

Accelist 3+1 Training

Competency Examination

ASP.NET WEB API EXAM 2023

Accelist 3 + 1 Exam merupakan soal ujian untuk mengetes pemahaman peserta memahami training yang telah dilakukan.

Durasi ujian berlangsung selama 1 hari **(8 jam)**, dan dilarang untuk menyontek **jawaban atau solusi dari partisipan lain yang mengikuti ujian ini**. Jika ketahuan partisipan telah melakukan kecurangan, **maka akan diberikan sanksi yang sesuai dengan yang telah diterapkan**, contohnya **mem-blacklist partisipan dari perusahaan Accelist dan perusahaan-perusahaan yang terafiliasi atau telah menjadi partner Accelist**.

Pengumpulan solusi harus di-push ke Git repository berikut:

Jika ada pertanyaan seputar soal proyek yang diberikan, silahkan ditanyakan ke trainer.

Rules

- Boleh buka refferensi dari slide dan googling.
- Saat penilaian, jika ada item pada point killer ditemukan maka untuk item penilaian akan **langsung jadi 0!**
- C# Coding Conventions:
 1. PascalCase pada setiap class names and class members (fields, properties, methods, etc.) names.
 2. camelCase pada setiap lokal variable (var) names and method parameter names.
 3. Not await-ing an asynchronous method, which includes blocking it using: .Result, .Wait()

- Accelist Coding Conventions:
 1. Semua struktur code (contoh if-else, foreach-in) harus di tulis menggunakan {}
 2. Gunakan LINQ dengan metode syntax (Lambda Expression / Q => ...) dari pada metode syntax query (from ...), kecuali saat join table

Point Killer

- Untuk project entities dan *SQLite DB* sudah dibuatkan jadi tidak perlu lagi dibuat dari awal.
- wajib menggunakan *MARVEL Pattern* menggunakan **MediatR** dan **FluentValidation** untuk validasinya.
- Untuk return *error handling* nya wajib menggunakan format **standard RFC 7807**.
- perlu diperhatikan penggunaan http response code harus benar dan sesuai standard.
- Pasang serilog sink file dengan konfigurasi sebagai berikut :
 1. Log level information
 2. Nama file Log-{tanggal hari ini}.txt contoh : Log-20230328.txt
 3. Simpen dalam folder /logs
- Setiap konek ke DB harus menggunakan **async await**

Good Luck!

Accelist ingin mengembangkan API untuk sistem booking tiket online dengan nama AccelTick. Sistem yang akan dibuatkan merupakan sistem untuk booking berbagai macam tiket mulai dari bioskop, kereta, kapal laut, dan juga event konser;

Buatlah sistem dengan rincian desain arsitektur berikut:

- **Aplikasi web API (ASP.NET 6)**
- **Database menggunakan SQLite**

Untuk project repository dapat di ambil di link berikut :

<https://gitlab.com/accelist-training-center/2023/exams/Web-API-Exam.git>

Tata cara pengumpulan exam :

- Dikumpulkan dengan nama branch dengan format exam-{nama_kalian} contoh : exam-crist-yanathan
- Push kerjaan kalian
- Merge request tag dan assign akun gitlab @stric
- Merge request di lakukan sebelum jam 19:00, diatas itu maka akan ada sanksi

Berikut fitur-fitur yang akan dikembangkan:

