

CONTRACT MODEL

IPONZ RESTFUL SERVICES VERSION 1

Author: Foster Moore

Date: 30 September 2011

Document Version: 1.7

FOSTER MOORE
the registry people



CONTENTS

BusinessData Authentication	3
Introduction	3
HMAC (Hashed Message Authentication Code)	3
Signature Generation	3
Request/Response Standards	6
RESTful Request Structure	6
Trademark RESTful Services	7
Services	7
XML Schema Definitions	8
iponz_trademark_information_v1.02.xsd	8
Design RESTful Services	14
Services	14
XML Schema Definitions	15
iponz_design_information_v1.00.xsd	15
Patent RESTful Services	21
Services	21
XML Schema Definitions	22
iponz_patent_information_v1.00.xsd	22

BUSINESSDATA AUTHENTICATION

Introduction

BusinessData is the authentication service layer through which all business services are exposed. Each business service transaction is first authenticated in BusinessData before forwarding the transaction to the appropriate business service for processing.

HMAC (Hashed Message Authentication Code)

HMAC is the authentication mechanism used by BusinessData. A shared **Secret Authentication Key** is used by the client to create a signature based upon predefined message properties. Upon receiving the signature with the request the service will recalculate the signature to authenticate the user's identity.

Each request should provide the signature and the data required as part of the source.

Two pieces of information are required to generate a signature.

- Authentication Key ID - This is your unique identifier to verify that you are a BusinessData user. This will not change.
- Secret Authentication Key - Used to generate a signature from the "String to Sign". This Secret Authentication Key is never transmitted in a message and should never be passed to a client machine.

Signature Generation

The "String to Sign" is the final string which is signed using a SHA256 digest to create the signature. It is built up from the following attributes of an HTTP request.

ATTRIBUTE	DESCRIPTION	REQUIRED/ OPTIONAL	SUPPLIED AS
HTTP Method	HTTP method used for the request. Only the GET method is allowed for this service. (Note: All RESTful web services in BusinessData support only the GET method.)	Required	
Host	Standard HTTP header.	Required	
Request URI	URI of service being requested.	Required	
Timestamp	Time that the signature was generated.	Required	Request header in the format yyyyMMddHHmmss z
Authentication Key ID	Identifies the BusinessData client.	Required	As part of the request "Authorization" header.

Context	A client identifier for the transaction. If supplied this must be unique to each service usage. (i.e., user and service combination).	Optional	Request header.
---------	---	----------	-----------------

The "String to Sign" is built up from these values separated by a new line character ('\n'). The construction of the String to Sign follows the template shown below:

```
<HTTP Method>\n<Host>\n<Request URI>\n<Timestamp>\n<Authentication Key ID>\n<Context>\n
```

Example Source String

```
GET
www.businessdata.govt.nz
/data/app/ws/rest/1.0/trademark/701802
20091221084626 NZDT
T4WZUVoufkaGfsrK
CTX001
```

A Base64 encoded SHA256 hash is then applied to this string using your Secret Authentication Key (not the Authentication Key ID) to produce the signature. An example signature may look like the following:

```
wFtZpfdDAh9kul6ZEnlvvmRYdWmvqjLxagNUcpMYGTM=
```

This signature must be added to the request as a SOAP header in the form <Authentication Key ID>:<Signature> :

```
Authorization:"T4WZUVoufkaGfsrK:4yDg0FE8nzqhY80K+5w1sHXVO+6UBXeMG4irvoUx6jI="
```

A request header would look as follows:

```
Authorization:"T4WZUVoufkaGfsrK:4yDg0FE8nzqhY80K+5w1sHXVO+6UBXeMG4irvoUx6jI=",Time
stamp:"20101202082124 NZDT",User-Agent:"Jakarta Commons-
HttpClient/3.0.1",Host:"www.businessdata.govt.nz"
```

If no context element is supplied the signature will only be valid for 20 seconds after its generation time. The timestamp used for generation on the server must match that to be supplied in the request header. It is recommended that the server be synchronized with a NTP server.

