

**IBM SPSS Collaboration and
Deployment Services 5 RESTful
Scoring Service Developer's Guide**



Note: Before using this information and the product it supports, read the general information under Notices on p. 54.

This edition applies to IBM SPSS Collaboration and Deployment Services 5 and to all subsequent releases and modifications until otherwise indicated in new editions.

Adobe product screenshot(s) reprinted with permission from Adobe Systems Incorporated.

Microsoft product screenshot(s) reprinted with permission from Microsoft Corporation.

Licensed Materials - Property of IBM

© Copyright IBM Corporation 2000, 2012.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Preface

This guide is intended for developers working with the web services available in IBM® SPSS® Collaboration and Deployment Services. Users should have experience writing web service client applications and are assumed to have knowledge of IBM Corp. applications, Java and .NET development, data modeling concepts, and related technologies.

About IBM Business Analytics

IBM Business Analytics software delivers complete, consistent and accurate information that decision-makers trust to improve business performance. A comprehensive portfolio of [business intelligence](#), [predictive analytics](#), [financial performance and strategy management](#), and [analytic applications](#) provides clear, immediate and actionable insights into current performance and the ability to predict future outcomes. Combined with rich industry solutions, proven practices and professional services, organizations of every size can drive the highest productivity, confidently automate decisions and deliver better results.

As part of this portfolio, IBM SPSS Predictive Analytics software helps organizations predict future events and proactively act upon that insight to drive better business outcomes. Commercial, government and academic customers worldwide rely on IBM SPSS technology as a competitive advantage in attracting, retaining and growing customers, while reducing fraud and mitigating risk. By incorporating IBM SPSS software into their daily operations, organizations become predictive enterprises – able to direct and automate decisions to meet business goals and achieve measurable competitive advantage. For further information or to reach a representative visit <http://www.ibm.com/spss>.

Technical support

Technical support is available to maintenance customers. Customers may contact Technical Support for assistance in using IBM Corp. products or for installation help for one of the supported hardware environments. To reach Technical Support, see the IBM Corp. web site at <http://www.ibm.com/support>. Be prepared to identify yourself, your organization, and your support agreement when requesting assistance.

Contents

1 RESTful Scoring Service 1

RESTful Scoring Service overview	1
Accessing the RESTful Scoring Service from a client	1
Specifying a request URL	1
Specifying request headers	2
Specifying a request body	4
Receiving response headers	4
Receiving a response body	5

2 API reference for the RESTful Scoring Service 8

Service resource	8
Service GET	8
Configuration resource	10
Configuration GET	10
Metadata resource	12
Metadata GET	13
Score resource	17
Score POST	19
Metric item resource	23
Metric item GET	24
Metric value resource	28
Metric value GET	28

3 JSON reference for the RESTful Scoring Service 30

The configurationReference object	30
The modelReference element	31
The configurationStatus element	32
The metadataResult object	32
The metadataInputTable element	34
The metadataContextTable element	35
The metadataOutputField element	36
The metricItem object	37
The metricValue object	38

The scoreRequest object	38
The requestInputTable element	39
The context element	42
The scoreResult object	44
The columnNames element	46
The rowValues element	46
The returnedRequestInputTable element	47
The returnedDPDOutputTable element	49
The scoringServiceDetails object	51
The scoreProviderDetails element	52

Appendix

<i>A Notices</i>	54
-------------------------	-----------

<i>Index</i>	57
---------------------	-----------

RESTful Scoring Service

RESTful Scoring Service overview

The RESTful Scoring Service allows client applications to employ real-time scores derived from predictive models developed in IBM® SPSS® Modeler, IBM® SPSS® Statistics, or third party tools. The service fetches a specified model, loads it, invokes the correct scoring implementation, and returns the result to the client. Supported models include regression (linear and logistic), decision trees, decision lists, neural networks, and naïve Bayes defined in SPSS Modeler streams or in PMML from SPSS Statistics.

Scoring can be either synchronous or asynchronous, depending on whether the client needs to wait for a score before proceeding or not. The service can load multiple models simultaneously for scoring and can be virtualized across multiple servers in a cluster configuration to handle large processing loads. The service logs all scoring activity for regulatory audit purposes. Configuring models for scoring and monitoring the service performance can be done using the IBM® SPSS® Collaboration and Deployment Services Deployment Manager.

Accessing the RESTful Scoring Service from a client

Access to the RESTful Scoring Service is through standard HTTP and HTTPS methods. A client initiates a request to a server; the server processes the requests and returns the appropriate responses. The request and response are built around the concept of a resource and its data.

- [Specifying a request URL](#). The base URL for accessing the service is the internet address of an enabled server.
- [Specifying request headers](#). Certain requests may require information in the request header.
- [Specifying a request body](#). For operations that transmit data to the server, specify a request body in JSON format.
- [Receiving response headers](#). Requests return headers as part of the response.
- [Receiving a response body](#). For a GET operation, the response body contains the requested information in the format of a JSON object or array of objects.

Specifying a request URL

The base URL for accessing the RESTful Scoring Service is the following:

```
http://{server}:{port}/scoring/rest
```

The following variables should be replaced with values for your system.

- *{server}* corresponds to the server name or IP address for your IBM® SPSS® Collaboration and Deployment Services Repository
- *{port}* is the port number on which to access the IBM SPSS Collaboration and Deployment Services Repository

Send the base URL appended with /configuration as a GET request to obtain information about the scoring configurations available in your IBM SPSS Collaboration and Deployment Services Repository. Most of the remaining URL requests include the identifier for a particular scoring configuration. The following sample URL requests illustrate typical use.

- GET `http://cdssvr:80/scoring/rest/service`
Gets information about the score providers available for your system. For more information, see the topic [Service resource](#) in Chapter 2 on p. 8.
- GET `http://cdssvr:80/scoring/rest/configuration`
Gets information about all available scoring configurations. Use this method to determine the identifier for the configuration you want to use for scoring or for performance analysis. For more information, see the topic [Configuration resource](#) in Chapter 2 on p. 10.
- GET `http://cdssvr:80/scoring/rest/configuration/KMeans/metadata`
Gets information about the inputs for the scoring configuration named *KMeans*. This information is useful for constructing the body of score requests. For more information, see the topic [Metadata resource](#) in Chapter 2 on p. 12.
- POST `http://cdssvr:80/scoring/rest/configuration/KMeans/score`
Retrieves a score using the scoring configuration named *KMeans*. The input data for the score is supplied in the request body. For more information, see the topic [Score resource](#) in Chapter 2 on p. 17.
- GET `http://cdssvr:80/scoring/rest/configuration/KMeans/metric`
Gets information about all available performance statistics for the scoring configuration named *KMeans*. For more information, see the topic [Metric item resource](#) in Chapter 2 on p. 23.
- GET `http://cdssvr:80/scoring/rest/configuration/KMeans/metric/CONFIGURATION_UPTIME`
Gets the current value for the amount of time the scoring configuration named *KMeans* has been available for scoring. For more information, see the topic [Metric value resource](#) in Chapter 2 on p. 28.

The URL encoding for your application server must be defined to support the characters used in your URL strings. For example, if a scoring configuration name contains multi-byte characters and the application server URL encoding is defined as ISO-8859-1, the server returns an error indicating that the configuration cannot be found. Setting the URL encoding to UTF-8 eliminates this problem. For information about setting the URL encoding for a specific server, see your application server documentation.

Specifying request headers

Certain requests may require headers. The following table identifies fields commonly used in request headers.

Table 1-1
Request header values

Name	Value
Content-type	application/json
Authorization	Authentication string encoded using Base64
Client-Accept-Language	Tag defining the language for the response information. This field takes precedence over the Accept-Language field.
Accept-Language	Tag defining the language for the response information

Content type

The *Content-type* field defines the MIME type of the information contained in the body of the request. The RESTful Scoring Service currently only supports a content type of *application/json*.

Authorization

The RESTful Scoring Service includes a security layer to limit access to authorized users. This layer uses HTTP basic access authentication, requiring the specification of a valid user name and password to be included in any service request.

To add authentication information to a request, include the *Authorization* field in the request header. The value for this field consists of the following components:

- the string *Basic* to indicate basic access authentication is being used
- a space
- the Base64 encoding of the concatenation of the user name, a colon, and the password

For example, the Base64 encoding for a user name of *user* and a password of *pass* is the following:

Base64 Encode(user:pass) = dXNlcjpwYXNz

The *Authorization* value for this user would be specified as the following string:

Basic dXNlcjpwYXNz

Note that the WebLogic Application Server enforces HTTP Basic Access Authentication at the domain level using its security realms. As a result, the application server authenticates the user name and password before passing it to the RESTful Scoring Service. To avoid this problem, either disable the Weblogic HTTP basic access authentication or define the same users in the Weblogic Admin Console.

Localization

The RESTful Scoring Service can return responses containing localized information. To specify a language for the response information, include either the *Client-Accept-Language* or the *Accept-Language* fields in the request header. For both fields, specify a language value in accordance with [RFC 1766](http://www.ietf.org/rfc/rfc1766.txt) (<http://www.ietf.org/rfc/rfc1766.txt>).

The *Client-Accept-Language* field is used by other components of IBM® SPSS® Collaboration and Deployment Services that interact with the RESTful Scoring Service. If your request includes both language fields, this field is evaluated before the *Accept-Language* field, which is the standard HTTP field for language specification.

For example, to create a response in German, include the following header field in your request:

```
Client-Accept-Language: de
```

If your request also includes the following header field:

```
Accept-Language: fr
```

the response will still be in German as the *Client-Accept-Language* field takes precedence. In this case, you would either need to change the value of *Client-Accept-Language* to *fr* or omit this field entirely to receive a response in French.

Specifying a request body

For operations that transmit data to the server, such as POST, specify a request body in JavaScript Object Notation, or JSON, format. For information on JSON formatting, see [Receiving a response body](#). For information about the specific JSON elements and objects used by the RESTful Scoring Service, see [JSON reference for the RESTful Scoring Service](#).

