Empire API Integration

SUPER CHARGE & BritNed

meet the team

Daniel Homoki-Farkas

Managing Director, UK

Balazs Kovacs

Chief Architect

Nikolett Szabo

Senior Project Manager

Adam Nogradi

Technical Product Owner

Anita Plesiu

Service Operations Manager

contents

01 integration overview

02 submitting nominations

03 submitting bids

04 sample JSON requests

05 XML file upload

06 JAO Syntax Emulator

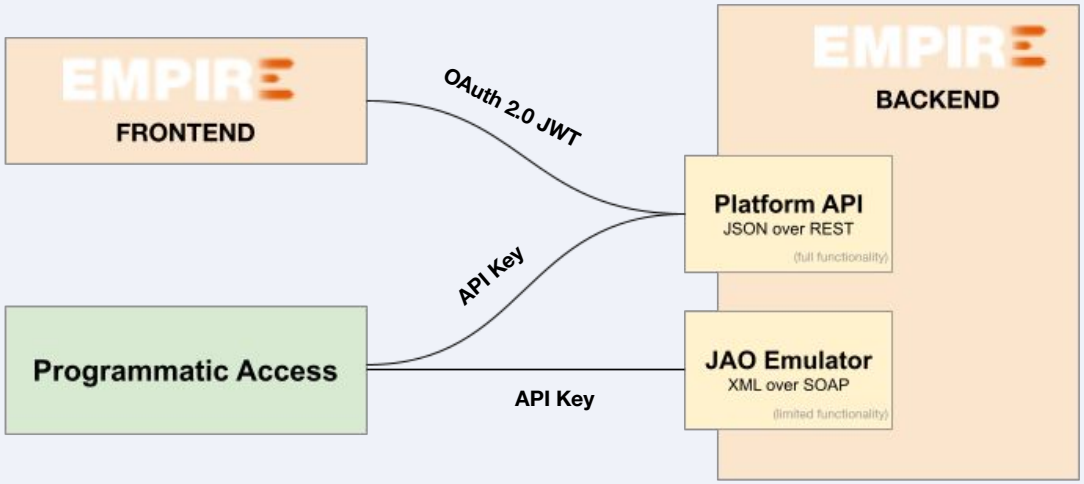
07 Customer Integration Testing

preface

- full walkthrough and guide about the JSON APIs
- provide sample XML files & Postman collection for the JSON API request - responses
- at the end we will provide details about the CIT & CAT schedule

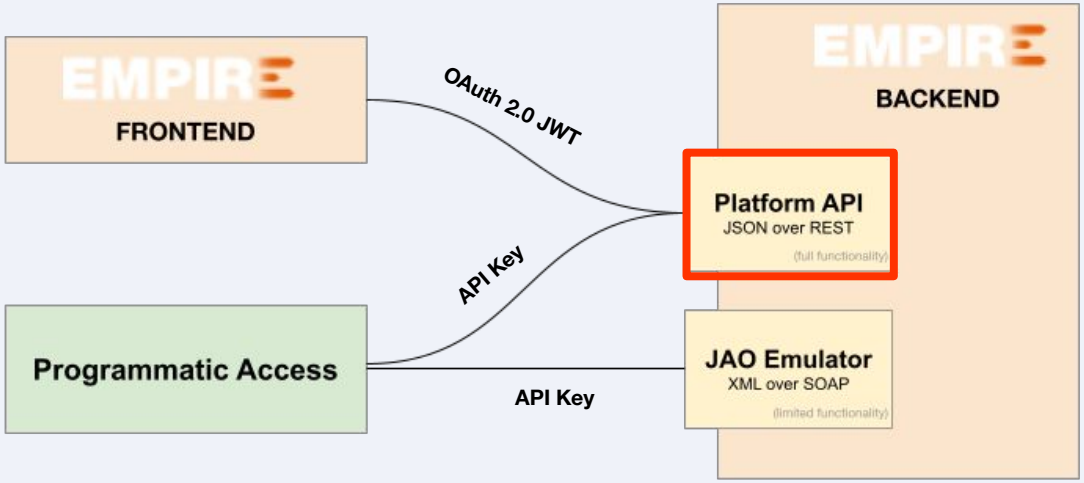
1. integration overview

Empire APIs

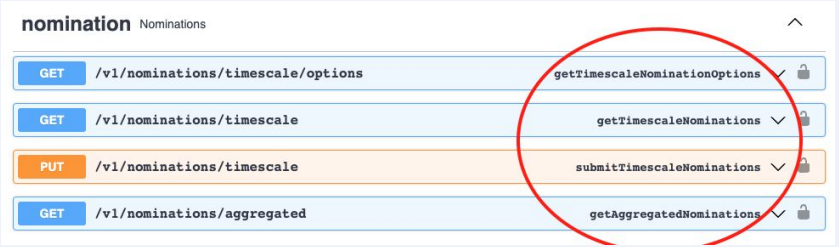


- Empire's **Backend** system offers two interfaces to the outside world
- the main interface is called **Platform API**
  - is the one that is powering the **Frontend** of Empire thus covers the *full functionality* of the platform
  - It is a modern REST style API primarily using JSON payloads
  - can be consumed programmatically using **API Keys**
- in addition to this, a **JAO Emulator** interface is available:
  - it is mimicking JAO's XML over SOAP API
  - can be consumed programmatically using **API Keys**
  - offers *limited functionality* (mostly around bidding and auctions)

