

Ендпоінт: Мінімальна/максимальна дата початку аукціону (auctionPeriod.startDate)

- Призначення
- URL
- Відповідь
 - Приклад (тільки minDate)
 - Приклад (minDate + maxDate)
 - Джерело правил: конфіг specs
 - Робочий/неробочий день
 - Мінімальна дата (validation.min)
 - Максимальна дата (validation.max)
 - Про startDate.time та conditions
 - Практичне використання на майданчиках
 - Висновок
- Change request
 - Мета зміни
 - Поточна поведінка (as-is)
 - Нова поведінка (to-be)
 - Backward compatibility
 - Правила вибору умов
 - Загальна логіка
 - Робота з case
 - Розширення формату конфігу
 - Boolean case (існує і зараз)
 - Numeric case для value.amount
 - Формат query параметрів
 - Boolean
 - Числовий value.amount
 - Відповідь ендпоінта (без змін)
 - Помилки / валідації
 - Некоректні значення в query параметрі - 400 Bad Request
 - Непідтримуваний параметр
 - Приклади сценаріїв
 - Perishable
 - Поріг стартової ціни
- Інформаційні повідомлення

Призначення

Ендпоінт надає довідкову інформацію щодо того, з якої дати (*і, за наявності, до якої дати*) Організатор може встановити auctionPeriod.startDate для процедури з конкретним sellingMethod.

Ендпоінт обчислює календарні межі згідно правил, описаних у конфігурації.

URL

GET /api/procedures/{sellingMethod}/auctionPeriod

де {sellingMethod} — технічна назва напрямку (*наприклад, basicSell-english, legitimatePropertyLease-english тощо*)

Приклад: <https://procedure.prozorro.sale/api/procedures/legitimatePropertyLease-english/auctionPeriod>

Відповідь

Ендпоінт повертає JSON з об'єктом startDate, який містить:

- minDate — **найраніша допустима дата/час** для auctionPeriod.startDate

- `maxDate` — **найпізніша допустима дата/час** (повертається лише якщо для `sellingMethod` у конфігу задано `validation.max`)

Приклад (тільки `minDate`)

```
{
  "startDate": {
    "minDate": "2026-03-04T09:00:00+00:00"
  }
}
```

Приклад (`minDate` + `maxDate`)

```
{
  "startDate": {
    "minDate": "2026-03-02T09:00:00+00:00",
    "maxDate": "2026-03-31T10:00:00+00:00"
  }
}
```

Джерело правил: конфіг specs

Посилання на конфіг: <https://procedure.prozorro.sale/api/specs>

Розрахунок дат виконується на основі конфігурації конкретного `sellingMethod`, секція: `periods.procedure.auctionPeriod.startDate.validation`

Ключові параметри:

Робочий/неробочий день

`is_business_day: true` означає, що результуюча дата має припадати на робочий день (після застосування `diff` і `direction`).

Мінімальна дата (`validation.min`)

Описує правило, як отримати `minDate`:

```
"min": {
  "diff": "8 days",
  "direction": "forward",
  "from": "now",
  "time": "11:00"
}
```

Логіка:

1. Береться базова точка відліку `from` (у прикладах це `now` — поточна дата/час на момент запиту).
2. Застосовується зміщення `diff` у напрямку `direction`:
 - `forward` — вперед від базової точки
 - `backward` — назад (для цього ендпоінту зазвичай не використовується, але механізм конфігураційно можливий)
3. Якщо задано `is_business_day: true` — дата нормалізується/підбирається так, щоб потрапити на робочий день.
4. Встановлюється час:
 - якщо `time` вказаний як **одна година** ("`11:00`") — використовується цей час;
 - якщо вказано інтервал ("`11:00 - 13:00`") — для `minDate` використовується **початок інтервалу** (логіка "найраніший можливий час").
5. Результат повертається в ISO-8601.

Максимальна дата (`validation.max`)

Якщо в конфігу присутній блок `validation.max`, ендпоінт додатково обчислює та повертає `maxDate`.

Приклад:

```
"max": {
  "diff": "35 days",
  "direction": "forward",
  "from": "now",
  "time": "13:00",
  "is_business_day": true
}
```

Логіка аналогічна `min`, але:

- для `maxDate` береться “найпізніший допустимий час”.
- якщо `time` — інтервал, то для `maxDate` використовується кінець інтервалу (логіка “найпізніший можливий час”).

