



MOBILNA PLATFORMA INFORMACYJNA

API coigdzie.pl

dostęp do największej bazy wydarzeń kulturalnych i rozrywkowych w Polsce

wersja 1.24

30 grudnia 2015 r.

support@coigdzie.pl

Spis treści

1 Wstęp.....	3
2 Architektura rozwiązania.....	4
3 Ogólny opis zapytań.....	5
3.1 Ograniczanie dostępu.....	5
3.2 Przykład.....	5
3.3 Stronicowanie wyników.....	6
3.4 Obrazki.....	7
3.5 Sortowanie wyników.....	7
4 Wyszukiwanie wydarzeń.....	8
4.1 Wyszukiwanie proste.....	8
4.2 Wyszukiwanie geograficzne.....	10
4.3 Pobieranie szczegółów wydarzenia.....	11
5 Sprzedaż biletów.....	12
6 Schematy XSD odpowiedzi.....	13
6.1 Lista wydarzeń.....	13
6.2 Szczegóły wydarzenia.....	14
7 Kody błędów.....	17
8 Planowany rozwój.....	19

1 Wstęp

Coigdzie.pl oferuje dwa modele współpracy w zakresie dostępu do bazy danych – poprzez XML oraz API. Model API powstał w odpowiedzi na potrzeby partnerów, dla których priorytetowe są dwa czynniki: dostęp do pełnej bazy wydarzeń oraz możliwość rozliczania się według liczby zapytań do serwera.

API umożliwia dostęp w czasie rzeczywistym do wszystkich danych o imprezach, wydarzeniach, koncertach i festiwalach znajdujących się w bazie coigdzie.pl poprzez oprogramowanie, serwis bądź aplikację mobilną partnera, a także budowę rozwiązań w oparciu o dane oraz narzędzie wyszukiwania oferowane przez API. Usługa udostępniana jest w ramach podstawowego abonamentu miesięcznego, a następnie rozliczana według faktycznego wykorzystania usługi w ramach pakietów zapytań.

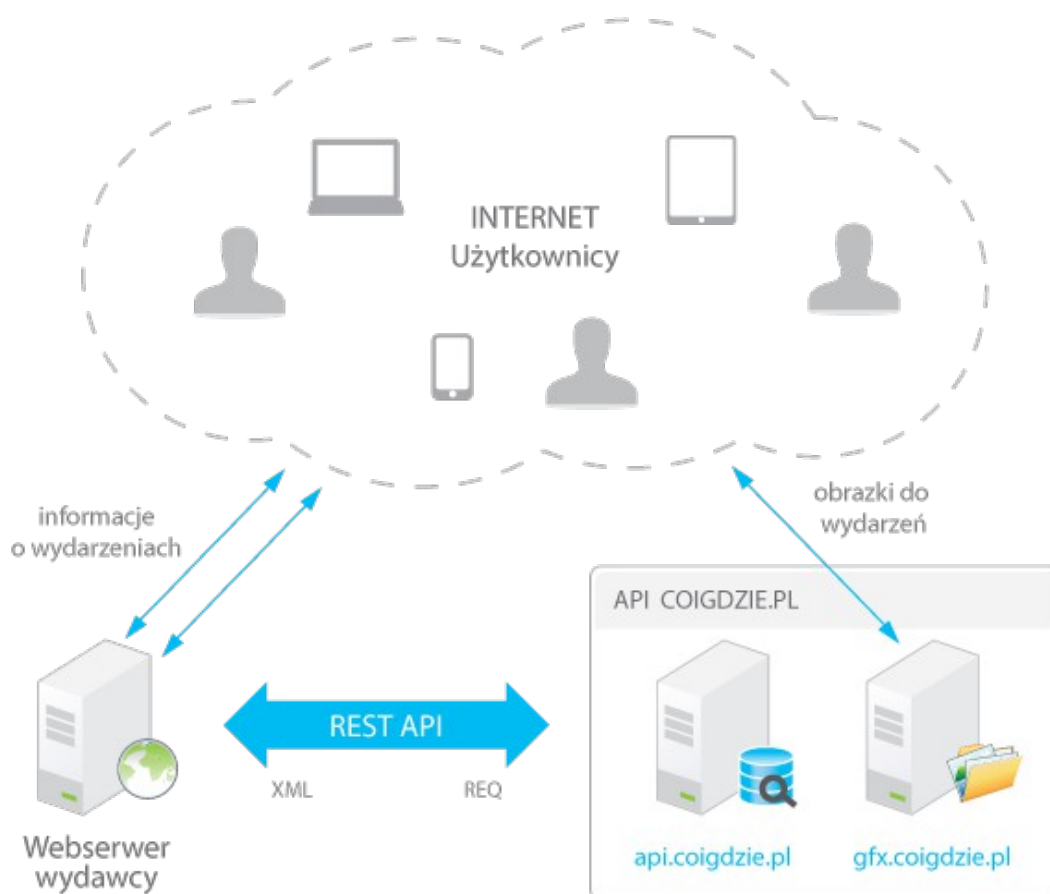
Główne korzyści API coigdzie.pl to:

- Dostęp do wszystkich baz imprez i wydarzeń coigdzie.pl w ramach podstawowego abonamentu.
- Możliwość wykorzystania gotowego, zaawansowanego narzędzia wyszukiwania.
- Skalowalny koszt dostępu do danych, zależny od ruchu na serwisie klienta oraz potrzeb użytkowników.
- Bezpośredni dostęp do aktualnych danych na serwerze coigdzie.pl.
- Wysoka jakość techniczna oferowanego rozwiązania.

Niniejszy dokument opisuje budowę i sposób działania API coigdzie.pl oraz prezentuje funkcje i przykłady jego użycia.

2 Architektura rozwiązania

API oparte jest o architekturę REST. Zapytania są kierowane do serwera jako wywołania GET protokołu HTTP, z parametrami przekazywanymi w URL-u. W odpowiedzi serwer generuje pliki XML zakodowane w UTF8. API coigdzie.pl powinno być wdrożone po stronie serwera Wydawcy – nie są akceptowane zapytania bezpośrednio od Użytkowników. Przy czym, w odpowiedziach znajdują się URL-e do obrazków na serwerze gfx.coigdzie.pl i ich to obostrzenie nie obowiązuje.



3 Ogólny opis zapytań

3.1 Ograniczanie dostępu

Dostęp do API jest ograniczony za pomocą unikalnego tokena, przydzielanego na czas trwania umowy. Przewidziany jest również testowy dostęp – tymczasowy token zostanie wygenerowany po wstępnych rozmowach. Poniżej znajduje się przykład komunikacji – wyszukanie przyszłych wydarzeń w Warszawie.

3.2 Przykład

Zapytanie

```
http://api.coigdzie.pl/happenings/search?token=123&locality=warszawa
```

Odpowiedź

```
<happenings count="3909">
<happening happeningId="2249441">
<name>T-Mobile Nowe Horyzonty Tournée 2014 - dzień 4</name>
<thumbnailImageUrl>http://gfx.coigdzie.pl/thumbnail/20140103/1388761426081.jpeg
</thumbnailImageUrl>
<startDate>2014-01-13T17:00:00Z</startDate>
<endDate>2014-01-13T22:59:00Z</endDate>
<recommended>true</recommended>
<venue venueId="8359">
<venueName>Kino Muranów</venueName>
<locality>Warszawa</locality>
<street>Gen. Władysława Andersa</street>
<houseNo>1</houseNo>
<venueGeom>
<lat>52.2447796109155</lat>
<lon>20.9998636984253</lon>
</venueGeom>
</venue>
</happening>
<happening happeningId="2242991">
```

```
<name>Koncert Kim Taylor - muzyka z filmu I USED TO BE DARKER i więcej</name>
<thumbnailImageUrl>http://gfx.coigdzie.pl/thumbnail/20131227/1388154675146.jpeg
</thumbnailImageUrl>
<startDate>2014-01-13T19:00:00Z</startDate>
<endDate>2014-01-13T22:59:00Z</endDate>
<recommended>true</recommended>
<venue venueId="1720">
<venueName>Klub Powiększenie w Warszawie</venueName>
<locality>Warszawa</locality>
<street>Nowy Świat</street>
<houseNo>27</houseNo>
<venueGeom>
<lat>52.23228</lat>
<lon>21.02025</lon>
</venueGeom>
</venue>
</happening>
</happenings>
```

3.3 Stronicowanie wyników

W głównym elemencie odpowiedzi, w tym wypadku happenings, znajduje się atrybut count, oznaczający liczbę rekordów spełniających kryteria.