RESTFUL SERVICE	SERVICE URI EXAMPLE
IPONZ List Trade Marks Filed	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/trademarks/iled/20100101-20101231
IPONZ List Trade Marks	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/trademarks/

Updated	updated/20100101-20101231
IPONZ Get Trade Mark Information	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/trademark/701802
IPONZ Get Trade Mark Object	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/trademark/701802/image.1
IPONZ List Designs Registered	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/design/registered/20000101-20071231
IPONZ List Designs Updated	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/design/updated/20000101-20071231
IPONZ Get Design Information	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/design/25260
IPONZ Get Design Object	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/design/25260/representation.1
IPONZ List Patents Updated	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/patent/updated/20000101-20071231
IPONZ Get Patent Information	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/patent/263821
IPONZ Get Patent Object	http://www.businessdata.govt.nz/data/app/ws/rest/1.0/patent/263121/abstract.1

REQUEST/RESPONSE STANDARDS

RESTful Request Structure

All the services described in this document are RESTful web services. Each transaction must contain the BusinessData client's authentication data and the input data required by the business service.

- Authentication Data – The client's authentication data must be supplied in the request header.
- Input Data – The service specific input data is embedded in the request URL.

TRADEMARK RESTFUL SERVICES

Services

There are four RESTful Trademark services as follows:

SERVICE	DESCRIPTION	PARAMETER(S)
IPONZ List Trade Marks Filed	To retrieve a list of Trade Marks filed within a given date range.	<ul style="list-style-type: none">Date range <p><u>Example:</u> trademarks/filed/20100101-20101231</p> <p><u>Date range validation rules:</u></p> <ol style="list-style-type: none">Must be in YYYYMMDD-YYYYMMDD format.The 'to date' must not be before the 'from date'.The date range must not be more than 365 days.
IPONZ List Trade Marks Updated	To retrieve a list of Trade Marks that have been updated within a given date range.	<ul style="list-style-type: none">Date range <p><u>Example:</u> trademarks/updated/20100101-20101231</p> <p><u>Date range validation rules:</u></p> <ol style="list-style-type: none">Must be in YYYYMMDD-YYYYMMDD format.The 'to date' must not be before the 'from date'.The date range must not be more than 365 days.The 'from date' must not be before 20100101 (i.e., Jan. 1, 2010)
IPONZ Get Trade Mark Information	To retrieve the details of a given Trade Mark.	<ul style="list-style-type: none">Trade Mark IP number <p><u>Example:</u> trademark/701802</p>
IPONZ Get Trade Mark Object	To retrieve an object of a specific type associated with a given Trade Mark. The object is returned as Base64-encoded binary data.	<ul style="list-style-type: none">Trade Mark IP numberObject type and index <p><u>Example:</u> trademark/701802/image.1</p>

		<p>The first parameter is the Trade Mark IP number for which the object needs to be returned.</p> <p>The second parameter is the object type and index. Valid object types for trademarks are image and sound.</p> <p>The .1 is the index of the object to be returned. So image.1 will return the first image of that IP. If the index is not supplied then the default is 1. So the path trademark/701802/image is equivalent to trademark/701802/image.1</p> <p>Note: The Get Trade Mark Information response will return a list of all objects that can be retrieved for a given IP.</p>
--	--	---

XML Schema Definitions

The following XSD file contains the schema definitions that form the basis of the response for all the Trademark RESTful services. These schema definitions conform to the WIPO standards. Note: This XSD file is published on the BusinessData production website.

