.info(s"${client.name} wishes to cancel a reservation at
        ${getPropertyname(action(0).propertyId)}.")
      searchActor ! CancelReservation(action(0).propertyId, client, reservationDate) // reservaionDate is
        date at this moment.
    }else{
      log.info(s"${client.name} wishes to reserve an accommodation at
        ${getPropertyname(action(0).propertyId)} on ${reservationDate}")
      searchActor ! MakeReservation(action(0).propertyId, client, reservationDate)
    }
  }
}
```

A **ClientActor** takes parameters client, propertyType(Hotel, Apartment, Resort, None), the reservationDate, the **searchActor** reference and an action ListBuffer which contains a single **Slug** of parameters cancellation(which determines if the request is a **CancelReservation** or **MakeReservation**), and the propertyId of the **Property** to reserve or cancel.

For a simple property search request, the action buffer is always empty and for a reservation request or cancellation request, it is not empty. A request made with non-empty action ListBuffer sends **MakeReservation** or **CancelReservation** messages directly to the **searchActor** for the corresponding property whilst a search request only does this after the **SearchResults** message is received.

```

case SearchResults(msg) =>
  val propertyToReserve = msg(rnd.nextInt(msg.size))
  log.info(s"${client.name} wishes to reserve an accommodation at ${propertyToReserve.name} on
${reservationDate}")
  searchActor ! MakeReservation(propertyToReserve.id, client, reservationDate)

```

When the results of a search are communicated, a property is reserved randomly from those properties.

```

case MakeReservation(propertyId, client, date) =>
  val childService: ActorRef = context.actorOf(Props(new ChildService(client, propertyId, date ,
self, sender())))
  childService ! SendReservation(sender())

```

When the `SystemService` receives a `MakeReservation` message from the `ClientActor`, it sends a `SendReservation` message to its ephemeral child actor to handle the request.

```

case SendReservation(sender_) =>
  if(isPropertyAvailable(propertyId, client, date)){
    log.info(s"A reservation request has been made by ${client.name}.")
    val reservationService: ActorRef = context.actorOf(Props(new ReservationService(propertyId,
client, date)))
    reservationService ! AddReservation(sender_)
  }else{
    val msg = s"Property cannot be reserved on ${date}. Please choose a different property or date."
    replyTo ! ChooseDifferentProperty(msg)
  }

```

Once the child actor receives the `SendReservation` message, it checks if the property has already been booked by calling the `isPropertyAvailable` method. A `ChooseDifferentProperty` message is sent back to the Client or `AddReservation` message is sent to the `ReservationService` actor accordingly.

```

case AddReservation(sender_) =>
  if(isReservationAvailable(propertyId, client, date)){
    // @ams -> This date might have already been reserved by someone from a different booking system
    RandomValues.reservations += Reservation(propertyId, client, date)
    sender() ! PropertyReserved(s"A reservation has been successfully made for ${client.name}",
    sender_)
  } else {
    sender() ! NotReserved(s"A reservation cannot be made for ${client.name} on ${date}.", sender_)
  }
case RemoveReservation(sender_) =>
  RandomValues.reservations -= Reservation(propertyId, client, date)
  sender() ! ReservationCancelled(client, propertyId, date, sender_)

```

When a [AddReservation](#) is received, the property is reserved if there is no reservation for that date or not reserved otherwise. Remember multiple booking systems could be using this reservation service. A [RemoveReservation](#) message also signals the actor to remove a reservation and send an acknowledgement.

```

case Reserved(propertyId, client, date) =>
  // update bookings and unavailability listBuffer of property
  RandomValues.bookings += Reservation(propertyId, client, date)
  RandomValues.properties.foreach(property =>
    if(property.id == propertyId)
      property.unavailability += date
  )

```

When the [SystemService](#) actor receives the [Reserved](#) message, it updates its bookings ListBuffer and adds the date to the [unavailability](#) ListBuffer parameter of the property to mark the Property as booked.

Cancelling a reservation

```
case CancelReservation(propertyId, client, date) =>
  val bookedDate = RandomValues.bookings.filter( booking =>
    booking.propertyId.equals(propertyId) && booking.client.equals(client))
    .map( booking => booking.date).head
  if((bookedDate.getDayOfMonth - date.getDayOfMonth) >= 2){ // 48rs or more before bookedDate
    val reservationService: ActorRef = context.actorOf(Props(new ReservationService(propertyId,
client, bookedDate)))
    reservationService ! RemoveReservation(sender())
  }else{
    sender() ! CancellationFailed(client.name, propertyId, bookedDate)
  }
}
```

Once the `SystemService` receive a `CancelReservation` message from the client, it checks if it meets the cancellation policy and base on that, sends a `RemoveReservation` message to the `ReservationService` or a `CancellationFailed` message to the client.