This sample POST request retrieves scoring output for a specified set of input values:

```
{
  "id":"config_1",
  "requestInputTable":[
    {
      "name":"Table 1",
      "requestInputRow":[
        "input":[
          {"name":"Age","value":"1"},
          {"name":"BP","value":"1"},
          {"name":"Cholesterol","value":"1"},
          {"name":"Drug","value":"1"},
          {"name":"K","value":"1"},
          {"name":"Na","value":"1"},
          {"name":"Gender","value":"1"}
        ]
      ]
    }
  ],
  "context":[]
}
```

Receiving response headers

Requests return headers as part of the response. The following table lists response header values.

Table 1-2
Response header values for GET requests

Name	Value
Status Code	200 OK or an error code
Server	The name of the server processing the request
X-Powered-By	Identifies the technology underlying the application
Content-Type	application/json;charset=ISO-8859-1
Content-Length	Length of the response body in bytes
Date	Current date and time
Set-Cookie	Specifies an HTTP session cookie

Receiving a response body

The response body contains the requested information in the format of a JSON object or array of objects. The following sections provide basic formatting rules for JSON. These rules also apply to the [request body](#).

See <http://www.json.org> for a complete description of JSON. For information about the specific JSON elements and objects used by the RESTful Scoring Service, see [JSON reference for the RESTful Scoring Service](#).

Elements

An element consists of a name/value pair having the following structure:

```
"<name>":<value>
```

- *<name>* is a string that identifies the element
- *<value>* corresponds to the value for the specific element. The value may be a boolean, string, number, object, or array.

String values must be enclosed in quotation marks and may contain backslash escape characters. Numeric values are not enclosed in quotation marks. A boolean value is either true or false, and is not enclosed in quotation marks.

For example, the following *label* element corresponds to a string value of *LATEST*:

```
"label":"LATEST"
```

The following *value* element corresponds to a numeric value of *38.0*:

```
"value":38.0
```

The following *isRequired* element corresponds to a boolean value of *true*:

```
"isRequired":true
```

Objects

An object represents an unordered collection of elements. Objects have the following structure:

```
{<element1>, ... ,<elementN>}
```

- *<element#>* corresponds to a particular element in the object. Individual object elements are separated by commas.

For example, the following object consists of the elements *state*, *modelReference*, *configurationStatus*, and *id*:

```
{
  "state":"ACTIVE",
  "modelReference":{
    "resourcePath":"\\june\\multiple_input_double_updated.str",
    "label":"LATEST",
    "id":"09775272cf8ee6e5000001328016b69284dc"
  },
  "configurationStatus":{
    "message":"Started",
    "statusCode":"INFORMATION"
  },
  "id":"AC2"
}
```

The values for the *modelReference* and *configurationStatus* elements are also objects consisting of three and two elements, respectively.

Arrays

An array represents an ordered collection of values, elements, objects, or other arrays. Arrays have the following structure:

```
[<entry1>, ... ,<entryN>]
```

- *<entry#>* corresponds to a particular entry in the array. Individual array entries are separated by commas.

For example, the value for the following *metadataOutputField* element is an array containing two objects:

```
"metadataOutputField":[
  {
    "categoricalValues":[],
    "isReturned":true,
    "type":"long",
    "name":"Age",
    "description":"Age"
  },
  {
    "categoricalValues":[],
    "isReturned":true,
```

```
"type":"string",  
"name":"$O-Anomaly",  
"description":"$O-Anomaly"  
}  
]
```

Each object in the array consists of five elements having values describing individual output fields. For instance, the first output field has a name of *Age* and is of type *long*. The second output field has a name of *\$O-Anomaly* and is of type *string*.

API reference for the RESTful Scoring Service

Service resource

Score providers provide the RESTful Scoring Service with the internal processing instructions necessary for specific model types. For the service to be able to create a scoring configuration for a model, a score provider for that type of model must be available. The provider registers itself with the service, indicating which MIME types it supports. Each provider has a unique name and identifier to distinguish it from other providers, and each is versioned separately. Providers available to the scoring service include:

- *SmartScore Score Provider*, for processing PMML files
- *Modeler Score Provider*, for processing IBM® SPSS® Modeler streams and scenario files

Service GET

Gets information about the available score providers.

Defined in

[Service resource](#)

Request

GET http://{server}:{port}/scoring/rest/service

The following variables should be replaced with values for your system.

- *{server}* corresponds to the server name or IP address for your IBM® SPSS® Collaboration and Deployment Services Repository
- *{port}* is the port number on which to access the IBM SPSS Collaboration and Deployment Services Repository

Table 2-1
Request header values

Name	Value
Content-type	application/json
Authorization	Authentication string encoded using Base64
Client-Accept-Language	Tag defining the language for the response information. This field takes precedence over the Accept-Language field.
Accept-Language	Tag defining the language for the response information

The HTTP request body should be empty.

Response

A successful request returns the following information:

- Status code 200
- Response body in JSON format containing the service representation for the request. The response object has the following general structure:

```
{
  "version":<string>,
  "scoreProviderDetails":[
    {
      "supportedMimeTypes":["<string>"],
      "version":<string>,
      "name":<string>,
      "id":<string>"
    }
  ]
}
```

For more information, see the topic [The `scoringServiceDetails` object](#) in Chapter 3 on p. 51.

An unsuccessful request returns a response code other than 200.

Example

This request retrieves information about which scoring providers are installed.

GET `http://cdssvr:80/scoring/rest/service`

This is the response object, indicating that there are two scoring providers installed.

```
{
  "version":"5.0.0.0.155",
  "scoreProviderDetails":[
    {
      "supportedMimeTypes":["application\x-vnd.spss-scenario","application\x-vnd.spss-clementine-stream"],
      "version":"1.0",
      "name":"Modeler Score Provider",
      "id":"091ed7cda16d295c000001327c563c4593de"
    },
    {
      "supportedMimeTypes":["application\x-vnd.spss-pmml"],
      "version":"1.0",
      "name":"Score Provider - SmartScore",
      "id":"091ed7cd82a9085b000001327c118593a316"
    }
  ]
}
```

The response indicates that the RESTful Scoring Service is able to generate scores for PMML files, IBM® SPSS® Modeler streams, and SPSS Modeler scenarios.

Configuration resource

Before a model can be used for scoring, supplemental information must be defined. For example, a IBM® SPSS® Modeler stream containing a IBM® SPSS® Collaboration and Deployment Services Enterprise View source node needs an associated real-time data provider definition specifying how to retrieve the required data. Such information constitutes a **scoring configuration** for the model, and defines scoring parameters such as the following:

- Identification information for the configuration itself
- Identification information for the model used for scoring
- The data provider for the input
- Settings for logging
- The order of the input attributes
- Cache size used for scoring models

A single model may be used in a variety of scoring situations that require different scoring parameters. For example, scores may be based on a test data provider for internal purposes and on a different data provider for production usage. Alternatively, the information being logged as result of scoring may depend on the scoring situation. To allow a model to be used in differing scoring circumstances, any model may be associated with multiple scoring configurations.

Scoring configurations can be suspended to temporarily prevent the processing of score requests. A suspended configuration must be reactivated before it can be used to generate scores.

Typically, you create scoring configurations using IBM® SPSS® Collaboration and Deployment Services Deployment Manager. You use the RESTful Scoring Service to generate scores or report performance metrics for those configurations.

Configuration GET

Gets information about the available scoring configurations.

Defined in

[Configuration resource](#)

Request

GET http://{server}:{port}/scoring/rest/configuration

The following variables should be replaced with values for your system.

- *{server}* corresponds to the server name or IP address for your IBM® SPSS® Collaboration and Deployment Services Repository
- *{port}* is the port number on which to access the IBM SPSS Collaboration and Deployment Services Repository

Table 2-2
Request header values

Name	Value
Content-type	application/json
Authorization	Authentication string encoded using Base64
Client-Accept-Language	Tag defining the language for the response information. This field takes precedence over the Accept-Language field.
Accept-Language	Tag defining the language for the response information

The HTTP request body should be empty.

Response

A successful request returns the following information:

- Status code 200
- Response body in JSON format containing the configuration representation for the request. The response array has the following general structure:

```
[
  {
    "state":"<string>",
    "modelReference":{
      "resourcePath":"<string>",
      "label":"<string>",
      "id":"<string>"
    },
    "configurationStatus":{
      "message":"<string>",
      "statusCode":"<string>"
    },
    "id":"<string>"
  }
]
```

For more information, see the topic [The configurationReference object](#) in Chapter 3 on p. 30.

An unsuccessful request returns a response code other than 200.

Example

This request retrieves information about the scoring configurations available on the *cdssvr* server.

GET <http://cdssvr:80/scoring/rest/configuration>

This is the response object, indicating that there are two scoring configurations.

```
[
  {
    "state":"ACTIVE",
    "modelReference":{
      "resourcePath":"\\zsi\\multiple_Input_Double_updated.str",
      "label":"LATEST",
      "id":"09775272cf8ee6e5000001328016b69284dc"
    },
    "configurationStatus":{"message":"Started","statusCode":"INFORMATION"},
    "id":"AC2"
  },
  {
    "state":"ACTIVE",
    "modelReference":{
      "resourcePath":"\\zsi\\KMeans.xml",
      "label":"LATEST",
      "id":"091ed68980eebc90000001317470503d82fb"
    },
    "configurationStatus":{
      "message":"Unable to load Data Provider Information settings
      Reason: com.spss.scoring.internal.engine.ScoringEngineException:
      Data Provider Definition failed validation",
      "statusCode":"ERROR"
    },
    "id":"SmartScore(RTDPD)"
  }
]
```

The first configuration has an identifier of *AC2*. The `modelReference` element indicates that the configuration uses the latest version of the IBM® SPSS® Modeler stream *multiple_Input_Double_updated.str* for score requests. The `configurationStatus` element reports that the configuration has been started.

In contrast, the second configuration, which has an identifier of *SmartScore(RTDPD)*, uses the latest version of the PMML file *KMeans.xml* for score requests. The `configurationStatus` element reports an error for the configuration due to problems with the data provider definition referenced in the configuration.

Metadata resource

Metadata describes the input and output structure used by a scoring model. Metadata for a model includes the following information:

- Name for each table of input fields
- Name and data type for every input field
- Name and data type for every output field

Use the metadata information to create your scoring requests in a structure required by the scoring model.