Aby umożliwić stronicowanie wyników zapytania, udostępnione są 2 opcjonalne parametry:

- `offset` – numer rekordu, od którego zaczynamy pobieranie (domyślnie 0),
- `limit` – liczba pobieranych rekordów (domyślnie 10, maksymalnie 20).

Zapytanie z rozdziału 3.2. można rozbudować do poniższej postaci.

```
http://api.coigdzie.pl/happenings/search?token=123&locality=warszawa&offset=5&limit=15
```

3.4 Obrazki

W odpowiedziach znajdują się elementy zawierające linki do obrazków na serwerze gfx.coigdzie.pl.

```
<imageUrl>http://gfx.coigdzie.pl/20131227/1388154675146.jpeg</imageUrl>  
<thumbnailImageUrl>http://gfx.coigdzie.pl/thumbnail/20131227/1388154675146.jpeg  
</thumbnailImageUrl>
```

Linki prowadzą do obrazków w formacie JPEG, kodowanych w RGB. Obrazki mają wymiary przynajmniej 288x384px dla obrazka pionowego, 576x384 dla poziomego.

Miniatury są dostępne również w formacie JPEG, o domyślnej wielkości 120x120px. Zapytania obsługujące pobieranie wydarzeń, udostępniają parametr `thumbnailWidth` umożliwiający modyfikację domyślnej wielkości miniaturki.

3.5 Sortowanie wyników

Doświadczenie wielu wdrożeń nauczyło nas, że dobre posortowanie wydarzeń z naszej bazy jest zadaniem dosyć złożonym. W związku z tym, postanowiliśmy ten ciężar wziąć na siebie. Wydarzenia są posortowane według następujących kryteriów:

- polecane przez coigdzie.pl – najpierw polecane,
- wydarzenia stałe – najpierw nie-stałe (wystawy na końcu),
- czas rozpoczęcia – najpierw starsze.

4 Wyszukiwanie wydarzeń

4.1 Wyszukiwanie proste

Wyszukiwanie proste wydarzeń dostępne jest pod adresem:

<http://api.coigdzie.pl/happenings/search>

Pozwala ono na filtrowanie bazy wydarzeń używając następujących kryteriów: lokalizacja administracyjna, czas trwania, słowa kluczowe.

Parametry serwisu - search		
Nazwa	Opis	Przykład
token* ¹	Token klienta	token=123
offset	Numer rekordu, od którego rozpoczynamy pobieranie.	offset=0, offset=1000
limit	Liczba jednorazowo pobieranych rekordów (maksymalnie 20).	limit=15
locality* ²	Miejscowość, w której odbywają się wydarzenia. Nie jest wymagane rozróżnianie wielkości znaków. Można zastępować polskie znaki znakami „polskawymi” (ą na a, etc.)	locality=Wrocław, locality=wroclaw, locality=bielsko+bia a
district*	Powiat, w którym odbywa się wydarzenie. Nie jest wymagane rozróżnianie wielkości znaków. Można zastępować polskie znaki znakami „polskawymi”.	district=kłodzki, district=warszawski
voivodeship*	Województwo, w którym odbywa się wydarzenie. Nie jest wymagane rozróżnianie wielkości znaków. Można zastępować polskie znaki znakami „polskawymi”.	voivodeship=dolnoslas kie, voivodeship=kujawsko+ pomorskie
startDate	Data, po której mają się kończyć wydarzenia. Wymagany jest format yyyy-MM-dd lub słowo now. Domyślnie ma wartość now, przy czym nie może przyjmować daty wcześniejszej niż dzisiaj.	startDate=2014-07-01

¹ Znak asterisk (*) oznacza, że parametr jest wymagany.