```
case ReservationCancelled(client, propertyId, date, sender_) =>
  //update bookings and unavailability listBuffer of property
  RandomValues.bookings -= Reservation(propertyId, client, date)
  RandomValues.properties.foreach(property =>
    if(property.id == propertyId)
      property.unavailability -= date
  )
  sender_ ! CancellationSuccessful(client.name, propertyId, date)
```

When a reservation is cancelled, the `SystemService` receives a message `ReservationCancelled`, and then it updates its bookings and the `unavailability` `ListBuffer` parameter of a `Property` to make that property available for booking again.

Result dump

```
akka://OnlineBooking
Welcome to Kijamii Booking
[INFO] [07/20/2022 07:54:11.654] [OnlineBooking-akka.actor.default-dispatcher-3]
[akka://OnlineBooking/user/searchHotel] Linoel Bettington wishes to reserve an accommodation at Hotel
Central on 2022-07-20
[INFO] [07/20/2022 07:54:11.660] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$a] A reservation request has been made by Linoel Bettington.
[INFO] [07/20/2022 07:54:11.667] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchHotel] A reservation has been successfully made for Linoel Bettington
to Hotel Central on 2022-07-20.
[INFO] [07/20/2022 07:54:12.111] [OnlineBooking-akka.actor.default-dispatcher-3]
[akka://OnlineBooking/user/searchApartment] Huberto Galpen wishes to reserve an accommodation at
International Students Complex - Old Town on 2022-07-29
[INFO] [07/20/2022 07:54:12.112] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$b] A reservation request has been made by Huberto Galpen.
[INFO] [07/20/2022 07:54:12.112] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchApartment] A reservation has been successfully made for Huberto Galpen
to International Students Complex - Old Town on 2022-07-29.
[INFO] [07/20/2022 07:54:13.115] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/searchResort] Way Berard wishes to reserve an accommodation at Castle of
eglantines on 2022-07-26
[INFO] [07/20/2022 07:54:13.116] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$c] A reservation request has been made by Way Berard.
[INFO] [07/20/2022 07:54:13.117] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchResort] A reservation has been successfully made for Way Berard to
Castle of eglantines on 2022-07-26.
[INFO] [07/20/2022 07:54:14.125] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/searchHotel2] Louis Matson wishes to reserve an accommodation at Hotel
Central on 2022-07-27
[INFO] [07/20/2022 07:54:14.126] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$d] A reservation request has been made by Louis Matson.
[INFO] [07/20/2022 07:54:14.127] [OnlineBooking-akka.actor.default-dispatcher-8]
[akka://OnlineBooking/user/searchHotel2] A reservation has been successfully made for Louis Matson to
Hotel Central on 2022-07-27.
[INFO] [07/20/2022 07:54:15.129] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/searchApartment2] Janet Betchley wishes to reserve an accommodation at
Smartflats Premium - Palace du Grand Sablon on 2022-07-30
[INFO] [07/20/2022 07:54:15.130] [OnlineBooking-akka.actor.default-dispatcher-8]
[akka://OnlineBooking/user/search/$e] A reservation request has been made by Janet Betchley.
[INFO] [07/20/2022 07:54:15.131] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchApartment2] A reservation has been successfully made for Janet
Betchley to Smartflats Premium - Palace du Grand Sablon on 2022-07-30.
```