- API GET untuk view available tiket yang **quotanya masih tersedia**
 - o Path nya api/v1/get-available-ticket.
 - o Input request querynya Nama Kategori, Kode Tiket, Nama Tiket, Harga, Tanggal Event Minimal, Tanggal Event Maksimal, Order By (semua kolom dari yang di tampilkan di result), Order State (kolom penentu untuk data akan di urutkan secara ascending atau descending).
 - o Harus bisa select semua data dan juga bisa searching by kolom Nama Kategori,Kode Tiket, Nama tiket, harga (untuk harga diambil harga yang \leq harga diinput), dan tanggal event(untuk tanggal event wajib bisa searching by mulai berlaku atau sampai berlaku atau keduanya menggunakan range).
 - o Searching harus bisa di 1 atau lebih kolom, jika semua kolom tidak ada yang disearching maka tampilkan semua data.
 - o Bisa Order data nya secara ascending dan descending berdasarkan nama kolom yang dipilih. Jika data Order By kosong by default order by kolom Kode Ticket secara ascending. Kolom Order State by Default ascending
 - o Return success nya harus menampilkan **list data** Nama Kategori, Kode Tiket, Nama Tiket, Tanggal Event, Harga, sisa quota berdasarkan hasil searching dan juga order by dan order state nya (untuk contoh outputnya liat gambar dibawah) di order berdasarkan tanggal paling baru dan harga paling rendah dan return error sesuai **standard RFC 7807** jika kena validasi.

```
{
  {
    "eventDate": "01-03-2022 12:00",
    "quota": 80,
    "ticketCode": "TD001",
    "ticketName": "Bus jawa-sumatra",
    "categoryName": "Transportasi Darat",
    "price": 50000000
  },
  {
    "eventDate": "01-03-2022 12:00",
    "quota": 70,
    "ticketCode": "TL001",
    "ticketName": "Kapal Ferri jawa-sumatra",
    "categoryName": "Transportasi Laut",
    "price": 50000000
  },
  {
    "eventDate": "01-03-2022 12:00",
    "quota": 99,
    "ticketCode": "C001",
    "ticketName": "Ironman CGV",
    "categoryName": "Cinema",
    "price": 50000000
  },
  {
    "eventDate": "01-03-2022 12:00",
    "quota": 76,
    "ticketCode": "H001",
    "ticketName": "Ibis Hotel Jakarta 21-23",
    "categoryName": "Hotel",
    "price": 50000000
  }
}
```

- o **BONUS POIN:** Tambahkan mekanisme pagination pada halaman ini dengan **item per page sebanyak 10 per halaman, contohnya bisa dilihat pada gambar dibawah)**

```

{
  "tickets": [
    {
      "eventDate": "02-03-2022 17:59",
      "quota": 80,
      "ticketCode": "TD001",
      "ticketName": "Bus jawa-sumatra",
      "categoryName": "Transportasi Darat",
      "price": 50000000
    },
    {
      "eventDate": "02-03-2022 17:59",
      "quota": 70,
      "ticketCode": "TL001",
      "ticketName": "Kapal Ferri jawa-sumatra",
      "categoryName": "Transportasi Laut",
      "price": 50000000
    },
    {
      "eventDate": "02-03-2022 17:59",
      "quota": 99,
      "ticketCode": "C001",
      "ticketName": "Ironman CGV",
      "categoryName": "Cinema",
      "price": 50000000
    },
    {
      "eventDate": "02-03-2022 17:59",
      "quota": 76,
      "ticketCode": "H001",
      "ticketName": "Ibis Hotel Jakarta 21-23",
      "categoryName": "Hotel",
      "price": 50000000
    }
  ],
  "totalTickets": 30
}

```

- API POST untuk booking tiket yang quotanya masih tersedia
 - Path nya api/v1/book-ticket.
 - Input request bodynya berupa objek yang didalamnya terdapat properti List Model yang berisi kolom Kode tiket, Quantity tiket.
 - Validasi
 - Kode tiket tidak terdaftar apabila yang diinput tidak ada di database.
 - Kode tiket quotanya habis apabila yang diinput Kode tiketnya quotanya sudah habis.
 - Quantity tiket yang diboeking melebihi sisa quota.
 - Kode ticket yang diboeking tanggal event nya tidak boleh \leq tanggal booking ticket.
 - Disimpan ke dalam table Bookedtiket
 - Return JSON nama tiket, kode tiket dan harga dari tiket yang di booking dan ditambah dengan 1 data hasil total penjumlahan dari semua harga tiketnya perkategori dan juga 1 data dari hasil penjumlahan harga semua kategori nya (untuk contoh outputnya liat gambar dibawah) jika berhasil dan return error sesuai **standard RFC 7807** jika kena validasi.

```

"priceSummary": 200000000,
"ticketsPerCategories": [
  {
    "categoryName": "Cinema",
    "summaryPrice": 100000000,
    "tickets": [
      {
        "ticketCode": "C001",
        "ticketName": "Ironman CGV",
        "price": 50000000
      },
      {
        "ticketCode": "C002",
        "ticketName": "Black Panther",
        "price": 50000000
      }
    ]
  },
  {
    "categoryName": "Transportasi Darat",
    "summaryPrice": 50000000,
    "tickets": [
      {
        "ticketCode": "TD001",
        "ticketName": "Bus jawa-sumatra",
        "price": 50000000
      }
    ]
  },
  {
    "categoryName": "Transportasi Laut",
    "summaryPrice": 50000000,
    "tickets": [
      {
        "ticketCode": "TL001",
        "ticketName": "Kapal Ferri jawa-sumatra",
        "price": 50000000
      }
    ]
  }
]