² Tylko jeden z parametrów locality, district lub voivodeship powinien być wypełniony.

endDate	Data, przed którą mają się zaczynać wydarzenia. Wymagany jest format yyyy-MM-dd. Domyślnie ma nieskończoną. Przy czym, nie może przyjmować daty wcześniejszej niż startDate.	endDate=2014-07-01
keyword	Słowa kluczowe	keyword=muzyka, keyword=sport, keyword=dzieci, keyword=jazz, keyword=wystawa+stala
thumbnailWidth	Szerokość miniaturki obrazka. Akceptowalne wartości od 100 do 999 pikseli.	thumbnailWidth=250

Poniżej przykładowe zapytania.

1. Przyszłe wydarzenia w Warszawie

<http://api.coigdzie.pl/happenings/search?token=123&locality=warszawa>

2. Przyszłe koncerty w Warszawie

<http://api.coigdzie.pl/happenings/search?token=123&locality=warszawa&keyword=koncert>

3. Koncerty w Warszawie, kończące się po 1 maja 2014

<http://api.coigdzie.pl/happenings/search?token=123&locality=warszawa&keyword=koncert&startDate=2014-05-01>

4.2 Wyszukiwanie geograficzne

Wyszukiwanie geograficzne wydarzeń dostępne jest pod adresem:

<http://api.coigdzie.pl/happenings/search-geo>

Pozwala ono na filtrowanie bazy wydarzeń używając następujących kryteriów: współrzędne geograficzne, promień poszukiwań, czas trwania, słowa kluczowe.

Parametry serwisu - search-geo		
Nazwa	Opis	Przykład
token*	Token klienta	token=123
offset	Numer rekordu, od którego rozpoczynamy pobieranie.	offset=0, offset=1000
limit	Liczba jednorazowo pobieranych rekordów (maksymalnie 20).	limit=15
lat*	Szerokość geograficzna w formacie dd.dddd ³	lat=52.12345
lon*	Długość geograficzna w formacie dd.dddd	lon=16.22234
radius*	Promień poszukiwań w kilometrach. Maksymalna wartość to 100km.	radius=40
startDate	Data, po której mają się kończyć wydarzenia. Wymagany jest format yyyy-MM-dd lub słowo now. Domyślnie ma wartość now, przy czym nie może przyjmować daty wcześniejszej niż dzisiaj.	startDate=2014-07-01
endDate	Data, przed którą mają się zaczynać wydarzenia. Wymagany jest format yyyy-MM-dd. Domyślnie ma nieskończoną. Przy czym, nie może przyjmować daty wcześniejszej niż startDate.	endDate=2014-07-01
keyword	Słowa kluczowe	keyword=koncert, keyword=teatr, keyword=festiwal, keyword=rock, keyword=kazik+koncert

3 Maksymalnie 6 miejsc po przecinku

thumbnailWidth	Szerokość miniaturki obrazka. Akceptowalne wartości od 100 do 999 pikseli.	thumbnailWidth=250
----------------	--	--------------------

Poniżej przykładowe zapytania.

1. Przyszłe wydarzenia w promieniu 40 km od centrum Warszawy

[http://api.coigdzie.pl/happenings/search-geo?
token=123&lat=52.236631&lon=21.009865&radius=40](http://api.coigdzie.pl/happenings/search-geo?token=123&lat=52.236631&lon=21.009865&radius=40)

2. Przyszłe koncerty w promieniu 40 km od centrum Warszawy

[http://api.coigdzie.pl/happenings/search-geo?
token=123&lat=52.236631&lon=21.009865&radius=40&keyword=koncert](http://api.coigdzie.pl/happenings/search-geo?token=123&lat=52.236631&lon=21.009865&radius=40&keyword=koncert)

3. Koncerty w promieniu 40 km od centrum Warszawy, kończące się po 1 maja 2014

[http://api.coigdzie.pl/happenings/search-geo?
token=123&lat=52.236631&lon=21.009865&radius=40&keyword=koncert&startDate=2014-
05-01](http://api.coigdzie.pl/happenings/search-geo?token=123&lat=52.236631&lon=21.009865&radius=40&keyword=koncert&startDate=2014-05-01)

4.3 Pobieranie szczegółów wydarzenia

Pobieranie szczegółów dostępne jest pod adresem:

<http://api.coigdzie.pl/happenings/details>

Pozwala na pobranie szczegółów wydarzenia, poprzez jego identyfikator.