```

[INFO] [07/20/2022 07:54:16.131] [OnlineBooking-akka.actor.default-dispatcher-8]
[akka://OnlineBooking/user/searchResort2] Vince Rosenfelt wishes to reserve an accommodation at Center
Parcs Les Ardennes on 2022-07-31
[INFO] [07/20/2022 07:54:16.132] [OnlineBooking-akka.actor.default-dispatcher-5]
[akka://OnlineBooking/user/search/$f] A reservation request has been made by Vince Rosenfelt.
[INFO] [07/20/2022 07:54:16.132] [OnlineBooking-akka.actor.default-dispatcher-5]
[akka://OnlineBooking/user/searchResort2] A reservation has been successfully made for Vince Rosenfelt
to Center Parcs Les Ardennes on 2022-07-31.
[INFO] [07/20/2022 07:54:17.145] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/searchHotel3] Glenna Celloni wishes to reserve an accommodation at Hotel
Central on 2022-07-25
[INFO] [07/20/2022 07:54:17.145] [OnlineBooking-akka.actor.default-dispatcher-8]
[akka://OnlineBooking/user/search/$g] A reservation request has been made by Glenna Celloni.
[INFO] [07/20/2022 07:54:17.146] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchHotel3] A reservation has been successfully made for Glenna Celloni to
Hotel Central on 2022-07-25.
[INFO] [07/20/2022 07:54:18.156] [OnlineBooking-akka.actor.default-dispatcher-2]
[akka://OnlineBooking/user/searchApartment3] Roxie Hardbattle wishes to reserve an accommodation at U-
Residence on 2022-07-22
[INFO] [07/20/2022 07:54:18.156] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$h] A reservation request has been made by Roxie Hardbattle.
[INFO] [07/20/2022 07:54:18.157] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchApartment3] A reservation has been successfully made for Roxie
Hardbattle to U-Residence on 2022-07-22.
[INFO] [07/20/2022 07:54:19.169] [OnlineBooking-akka.actor.default-dispatcher-2]
[akka://OnlineBooking/user/searchResort3] Eleanore Cohane wishes to reserve an accommodation at Castle
of eglantines on 2022-07-25
[INFO] [07/20/2022 07:54:19.169] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$i] A reservation request has been made by Eleanore Cohane.
[INFO] [07/20/2022 07:54:19.170] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchResort3] A reservation has been successfully made for Eleanore Cohane
to Castle of eglantines on 2022-07-25.
[INFO] [07/20/2022 07:54:20.179] [OnlineBooking-akka.actor.default-dispatcher-2]
[akka://OnlineBooking/user/searchHotel4] Kizzie Hillum wishes to reserve an accommodation at Hotel
Central on 2022-07-22
[INFO] [07/20/2022 07:54:20.180] [OnlineBooking-akka.actor.default-dispatcher-2]
[akka://OnlineBooking/user/search/$j] A reservation request has been made by Kizzie Hillum.
[INFO] [07/20/2022 07:54:20.180] [OnlineBooking-akka.actor.default-dispatcher-5]
[akka://OnlineBooking/user/searchHotel4] A reservation has been successfully made for Kizzie Hillum to
Hotel Central on 2022-07-22.

```

The above two snapshots are the results of 10 clients booking concurrently.

```

akka://OnlineBooking
Welcome to Kijamii Booking
[INFO] [07/20/2022 07:57:26.245] [OnlineBooking-akka.actor.default-dispatcher-3]
[akka://OnlineBooking/user/searchHotel] Linoel Bettington wishes to reserve an accommodation at Juliana
Hotel on 2022-07-20
[INFO] [07/20/2022 07:57:26.252] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$a] A reservation request has been made by Linoel Bettington.
[INFO] [07/20/2022 07:57:26.260] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchHotel] A reservation has been successfully made for Linoel Bettington
to Juliana Hotel on 2022-07-20.
[INFO] [07/20/2022 07:57:26.625] [OnlineBooking-akka.actor.default-dispatcher-2]
[akka://OnlineBooking/user/searchApartment] Huberto Galpen wishes to reserve an accommodation at
Smartflats Premium - Palace du Grand Sablon on 2022-07-29
[INFO] [07/20/2022 07:57:26.626] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$b] A reservation request has been made by Huberto Galpen.
[INFO] [07/20/2022 07:57:26.626] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchApartment] A reservation has been successfully made for Huberto Galpen
to Smartflats Premium - Palace du Grand Sablon on 2022-07-29.
[INFO] [07/20/2022 07:57:27.639] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/searchResort] Way Berard wishes to reserve an accommodation at Center Parcs
Les Ardennes on 2022-07-26
[INFO] [07/20/2022 07:57:27.639] [OnlineBooking-akka.actor.default-dispatcher-2]
[akka://OnlineBooking/user/search/$c] A reservation request has been made by Way Berard.
[INFO] [07/20/2022 07:57:27.640] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/searchResort] A reservation has been successfully made for Way Berard to
Center Parcs Les Ardennes on 2022-07-26.
[INFO] [07/20/2022 07:57:28.647] [OnlineBooking-akka.actor.default-dispatcher-3]
[akka://OnlineBooking/user/searchHotel2] Louis Matson wishes to reserve an accommodation at Hotel
Central on 2022-07-27
[INFO] [07/20/2022 07:57:28.648] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$d] A reservation request has been made by Louis Matson.
[INFO] [07/20/2022 07:57:28.649] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchHotel2] A reservation has been successfully made for Louis Matson to
Hotel Central on 2022-07-27.
[INFO] [07/20/2022 07:57:29.652] [OnlineBooking-akka.actor.default-dispatcher-2]
[akka://OnlineBooking/user/searchApartment2] Janet Betchley wishes to reserve an accommodation at
Smartflats Premium - Palace du Grand Sablon on 2022-07-30
[INFO] [07/20/2022 07:57:29.653] [OnlineBooking-akka.actor.default-dispatcher-2]
[akka://OnlineBooking/user/search/$e] A reservation request has been made by Janet Betchley.
[INFO] [07/20/2022 07:57:29.654] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/searchApartment2] A reservation has been successfully made for Janet
Betchley to Smartflats Premium - Palace du Grand Sablon on 2022-07-30.

```