Platform API



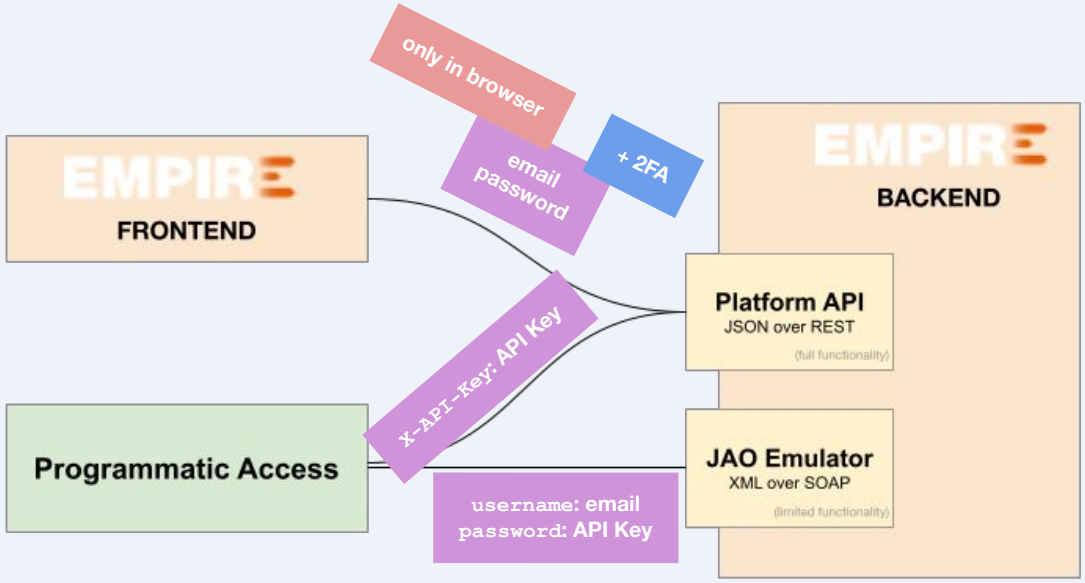
- more about **Platform API**
  - :: the latest **OpenAPI documentation** is always available on GitHub*
  - :: generated **Python SDK** available for the public on PyPI.org**
  - :: timestamps are to be sent in ISO 8601 format in UTC time standard:
    - -- eg: `2023-07-11T09:27:54Z`
  - :: we'll refer to Platform API endpoints using their **Operation ID**



* Swagger UI: https://britned.github.io/empire-platform-api/
OpenAPI description: https://github.com/britned/empire-platform-api/blob/main/openapi.yaml

** Python Client SDK on PyPI.org: https://pypi.org/project/empire-platform-api-public-client/

API Authentication



- **Empire Frontend** handles the authentication in the browser with email + password and additional 2FA
- on the Frontend every user can generate **API Keys** for themselves
  - API Keys are long strings in `emp_...` format
  - the generated API Key **“inherits” the permissions** of the user it belongs to
- **Platform API** can be consumed programmatically by setting the `X-API-Key` header to a valid API Key value (`emp_...`)
- the **JAO Emulator** mimics utilises username + password login:
  - `username` should be the email address of the user
  - `password` should be a generated API Key (`emp_...`)

fetching user profile from Platform API

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/profile' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...'
```


```
{
  "user": {
    "id": "3c4e8e61-5185-40d8-a9b2-89bfcba21d80",
    "name": "Apple Test User",
    "email": "apple_user@empire.scapps.io",
    "role": "PART_ADMIN",
    "organisation": {
      "id": "5c6a088a-93b6-433c-8538-b7488207df39",
      "name": "Test A Participant",
      "authMethod": "PLATFORM"
    }
  },
  "permissions": [
    "VIEW_PUBLISHED_AUCTIONS",
    "MANAGE_LT_AUCTION_BIDS",
    ...
  ],
  "appearance": "DEFAULT",
  "impersonatedBy": null
}
```


Test A Participant

fetching user profile from Platform API

⬅️ make sure to use the right environment

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/profile' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...'
```


```
{
  "user": {
    "id": "3c4e8e61-5185-40d8-a9b2-89bfcba21d80",
    "name": "Apple Test User",
    "email": "apple_user@empire.scapps.io",
    "role": "PART_ADMIN",
    "organisation": {
      "id": "5c6a088a-93b6-433c-8538-b7488207df39",
      "name": "Test A Participant",
      "authMethod": "PLATFORM"
    }
  },
  "permissions": [
    "VIEW_PUBLISHED_AUCTIONS",
    "MANAGE_LT_AUCTION_BIDS",
    ...
  ],
  "appearance": "DEFAULT",
  "impersonatedBy": null
}
```


Test A Participant

fetching user profile from Platform API

⬅️ send API Key in `X-API-Key` header

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/profile' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...'
```


```
{
  "user": {
    "id": "3c4e8e61-5185-40d8-a9b2-89bfcba21d80",
    "name": "Apple Test User",
    "email": "apple_user@empire.scapps.io",
    "role": "PART_ADMIN",
    "organisation": {
      "id": "5c6a088a-93b6-433c-8538-b7488207df39",
      "name": "Test A Participant",
      "authMethod": "PLATFORM"
    }
  },
  "permissions": [
    "VIEW_PUBLISHED_AUCTIONS",
    "MANAGE_LT_AUCTION_BIDS",
    ...
  ],
  "appearance": "DEFAULT",
  "impersonatedBy": null
}
```


Test A Participant

fetching user profile from Platform API

⬅️ fetch user of the API Key using `getProfile`

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/profile' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...'
```


```
{
  "user": {
    "id": "3c4e8e61-5185-40d8-a9b2-89bfcba21d80",
    "name": "Apple Test User",
    "email": "apple_user@empire.scapps.io",
    "role": "PART_ADMIN",
    "organisation": {
      "id": "5c6a088a-93b6-433c-8538-b7488207df39",
      "name": "Test A Participant",
      "authMethod": "PLATFORM"
    }
  },
  "permissions": [
    "VIEW_PUBLISHED_AUCTIONS",
    "MANAGE_LT_AUCTION_BIDS",
    ...
  ],
  "appearance": "DEFAULT",
  "impersonatedBy": null
}
```