Parametry serwisu - details		
Nazwa	Opis	Przykład
token*	Token klienta	token=123
happeningId*	Identyfikator wydarzenia	happeningId=20040635
thumbnailWidth	Szerokość miniaturki obrazka. Akceptowalne wartości od 100 do 999 pikseli.	thumbnailWidth=250

Poniżej przykładowe zapytanie.

Pobranie szczegółów wydarzenia o identyfikatorze 20040635

<http://api.coigdzie.pl/happenings/details?token=123&happeningId=20040635>

5 Sprzedaż biletów

API udostępnia mechanizmy umożliwiające zakup biletów na wydarzenia za pośrednictwem wiodących platform biletowych. Domyślnie mechanizmy te są wyłączone, aby je włączyć, należy w zapytaniach wykorzystywanych przez serwisy (wyszukiwania prostego, wyszukiwania geograficznego i pobierania szczegółów wydarzenia) ustawić parametr `tickets=1`.

Pobranie szczegółów wydarzenia o identyfikatorze 20048460 oraz włączenie mechanizmów umożliwiających zakup biletów

<http://api.coigdzie.pl/happenings/details?token=123&happeningId=20048460&tickets=1>

W zwracanych XML-ach, dla wydarzeń dla których możliwa jest internetowa sprzedaż biletów, obiekt `happening` będzie wzbogacony o element `ticketingSystems`.

Przykładowy element `ticketingSystems`

```
<ticketingSystems>
  <ticketingSystem name="KupBilecik.pl" priority="1"
    logo="http://gfx.coigdzie.pl/ticketing/kupbilecik/logo.png">
    <ticketUrl>http://www.kupbilecik.pl/impreza-2971-cezik,i.klejnuty-63.html</ticketUrl>
  </ticketingSystem>
  <ticketingSystem name="eBilet Sp. z o.o." priority="3"
    logo="http://gfx.coigdzie.pl/ticketing/ebilet/logo.png">
    <ticketUrl>https://coigdzie.ebilet.pl/bilety.php?id=84181</ticketUrl>
  </ticketingSystem>
</ticketingSystems>
```

Platforma opisywana jest poprzez `ticketingSystem`. Kluczowym jest tutaj znacznik `ticketUrl`, jest to adres pozwalający rozpocząć proces zakupu. Budowa tego adresu, pozwala platformie biletowej rozpoznać skąd przyszedł kupujący i odpowiednio podzielić zysk ze sprzedaży.

Możliwym jest, że na jedno wydarzenie bilety są dystrybuowane za pośrednictwem kilku platform biletowych. W powyższym przypadku bilety dystrybuują dwie platformy. Sugerowanym rozwiązaniem jest umożliwienie Użytkownikowi wyboru, z usług której platformy chce skorzystać. Wybór taki można zrealizować, na przykład, poprzez okienko typu popup. Aby ułatwić implementację takiego mechanizmu, dodatkowo dostępne są:

- `name` – oficjalna nazwa platformy,
- `logo` – logo platformy w formacie png, stałej wysokości 50px, oraz szerokości od 100px do 200px,
- `priority` – wartość, po której powinny zostać posortowane (rosnąco) platformy biletowe, jeżeli Użytkownik skorzysta z usługi platformy o najmniejszej wartości `priority`, to zysk ze sprzedaży będzie największy.