iponz_trademark_information_v1.02.xsd

URL: http://www.businessdata.govt.nz/data/app/ws/iponz_trademark_information_v1.02.xsd

```
<?xml version="1.0" encoding="utf-8"?>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.iponz.govt.nz/XMLSchema/trademarks/information"
    xmlns:iponz="http://www.iponz.govt.nz/XMLSchema/trademarks"

    targetNamespace="http://www.iponz.govt.nz/XMLSchema/trademarks/information"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.02">

    <!--<![CDATA[
*****
*
*      IPONZ Trade Mark Information Schema
*
*      Filename      : iponz_trademark_information_v1.02.xsd
*      Author       : IPONZ
*****
```



```

*      Version      : 1.02                                     *
*      Last Update  : 2010-05-23                               *
*      Contact      : www.iponz.govt.nz                         *
*                                                           *
*****
]]>-->

<xs:annotation>
  <xs:documentation>IPONZ Trade Mark Information Schema </xs:documentation>
</xs:annotation>

<!--<![CDATA[
=====

Import IPONZ Trade Mark Common Types Library
=====

]]>-->

<xs:import namespace="http://www.iponz.govt.nz/XMLSchema/trademarks"
schemaLocation="common/iponz_trademark_common_types_v1.09.xsd" />

<!--<![CDATA[
=====

List Trade Marks Filed Transaction Response Root Element
=====

]]>-->

<xs:element name="TradeMarksFiledResponse"
type="TradeMarksFiledResponseType"/>

<!--<![CDATA[
=====

List Trade Marks Updated Transaction Response Root Element
=====

]]>-->

<xs:element name="TradeMarksUpdatedResponse"
type="TradeMarksUpdatedResponseType"/>

<!--<![CDATA[
=====

Get Trade Mark Information Transaction Response Root Element

```

```

=====
]]>-->

<xs:element name="TradeMarkInformationResponse"
type="TradeMarkInformationResponseType"/>

<!--<![CDATA[
=====
Get Trade Mark Object Transaction Response Root Element
=====
]]>-->

<xs:element name="TradeMarkObjectResponse"
type="TradeMarkObjectResponseType"/>

<!--<![CDATA[
*****
* TRANSACTION/MESSAGE TYPES (in alphabetical order by name)      *
*****
]]>-->

<xs:complexType name="TradeMarkInformationResponseType">
  <xs:sequence>
    <xs:element name="TransactionDateTime" type="xs:dateTime" />
    <xs:choice>
      <xs:element name="TransactionError" type="TransactionErrorType" />
      <xs:element name="TradeMarkInformation"
type="iponz:NZ_TradeMarkInformationType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="TradeMarkObjectResponseType">
  <xs:sequence>
    <xs:element name="TransactionDateTime" type="xs:dateTime" />
    <xs:choice>
      <xs:element name="TransactionError" type="TransactionErrorType" />
      <xs:element name="TradeMarkObject" type="TradeMarkObjectType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="TradeMarkObjectType">
  <xs:sequence>
    <xs:element name="ApplicationNumber" type="xs:positiveInteger" />
    <xs:element name="ObjectKey" type="xs:token" />
    <xs:element name="ObjectFilename" type="xs:token" />
    <xs:element name="ObjectFormat" type="iponz:FileFormatType" />
    <xs:element name="ObjectBinary" type="xs:base64Binary" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="TradeMarksFiledResponseType">
  <xs:sequence>
    <xs:element name="TransactionDateTime" type="xs:dateTime" />
    <xs:choice>
      <xs:element name="TransactionError" type="TransactionErrorType" />
      <xs:element name="TradeMarksFiled" type="TradeMarksFiledType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="TradeMarksUpdatedResponseType">
  <xs:sequence>
    <xs:element name="TransactionDateTime" type="xs:dateTime" />
    <xs:choice>
      <xs:element name="TransactionError" type="TransactionErrorType" />
      <xs:element name="TradeMarksUpdated" type="TradeMarksUpdatedType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="TradeMarksFiledType">
  <xs:sequence>
    <xs:element name="FilingDateRange">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="FromDate" type="xs:date" />
          <xs:element name="ToDate" type="xs:date" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

</xs:element>

<xs:element name="TradeMarks">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="TradeMark" type="TradeMarkSummaryType"
minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="TradeMarksUpdatedType">
  <xs:sequence>
    <xs:element name="UpdatedDateRange">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="FromDate" type="xs:date" />
          <xs:element name="ToDate" type="xs:date" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="TradeMarks">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="TradeMark" type="TradeMarkSummaryType"
minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="TradeMarkSummaryType">
  <xs:sequence>
    <xs:element name="ApplicationNumber" type="xs:positiveInteger" />
    <xs:element name="MarkCurrentStatusCode" type="xs:token" />
    <xs:element name="FilingDate" type="xs:date" />
  </xs:sequence>
</xs:complexType>