Про `startDate.time` та `conditions`

У деяких `sellingMethod` у `startDate` додатково є:

- `time`: "11:00 - 13:00" — допустиме вікно часу проведення аукціону (впливає на вибір часу в межах доби)
- `conditions[]` — правила для автостановлення або альтернативної логіки (наприклад, коли `auto_set: true` і `case.isPerishable: true`)

Практичне використання на майданчиках

Майданчик може використовувати відповідь ендпоінту для:

- обмеження вибору дати в UI:
 - `minDate` як мінімально доступна дата,
 - `maxDate` як максимально доступна дата (якщо присутня)
- валідації введеного значення до відправки в API запиті
- відображення підказки Організатору типу:
 - “Найраніша дата початку аукціону: ...”
 - “Дозволений діапазон: ... - ...”

Висновок

- Ендпоінт повертає **розрахунок на момент запиту** (прив’язаний до `now`).
- Ендпоінт є **допоміжним**: фінальна валідність `auctionPeriod.startDate` все одно перевіряється ЦБД при створенні/оновленні процедури згідно тих самих правил конфігу.
- Якщо для `sellingMethod` не визначено `validation.max`, то `maxDate` не повертається.

Change request

Необхідно реалізувати підтримку `query`-параметрів у `GET /api/procedures/{sellingMethod}/auctionPeriod`

Мета зміни

Розширити ендпоінт `GET /api/procedures/{sellingMethod}/auctionPeriod`, щоб він міг обчислювати `min/max` дати з урахуванням параметрів конкретного кейсу, переданих у `query-string` (наприклад: `isPerishable`, `value.amount`), а не лише базової `validation.min/max`.

Це потрібно для того, щоб майданчик міг коректно показувати допустимий діапазон дат до створення процедури, але вже з урахуванням бізнес-умов.

Поточна поведінка (as-is)

Ендпоінт повертає `startDate.minDate` (та опційно `startDate.maxDate`) на основі `periods.procedure.auctionPeriod.startDate.validation`.

`conditions[]` наразі не параметризовані і не залежать від введених даних користувача (бо немає механізму передати ці дані у запит)

Нова поведінка (to-be)

Ендпоінт має приймати query параметри, які будуть використані при визначенні релевантного condition.case та/або правил validation

Базовий приклад:

```
GET /api/procedures/basicSell-english/auctionPeriod?isPerishable=true
```

Розширений приклад:

```
GET /api/procedures/sanctionedAssets-english/auctionPeriod?value.amount=300000000.00
```



Примітка

Параметри можуть комбінуватися

Backward compatibility

Якщо query параметри не передані, ендпоінт працює як зараз: повертає межі з startDate.validation.min/max (або тільки min, якщо max відсутній)

Якщо sellingMethod не має conditions[] для auctionPeriod.startDate — query параметри не впливають (результат як зараз)

Правила вибору умов

Загальна логіка

1. Система завантажує конфіг для {sellingMethod}
2. Якщо в startDate є масив conditions[], система має:
 - пройтись по conditions[] у вказаному у відповіді порядку
 - знайти першу умову, у якої case матиметься із якимось параметром запиту
 - застосувати її правила розрахунку дат
3. Якщо жоден case не матиметься — використовується дефолтна логіка з startDate.validation (*як зараз*)

Якщо матиметься кілька conditions - застосовується **перший у масиві** (тобто порядок у конфігу = пріоритет).

Це важливо, бо може бути комбінація:

```
GET /api/procedures/basicSell-english/auctionPeriod?value.amount=300000000.00&isPerishable=true
```

то в такому випадку застосовується правило, яке **перше в списку в specs**.

Наприклад, конфіг такий:

```

"active_tendering": {
  "periods": {
    "procedure": {
      "auctionPeriod": {
        "startDate": {
          "conditions": [
            {
              "auto_set": true,
              "case": {
                "isPerishable": true
              },
              "diff": "2 business days",
              "direction": "forward",
              "error": "raise",
              "from": "now",
              "time": "11:00 - 13:00"
            },
            {
              "case": {
                "value.amount": { "gt": 250000000 }
              },
              "validation": {
                "min": {
                  "diff": "30 days",
                  "direction": "forward",
                  "error": "raise",
                  "from": "now",
                  "time": "11:00",
                  "is_business_day": true
                },
                "max": {
                  "diff": "60 days",
                  "direction": "forward",
                  "error": "raise",
                  "from": "now",
                  "time": "13:00",
                  "is_business_day": true
                }
              }
            }
          ]
        },
        "time": "11:00 - 13:00",
        "validation": {
          "is_business_day": true,
          "min": {
            "diff": "8 days",
            "direction": "forward",
            "error": "raise",
            "from": "now",
            "time": "11:00"
          }
        }
      }
    }
  }
}

```

У прикладі

- isPerishable=true **матчиться з condition №1**
- value.amount=300000000.00 також **матчиться з condition №2** (бо 300000000 > 250000000)
- але оскільки **умови перевіряються послідовно**, застосовується **перший condition**, який матчиться.