Test A Participant

fetching user profile from Platform API

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/profile' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...'
```

```
{
  "user": {
    "id": "3c4e8e61-5185-40d8-a9b2-89bfcba21d80",
    "name": "Apple Test User",
    "email": "apple_user@empire.scapps.io",
    "role": "PART_ADMIN",
    "organisation": {
      "id": "5c6a088a-93b6-433c-8538-b7488207df39",
      "name": "Test A Participant",
      "authMethod": "PLATFORM"
    }
  },
  "permissions": [
    "VIEW_PUBLISHED_AUCTIONS",
    "MANAGE_LT_AUCTION_BIDS",
    ...
  ],
  "appearance": "DEFAULT",
  "impersonatedBy": null
}
```

⬅️ to be used as `participantId` in subsequent requests

⬅️ permissions (refer to API specification)

Test A Participant

2.
submitting nominations

nominations overview

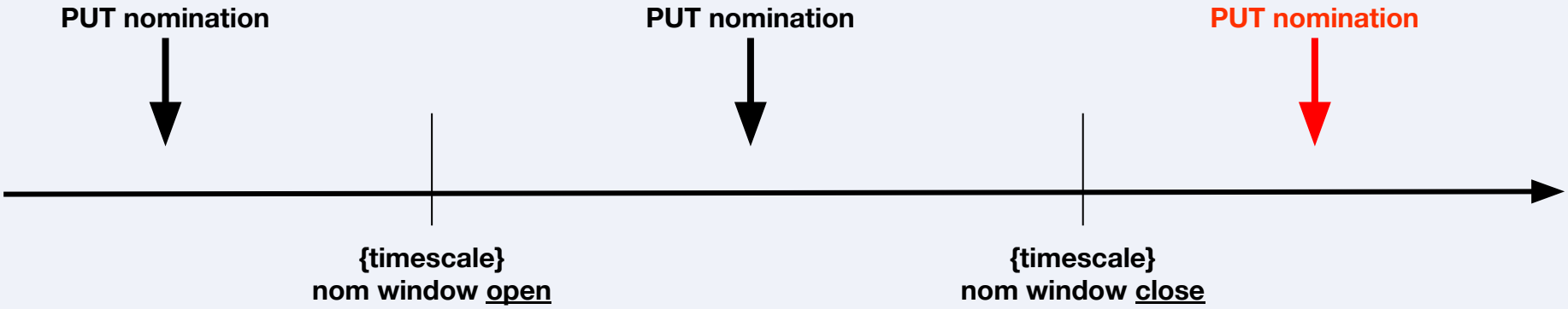
- multiple ways to submit nominations
- our recommendation is to use the JSON API
- provides
  - the most transparency and feedback for Participants
  - detailed feedback in case of submission issues
- use `getTimescaleNominations` / `submitTimescaleNomiations` to achieve the desired outcomes (see details later)

nominations endpoints details

- use `getTimescaleNominations` to find out MTU status, available TRs, curtailment feedback, etc…
- all capacity values are sent / received in kW
- when using the `submitTimescaleNomiations` endpoint always submit all MTUs received in GET
  - for locked MTUs send `null`
  - in case you do not want to nominate an MTU send `null`
  - for editable MTU send nomination value in kW
- full delivery day MTUs requirement caters for clock change and later MTU resolution change as well
- any nominations received (*if accepted*) will override all previous records automatically

more details & examples on the following slides

pre & draft nominations

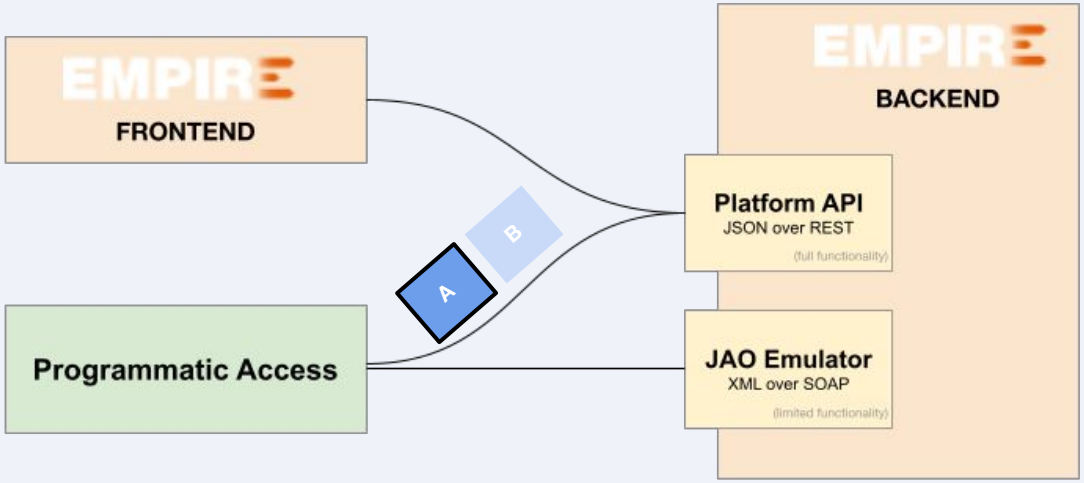


nomination saved as:
PRE_NOMINATION

nomination saved as:
DRAFT_NOMINATION

nomination not possible and will be rejected

submitting nominations



There are 2 different ways of submitting Bids to **Empire**

A) using the Platform API JSON endpoints

- fetch current nominations with `getTimescaleNominations`
- submit nominations with `submitTimescaleNominations`

A) LT nominations using JSON endpoints

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2023-07-10 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_zudla5...'
```


```
{
    "mtuSize": "MTU_60_MINS",
    "mtus": [
        {
            "mtu": "2023-07-12T22:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        {
            "mtu": "2023-07-12T23:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        ...
    ]
}
```