```

```
<xs:complexType name="TransactionErrorType">
  <xs:sequence>
    <xs:element name="TransactionErrorCode" type="xs:token" />
    <xs:element name="TransactionErrorText" type="iponz:TextType" />
  </xs:sequence>
</xs:complexType>

</xs:schema>
```

DESIGN RESTFUL SERVICES

Services

There are four RESTful Design services as follows:

SERVICE	DESCRIPTION	PARAMETERS
IPONZ List Designs Registered	To retrieve a list of Designs registered within a given date range.	<ul style="list-style-type: none">Date range <p><u>Example:</u> designs/registered/20100101-20101231</p> <p><u>Date range validation rules:</u></p> <ol style="list-style-type: none">Must be in YYYYMMDD-YYYYMMDD format.The 'to date' must not be before the 'from date'.The date range must not be more than 365 days.
IPONZ List Designs Updated	To retrieve a list of Designs that have been updated within a given date range.	<ul style="list-style-type: none">Date range <p><u>Example:</u> designs/updated/20100101-20101231</p> <p><u>Date range validation rules:</u></p> <ol style="list-style-type: none">Must be in YYYYMMDD-YYYYMMDD format.The 'to date' must not be before the 'from date'.The date range must not be more than 365 days.The 'from date' must not be before 20100101 (i.e., Jan. 1, 2010)
IPONZ Get Design Information	To retrieve the details of a given Design.	<ul style="list-style-type: none">Design IP number <p><u>Example:</u> design/701802</p>
IPONZ Get Design Object	To retrieve an object of a specific type associated with a given Design. The object is returned as Base64-encoded binary data.	<ul style="list-style-type: none">Design IP numberObject type, article index, and object index within the article <p><u>Example:</u> design/701802/representation.1.1</p>


```

*
*****

]]>-->

<xs:annotation>
  <xs:documentation>IPONZ Design Information Schema </xs:documentation>
</xs:annotation>

<!--<![CDATA[
=====
Import IPONZ Design Common Types Library
=====
]]>-->

<xs:import namespace="http://www.iponz.govt.nz/XMLSchema/designs"
schemaLocation="common/iponz_design_common_types_v1.07.xsd" />

<!--<![CDATA[
=====
List Designs Registered Transaction Response Root Element
=====
]]>-->

<xs:element name="DesignsRegisteredResponse"
type="DesignsRegisteredResponseType"/>

<!--<![CDATA[
=====
List Designs Updated Transaction Response Root Element
=====
]]>-->

<xs:element name="DesignsUpdatedResponse"
type="DesignsUpdatedResponseType"/>

<!--<![CDATA[
=====
Get Design Information Transaction Response Root Element
=====
]]>-->

```



```

    <xs:element name="DesignInformationResponse"
type="DesignInformationResponseType"/>

<!--<![CDATA[
=====
Get Design Object Transaction Response Root Element
=====
]]>-->

    <xs:element name="DesignObjectResponse" type="DesignObjectResponseType"/>

<!--<![CDATA[
*****
* TRANSACTION/MESSAGE TYPES (in alphabetical order by name)      *
*****
]]>-->

    <xs:complexType name="DesignInformationResponseType">
        <xs:sequence>
            <xs:element name="TransactionDateTime" type="xs:dateTime" />
            <xs:choice>
                <xs:element name="TransactionError" type="TransactionErrorType" />
                <xs:element name="DesignInformation"
type="iponz:NZ_DesignInformationType" />
            </xs:choice>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="DesignObjectResponseType">
        <xs:sequence>
            <xs:element name="TransactionDateTime" type="xs:dateTime" />
            <xs:choice>
                <xs:element name="TransactionError" type="TransactionErrorType" />
                <xs:element name="DesignObject" type="DesignObjectType" />
            </xs:choice>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="DesignObjectType">
        <xs:sequence>
            <xs:element name="RegistrationNumber" type="xs:positiveInteger" />