Отже результат має бути розрахований **за condition №1** (isPerishable), а не за "високою ціною":

- minDate = now + 2 business days, час = початок інтервалу 11:00

- `maxDate` у цьому прикладі не повертається, бо `condition №1` не задає `validation.max`, а визначає лише правило для найранішої дати (як у поточних спеках `basicSell`).

Робота з case

`case` — це набір умов над параметрами, переданими у `query`

Підтримувані типи перевірок:

1. **boolean equality**
`isPerishable: true` → очікує `?isPerishable=true`
2. **numeric comparisons** для чисел
Наприклад:

- `value.amount > 250000000`
- `value.amount >= 250000000`
- `value.amount < 250000000`
- `value.amount <= 250000000`

Розширення формату конфігу

Boolean case (існує і зараз)

```
"conditions": [  
  {  
    "auto_set": true,  
    "case": {  
      "isPerishable": true  
    },  
    "diff": "2 business days",  
    "direction": "forward",  
    "from": "now",  
    "time": "11:00 - 13:00"  
  }  
]
```

Numeric case для `value.amount`

```
"conditions": [  
  {  
    "case": {  
      "value.amount": {  
        "gt": 250000000.00 // gt/gte/lt/lte  
      }  
    },  
    "validation": {  
      "min": {  
        "diff": "30 days",  
        "direction": "forward",  
        "from": "now",  
        "time": "11:00"  
      },  
      "max": {  
        "diff": "60 days",  
        "direction": "forward",  
        "from": "now",  
        "time": "13:00"  
      }  
    }  
  }  
]
```

`conditions[]` мають мати можливість:

- задавати "коротке" правило (diff/from/time) — як у прикладі з isPerishable

АБО

- перевизначати validation.min/max локально для цього кейсу

Тобто реалізмція має дозволити робити один із варіантів:

- Варіант А: condition має власні поля diff/from/... і повертає тільки minDate (або min+max, якщо є)
- Варіант Б: condition містить вкладений validation і повністю замінює дефолтний startDate.validation для обчислення min/max.

Приклад спеки:

```
"auctionPeriod": {
  "startDate": {
    "conditions": [
      {
        "case": { "value.amount": { "gt": 250000000 } },
        "validation": {
          "min": { "diff": "30 days", "direction": "forward", "from": "now", "time": "11:00" },
          "max": { "diff": "60 days", "direction": "forward", "from": "now", "time": "13:00" }
        }
      },
      {
        "case": { "value.amount": { "lte": 250000000 } },
        "validation": {
          "min": { "diff": "20 days", "direction": "forward", "from": "now", "time": "11:00" },
          "max": { "diff": "35 days", "direction": "forward", "from": "now", "time": "13:00" }
        }
      }
    ],
    "time": "11:00 - 13:00",
    "validation": {
      "is_business_day": true,
      "min": {
        "diff": "20 days",
        "direction": "forward",
        "error": "raise",
        "from": "now",
        "time": "11:00"
      }
    }
  }
}
```

Формат query параметрів

Boolean

- isPerishable=true|false (*case-insensitive*)
- якщо параметр переданий не як boolean → 400 помилка

Числовий value.amount

- value.amount — decimal
- якщо не число → 400 помилка

Відповідь ендпоінта (без змін)

Формат відповіді **не змінюється**:

```
{
  "startDate": {
    "minDate": "...",
    "maxDate": "..."
  }
}
```

Помилки / валідації

Некоректні значення в query параметрі - 400 Bad Request

Приклад:

- isPerishable = abc
- value.amount = not_a_number

Непідтримуваний параметр

Lenient: невідомі параметри ігноруються (*щоб не ламати майданчики*)

Приклади сценаріїв

Perishable

Запит:

GET /api/procedures/basicSell-english/auctionPeriod?isPerishable=true

Очікування:

- якщо в конфігу є condition з case.isPerishable=true, повертається min/max за ним (наприклад +2 business days)
- інакше — дефолтний validation.min/max

Поріг стартової ціни

Запит:

GET /api/procedures/sanctionedAssets-english/auctionPeriod?value.amount=300000000.65

Очікування:

- якщо в конфігу є condition value.amount gt 250000000, тоді:
 - minDate = now + 30 days
 - maxDate = now + 60 days (або як задано)
- якщо amount менший — інший condition або дефолтні правила

Інформаційні повідомлення

Ендпоінт повертає min max дату першого аукціону та tenderPeriod.duration для другого і наступних аукціонів в ланцюжку

<https://procedure.prozorro.sale/api/jobber/announcements/jas/hints/announcement>