step #1 - `getTimescaleNominations`

A) LT nominations using JSON endpoints

⬅️ make sure to use the right environment & API Key

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2023-07-10 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_zudla5...'
```


```
{
    "mtuSize": "MTU_60_MINS",
    "mtus": [
        {
            "mtu": "2023-07-12T22:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        {
            "mtu": "2023-07-12T23:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        ...
    ]
}
```


step #1 - `getTimescaleNominations`

A) LT nominations using JSON endpoints

⬅️ request parameters for `getTimescaleNominations`

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2023-07-10 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_zudla5...'
```

```
{
    "mtuSize": "MTU_60_MINS",
    "mtus": [
        {
            "mtu": "2023-07-12T22:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        {
            "mtu": "2023-07-12T23:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        ...
    ]
}
```

step #1 - `getTimescaleNominations`

A) LT nominations using JSON endpoints

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2023-07-10 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_zudla5...'
```

⬅️ `participantId` obtained from `getProfile`

```
{
    "mtuSize": "MTU_60_MINS",
    "mtus": [
        {
            "mtu": "2023-07-12T22:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        {
            "mtu": "2023-07-12T23:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        ...
    ]
}
```

step #1 - `getTimescaleNominations`

A) LT nominations using JSON endpoints

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2023-07-10 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_zudla5...'
```

```
{
    "mtuSize": "MTU_60_MINS",
    "mtus": [
        {
            "mtu": "2023-07-12T22:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        {
            "mtu": "2023-07-12T23:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        ...
    ]
}
```

⬅️ `mtus` calculated on backend-side

- always use the MTU list provided and **map these values** when sending `submitTimescaleNominations`endpoint
- clock changes are handled automatically, `mtus` array have 23-24-25 items depending on the requested delivery day

step #1 - `getTimescaleNominations`

A) LT nominations using JSON endpoints

```
curl -X 'GET' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2023-07-10 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_zudla5...'
```

```
{
    "mtuSize": "MTU_60_MINS",
    "mtus": [
        {
            "mtu": "2023-07-12T22:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        {
            "mtu": "2023-07-12T23:00:00Z",
            "mtuStatus": "EDITABLE",
            "totalTrs": 30000,
            "notNominatedTrs": 30000,
            "currentNomination": null
        },
        ...
    ]
}
```

⬅️ properties for each MTU's current nominations

- available transmission rights: 30.000 kW = 30 MW
- all of it is yet un-nominated
- `currentNomination` value is `null` (which means "no nominations", not 0 kW nominations!)

internal business logic to be implemented at this step

step #1 - `getTimescaleNominations`

A) LT nominations using JSON endpoints

```
curl -X 'PUT' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2022-01-04 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: */*' \
  -H 'X-API-Key: emp_zudla5...' \
  -H 'Content-Type: application/json' \
  -d '{
    "mtus": [
        { "mtu": "2023-07-12T22:00:00.000Z" },
        { "mtu": "2023-07-12T23:00:00.000Z" },
        {
            "mtu": "2023-07-13T00:00:00.000Z",
            "value": 30000
        },
        {
            "mtu": "2023-07-13T01:00:00.000Z",
            "value": 30000
        },
        { "mtu": "2023-07-13T02:00:00.000Z" },
        ...
    ]
}'
```

step #2 - `submitTimescaleNominations`

A) LT nominations using JSON endpoints

```
curl -X 'PUT' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2022-01-04 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: */*' \
  -H 'X-API-Key: emp_zudla5...' \
  -H 'Content-Type: application/json' \
  -d '{
    "mtus": [
        { "mtu": "2023-07-12T22:00:00.000Z" },
        { "mtu": "2023-07-12T23:00:00.000Z" },
        {
            "mtu": "2023-07-13T00:00:00.000Z",
            "value": 30000
        },
        {
            "mtu": "2023-07-13T01:00:00.000Z",
            "value": 30000
        },
        { "mtu": "2023-07-13T02:00:00.000Z" },
        ...
    ]
}'
```

⬅️ request parameters for `submitTimescaleNominations`

⌟ use the same parameters as in the previous step

⬅️ indicate sending JSON payload

step #2 - `submitTimescaleNominations`

A) LT nominations using JSON endpoints

```
curl -X 'PUT' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2022-01-04 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: */*' \
  -H 'X-API-Key: emp_zudla5...' \
  -H 'Content-Type: application/json' \
  -d '{
    "mtus": [
        { "mtu": "2023-07-12T22:00:00.000Z" },
        { "mtu": "2023-07-12T23:00:00.000Z" },
        {
            "mtu": "2023-07-13T00:00:00.000Z",
            "value": 30000
        },
        {
            "mtu": "2023-07-13T01:00:00.000Z",
            "value": 30000
        },
        { "mtu": "2023-07-13T02:00:00.000Z" },
        ...
    ]
}'
```