```

```

    <xs:element name="ObjectKey" type="xs:token" />
    <xs:element name="ObjectFilename" type="xs:token" />
    <xs:element name="ObjectFormat" type="iponz:FileFormatType" />
    <xs:element name="ObjectBinary" type="xs:base64Binary" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DesignsRegisteredResponseType">
  <xs:sequence>
    <xs:element name="TransactionDateTime" type="xs:dateTime" />
    <xs:choice>
      <xs:element name="TransactionError" type="TransactionErrorType" />
      <xs:element name="DesignsRegistered" type="DesignsRegisteredType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DesignsUpdatedResponseType">
  <xs:sequence>
    <xs:element name="TransactionDateTime" type="xs:dateTime" />
    <xs:choice>
      <xs:element name="TransactionError" type="TransactionErrorType" />
      <xs:element name="DesignsUpdated" type="DesignsUpdatedType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DesignsRegisteredType">
  <xs:sequence>
    <xs:element name="RegistrationDateRange">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="FromDate" type="xs:date" />
          <xs:element name="ToDate" type="xs:date" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Designs">
      <xs:complexType>
        <xs:sequence>

```

```

        <xs:element name="Design" type="DesignSummaryType" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="DesignsUpdatedType">
    <xs:sequence>
        <xs:element name="UpdatedDateRange">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="FromDate" type="xs:date" />
                    <xs:element name="ToDate" type="xs:date" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="Designs">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="Design" type="DesignSummaryType" minOccurs="0"
maxOccurs="unbounded" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="DesignSummaryType">
    <xs:sequence>
        <xs:element name="RegistrationNumber" type="xs:positiveInteger" />
        <xs:element name="DesignCurrentStatusCode" type="xs:token" />
        <xs:element name="RegistrationDate" type="xs:date" minOccurs="0" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="TransactionErrorType">
    <xs:sequence>
        <xs:element name="TransactionErrorCode" type="xs:token" />
        <xs:element name="TransactionErrorText" type="iponz:TextType" />
    </xs:sequence>
</xs:complexType>

```

```
        </xs:sequence>
    </xs:complexType>

</xs:schema>
```

PATENT RESTFUL SERVICES

Services

There are three RESTful Patent services as follows:

SERVICE	DESCRIPTION	PARAMETER(S)
IPONZ List Patent Updated	To retrieve a list of Patents that have been updated within a given date range.	<ul style="list-style-type: none">• Date range <p><u>Example:</u> patents/updated/20100101-20101231</p> <p><u>Date range validation rules:</u></p> <ol style="list-style-type: none">1. Must be in YYYYMMDD-YYYYMMDD format.2. The 'to date' must not be before the 'from date'.3. The date range must not be more than 365 days.4. The 'from date' must not be before 20100101 (i.e., Jan. 1, 2010)
IPONZ Get Patent Information	To retrieve the details of a given Patent.	<ul style="list-style-type: none">• Patent IP number <p><u>Example:</u> patent/701802</p>
IPONZ Get Patent Object	To retrieve an object of a specific type associated with a given Patent. The object is returned as Base64-encoded binary data.	<ul style="list-style-type: none">• Patent IP number• Object type and index <p><u>Example:</u> patent/701802/drawing.1</p> <p>The first parameter is the Patent IP number for which the object needs to be returned.</p> <p>The second parameter is the object type and index. Valid object types for patents are drawing, provisional, complete, and abstract.</p> <p>The .1 is the index of the object to be returned. So drawing.1 will return the first drawing of that IP. If the index is not supplied then the default is 1. So the path patent/701802/drawing is equivalent to patent/701802/drawing.1</p> <p>Note: The Get Patent Information response will return a list of all objects that can be</p>

XML Schema Definitions

The following XSD file contains the schema definitions that form the basis of the response for all the Design RESTful services. These schema definitions conform to the WIPO standards. Note: This XSD file is published on the BusinessData production website.