Metadata GET

Gets information about the input and output fields used by a specific scoring configuration.

Defined in

[Metadata resource](#)

Request

GET `http://{server}:{port}/scoring/rest/configuration/{config_id}/metadata`

The following variables should be replaced with values for your system.

- *{server}* corresponds to the server name or IP address for your IBM® SPSS® Collaboration and Deployment Services Repository
- *{port}* is the port number on which to access the IBM SPSS Collaboration and Deployment Services Repository
- *{config_id}* is the identifier for the scoring configuration. You can obtain the identifiers for the available scoring configurations by using a configuration request. For more information, see the topic [Configuration GET](#) on p. 10.

Table 2-3
Request header values

Name	Value
Content-type	application/json
Authorization	Authentication string encoded using Base64
Client-Accept-Language	Tag defining the language for the response information. This field takes precedence over the Accept-Language field.
Accept-Language	Tag defining the language for the response information

The HTTP request body should be empty.

Response

A successful request returns the following information:

- Status code 200
- Response body in JSON format containing the metadata representation for the request. The response object has the following general structure:

```
{
  "metadataInputTable":[
    {
      "name": "<string>",
```

```

    "id":<string>,
    "metadataInputField":[
      {
        "categoricalValues":<string>,
        "isRequired":<boolean>,
        "type":<string>,
        "name":<string>,
        "description":<string>"
      }
    ]
  },
  "metadataOutputField":[
    {
      "categoricalValues":<string>,
      "isReturned":<boolean>,
      "type":<string>,
      "name":<string>,
      "description":<string>"
    }
  ],
  "metadataContextTable":[
    {
      "table":<string>,
      "contextColumn":[
        {
          "categoricalValues":<string>,
          "type":<string>,
          "name":<string>,
          "description":<string>"
        }
      ]
    }
  ]
}

```

For more information, see the topic [The metadataResult object](#) in Chapter 3 on p. 32.

An unsuccessful request returns a response code other than 200.

Example

This request retrieves the metadata for the *Kmeans* scoring configuration.

GET http://cdssvr:80/scoring/rest/configuration/KMeans/metadata

This is the response object.

```

{
  "metadataInputTable":[
    {
      "name":"DRUG1n",
      "id":"id5JAYKBB9IXD",

```

```
"metadataInputField":[
  {
    "categoricalValues":[],"isRequired":true,"type":"long",
    "name":"Age","description":"Age"
  },
  {
    "categoricalValues":[],"isRequired":true,"type":"string",
    "name":"BP","description":"BP"
  },
  {
    "categoricalValues":[],"isRequired":true,"type":"string",
    "name":"Cholesterol","description":"Cholesterol"
  },
  {
    "categoricalValues":[],"isRequired":true,"type":"string",
    "name":"Drug","description":"Drug"
  },
  {
    "categoricalValues":[],"isRequired":true,"type":"double",
    "name":"K","description":"K"
  },
  {
    "categoricalValues":[],"isRequired":true,"type":"double",
    "name":"Na","description":"Na"
  },
  {
    "categoricalValues":[],"isRequired":true,"type":"string",
    "name":"Gender","description":"Gender"
  }
],
"metadataOutputField":[
  {
    "categoricalValues":[],"isReturned":true,"type":"long",
    "name":"Age","description":"Age"
  },
  {
    "categoricalValues":[],"isReturned":true,"type":"string",
    "name":"Sex","description":"Sex"
  },
  {
    "categoricalValues":[],"isReturned":true,"type":"string",
    "name":"BP","description":"BP"
  },
  {
    "categoricalValues":[],"isReturned":true,"type":"string",
    "name":"Cholesterol","description":"Cholesterol"
  },
  {
    "categoricalValues":[],"isReturned":true,"type":"double",
    "name":"Na","description":"Na"
  }
]
```

```

    },
    {
      "categoricalValues":[],"isReturned":true,"type":"double",
      "name":"K","description":"K"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"string",
      "name":"Drug","description":"Drug"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"string",
      "name":"$0-Anomaly","description":"$0-Anomaly"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"double",
      "name":"$0-AnomalyIndex","description":"$0-AnomalyIndex"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"long",
      "name":"$0-PeerGroup","description":"$0-PeerGroup"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"string",
      "name":"$0-Field-1","description":"$0-Field-1"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"double",
      "name":"$0-FieldImpact-1","description":"$0-FieldImpact-1"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"string",
      "name":"$0-Field-2","description":"$0-Field-2"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"double",
      "name":"$0-FieldImpact-2","description":"$0-FieldImpact-2"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"string",
      "name":"$0-Field-3","description":"$0-Field-3"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"double",
      "name":"$0-FieldImpact-3","description":"$0-FieldImpact-3"
    }
  ],
  "metadataContextTable":[]
}

```

The `metadataInputTable` element indicates that the configuration expects score requests to include seven input values for a single table. The following list identifies the input fields and their corresponding data types:

- *Age* is of type *long*
- *BP* is of type *string*
- *Cholesterol* is of type *string*
- *Drug* is of type *string*
- *K* is of type *double*
- *Na* is of type *double*
- *Gender* is of type *string*

The `metadataOutputTable` object indicates that the configuration returns score responses that include all input values plus several additional output values containing the scoring results.

The empty array for the `metadataContextTable` object indicates that the configuration does not use any context data.

Score resource

Applying a predictive model to a set of data can produce a variety of scores, such as predicted values, predicted probabilities, and other values based on that model. The type of score produced is referred to as the **scoring function**. The following scoring functions are available:

PREDICT	<i>Returns the predicted value of the target variable.</i>
STDDEV	<i>Standard deviation.</i>
PROBABILITY	<i>Probability associated with a particular category of a target variable. Applies only to categorical variables. In the absence of the optional third parameter, <i>category</i>, this is the probability that the predicted category is the correct one for the target variable. If a particular category is specified, then this is the probability that the specified category is the correct one for the target variable.</i>
CONFIDENCE	<i>A probability measure associated with the predicted value of a categorical target variable. Applies only to categorical variables.</i>
NODEID	<i>The terminal node number. Applies only to tree models.</i>
CUMHAZARD	<i>Cumulative hazard value. Applies only to Cox regression models.</i>
NEIGHBOR	<i>The ID of the <i>k</i>th nearest neighbor. Applies only to nearest neighbor models. In the absence of the optional third parameter, <i>k</i>, this is the ID of the nearest neighbor. The ID is the value of the case labels variable, if supplied, and otherwise the case number.</i>
DISTANCE	<i>The distance to the <i>k</i>th nearest neighbor. Applies only to nearest neighbor models. In the absence of the optional third parameter, <i>k</i>, this is the distance to the nearest neighbor. Depending on the model, either Euclidean or City Block distance will be used.</i>

The following table lists the set of scoring functions available for each type of model that supports scoring. The function type denoted as `PROBABILITY (category)` refers to specification of a particular category for the `PROBABILITY` function.

Table 2-4
Supported functions by model type

Model type	Supported functions
Tree (categorical target)	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE, NODEID
Tree (scale target)	PREDICT, NODEID, STDDEV
Boosted Tree (C5.0)	PREDICT, CONFIDENCE
Linear Regression	PREDICT, STDDEV
Automatic Linear Models	PREDICT
Binary Logistic Regression	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
Conditional Logistic Regression	PREDICT
Multinomial Logistic Regression	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
General Linear Model	PREDICT, STDDEV
Discriminant	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
TwoStep Cluster	PREDICT
K-Means Cluster	PREDICT, CONFIDENCE
Kohonen	PREDICT
Neural Net (categorical target)	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
Neural Net (scale target)	PREDICT
Naive Bayes	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
Anomaly Detection	PREDICT
Ruleset	PREDICT, CONFIDENCE
Generalized Linear Model (categorical target)	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
Generalized Linear Model (scale target)	PREDICT, STDDEV
Generalized Linear Mixed Model (categorical target)	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
Generalized Linear Mixed Model (scale target)	PREDICT
Ordinal Multinomial Regression	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
Cox Regression	PREDICT, CUMHAZARD
Nearest Neighbor (scale target)	PREDICT, NEIGHBOR, NEIGHBOR(K), DISTANCE, DISTANCE(K)
Nearest Neighbor (categorical target)	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE, NEIGHBOR, NEIGHBOR(K), DISTANCE, DISTANCE(K)
Bayesian Network	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
Support Vector Machine (categorical target)	PREDICT, PROBABILITY, PROBABILITY (category), CONFIDENCE
Support Vector Machine (scale target)	PREDICT, STDDEV

- For the Binary Logistic Regression, Multinomial Logistic Regression, and Naive Bayes models, the value returned by the CONFIDENCE function is identical to that returned by the PROBABILITY function.

- For the K-Means model, the value returned by the CONFIDENCE function is the least distance.
- For tree and ruleset models, the confidence can be interpreted as an adjusted probability of the predicted category and is always less than the value given by PROBABILITY. For these models, the confidence value is more reliable than the value given by PROBABILITY.
- For neural network models, the confidence provides a measure of whether the predicted category is much more likely than the second-best predicted category.
- For Ordinal Multinomial Regression and Generalized Linear Model, the PROBABILITY function is supported when the target variable is binary.
- For nearest neighbor models without a target variable, the available functions are NEIGHBOR and DISTANCE.

Score POST

Retrieves one or more scores for a specific scoring configuration.

Defined in

[Score resource](#)

Request

POST `http://{server}:{port}/scoring/rest/configuration/{config_id}/score`

The following variables should be replaced with values for your system.

- *{server}* corresponds to the server name or IP address for your IBM® SPSS® Collaboration and Deployment Services Repository
- *{port}* is the port number on which to access the IBM SPSS Collaboration and Deployment Services Repository
- *{config_id}* is the identifier for the scoring configuration. You can obtain the identifiers for the available scoring configurations by using a configuration request. For more information, see the topic [Configuration GET](#) on p. 10.