⬅️ list of `mtus` mapped from previous step

- always the full list need to be sent
- subsequent requests **overwrite** each other

step #2 - `submitTimescaleNominations`

A) LT nominations using JSON endpoints

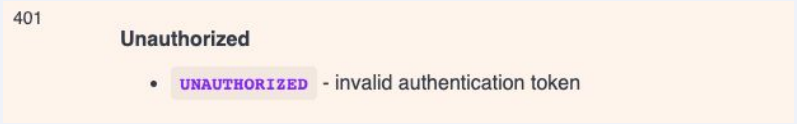
```
curl -X 'PUT' \
  'https://api.dev.empire.scapps.io/v1/nominations/timescale \
  ?deliveryDay=2022-01-04 \
  &type=LONG_TERM_NOMINATION \
  &borderDirection=NL_GB \
  &participantId=5c6a088a-93b6-433c-8538-b7488207df39' \
  -H 'accept: */*' \
  -H 'X-API-Key: emp_zudla5...' \
  -H 'Content-Type: application/json' \
  -d '{
    "mtus": [
        { "mtu": "2023-07-12T22:00:00.000Z" },
        { "mtu": "2023-07-12T23:00:00.000Z" },
        {
            "mtu": "2023-07-13T00:00:00.000Z",
            "value": 30000
        },
        {
            "mtu": "2023-07-13T01:00:00.000Z",
            "value": 30000
        },
        { "mtu": "2023-07-13T02:00:00.000Z" },
        ...
    ]
}'
```

⬅️ send nomination values in kW

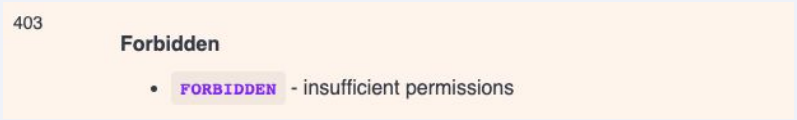
- only in the MTUs where nominations are intended to be submitted
- subsequent requests **overwrite** each other!
- In order to **update** existing nominations:
  - :: fetch current values with `getTimescaleNominations`
  - :: apply updates based on internal business logic
  - :: re-submit with `submitTimescaleNominations`

step #2 - `submitTimescaleNominations`

validations & error handling



- `401` status with `UNAUTHORISED` error code for invalid API Key



- `403` status with `FORBIDDEN` error code for permission problems
  - the user does not have the required permissions
  - also for e.g submitting someone else's participantId

validations & error handling

400

Bad Request

- `INVALID_PARTICIPANT_ID` - specified Participant not found in the system

↳ `400` status with custom error codes for fundamental problems

422

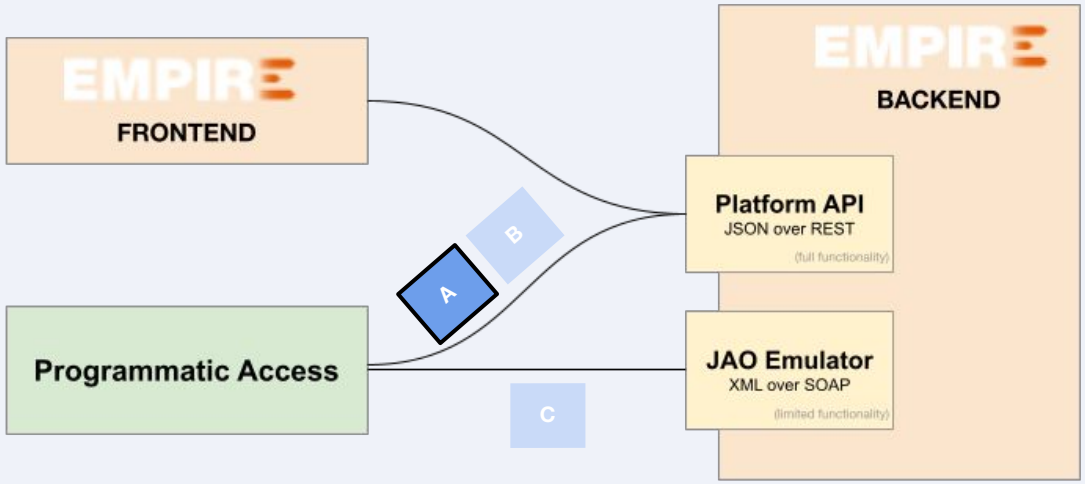
Unprocessable Entity

- `MTU_LIST_MISALIGNED`
  - for the values to be sent
  - one or more MTU start times are not aligned to MTU borders (considering MTU size)
  - the number of MTUs defined does not match the required number (pay special attention to daylight saving changes!)
- `MTU_NOT_EDITABLE`
  - only MTUs that are in `mtuState = EDITABLE` can contain not-empty values
  - an MTU can be non-editable because:
    - the MTU nomination window for the selected timescale is closed
    - has already FIRM or CURTAILED nominations
- `INVALID_NOMINATION_VALUE`
  - nomination values can be empty for any MTU, where not empty has to be
  - in CAPPED state:
    - 0 <= value <= Cap Value
  - in FLOORED state:
    - Floor Value <= value <= Total TRs
  - otherwise:
    - 0 <= value <= Total TRs