iponz_patent_information_v1.00.xsd

URL: http://www.businessdata.govt.nz/data/app/ws/iponz_patent_information_v1.00.xsd

```
<?xml version="1.0" encoding="utf-8"?>

  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.iponz.govt.nz/XMLSchema/patents/information"
    xmlns:iponz="http://www.iponz.govt.nz/XMLSchema/patents"
    targetNamespace="http://www.iponz.govt.nz/XMLSchema/patents/information"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.00">

    <!--<![CDATA[
*****
*
*      IPONZ Patent Information Schema
*
*      Filename      : iponz_patent_information_v1.00.xsd
*      Author        : IPONZ
*      Version       : 1.00
*      Last Update   : 2010-08-04
*      Contact       : www.iponz.govt.nz
*
*****
]]>-->

    <xs:annotation>
      <xs:documentation>IPONZ Patent Information Schema </xs:documentation>
    </xs:annotation>
```

```

<!--<![CDATA[
=====
Import IPONZ Patent Common Types Library
=====
]]>-->

<xs:import namespace="http://www.iponz.govt.nz/XMLSchema/patents"
schemaLocation="common/iponz_patent_common_types_v1.10.xsd" />

<!--<![CDATA[
=====
List Patents Updated Transaction Response Root Element
=====
]]>-->

<xs:element name="PatentsUpdatedResponse"
type="PatentsUpdatedResponseType" />

<!--<![CDATA[
=====
Get Patent Information Transaction Response Root Element
=====
]]>-->

<xs:element name="PatentInformationResponse"
type="PatentInformationResponseType" />

<!--<![CDATA[
=====
Get Patent Object Transaction Response Root Element
=====
]]>-->

<xs:element name="PatentObjectResponse" type="PatentObjectResponseType" />

<!--<![CDATA[
*****
* TRANSACTION/MESSAGE TYPES (in alphabetical order by name) *
*****
]]>-->

```

```

<xs:complexType name="PatentInformationResponseType">
  <xs:sequence>
    <xs:element name="TransactionDateTime" type="xs:dateTime" />
    <xs:choice>
      <xs:element name="TransactionError" type="TransactionErrorType" />
      <xs:element name="PatentInformation"
type="iponz:NZ_PatentInformationType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PatentObjectResponseType">
  <xs:sequence>
    <xs:element name="TransactionDateTime" type="xs:dateTime" />
    <xs:choice>
      <xs:element name="TransactionError" type="TransactionErrorType" />
      <xs:element name="PatentObject" type="PatentObjectType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PatentObjectType">
  <xs:sequence>
    <xs:element name="PatentNumber" type="xs:positiveInteger" />
    <xs:element name="ObjectKey" type="xs:token" />
    <xs:element name="ObjectFilename" type="xs:token" />
    <xs:element name="ObjectFormat" type="iponz:FileFormatType" />
    <xs:element name="ObjectBinary" type="xs:base64Binary" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PatentsUpdatedResponseType">
  <xs:sequence>
    <xs:element name="TransactionDateTime" type="xs:dateTime" />
    <xs:choice>
      <xs:element name="TransactionError" type="TransactionErrorType" />
      <xs:element name="PatentsUpdated" type="PatentsUpdatedType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```



```

<xs:complexType name="PatentsUpdatedType">
  <xs:sequence>
    <xs:element name="UpdatedDateRange">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="FromDate" type="xs:date" />
          <xs:element name="ToDate" type="xs:date" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Patents">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Patent" type="PatentSummaryType" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PatentSummaryType">
  <xs:sequence>
    <xs:element name="PatentNumber" type="xs:positiveInteger" />
    <xs:element name="PatentCurrentStatusCode" type="xs:token" />
    <xs:element name="FilingDate" type="xs:date" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="TransactionErrorType">
  <xs:sequence>
    <xs:element name="TransactionErrorCode" type="xs:token" />
    <xs:element name="TransactionErrorText" type="iponz:TextType" />
  </xs:sequence>
</xs:complexType>

</xs:schema>

```