Table 2-5
Request header values

Name	Value
Content-type	application/json
Authorization	Authentication string encoded using Base64
Client-Accept-Language	Tag defining the language for the response information. This field takes precedence over the Accept-Language field.
Accept-Language	Tag defining the language for the response information

The HTTP request body should contain the input data for the score request in JSON format. The request object has the following general structure:

```
{
  "id": "<string>",
  "requestInputTable": [
    {
      "name": "<string>",
      "requestInputRow": [
        "input": [
          {
            "name": "<string>",
            "value": "<string>"
          }
        ]
      ]
    }
  ],
  "context": [
    {
      "name": "<string>",
      "columnName": ["<string>"],
      "rowValues": [
        {
          "value": [
            {
              "value": "<string>"
            }
          ]
        }
      ]
    }
  ]
}
```

For more information, see the topic [The scoreRequest object](#) in Chapter 3 on p. 38.

Response

A successful request returns the following information:

- Status code 200
- Response body in JSON format containing the score information for the request. The response object has the following general structure:

```
{
  "id": "<string>",
  "columnNames": {
    "name": ["<string>"]
  },
  "rowValues": [
    {
```

```

    "reserved":<string>
    "value":[
      {
        "value":<string>
      }
    ]
  }
],
"returnedRequestInputTable":[
  {
    "id":<string>,
    "name":<string>,
    "returnedRequestInputRow":[
      {
        "returnedRequestInputValue":[
          {
            "name":<string>,
            "type":<string>,
            "value":<string>
          }
        ]
      }
    ]
  }
]
}
],
"returnedDPDOutputTable":[
  {
    "id":<string>,
    "name":<string>,
    "returnedDPDOutputRow":[
      {
        returnedDPDOutputValue[
          {
            "name":<string>,
            "type":<string>,
            "value":<string>
          }
        ]
      }
    ]
  }
]
}
]
}
}

```

For more information, see the topic [The scoreResult object](#) in Chapter 3 on p. 44.

An unsuccessful request returns a response code other than 200.

Example

This request retrieves a score for the *config_1* scoring configuration.

GET `http://cdssvr:80/scoring/rest/configuration/config_1/metric`

This is the request body, which includes a single table consisting of one row. This row specifies the values for the seven model inputs for the configuration.

```
{
  "id":"config_1",
  "requestInputTable":[
    {
      "name":"Table 1",
      "requestInputRow":[
        "input":[
          {"name":"Age","value":"1"},
          {"name":"BP","value":"1"},
          {"name":"Cholesterol","value":"1"},
          {"name":"Drug","value":"1"},
          {"name":"K","value":"1"},
          {"name":"Na","value":"1"},
          {"name":"Gender","value":"1"}
        ]
      ]
    }
  ],
  "context":[]
}
```

This is the response object. The `columnNames` element indicates that the configuration returned scoring function values for *Prediction* and *Confidence*. The `rowValues` element contains a single object that reports the actual scoring values of 2 and 1.34153.

```
{
  "id":"954d9db3-049f-464c-b218-eadd628cf349",
  "columnNames":[
    "name":["Prediction","Confidence"]
  ],
  "rowValues":[
    {
      "reserved":null,
      "value":{"value":"2"},{"value":"1.3415297614991626"}
    }
  ],
  "returnedRequestInputTable":[
    {
      "name":"Table 1",
      "id":"Table 1",
      "returnedRequestInputRow":[
        {
          "returnedRequestInputValue":[
            {"name":"Age","type":"double","value":"1"},
            {"name":"BP","type":"string","value":"1"},
            {"name":"Cholesterol","type":"string","value":"1"},
            {"name":"Drug","type":"string","value":"1"},

```

```

        {"name":"K","type":"double","value":"1"},
        {"name":"Na","type":"double","value":"1"},
        {"name":"Gender","type":"string","value":"1"}
    ]
}
],
"returnedDPDOutputTable":[]
}

```

The returnedRequestInputTable element contains the input values that resulted in these scores.

Metric item resource

Scoring metrics are measurements that reflect the performance of a scoring configuration or the service itself. Available metric items include the following:

- **Service Scores.** Total number of scores produced by the service.
- **Service Uptime.** Amount of time, measured in seconds, that the service has been running.
- **Average Latency.** Average amount of time, measured in milliseconds, between the request for a score and the generation of the score.
- **Minimum Latency.** Smallest amount of time, measured in milliseconds, between the request for a score and the generation of the score.
- **Maximum Latency.** Largest amount of time, measured in milliseconds, between the request for a score and the generation of the score.
- **Score Data Initialization Time.** Amount of time, measured in milliseconds, that the data service takes to be initialized for a score request.
- **Average Data Initialization Time.** Average amount of time, measured in milliseconds, that the data service takes to be initialized for a score request.
- **Minimum Data Initialization Time.** Smallest amount of time, measured in milliseconds, that the data service takes to be initialized for a score request.
- **Maximum Data Initialization Time.** Largest amount of time, measured in milliseconds, that the data service takes to be initialized for a score request.
- **Score Data Access Time.** Amount of time, measured in milliseconds, that the data service takes to access the data.
- **Average Data Access Time.** Average amount of time, measured in milliseconds, that the data service takes to access the data.
- **Minimum Data Access Time.** Smallest amount of time, measured in milliseconds, that the data service takes to access the data.
- **Maximum Data Access Time.** Largest amount of time, measured in milliseconds, that the data service takes to access the data.
- **Score Computation Wait Time.** Amount of time, measured in milliseconds, that the score provider worker spends waiting for the data service.

- **Average Computation Wait Time.** Average amount of time, measured in milliseconds, that the score provider worker spends waiting for the data service.
- **Minimum Computation Wait Time.** Smallest amount of time, measured in milliseconds, that the score provider worker spends waiting for the data service.
- **Maximum Computation Wait Time.** Largest amount of time, measured in milliseconds, that the score provider worker spends waiting for the data service.
- **Score Computation Time.** Amount of time, measured in milliseconds, that the score provider worker spends computing the score.
- **Average Computation Time.** Average amount of time, measured in milliseconds, that the score provider worker spends computing the score.
- **Minimum Computation Time.** Smallest amount of time, measured in milliseconds, that the score provider worker spends computing the score.
- **Maximum Computation Time.** Largest amount of time, measured in milliseconds, that the score provider worker spends computing the score.
- **Average Log Serialization Time.** Average amount of time, measured in milliseconds, to create a log entry in XML format.
- **Minimum Log Serialization Time.** Smallest amount of time, measured in milliseconds, to create a log entry in XML format.
- **Maximum Log Serialization Time.** Largest amount of time, measured in milliseconds, to create a log entry in XML format.
- **Average Log Queue Time.** Average amount of time, measured in milliseconds, to place the XML logging data onto the JMS queue.
- **Minimum Log Queue Time.** Smallest amount of time, measured in milliseconds, to place the XML logging data onto the JMS queue.
- **Maximum Log Queue Time.** Largest amount of time, measured in milliseconds, to place the XML logging data onto the JMS queue.
- **Configuration Scores.** Total number of scores produced by a specific scoring configuration.
- **Score Elapsed Time.** Amount of time, measured in milliseconds, since the previous score generation.
- **Configuration Uptime.** Amount of time, measured in seconds, that the scoring configuration has been available for scoring.
- **Cache Hits.** Number of successful attempts to retrieve data from the memory cache for a scoring configuration.
- **Cache Misses.** Number of failed attempts to retrieve data from the memory cache for a scoring configuration. Each failed attempt results in a new service call to retrieve the necessary data.

The scale for a particular metric item determines the number of decimal points included in the measurement.

Metric item GET

Retrieves a list of available scoring metric items for a specific scoring configuration.

Defined in

[Metric item resource](#)

Request

GET http://{server}:{port}/scoring/rest/configuration/{config_id}/metric

The following variables should be replaced with values for your system.

- *{server}* corresponds to the server name or IP address for your IBM® SPSS® Collaboration and Deployment Services Repository
- *{port}* is the port number on which to access the IBM SPSS Collaboration and Deployment Services Repository
- *{config_id}* is the identifier for the scoring configuration. You can obtain the identifiers for the available scoring configurations by using a configuration request. For more information, see the topic [Configuration GET](#) on p. 10.

Table 2-6
Request header values

Name	Value
Content-type	application/json
Authorization	Authentication string encoded using Base64
Client-Accept-Language	Tag defining the language for the response information. This field takes precedence over the Accept-Language field.
Accept-Language	Tag defining the language for the response information

The HTTP request body should be empty.

Response

A successful request returns the following information:

- Status code 200
- Response body in JSON format containing the metric item information for the request. The response array has the following general structure:

```
[
  {
    "unit": "<string>",
    "scale": <number>,
    "id": "<string>",
    "name": "<string>"
  }
]
```

For more information, see the topic [The metricItem object](#) in Chapter 3 on p. 37.

An unsuccessful request returns a response code other than 200.

Example

This request retrieves information about all available metrics for the *Kmeans* scoring configuration.