```

[INFO] [07/20/2022 07:57:30.660] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchResort2] Vince Rosenfelt wishes to reserve an accommodation at Village
Sunparks Kempense Meren on 2022-07-31
[INFO] [07/20/2022 07:57:30.660] [OnlineBooking-akka.actor.default-dispatcher-6]
[akka://OnlineBooking/user/search/$f] A reservation request has been made by Vince Rosenfelt.
[INFO] [07/20/2022 07:57:30.661] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/searchResort2] A reservation has been successfully made for Vince Rosenfelt
to Village Sunparks Kempense Meren on 2022-07-31.
[INFO] [07/20/2022 07:57:31.673] [OnlineBooking-akka.actor.default-dispatcher-3]
[akka://OnlineBooking/user/searchHotel3] Glenna Celloni wishes to reserve an accommodation at Boutique
Hotel on 2022-07-25
[INFO] [07/20/2022 07:57:31.674] [OnlineBooking-akka.actor.default-dispatcher-6]
[akka://OnlineBooking/user/search/$g] A reservation request has been made by Glenna Celloni.
[INFO] [07/20/2022 07:57:31.675] [OnlineBooking-akka.actor.default-dispatcher-6]
[akka://OnlineBooking/user/searchHotel3] A reservation has been successfully made for Glenna Celloni to
Boutique Hotel on 2022-07-25.
[INFO] [07/20/2022 07:57:32.673] [OnlineBooking-akka.actor.default-dispatcher-3]
[akka://OnlineBooking/user/searchApartment3] Roxie Hardbattle wishes to reserve an accommodation at
Smartflats Diamant on 2022-07-22
[INFO] [07/20/2022 07:57:32.674] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/search/$h] A reservation request has been made by Roxie Hardbattle.
[INFO] [07/20/2022 07:57:32.674] [OnlineBooking-akka.actor.default-dispatcher-3]
[akka://OnlineBooking/user/searchApartment3] A reservation has been successfully made for Roxie
Hardbattle to Smartflats Diamant on 2022-07-22.
[INFO] [07/20/2022 07:57:33.678] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/cancelReservation] Linoel Bettington wishes to cancel a reservation at
Juliana Hotel.
[INFO] [07/20/2022 07:57:33.687] [OnlineBooking-akka.actor.default-dispatcher-4]
[akka://OnlineBooking/user/cancelReservation] The reservation by Linoel Bettington for Juliana Hotel on
2022-07-24 was successfully cancelled.
[INFO] [07/20/2022 07:57:34.683] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/reserveUponCancelled] Glenna Celloni wishes to reserve an accommodation at
Juliana Hotel on 2022-07-24
[INFO] [07/20/2022 07:57:34.683] [OnlineBooking-akka.actor.default-dispatcher-7]
[akka://OnlineBooking/user/search/$j] A reservation request has been made by Glenna Celloni.
[INFO] [07/20/2022 07:57:34.684] [OnlineBooking-akka.actor.default-dispatcher-6]
[akka://OnlineBooking/user/reserveUponCancelled] A reservation has been successfully made for Glenna
Celloni to Juliana Hotel on 2022-07-24.

Process finished with exit code 0

```

These two snapshots are results of results of 10 clients running concurrently and the last two clients cancelled a reservation and booked a cancelled reservation accordingly. Uncomment the last two client values to test this process.