↳ `422` status with custom error codes for validation errors

3.
submitting bids

submitting bids



There are 3 different ways of submitting Bids to **Empire**

A) using the Platform API JSON endpoints

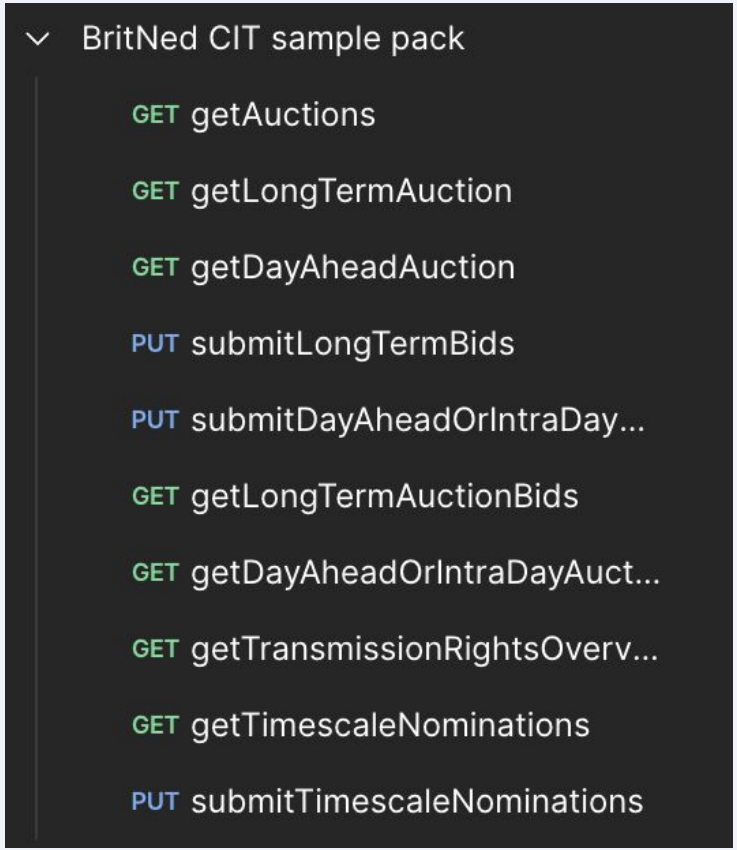
- fetch current with `getLongTermAuctionBids` or `getDayAheadOrIntraDayAuctionBids`
- submit updates with `submitLongTermBids` or `submitDayAheadOrIntraDayAuctionBids`

A) submitting bids over Platform API

- when using Platform API endpoints `submitLongTermBids` or `submitDayAheadOrIntraDayAuctionBids` all bids always have to be submitted in the `bids` array
- **adding a new bid:**
  - add a new entry to the `bids` array with `id: null`
- **updating or keeping an existing bid:**
  - include the entry in the bids array with its previous `id: <uuid>`
- **deleting a bid:**
  - remove the entry from the `bids` array
- **shortcut:** if you want to completely replace the list of bids just submit a “new” `bids` array with every item having `id: null`

4. sample JSON request

postman collection

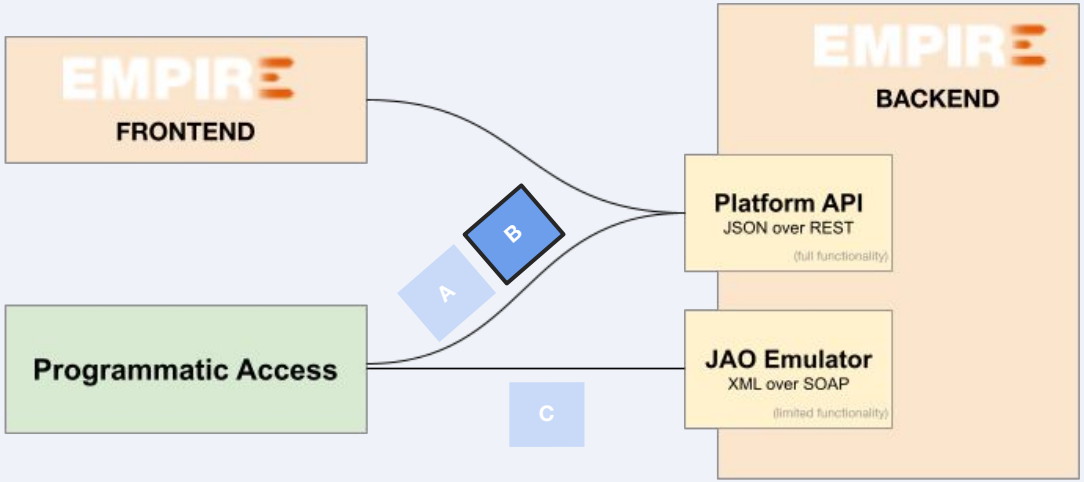


- Postman is a tool that enables you to run API calls from an easy to use GUI - https://www.postman.com/
- Postman is free to download & use
- collection has been created for the following Operation IDs
- available via. the BritNed website, you can download and import it
- allows you to execute these API calls at ease to test the request / responses

demo

5. manual file upload

submitting files



B) using Platform API "manual file upload"

- upload XML file with `uploadAttachments`
- submit bids with `uploadLongTermBids`or `uploadDayAheadOrIntraDayBids`
  - `BidDocument` XML schema
- submit nominations with `uploadTimescaleNominations`
  - `Schedule_MarketDocument`XML schema