GET <http://cdssvr:80/scoring/rest/configuration/KMeans/metric>

This is the response array. Each object in the array corresponds to a separate metric.

```
[
  {
    "unit":"milliseconds","scale":3,
    "id":"CONFIGURATION_COMPUTATION_WAIT_TIME_MINIMUM","name":"Minimum Computation Wait Time"
  },
  {
    "unit":"hits","scale":0,
    "id":"CONFIGURATION_CACHE_HITS","name":"Cache Hits"
  },
  {
    "unit":"scores","scale":0,
    "id":"SERVICE_TOTAL_SCORES","name":"Service Scores"
  },
  {
    "unit":"milliseconds","scale":3,
    "id":"CONFIGURATION_DATA_INIT_TIME_AVERAGE","name":"Average Data Initialization Time"
  },
  {
    "unit":"milliseconds","scale":3,
    "id":"CONFIGURATION_LOG_QUEUE_TIME_MINIMUM","name":"Minimum Log Queue Time"
  },
  {
    "unit":"milliseconds","scale":3,
    "id":"CONFIGURATION_DATA_INIT_TIME_MAXIMUM","name":"Maximum Data Initialization Time"
  },
  {
    "unit":"milliseconds","scale":3,
    "id":"CONFIGURATION_RESPONSE_TIME_MINIMUM","name":"Minimum Latency"
  },
  {
    "unit":"seconds","scale":0,
    "id":"SERVICE_UPTIME","name":"Service Uptime"
  },
  {
    "unit":"milliseconds","scale":3,
    "id":"CONFIGURATION_COMPUTATION_WAIT_TIME_AVERAGE","name":"Average Computation Wait Time"
  },
  {
    "unit":"milliseconds","scale":3,
    "id":"CONFIGURATION_COMPUTATION_WAIT_TIME_MAXIMUM","name":"Maximum Computation Wait Time"
  },
  {
    "unit":"milliseconds","scale":3,
    "id":"CONFIGURATION_LOG_QUEUE_TIME_AVERAGE","name":"Average Log Queue Time"
  },
]
```



```

{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_COMPUTATION_TIME_MINIMUM","name":"Minimum Computation Time"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_LOG_SERIALIZE_TIME_MINIMUM","name":"Minimum Log Serialization Time"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_RESPONSE_TIME_AVERAGE","name":"Average Latency"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_LOG_QUEUE_TIME_MAXIMUM","name":"Maximum Log Queue Time"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_RESPONSE_TIME_MAXIMUM","name":"Maximum Latency"
},
{
  "unit":"scores","scale":0,
  "id":"CONFIGURATION_TOTAL_SCORES","name":"Configuration Scores"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_DATA_ACCESS_TIME_MINIMUM","name":"Minimum Data Access Time"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_COMPUTATION_TIME_AVERAGE","name":"Average Computation Time"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_LOG_SERIALIZE_TIME_AVERAGE","name":"Average Log Serialization Time"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_COMPUTATION_TIME_MAXIMUM","name":"Maximum Computation Time"
},
{
  "unit":"misses","scale":0,
  "id":"CONFIGURATION_CACHE_MISSES","name":"Cache Misses"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_LOG_SERIALIZE_TIME_MAXIMUM","name":"Maximum Log Serialization Time"
},
{
  "unit":"seconds","scale":0,
  "id":"CONFIGURATION_UPTIME","name":"Configuration Uptime"
},
}

```

```

{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_DATA_ACCESS_TIME_AVERAGE","name":"Average Data Access Time"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_DATA_ACCESS_TIME_MAXIMUM","name":"Maximum Data Access Time"
},
{
  "unit":"milliseconds","scale":3,
  "id":"CONFIGURATION_DATA_INIT_TIME_MINIMUM","name":"Minimum Data Initialization Time"
}
]

```

Use the [Metric value GET](#) method to retrieve the value for a specific metric item.

Metric value resource

A metric value corresponds to actual value for a specific metric item. The *id* for any metric item can be used to retrieve the value for that metric.

Metric value GET

Retrieves the value for a specific scoring configuration metric item.

Defined in

[Metric value resource](#)

Request

GET `http://{server}:{port}/scoring/rest/configuration/{config_id}/metric/{name}`

The following variables should be replaced with values for your system.

- *{server}* corresponds to the server name or IP address for your IBM® SPSS® Collaboration and Deployment Services Repository
- *{port}* is the port number on which to access the IBM SPSS Collaboration and Deployment Services Repository
- *{config_id}* is the identifier for the scoring configuration. You can obtain the identifiers for the available scoring configurations by using a configuration request. For more information, see the topic [Configuration GET](#) on p. 10.
- *{name}* identifies the metric item whose value is being retrieved. You can obtain a list of available metric items by using a metric item request. For more information, see the topic [Metric item GET](#) on p. 24.

Table 2-7
Request header values

Name	Value
Content-type	application/json
Authorization	Authentication string encoded using Base64
Client-Accept-Language	Tag defining the language for the response information. This field takes precedence over the Accept-Language field.
Accept-Language	Tag defining the language for the response information

The HTTP request body should be empty.

Response

A successful request returns the following information:

- Status code 200
- Response body in JSON format containing the metric value for the request. The response object has the following general structure:

```
{  
  "value":<number>  
}
```

For more information, see the topic [The metricValue object](#) in Chapter 3 on p. 38.

An unsuccessful request returns a response code other than 200.

Example

This request retrieves the value of the *CONFIGURATION_UPTIME* metric for the *KMeans* scoring configuration.

```
GET http://cdssvr:80/scoring/rest/configuration/KMeans/metric/CONFIGURATION_UPTIME
```

This is the response object, indicating that the scoring configuration has been available for scoring for 38 seconds.

```
{  
  "value":38.0  
}
```

JSON reference for the RESTful Scoring Service

Describes all JSON objects used in the RESTful Scoring Service REST API.

The configurationReference object

Describes a scoring configuration available in the IBM® SPSS® Collaboration and Deployment Services Repository. This object contains a reference to the model on which the configuration is based, as well as the identifier of the scoring configuration itself.

The [Configuration GET](#) method returns an array of configurationReference objects in the response body.

Table 3-1
Child elements for configurationReference

Element name	Value type	Description
id	string	Non-localized identifier for the configuration. Use the id value in URL requests to identify a specific configuration.
state	string	Indicates the current status of the configuration. A state value of <i>ACTIVE</i> indicates the configuration is consuming resources and can process incoming scoring requests. In contrast, a state value of <i>SUSPENDED</i> indicates the configuration is not consuming resources and cannot process any scoring requests.
modelReference	object	Identifies the model used by a scoring configuration.
configurationStatus	object	Reports the status of a scoring configuration.

Example

The following array includes two configurationReference objects. The first has an identifier of AC2 and uses latest version of the file *multiple_Input_Double_updated.str* for scoring requests. The second has an identifier of *withRTDPD* and uses the version of the file *MultipleSourceNodes.str* that is labeled *PRODUCTION*. The configurationStatus object for the *withRTDPD* configuration indicates that the configuration is unable to process any scoring requests due to a problem with the specified data provider.

```
[
  {
    "state":"ACTIVE",
    "modelReference":{
      "resourcePath":"\june\multiple_Input_Double_updated.str",
      "label":"LATEST",
      "id":"09775272cf8ee6e5000001328016b69284dc"
    },
    "configurationStatus":{
      "message":"Started",
      "statusCode":"INFORMATION"
    },
    "id":"AC2"
  },
  {
    "state":"ACTIVE",
    "modelReference":{
      "resourcePath":"\july\MultipleSourceNodes.str",
      "label":"PRODUCTION",
      "id":"091ed68980eebc90000001317470503d84a5"
    },
    "configurationStatus":{
      "message":"Unable to load Data Provider Information settings Reason:
      com.spss.scoring.internal.engine.ScoringEngineException: Data Provider Definition
      failed validation",
      "statusCode":"ERROR"
    },
    "id":"withRTDPD"
  }
]
```

The modelReference element

Provides all of the information necessary to locate a specific version of a model file in the IBM® SPSS® Collaboration and Deployment Services Repository.

Table 3-2
Child elements for modelReference

Element name	Value type	Description
id	string	Non-localized identifier for the scoring model
label	string	The label corresponding to the version of the model file
resourcePath	string	Full path to the file in the content repository

Example

The following `modelReference` element corresponds to the latest version of the file `multiple_Input_Double_updated.str` in the `june` folder. The path includes the backslash (\) escape character before each slash (/) to optimize client processing.

```

"modelReference":{
  "resourcePath":"\\june\\multiple_Input_Double_updated.str",
  "label":"LATEST",
  "id":"09775272cf8ee6e500001328016b69284dc"
}

```

The configurationStatus element

Provides information about the current status of a scoring configuration.

Table 3-3

Child elements for configurationStatus

Element name	Value type	Description
message	string	Describes the status of the configuration.
statusCode	string	Indicates the type of message. Valid values include the following strings: <ul style="list-style-type: none"> ■ INFORMATION. The message is informational, describing the current status of the configuration in the system. ■ WARNING. A problem exists that may affect the configuration results. ■ ERROR. A problem exists that prevents the configuration from processing requests.

Example

The following configurationStatus element indicates that the configuration has started and is available for processing scoring requests.

```

"configurationStatus":{
  "message":"Started",
  "statusCode":"INFORMATION"
}

```

The metadataResult object

Describes the input and output fields used by a scoring configuration.

The [Metadata GET](#) method returns the metadataResult object in the response body.

Table 3-4

Child elements for metadataResult

Element name	Value type	Description
metadataInputTable	object array	The input tables for a scoring configuration

Element name	Value type	Description
metadataContextTable	object array	The context tables for a scoring configuration
metadataOutputField	object array	The output fields for a scoring configuration

Example

The following `metadataResult` object contains one input table, *DRUG*, consisting of three input fields. A scoring request returns four output fields. The empty `metadataContextTable` object indicates that the configuration does not use any context data.

```
{
  "metadataInputTable":[
    {
      "name":"DRUG",
      "id":"id5JAYKBB9IXD",
      "metadataInputField":[
        {
          "categoricalValues":[],"isRequired":true,"type":"long",
          "name":"Age","description":"Age"
        },
        {
          "categoricalValues":[],"isRequired":true,"type":"string",
          "name":"BP","description":"BP"
        },
        {
          "categoricalValues":[],"isRequired":true,"type":"string",
          "name":"Cholesterol","description":"Cholesterol"
        }
      ]
    }
  ],
  "metadataOutputField":[
    {
      "categoricalValues":[],"isReturned":true,"type":"string",
      "name":"BP","description":"BP"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"string",
      "name":"Cholesterol","description":"Cholesterol"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"string",
      "name":"$O-Anomaly","description":"$O-Anomaly"
    },
    {
      "categoricalValues":[],"isReturned":true,"type":"double",
      "name":"$O-AnomalyIndex","description":"$O-AnomalyIndex"
    }
  ],
  "metadataContextTable":[]
}
```

```
}

```

The `metadataInputTable` element

Describes an input table for a scoring configuration.