6 Schematy XSD odpowiedzi

6.1 Lista wydarzeń

Serwisy search oraz search-geo zwracają XML-e posiadające tylko skrótowe informacje o wydarzeniach. Poniżej ich schemat XSD.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="happenings">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="happening" maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="count" use="required" type="xs:integer"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="happening">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="imageUrl" minOccurs="0" type="xs:anyURI"/>
        <xs:element name="thumbnailImageUrl" minOccurs="0" type="xs:anyURI"/>
        <xs:element name="startDate" type="xs:dateTime"/>
        <xs:element name="endDate" type="xs:dateTime"/>
        <!-- godziny rozpoczęcia i zakończenia wydarzenia nie są znane -->
        <!-- data jest podana w formacie "YYYY-mm-DDTHH:MM:SSZ". Data jest
godziny rozpoczęcia i zakończenia wydarzenia nie są znane -->
        <!-- data jest podana w formacie "YYYY-mm-DDTHH:MM:SSZ". Data jest
podana dla strefy czasowej UTC -->
        <xs:element name="recommended" type="xs:boolean"/>
        <xs:element ref="ticketingSystems" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="venue"/>
      </xs:sequence>
      <xs:attribute name="happeningId" use="required" type="xs:integer"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="venue">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="venueName" minOccurs="0" type="xs:string"/>
        <xs:element name="locality" type="xs:string"/>
        <xs:element name="street" minOccurs="0" type="xs:string"/>
        <xs:element name="houseNo" minOccurs="0" type="xs:string"/>
        <xs:element ref="venueGeom"/>
      </xs:sequence>
      <xs:attribute name="venueId" use="optional" type="xs:integer"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="venueGeom">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="lat" type="xs:double"/>
        <xs:element name="lon" type="xs:double"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

```

        <xs:element name="ticketingSystems">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="ticketingSystem" minOccurs="1"
maxOccurs="unbounded">
                    </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ticketingSystem">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="ticketUrl" type="xs:anyURI"/>
            </xs:sequence>
            <xs:attribute name="name" type="xs:string" use="required" />
            <xs:attribute name="logo" type="xs:anyURI" use="required" />
            <xs:attribute name="priority" type="xs:int" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>

```

6.2 Szczegóły wydarzenia

Serwis details zwraca szczegółowe informacje o wydarzeniu.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="happenings">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="happening" maxOccurs="1" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="count" use="required" type="xs:integer"/>
        </xs:complexType>
    </xs:element>

    <xs:element name="happening">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="name" type="xs:string"/>
                <xs:element name="categoryName" type="eventCategoryName"/>
                <xs:element name="description" minOccurs="0" type="xs:string"/>
                <xs:element name="happeningUrl" minOccurs="0" type="xs:anyURI"/>
                <xs:element name="otherHappeningUrl" minOccurs="0" type="xs:anyURI"/>
                <xs:element name="imageUrl" minOccurs="0" type="xs:anyURI"/>
                <xs:element name="thumbnailImageUrl" minOccurs="0" type="xs:anyURI"/>
                <xs:element name="recommended" type="xs:boolean"/>
                <xs:element name="price" type="xs:integer"/>
                <!-- "-1" oznacza brak danych, 0 bezpłatna, inna liczba cenę biletu
-->
                <xs:element name="startDate" type="xs:dateTime"/>
                <xs:element name="endDate" type="xs:dateTime"/>
                <!-- godziny 00:00 dla startDate i 23:59 dla endDate oznaczają, że
godziny rozpoczęcia i zakończenia wydarzenia nie są znane -->
                <!-- data jest podana w formacie "YYYY-mm-DDTHH:MM:SSZ". Data jest
podana dla strefy czasowej UTC -->
                <xs:element ref="ticketingSystems" minOccurs="0" maxOccurs="1"/>
                <xs:element ref="venue"/>
                <xs:element ref="organizer" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="happeningId" use="required" type="xs:integer"/>
        </xs:complexType>

```