```

- API GET untuk view detail tiket yang sudah pernah di booking
 - Path nya api/v1/get-booked-ticket/{BookedTicketId}.
 - Input parameternya berupa parameter Booked tiket Id.
 - Validasi
 - BookedticketId tidak terdaftar apabila yang diinput tidak ada di database.
 - Return success nya harus menampilkan data JSON berupa , Kode Tiket, Nama Tiket, Tanggal Event, Quantity tiket yang di telah di book di group berdasarkan kategori (untuk contohnya liat gambar dibawah ini) dan return error sesuai **standard RFC 7807** jika kena validasi.

```

{
  "qtyPerCategory": 10,
  "categoryName": "Cinema",
  "tickets": [
    {
      "ticketCode": "C001",
      "ticketName": "Ironman CGV",
      "eventDate": "01-03-2022 11:58"
    },
    {
      "ticketCode": "C002",
      "ticketName": "Black Panther",
      "eventDate": "01-03-2022 11:58"
    }
  ]
},
{
  "qtyPerCategory": 10,
  "categoryName": "Transportasi Darat",
  "tickets": [
    {
      "ticketCode": "TD001",
      "ticketName": "Bus jawa-sumatra",
      "eventDate": "01-03-2022 11:58"
    }
  ]
}
}

```

- API DELETE untuk revoke tiket yang sudah pernah di booking
 - Path nya api/v1/revoke-ticket/{BookedTicketId}/{KodeTicket}/{Qty}.
 - Input parameternya berupa parameter Booked tiket Id, Kode Tiket dan Quantity sebanyak apa mau di hapus tiket yang telah di booking nya.
 - Validasi
 - Booked tiketId tidak terdaftar apabila yang diinput tidak ada di database.
 - Kode tiket tidak terdaftar apabila yang diinput tidak ada di database.
 - Qty harus <= sebanyak tiket yang sudah pernah di booking sebelumnya, jika lebih maka akan tervalidasi.
 - Update data Bookedtiket kolom qty pada tiket yang di revoke, jika qty menjadi 0 maka hapus row nya
 - Apa bila dari semua kode tiket pada Booked tiket Id tersebut sudah menjadi 0 maka hapus seluruh row dari Booked tiket Id tersebut pada table Bookedtiket.
 - Return success nya berupa Kode Ticket, Nama Ticket, Nama Kategori, dan sisa Quantity dari tiket yang di revoke dari BookedTicketId tersebut jika berhasil dan return error sesuai **standard RFC 7807** jika kena validasi.

```
[
  {
    "ticketCode": "TD001",
    "ticketName": "Bus jawa-sumatra",
    "quantity": 5,
    "categoryName": "Transportasi Darat"
  },
  {
    "ticketCode": "TL001",
    "ticketName": "Kapal Ferri jawa-sumatra",
    "quantity": 7,
    "categoryName": "Transportasi Laut"
  },
  {
    "ticketCode": "C001",
    "ticketName": "Ironman CGV",
    "quantity": 1,
    "categoryName": "Cinema"
  }
]
```

- API PUT untuk edit quantity tiket yang sudah pernah di booking
 - Path nya api/v1/edit-booked-ticket/{BookedTicketId}.
 - Input parameter nya berupa parameter Booked tiket Id.
 - Input Request Bodynya berupa object yang didalamnya terdapat sebuah list model yang berisi kolom Kode tiket nya dan quantity jumlah yang terbaru.
 - Validasi
 - Booked tiketId tidak terdaftar apabila yang diinput tidak ada di database.
 - Kode tiket tidak terdaftar apabila yang diinput tidak ada di database pada table Bookedtiket.
 - Quantity harus \leq total sisa quota tiket pada table tiket.
 - Quantity minimal 1.
 - Update data Bookedtiket kolom qty pada tiket yang telah di edit quantity.
 - Return success nya berupa Kode Ticket, Nama Ticket, Nama Kategori dan sisa Quantity dari tiket yang di edit dari BookedTicketId tersebut jika berhasil dan return error sesuai **standard RFC 7807** jika kena validasi.

```
[
  {
    "ticketCode": "TD001",
    "ticketName": "Bus jawa-sumatra",
    "quantity": 5,
    "categoryName": "Transportasi Darat"
  },
  {
    "ticketCode": "TL001",
    "ticketName": "Kapal Ferri jawa-sumatra",
    "quantity": 7,
    "categoryName": "Transportasi Laut"
  },
  {
    "ticketCode": "C001",
    "ticketName": "Ironman CGV",
    "quantity": 1,
    "categoryName": "Cinema"
  }
]
```