Table 3-5

Child elements for `metadataInputTable`

Element name	Value type	Description
id	string	Non-localized identifier for the input table
name	string	Name, possibly localized, for the input table
<code>metadataInputField</code>	object array	The input fields used for scoring

Example

The following `metadataInputTable` element describes an input table named *DRUG*. The table consists of the input fields *Age*, *BP*, and *Cholesterol*.

```
"metadataInputTable":[
  {
    "name":"DRUG",
    "id":"id5JAYKBB9IXD",
    "metadataInputField":[
      {
        "categoricalValues":[],"isRequired":true,"type":"long",
        "name":"Age","description":"Age"
      },
      {
        "categoricalValues":[],"isRequired":true,"type":"string",
        "name":"BP","description":"BP"
      },
      {
        "categoricalValues":[],"isRequired":true,"type":"string",
        "name":"Cholesterol","description":"Cholesterol"
      }
    ]
  }
]
```

The `metadataInputField` element

Describes the scoring input fields for a model. Each object in the array corresponds to an input field. For models that are based upon an Application View, the objects in this array should match the real time Data Provider Definition key and context data inputs. The inputs that make up the Real Time Data Provider Definition key are required, whereas the other inputs are optional. Any missing, but optional, fields may result in an exception. For models that are not based upon an Application View, the objects in this array come directly from the model and all fields will be required.

Table 3-6
Child elements for `metadataInputField`

Element name	Value type	Description
<code>description</code>	string	Optional description of the field
<code>isRequired</code>	boolean	Indicates whether or not a value for the field is required for scoring requests
<code>name</code>	string	Name of the field
<code>type</code>	string	Data type of the field. <ul style="list-style-type: none"> ■ boolean ■ date ■ daytime ■ decimal ■ double ■ float ■ integer ■ long ■ string ■ timestamp
<code>categoricalValues</code>	string array	Set of expected values, if any, for a categorical field

Example

The following `metadataInputField` element consists of an array of objects corresponding to the three fields in the table. The first input field, *Age*, has a type of *long*. The other two fields, *BP* and *Cholesterol*, are both of the *string* type. Values for all three fields are required in scoring requests.

```
"metadataInputField":[
  {
    "categoricalValues":[],"isRequired":true,"type":"long",
    "name":"Age","description":"Age"
  },
  {
    "categoricalValues":[],"isRequired":true,"type":"string",
    "name":"BP","description":"BP"
  },
  {
    "categoricalValues":[],"isRequired":true,"type":"string",
    "name":"Cholesterol","description":"Cholesterol"
  }
]
```

The `metadataContextTable` element

Describes a context table for a scoring configuration.

Table 3-7
Child elements for `metadataContextTable`

Element name	Value type	Description
table	string	Name for the context table
<code>contextColumn</code>	object array	The context data columns for the context table

The contextColumn element

Describes the context data columns for a context table.

Table 3-8
Child elements for `contextColumn`

Element name	Value type	Description
description	string	Optional description of the field
name	string	Name of the field
type	string	Data type of the field. <ul style="list-style-type: none"> ■ boolean ■ date ■ daytime ■ decimal ■ double ■ float ■ integer ■ long ■ string ■ timestamp
categoricalValues	string array	Set of expected values, if any, for a categorical field

The metadataOutputField element

Describes the output fields for a scoring configuration. Each object in this array corresponds to an output field. The model definition is directly responsible for the objects in this array.

Table 3-9
Child elements for `metadataOutputField`

Element name	Value type	Description
description	string	Optional description of the field
isReturned	boolean	Indicates if the output is returned in a score request. A true value is used if the output is returned in a score request, false otherwise. If this value is not present, the output will be returned.
name	string	Name of the field

Element name	Value type	Description
type	string	Data type of the field. <ul style="list-style-type: none"> ■ boolean ■ date ■ daytime ■ decimal ■ double ■ float ■ integer ■ long ■ string ■ timestamp
categoricalValues	string array	Set of expected values, if any, for a categorical field

Example

The following `metadataOutputField` element consists of an array of objects corresponding to the fields returned as output from a scoring request. The first field `Age`, has a type of `long`. The other field, `$O-Anomaly`, is of the `string` type.

```
"metadataOutputField":[
  {
    "categoricalValues":[],"isReturned":true,"type":"long",
    "name":"Age","description":"Age"
  },
  {
    "categoricalValues":[],"isReturned":true,"type":"string",
    "name":"$O-Anomaly","description":"$O-Anomaly"
  }
]
```

The `metricItem` object

Identifies all performance measurements available for a scoring configuration. Each object in this array corresponds to a performance metric. You can retrieve the value for a specific metric by supplying the metric identifier in a request URL. For more information, see the topic [Metric value GET](#) in Chapter 2 on p. 28.

The [Metric item GET](#) method returns the `metricItem` object in the response body.

Table 3-10
Child elements for `metricItem`

Element name	Value type	Description
id	string	Non-localized identifier for the item
name	string	Name, possibly localized, for the item

Element name	Value type	Description
scale	number	Identifies the number of digits to the right of the decimal point for the item value
unit	string	Unit of measurement for the item, such as milliseconds

Example

The following `metricItem` object corresponds to the *Service Scores* metric for a configuration. The `unit` property indicates the metric is measured in *scores*. The `scale` property reports that the value of this metric has 0 decimal places. You can retrieve the current value of this metric by including the `id` value, `SERVICE_TOTAL_SCORES`, in a request URL.

```
{
  "unit":"scores",
  "scale":0,
  "id":"SERVICE_TOTAL_SCORES",
  "name":"Service Scores"
}
```

The `metricValue` object

Reports the current value for a performance metric. For more information, see the topic [The `metricItem` object](#) on p. 37.

The [Metric value GET](#) method returns the `metricValue` object in the response body.

Table 3-11
Child elements for `metricValue`

Element name	Value type	Description
value	number	The value for a metric item

Example

The `value` element of the following `metricValue` object indicates the current value of the requested metric item is 38.

```
{
  "value":38.0
}
```

The `scoreRequest` object

Represents a request for a score as an object containing the input values for a scoring model. You can obtain the structure for the input tables by using the [Metadata GET](#) method.

The [Score POST](#) method uses the `scoreResult` object in the request body.

Table 3-12
Child elements for *scoreRequest*

Element name	Value type	Description
id	string	Identifier for the scoring configuration being used to generate scores
requestInputTable	object array	Input tables for the scoring model
context	object array	Context tables containing indirect inputs for the scoring model

Example

The following object corresponds to a scoring request for the *config_1* scoring configuration. The request includes one table named *Table 1* that contains a single input row.

```
{
  "id": "config_1",
  "requestInputTable": [
    {
      "name": "Table 1",
      "requestInputRow": [
        "input": [
          {"name": "Age", "value": "1"},
          {"name": "BP", "value": "1"},
          {"name": "Cholesterol", "value": "1"},
          {"name": "Drug", "value": "1"},
          {"name": "K", "value": "1"},
          {"name": "Na", "value": "1"},
          {"name": "Gender", "value": "1"}
        ]
      ]
    }
  ],
  "context": []
}
```

The *requestInputTable* element

Describes the input tables used in a score request. The value for this element is an object array in which each object represents an input table.

Table 3-13
Child elements for *requestInputTable*

Element name	Value type	Description
name	string	Name, possibly localized, for the input table
requestInputRow	element array	One or more rows of input data

Example

The value for the following `requestInputTable` element corresponds to an array containing one object. That object represents a table named *Table 1* that contains a single input row consisting of the values for seven input fields.

```
"requestInputTable":[
  {
    "name":"Table 1",
    "requestInputRow":[
      "input":[
        {"name":"Age","value":"1"},
        {"name":"BP","value":"1"},
        {"name":"Cholesterol","value":"1"},
        {"name":"Drug","value":"1"},
        {"name":"K","value":"1"},
        {"name":"Na","value":"1"},
        {"name":"Gender","value":"1"}
      ]
    ]
  }
]
```

If the model required data from two input tables, the array would contain multiple objects. For example, the following element represents two tables.

```
"requestInputTable":[
  {
    "name":"Table 1",
    "requestInputRow":[
      "input":[
        {"name":"Age","value":"1"},
        {"name":"BP","value":"1"},
        {"name":"Cholesterol","value":"1"},
      ]
    ]
  },
  {
    "name":"Table 2",
    "requestInputRow":[
      "input":[
        {"name":"Drug","value":"1"},
        {"name":"K","value":"1"},
        {"name":"Na","value":"1"},
        {"name":"Gender","value":"1"}
      ]
    ]
  }
]
```

The requestInputRow element

Represents rows of input values used to produce score results. The value for this element is an element array in which each element represents an input row.

Table 3-14
Child elements for requestInputRow

Element name	Value type	Description
input	object array	Values for input fields

Example

The value for the following requestInputRow element corresponds to an array containing one element. The element represents a single input row consisting of the values for seven input fields.

```
"requestInputRow":[
  "input":[
    {"name":"Age","value":"1"},
    {"name":"BP","value":"1"},
    {"name":"Cholesterol","value":"1"},
    {"name":"Drug","value":"1"},
    {"name":"K","value":"1"},
    {"name":"Na","value":"1"},
    {"name":"Gender","value":"1"}
  ]
]
```

To include multiple input rows, add input elements to the requestInputRow element. For example, the following element contains two rows of input data.

```
"requestInputRow":[
  "input":[
    {"name":"Age","value":"1"},
    {"name":"BP","value":"1"},
    {"name":"Cholesterol","value":"1"},
    {"name":"Drug","value":"1"},
    {"name":"K","value":"1"},
    {"name":"Na","value":"1"},
    {"name":"Gender","value":"1"}
  ],
  "input":[
    {"name":"Age","value":"3"},
    {"name":"BP","value":"2"},
    {"name":"Cholesterol","value":"4"},
    {"name":"Drug","value":"2"},
    {"name":"K","value":"3"},
    {"name":"Na","value":"1"},
    {"name":"Gender","value":"2"}
  ]
]
```

The input element

Defines the scoring input data values. The value for this element is an object array in which each object represents a value for an input field. The format of the values must match the data type, as specified in the scoring configuration metadata.

Table 3-15
Child elements for input

Element name	Value type	Description
name	string	Name of the input item
value	string	A value, in string representation. If this attribute is not specified, the value is considered to be null. The text representation of the numeric types is obvious, but several types are not. The format of the non-numeric types must be as follows: <ul style="list-style-type: none"> ■ boolean=<i>true</i>(case insensitive) or <i>1</i> or <i>false</i>(case insensitive) or <i>0</i> ■ date=<i>yyyy-MM-dd</i> ■ daytime=<i>HH:mm:ss</i> ■ timestamp=<i>yyyy-MM-dd'T'HH:mm:ss</i>

Example

The value for the following input element corresponds to an array containing the values for seven input fields.

```
"input":[
  {"name":"Age","value":"1"},
  {"name":"BP","value":"1"},
  {"name":"Cholesterol","value":"1"},
  {"name":"Drug","value":"1"},
  {"name":"K","value":"1"},
  {"name":"Na","value":"1"},
  {"name":"Gender","value":"1"}
]
```

The context element

Describes the indirect inputs to the scoring model. The value for this element is an object array in which each object represents an input table.