B) nominations using “manual file upload”

```
curl -X 'POST' \
  'https://api.dev.empire.scapps.io/v1/attachments' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...' \
  -H 'Content-Type: multipart/form-data' \
  -F 'fileName=nominations.xml' \
  -F 'file=@nominations.xml;type=text/xml'
```

```
{
  "id": "92076318-955d-4e32-b01a-fe7587f2a83e",
  "fileName": "nominations.xml",
  "fileSize": 12228,
  "mimeType": "text/xml",
  "url": "https://britned-nonprod-dev-bucket.s3.eu-west-1.amazonaws.com/attachments/..."
}
```

step #1 - `uploadAttachment`

B) nominations using “manual file upload”

```
curl -X 'POST' \
  'https://api.dev.empire.scapps.io/v1/attachments' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...' \
  -H 'Content-Type: multipart/form-data' \
  -F 'fileName=nominations.xml' \
  -F 'file=@nominations.xml;type=text/xml'
```

```
{
  "id": "92076318-955d-4e32-b01a-fe7587f2a83e",
  "fileName": "nominations.xml",
  "fileSize": 12228,
  "mimeType": "text/xml",
  "url": "https://britned-nonprod-dev-bucket.s3.eu-west-1.amazonaws.com/attachments/..."
}
```

⬅️ make sure to use the right environment & API Key

step #1 - `uploadAttachment`

B) nominations using “manual file upload”

```
curl -X 'POST' \
  'https://api.dev.empire.scapps.io/v1/attachments' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...' \
  -H 'Content-Type: multipart/form-data' \
  -F 'fileName=nominations.xml' \
  -F 'file=@nominations.xml;type=text/xml'
```

```
{
  "id": "92076318-955d-4e32-b01a-fe7587f2a83e",
  "fileName": "nominations.xml",
  "fileSize": 12228,
  "mimeType": "text/xml",
  "url": "https://britned-nonprod-dev-bucket.s3.eu-west-1.amazonaws.com/attachments/..."
}
```

⬅️ request parameters for file upload

step #1 - `uploadAttachment`

B) nominations using “manual file upload”

```
curl -X 'POST' \
  'https://api.dev.empire.scapps.io/v1/attachments' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...' \
  -H 'Content-Type: multipart/form-data' \
  -F 'fileName=nominations.xml' \
  -F 'file=@nominations.xml;type=text/xml'

{
  "id": "92076318-955d-4e32-b01a-fe7587f2a83e",
  "fileName": "nominations.xml",
  "fileSize": 12228,
  "mimeType": "text/xml",
  "url": "https://britned-nonprod-dev-bucket.s3.eu-west-1.amazonaws.com/attachments/..."
}
```

⬅️ `fileName` can be arbitrary

step #1 - `uploadAttachment`

B) nominations using “manual file upload”

```
curl -X 'POST' \
  'https://api.dev.empire.scapps.io/v1/attachments' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...' \
  -H 'Content-Type: multipart/form-data' \
  -F 'fileName=nominations.xml' \
  -F 'file=@nominations.xml;type=text/xml'
```

⬅️ send (possible binary) file contents in `file` attribute

```
{
  "id": "92076318-955d-4e32-b01a-fe7587f2a83e",
  "fileName": "nominations.xml",
  "fileSize": 12228,
  "mimeType": "text/xml",
  "url": "https://britned-nonprod-dev-bucket.s3.eu-west-1.amazonaws.com/attachments/..."
}
```

step #1 - `uploadAttachment`

B) nominations using “manual file upload”

```
curl -X 'POST' \
  'https://api.dev.empire.scapps.io/v1/attachments' \
  -H 'accept: application/json' \
  -H 'X-API-Key: emp_lELzGKe...' \
  -H 'Content-Type: multipart/form-data' \
  -F 'fileName=nominations.xml' \
  -F 'file=@nominations.xml;type=text/xml'
```

```
{
  "id": "92076318-955d-4e32-b01a-fe7587f2a83e",
  "fileName": "nominations.xml",
  "fileSize": 12228,
  "mimeType": "text/xml",
  "url": "https://britned-nonprod-dev-bucket.s3.eu-west-1.amazonaws.com/attachments/..."
}
```

⬅️ save for the future, will be used as `attachmentId`

step #1 - `uploadAttachment`

B) nominations using “manual file upload”

```
curl -X 'POST' \
  'https://api.dev.empire.scapps.io/v1/manual-file-upload/nominations/timescale' \
  -H 'accept: */*' \
  -H 'X-API-Key: emp_lELzGKe...' \
  -H 'Content-Type: application/json' \
  -d '{
  "attachmentId": "92076318-955d-4e32-b01a-fe7587f2a83e"
}'
```

step #2 - `uploadTimescaleNominations`

B) nominations using “manual file upload”

```
curl -X 'POST' \
  'https://api.dev.empire.scapps.io/v1/manual-file-upload/nominations/timescale' \
  -H 'accept: */*' \
  -H 'X-API-Key: emp_lELzGKe...' \
  -H 'Content-Type: application/json' \
  -d '{
  "attachmentId": "92076318-955d-4e32-b01a-fe7587f2a83e"
}'
```