```

</xs:element>

<xs:element name="venue">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="venueName" minOccurs="0" type="xs:string"/>
      <xs:element name="locality" type="xs:string"/>
      <xs:element name="street" minOccurs="0" type="xs:string"/>
      <xs:element name="houseNo" minOccurs="0" type="xs:string"/>
      <xs:element name="postCode" minOccurs="0" type="xs:string"/>
      <xs:element name="telephone" minOccurs="0" type="xs:string"/>
      <xs:element name="venueUrl" minOccurs="0" type="xs:anyURI"/>
      <xs:element name="venueEmail" minOccurs="0" type="xs:string"/>
      <xs:element name="venueImageUrl" minOccurs="0" type="xs:anyURI"/>
      <xs:element ref="venueGeom"/>
    </xs:sequence>
    <xs:attribute name="venueId" use="optional" type="xs:integer"/>
  </xs:complexType>
</xs:element>
<xs:element name="venueGeom">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="lat" type="xs:double"/>
      <xs:element name="lon" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

  <xs:element name="ticketingSystems">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ticketingSystem" minOccurs="1"
maxOccurs="unbounded">
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ticketingSystem">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ticketUrl" type="xs:anyURI"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required" />
        <xs:attribute name="logo" type="xs:anyURI" use="required" />
        <xs:attribute name="priority" type="xs:int" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="organizer">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="organizerName" type="xs:string"/>
          <xs:element name="organizerLocality" type="xs:string"/>
          <xs:element name="organizerStreet" type="xs:string"/>
          <xs:element name="organizerHouseNo" type="xs:string"/>
          <xs:element name="organizerPostCode" type="xs:string"/>
          <xs:element name="organizerTelephone" type="xs:string"/>
          <xs:element name="organizerUrl" minOccurs="0" type="xs:anyURI"/>
          <xs:element name="organizerImageUrl" minOccurs="0" type="xs:anyURI"/>
        </xs:sequence>
        <xs:attribute name="organizerId" use="required" type="xs:integer"/>
      </xs:complexType>
    </xs:element>

    <xs:simpleType name="eventCategoryName">
      <xs:restriction base="xs:string">
        <xs:enumeration value="gospodarka i nauka/ekologia"/>
        <xs:enumeration value="gospodarka i nauka/ekonomia"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

```