Table 3-16
Child elements for context

Element name	Value type	Description
name	string	Name, possibly localized, for the context table

Element name	Value type	Description
columnName	string array	An ordered list of column names
rowValues	object array	A row of table data, following the order in the columnName list

Example

The value for the following context element is an array containing an object representing a context table named *Context 1*. The table contains two columns named *Education* and *Area* having values of 2 and 3 respectively.

```
"context":[
  {
    "name":"Context 1",
    "columnName":["Education","Area"],
    "rowValues":[
      {
        "value":[
          {"value":"2"},
          {"value":"3"}
        ]
      }
    ]
  }
]
```

The rowValues element

Represents a row of context field values. The value for this element is an object array in which each object represents a set of values. The order of the values corresponds to the order of the field names in the [columnName element](#).

Table 3-17
Child elements for rowValues

Element name	Value type	Description
value	object array	Values for the indirect model inputs

Example

The value for the following rowValues element corresponds to an array containing a single object representing two values.

```
"rowValues":[
  {
    "value":[
      {"value":"2"},
      {"value":"3"}
    ]
  }
]
```

The value element

Provides values for the context fields. The value for this element is an object array in which each object represents a value.

Table 3-18
Child elements for value

Element name	Value type	Description
value	string	<p>A value, in string representation. If this attribute is not specified, the value is considered to be null. The text representation of the numeric types is obvious, but several types are not. The format of the non-numeric types must be as follows:</p> <ul style="list-style-type: none"> ■ boolean=<i>true</i>(case insensitive) or <i>1</i> or <i>false</i>(case insensitive) or <i>0</i> ■ date=<i>yyyy-MM-dd</i> ■ daytime=<i>HH:mm:ss</i> ■ timestamp=<i>yyyy-MM-dd'T'HH:mm:ss</i>

Example

The array for the following value element contains two values.

```
"value":[
  {"value":"2"},
  {"value":"3"}
]
```

The scoreResult object

Represents the results of a scoring request as an object containing the output specified by the scoring configuration.

The [Score POST](#) method returns the scoreResult object in the response body.

Table 3-19
Child elements for scoreResult

Element name	Value type	Description
id	string	Internal identifier for the score result
columnNames	element array	Names of the scoring functions included in the result
rowValues	object array	Values for the scoring functions

Element name	Value type	Description
returnedRequestInputTable	object array	Tables containing the input values on which the scores are based. The presence of this array is determined by the scoring configuration settings.
returnedDPDOutputTable	object array	Tables containing data that was output from a data provider definition. The presence of this array is determined by the scoring configuration settings.

Example

The following object corresponds to a scoring response for a model having seven input fields. The response contains a value of 2 for the *Prediction* scoring function and a value of 1.34153 for the *Confidence* scoring function.

```
{
  "id":"954d9db3-049f-464c-b218-eadd628cf349",
  "columnNames":[
    "name":["Prediction","Confidence"]
  ],
  "rowValues":[
    {
      "reserved":null,
      "value":{"value":"2"},{"value":"1.3415297614991626"}
    }
  ],
  "returnedRequestInputTable":[
    {
      "name":"Table 1",
      "id":"Table 1",
      "returnedRequestInputRow":[
        {
          "returnedRequestInputValue":[
            {"name":"Age","type":"double","value":"1"},
            {"name":"BP","type":"string","value":"1"},
            {"name":"Cholesterol","type":"string","value":"1"},
            {"name":"Drug","type":"string","value":"1"},
            {"name":"K","type":"double","value":"1"},
            {"name":"Na","type":"double","value":"1"},
            {"name":"Gender","type":"string","value":"1"}
          ]
        }
      ]
    }
  ],
  "returnedDPDOutputTable":[]
}
```

The *columnNames* element

Identifies the scoring functions included in a score result. The value for this element is an element array in which each element represents a set of scoring functions.

Table 3-20
Child elements for *columnNames*

Element name	Value type	Description
name	string array	Scoring function names.

Example

The value for the following *columnNames* element identifies two scoring functions, *Prediction* and *Confidence*.

```
"columnNames":[
  "name":["Prediction","Confidence"]
]
```

The *rowValues* element

Represents a row of scoring result values. The value for this element is an object array in which each object represents a set of scoring values.

Table 3-21
Child elements for *rowValues*

Element name	Value type	Description
reserved	string	Internal string that is currently unused.
value	object array	Scoring function values

Example

The value for the following *rowValues* element reports scoring function values of 2 and 1.34153.

```
"rowValues":[
  {
    "reserved":null,
    "value":[{"value":"2"},{"value":"1.3415297614991626"}]
  }
]
```

The *value* element

Provides the score result values. The value for this element is an object array in which each object represents a scoring value.

Table 3-22
Child elements for value

Element name	Value type	Description
value	string	A scoring value, in string representation.

Example

The value for the following value element is an array containing two objects. The first object reports a value of 4 and the second object reports a value of 2.434156.

```
"value":[
  {"value":"4"},
  {"value":"2.434156"}
]
```

The returnedRequestInputTable element

Describes the input tables used in a score request. The value for this element is an object array in which each object represents an input table.

Table 3-23
Child elements for returnedRequestInputTable

Element name	Value type	Description
id	string	Non-localized identifier for the input table
name	string	Name, possibly localized, for the input table
returnedRequestInputRow	object array	One or more rows of input values used for scoring

Example

The value for the following returnedRequestInputTable element is an array containing one object representing a table named *Table 1*.

```
"returnedRequestInputTable":[
  {
    "name":"Table 1",
    "id":"Table 1",
    "returnedRequestInputRow":[
      {
        "returnedRequestInputValue":[
          {"name":"Age","type":"double","value":"1"},
          {"name":"BP","type":"string","value":"1"},
          {"name":"Cholesterol","type":"string","value":"1"},
          {"name":"Drug","type":"string","value":"1"},
          {"name":"K","type":"double","value":"1"},
          {"name":"Na","type":"double","value":"1"},
          {"name":"Gender","type":"string","value":"1"}
        ]
      }
    ]
  }
]
```

```

    }
  ]
}
]

```

The returnedRequestInputRow element

Represents rows of input values used to produce score results. The value for this element is an object array in which each object represents an input row.

Table 3-24

Child elements for returnedRequestInputRow

Element name	Value type	Description
returnedRequestInputValue	object array	Values used to produce the score results. A value might be null.

Example

The value for the following returnedRequestInputRow element is an array containing one object representing a single row of input values.

```

"returnedRequestInputRow":[
  {
    "returnedRequestInputValue":[
      {"name":"Age","type":"double","value":"1"},
      {"name":"BP","type":"string","value":"1"},
      {"name":"Cholesterol","type":"string","value":"1"},
      {"name":"Drug","type":"string","value":"1"},
      {"name":"K","type":"double","value":"1"},
      {"name":"Na","type":"double","value":"1"},
      {"name":"Gender","type":"string","value":"1"}
    ]
  }
]

```

The returnedRequestInputValue element

Represents the input values used to produce score results. The value for this element is an object array in which each object represents an input field.

Table 3-25

Child elements for returnedRequestInputValue

Element name	Value type	Description
name	string	Name of the input item

Element name	Value type	Description
type	string	Data type of the field. Valid entries include the following values: <ul style="list-style-type: none"> ■ boolean ■ date ■ daytime ■ decimal ■ double ■ float ■ integer ■ long ■ string ■ timestamp
value	string	A value, in string representation.

Example

The value for the following `returnedRequestInputValue` element is an array containing several objects. Each object represents the value for a single input field. The inputs consist of four string fields and three double fields.

```
"returnedRequestInputValue":[
  {"name":"Age","type":"double","value":"1"},
  {"name":"BP","type":"string","value":"2"},
  {"name":"Cholesterol","type":"string","value":"3"},
  {"name":"Drug","type":"string","value":"4"},
  {"name":"K","type":"double","value":"3"},
  {"name":"Na","type":"double","value":"2"},
  {"name":"Gender","type":"string","value":"1"}
]
```

The returnedDPDOutputTable element

Describes tables of input values that were generated from a data provider definition. The value for this element is an object array in which each object represents a table of generated data.

Table 3-26

Child elements for `returnedDPDOutputTable`

Element name	Value type	Description
id	string	Non-localized identifier for the table
name	string	Name, possibly localized, for the table
returnedDPDOutputRow	object array	One or more rows of data from a data provider definition

Example

The value for the following `returnedDPDOutputTable` element is an array containing one object representing a table named *DPD 1*.

```
"returnedDPDOutputTable":[
  {
    "name":"DPD 1",
    "id":"DPD 1",
    "returnedDPDOutputRow":[
      {
        "returnedDPDOutputValue":[
          {"name":"Education","type":"double","value":"1"},
          {"name":"Area","type":"string","value":"1"},
          {"name":"Loyalty","type":"string","value":"1"}
        ]
      }
    ]
  }
]
```

The returnedDPDOutputRow element

Represents a row of data generated from a data provider definition. The value for this element is an object array in which each object represents a row of inputs that were output by the data provider definition.

Table 3-27

Child elements for `returnedDPDOutputRow`

Element name	Value type	Description
<code>returnedDPDOutputValue</code>	object array	Values generated from a data provider definition that were used to produce the score results.

Example

The value for the following `returnedDPDOutputRow` element is an array containing one object representing a single row of generated values.

```
"returnedDPDOutputRow":[
  {
    "returnedDPDOutputValue":[
      {"name":"Education","type":"double","value":"1"},
      {"name":"Area","type":"string","value":"1"},
      {"name":"Loyalty","type":"string","value":"1"}
    ]
  }
]
```


The returnedDPDOutputValue element

Represents the values generated from a data provider definition that were used to produce score results. The value for this element is an object array in which each object represents a generated field.