⬅️ request parameters for `uploadTimescaleNominations`

⬅️ `attachmentId` saved from previous step

step #2 - `uploadTimescaleNominations`

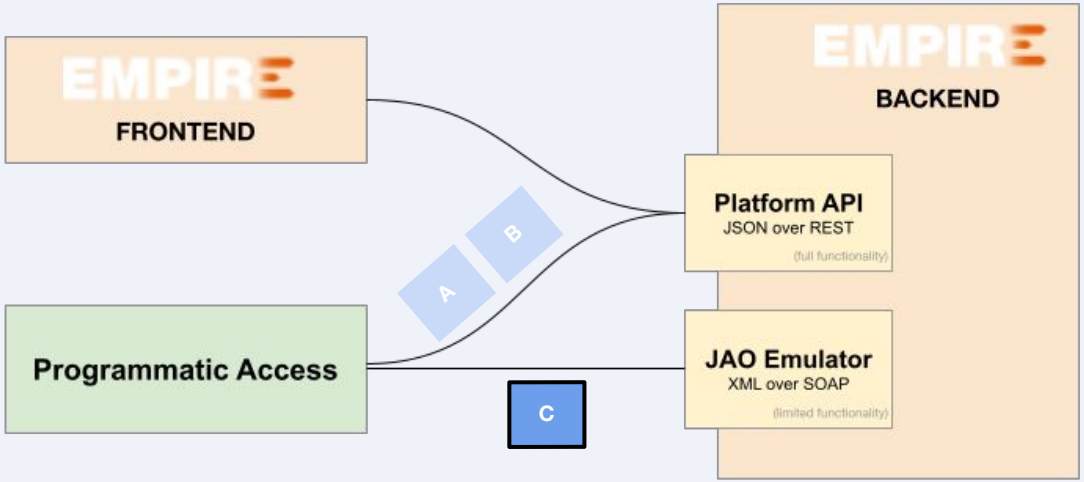
nominations using “manual file upload”

- XML validation rules as required by BritNed
  - Bids
  - Nominations
  - Validation helper Excel files will be provided
- Can be uploaded from UI or via API
- Syntax and validation rules will be familiar from JAO
  - WSDL for XML available here
- 6 sample files will be provided

demo

6.
JAO Syntax Emulator

submitting bids



C) using the JAO Emulator (XML only)

- submit LT / DA bids via SOAP API
  - `BidDocument` XML schema

JAO Syntax Emulator

- very different from JAO as there is no customer folder on an SFTP server
- we do not know the inner workings of the JAO endpoints
- received limited example files from participants
- processing of those example files have been built
- as there was no input or support we can not guarantee that all the flows and behaviours will be the same as they are on JAO
- recommendation for guaranteed success and results is to use either
  - JSON API endpoints (*option A*)
  - manual file upload (*option B*)
- both will work with example files, both will be processed, both will provide real time feedback on either successful processing or specific issues that can be rectified

JAO Syntax Emulator testing

- if a participant wishes to only use the JAO emulator
- we strongly recommend to start using it next week as part of this early Customer Integration Testing
- fixing will be at best efforts prior to go live

7. Customer Integration Testing

Customer Integration Testing

- brought forward to provide time and opportunity for testing / readiness
- prepare for a successful/smooth integration
- understand differences
- more time to ask questions & clarifications
- environment will be available & configured approx. EOD tomorrow
- https://api.training.empire.britned.com/

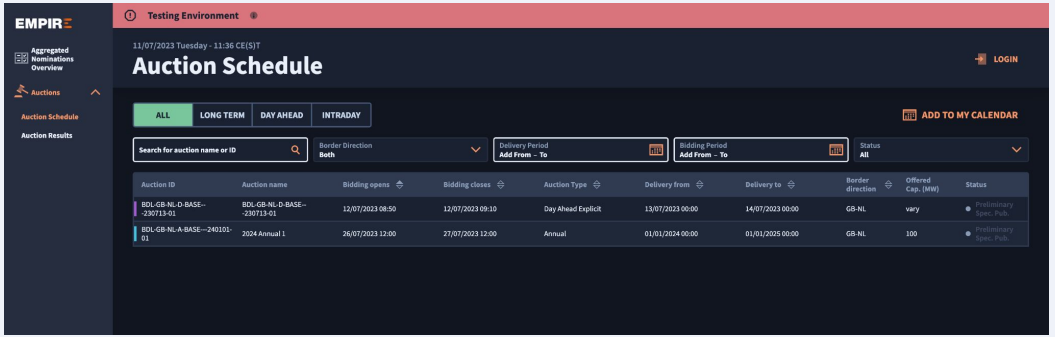
timelines & plan

- Dates: **July 10th - August 18th**
- Two messages go out to all Customers:
  - **07.10** Giving context/head's up - by Tom
  - **07.12** Call to Action message - by us (Zendesk)
- Upon request (Opt-in)
  - Two API keys will be generated for each customer - two user roles available: Participant Admin & Nominator
  - During the whole period - API documentation and test API Key

- Customer Acceptance Testing
- Starting: **August 21st**
  - full GUI access & production setup

auctions & nomination gates

- DA & ID auctions run according to schedule every day (*1 DA & 4 ID / delivery day in both directions*)
- LT auctions will be scheduled by BritNed (*at least one always running*)
- all nomination gates (LT, DA, ID) will run as normal
- enables participants to test the system in its functioning form
- public access & Auction Schedule & Details available via. GUI already: https://training.empire.britned.com/



CIT support

- **Helpdesk Support** availability
  - Monday - Friday
  - 9 am - 5 pm CEST
  - email Support
- **contact us** either:
  - replying to the initial message we've sent out
  - sending an email to:
    support@britned.zendesk.com

Customer Acceptance Testing

- Starting: **August 21st**
- Helpdesk support available:
  - 24/7 Support
  - Chat through GUI, Phone & E-mail
- Product support available
  - users perspective, user interviews, feedback gathering, etc…
- more details will follow later

SUPER
CHARGE