```

<xs:enumeration value="gospodarka i nauka/konferencja"/>
<xs:enumeration value="gospodarka i nauka/targi"/>
<xs:enumeration value="gospodarka i nauka/technika"/>
<xs:enumeration value="inne inne"/>
<xs:enumeration value="inne kulinarna"/>
<xs:enumeration value="inne zdrowie"/>
<xs:enumeration value="kultura dla dzieci"/>
<xs:enumeration value="kultura edukacja"/>
<xs:enumeration value="kultura film"/>
<xs:enumeration value="kultura fotografia"/>
<xs:enumeration value="kultura historia"/>
<xs:enumeration value="kultura inne"/>
<xs:enumeration value="kultura jubileusz"/>
<xs:enumeration value="kultura kabaret"/>
<xs:enumeration value="kultura koncert"/>
<xs:enumeration value="kultura konkurs"/>
<xs:enumeration value="kultura literatura"/>
<xs:enumeration value="kultura ludowa"/>
<xs:enumeration value="kultura malarstwo"/>
<xs:enumeration value="kultura muzyka"/>
<xs:enumeration value="kultura opera"/>
<xs:enumeration value="kultura religia"/>
<xs:enumeration value="kultura rozrywka"/>
<xs:enumeration value="kultura spotkanie"/>
<xs:enumeration value="kultura taniec"/>
<xs:enumeration value="kultura teatr"/>
<xs:enumeration value="kultura warsztaty"/>
<xs:enumeration value="kultura wernisaż"/>
<xs:enumeration value="kultura widowisko"/>
<xs:enumeration value="kultura wykład"/>
<xs:enumeration value="kultura wystawa"/>
<xs:enumeration value="motoryzacja inne"/>
<xs:enumeration value="motoryzacja rajd"/>
<xs:enumeration value="motoryzacja wyścigi"/>
<xs:enumeration value="sport badminton"/>
<xs:enumeration value="sport biegi"/>
<xs:enumeration value="sport dla dzieci"/>
<xs:enumeration value="sport gry planszowe"/>
<xs:enumeration value="sport hokej"/>
<xs:enumeration value="sport inne"/>
<xs:enumeration value="sport kajaki"/>
<xs:enumeration value="sport karciane"/>
<xs:enumeration value="sport kolarstwo"/>
<xs:enumeration value="sport koszykówka"/>
<xs:enumeration value="sport lekkoatletyka"/>
<xs:enumeration value="sport łyżwiarstwo"/>
<xs:enumeration value="sport narty"/>
<xs:enumeration value="sport niepełnosprawni"/>
<xs:enumeration value="sport piłka nożna"/>
<xs:enumeration value="sport piłka ręczna"/>
<xs:enumeration value="sport piłka siatkowa"/>
<xs:enumeration value="sport pływanie"/>
<xs:enumeration value="sport samolotowy"/>
<xs:enumeration value="sport snowboard"/>
<xs:enumeration value="sport strzelectwo sportowe"/>
<xs:enumeration value="sport szachy"/>
<xs:enumeration value="sport sztuki walki"/>
<xs:enumeration value="sport tenis"/>
<xs:enumeration value="sport tenis stołowy"/>
<xs:enumeration value="sport turystyka i rekreacja"/>
<xs:enumeration value="sport żeglarsstwo"/>
</xs:restriction>
</xs:simpleType>

```

</xs:schema>

7 Kody błędów

Kod HTTP	Treść błędu	Opis
400	Parameter token can not be empty	Pusta wartość tokena.
403	Permission denied, for token:	Błędna wartość tokena.
500	Internal CURL error:	Wewnętrzny błąd serwisu.
500	Internal server error	Wewnętrzny błąd serwisu.
400	Parameter happeningId can not be empty	Niewypełniony parametr happeningId podczas pobierania szczegółów wydarzenia.
400	Parameter happeningId should be a number	Parametr happeningId powinien być liczbą.
400	Parameter lat can not be empty	Niewypełniona szerokość geograficzna podczas wyszukiwania geometrycznego.
400	Invalid latitude format, should look like: 52.112	Niepoprawnie sformatowana szerokość geograficzna.
400	Parameter lon can not be empty	Niewypełniona długość geograficzna podczas wyszukiwania geometrycznego.
400	Invalid longitude format, should look like: 17.112	Niepoprawnie sformatowana długość geograficzna podczas wyszukiwania geometrycznego.
400	Parameter radius can not be empty	Niewypełniony promień podczas wyszukiwania geometrycznego.
400	Radius should be a number	Promień musi być liczbą
400	Radius shouldn't be more than 100	Promień nie może być większy niż 100km podczas wyszukiwania geometrycznego.
400	Limit should be a number	Limit musi być liczbą
400	Limit shouldn't be more than 20	Liczba rekordów zwracanych jednorazowo przez serwer nie może być większa niż 20.
400	Offset is not a number	Parametr offset musi być liczbą.
400	Incorrect thumbnailWidth (100-999 pix supported)	Niepoprawna szerokość miniaturki – akceptowalne są tylko liczby 3 cyfrowe.

Kod HTTP	Treść błędu	Opis
400	StartDate should be in format yyyy-MM-dd or now	Niepoprawny format daty startDate.
400	StartDate shouldn't be before midnight	Data startDate nie może być wcześniejsza niż „dziś”.
400	EndDate should be in format yyyy-MM-dd	Niepoprawny format daty endDate.
400	EndDate should be after startDate	Data endDate powinna być późniejsza niż startDate.
400	One of parameters: locality, voivodeship or district should be filled	Żaden z parametrów: locality, voivodeship lub district nie został wypełniony.
400	Only one of parameters: locality, voivodeship or district should be filled	Tylko jeden z parametrów locality, voivodeship lub district może zostać wypełniony.

8 Planowany rozwój

W przyszłych wersjach API, planujemy umożliwić:

- przeszukiwanie repertuaru kin,
- autouzupełnianie nazw miejscowości,
- przeszukiwanie repertuaru teatrów,
- zaawansowane wyszukiwanie koncertów.

Czekamy również na Państwa propozycje.