Table 3-28
Child elements for returnedDPDOutputValue

Element name	Value type	Description
name	string	The name of the item
type	string	Data type of the field. Valid entries include the following values: <ul style="list-style-type: none"> ■ boolean ■ date ■ daytime ■ decimal ■ double ■ float ■ integer ■ long ■ string ■ timestamp
value	string	A value, in string representation.

Example

The value for the following returnedDPDOutputValue element is an array containing several objects. Each object represents the value for a single input field. The inputs consist of two string fields and one double field.

```
"returnedDPDOutputValue":[
  {"name":"Education","type":"double","value":"1"},
  {"name":"Area","type":"string","value":"1"},
  {"name":"Loyalty","type":"string","value":"1"}
]
```

The scoringServiceDetails object

Provides details about the capabilities of the RESTful Scoring Service. In addition to reporting the service version, this object includes information about each individual score provider currently installed.

The [Service GET](#) method returns the scoringServiceDetails object in the response body.

Table 3-29
Child elements for *scoringServiceDetails*

Element name	Value type	Description
version	string	The service version number
scoreProviderDetails	object array	Information about the score providers available in the system

Example

The following object indicates that version *5.0.0.0.155* of the RESTful Scoring Service is in use. The *Modeler Score Provider* score provider is available for scoring requests using IBM® SPSS® Modeler streams or scenarios. In addition, the *Score Provider - SmartScore* score provider is available for scoring requests using PMML files.

```
{
  "version":"5.0.0.0.155",
  "scoreProviderDetails":[
    {
      "supportedMimeTypes":["application/x-vnd.spss-scenario","application/x-vnd.spss-clementine-stream"],
      "version":"1.0",
      "name":"Modeler Score Provider",
      "id":"091ed7cda16d295c000001327c563c4593de"
    },
    {
      "supportedMimeTypes":["application/x-vnd.spss-pmml"],
      "version":"1.0",
      "name":"Score Provider - SmartScore",
      "id":"091ed7cd82a9085b000001327c118593a316"
    }
  ]
}
```

The *scoreProviderDetails* element

Provides details about the score providers in the system. The value of this element consists of an object array in which each object corresponds to a score provider. The following table identifies the elements comprising each object in the array.

Table 3-30
Child elements for *scoreProviderDetails*

Element name	Value type	Description
id	string	Non-localized identifier for the score provider
name	string	Name, possibly localized, for the score provider
version	string	The score provider version number
supportedMimeValue types	string array	One or more MIME types indicating the type of files that can be used with the score provider

Example

The following `scoreProviderDetails` element consists of an array containing one object. The object corresponds to version *1.0* of the *Modeler Score Provider* score provider used for scenario and stream files from IBM® SPSS® Modeler.

```
"scoreProviderDetails":[
  {
    "supportedMimeTypes":[
      "application/vnd.spss-scenario",
      "application/vnd.spss-clementine-stream"
    ],
    "version":"1.0",
    "name":"Modeler Score Provider",
    "id":"091ed7cda16d295c000001327c563c4593de"
  }
]
```

Notices

This information was developed for products and services offered worldwide.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japan.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, [ibm.com](http://www.ibm.com), and SPSS are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.



Index

- anomaly detection models
 - scoring, 17
- arrays
 - in JSON, 6
- authorization
 - requests, 3
- Bayesian networks
 - scoring, 17
- cache hits
 - for scoring configurations, 24
- cache misses
 - for scoring configurations, 24
- categoricalValues element
 - for contextColumn elements, 36
 - for metadataInputField elements, 35
 - for metadataOutputField elements, 36
- cluster models
 - scoring, 17
- columnName element
 - for context elements, 42
- columnNames elements, 44, 46
- computation time
 - for scoring configurations, 24
- computation wait time
 - for scoring configurations, 23
- confidences
 - in scoring, 17
- configuration resources, 10
 - GET method, 10
 - requests, 10
 - responses, 11
 - retrieving, 10
 - URL, 10
- configuration responses
 - configurationReference objects, 30
 - configurationStatus elements, 32
 - modelReference elements, 31
- configurationReference objects, 30
- configurations
 - for scoring models, 10
- configurationStatus elements, 32
 - for configurationReference objects, 30
- content type
 - requests, 3
- context elements, 39, 42
- contextColumn elements, 36
- Cox regression models
 - scoring, 17
- cumulative hazard values
 - in scoring, 17
- data access time
 - for scoring configurations, 23
- data initialization time
 - for scoring configurations, 23
- description element
 - for contextColumn elements, 36
 - for metadataInputField elements, 35
 - for metadataOutputField elements, 36
- discriminant models
 - scoring, 17
- elements
 - in JSON, 5
- general linear models
 - scoring, 17
- generalized linear models
 - scoring, 17
- GET method
 - for configuration resources, 10
 - for metadata resources, 13
 - for metric item resources, 25
 - for metric value resources, 28
 - for service resources, 8
 - response headers, 5
- headers
 - for requests, 2
 - for responses, 4
- id element
 - for configurationReference objects, 30
 - for metadataInputTable elements, 34
 - for metricItem objects, 37
 - for modelReference elements, 31
 - for returnedDPDOutputTable elements, 49
 - for returnedRequestInputTable elements, 47
 - for scoreProviderDetails elements, 52
- input elements, 41–42
- isRequired element
 - for metadataInputField elements, 35
 - for metadataOutputField elements, 36
- JSON, 4–5
 - arrays, 6
 - elements, 5
 - objects, 6
- Kohonen models
 - scoring, 17
- label element
 - for modelReference elements, 31
- latency
 - for scoring configurations, 23

- legal notices, 54
- linear regression models
 - scoring, 17
- localization
 - requests, 3
- log queue
 - for scoring configurations, 24
- log serialization
 - for scoring configurations, 24
- logistic regression models
 - scoring, 17

- message element
 - for configurationStatus elements, 32
- metadata resources, 12
 - GET method, 13
 - requests, 13
 - responses, 13
 - retrieving, 13
 - URL, 13
- metadata responses
 - contextColumn elements, 36
 - metadataContextTable elements, 35
 - metadataInputField elements, 34
 - metadataInputTable elements, 34
 - metadataOutputField elements, 36
 - metadataResult objects, 32
- metadataContextTable elements, 32, 35
- metadataInputField elements, 34
- metadataInputTable elements, 32, 34
- metadataOutputField elements, 32, 36
- metadataResult objects, 32
- metric item resources, 23
 - GET method, 25
 - requests, 25
 - responses, 25
 - retrieving, 24
 - URL, 25
- metric item responses
 - metricItem objects, 37
- metric value resources, 28
 - GET method, 28
 - requests, 28
 - responses, 29
 - retrieving, 28
 - URL, 28
- metric value responses
 - metricValue objects, 38
- metricItem objects, 37
- metrics
 - for scoring performance, 23
- metricValue objects, 38
- modelReference elements, 31
 - for configurationReference objects, 30

- naive Bayes models
 - scoring, 17

- name element
 - for columnNames elements, 46
 - for context elements, 42
 - for contextColumn elements, 36
 - for input elements, 42
 - for metadataInputField elements, 35
 - for metadataInputTable elements, 34
 - for metadataOutputField elements, 36
 - for metricItem objects, 37
 - for requestInputTable elements, 39
 - for returnedDPDOutputTable elements, 49
 - for returnedDPDOutputValue elements, 51
 - for returnedRequestInputTable elements, 47
 - for returnedRequestInputValue elements, 48
 - for scoreProviderDetails elements, 52
- nearest neighbors
 - in scoring, 17
- neural networks
 - scoring, 17
- node identifiers
 - in scoring, 17

- objects
 - in JSON, 6
- ordinal multinomial regression models
 - scoring, 17

- POST method
 - for score resources, 19
- predicted values, 17
- probabilities
 - in scoring, 17

- requestInputRow elements, 39, 41
- requestInputTable elements, 39
- requests
 - authorization, 3
 - content type, 3
 - for configuration resources, 10
 - for metadata resources, 13
 - for metric item resources, 25
 - for metric value resources, 28
 - for score resources, 19
 - for service resources, 8
 - headers, 2
 - localization, 3
 - URL, 1
- resourcePath element
 - for modelReference elements, 31
- responses
 - for configuration requests, 11
 - for metadata requests, 13
 - for metric item requests, 25
 - for metric value requests, 29
 - for score requests, 20
 - for service requests, 9

- headers, 4
- retrieving
 - configuration resources, 10
 - metadata resources, 13
 - metric item resources, 24
 - metric value resources, 28
 - score resources, 19
 - service resources, 8
- returnedDPDOutputRow elements, 49–50
- returnedDPDOutputTable elements, 44, 49
- returnedDPDOutputValue elements, 50–51
- returnedRequestInputRow elements, 44, 47–48
- returnedRequestInputTable elements, 47
- returnedRequestInputValue elements, 48
- rowValues elements, 42–44, 46
- ruleset models
 - scoring, 17
- scale
 - for performance metrics, 24
- scale element
 - for metricItem objects, 37
- score providers, 8
- score requests
 - context elements, 42
 - input elements, 42
 - requestInputRow elements, 41
 - requestInputTable elements, 39
 - rowValues elements, 43
 - scoreRequest objects, 38
 - value elements, 44
- score resources, 17
 - POST method, 19
 - requests, 19
 - responses, 20
 - retrieving, 19
 - URL, 19
- score responses
 - columnNames elements, 46
 - returnedDPDOutputRow elements, 50
 - returnedDPDOutputTable elements, 49
 - returnedDPDOutputValue elements, 51
 - returnedRequestInputRow elements, 48
 - returnedRequestInputTable elements, 47
 - returnedRequestInputValue elements, 48
 - rowValues elements, 46
 - scoreResult objects, 44
 - value elements, 46
- scoreProviderDetails elements, 52
- scoreRequest objects, 38
- scoreResult objects, 44
- scoring
 - configurations, 10
- scoring configurations, 10
 - performance metrics, 23
- scoring functions, 17
- scoringServiceDetails objects, 51
- service resources, 8
 - GET method, 8
 - requests, 8
 - responses, 9
 - retrieving, 8
 - URL, 8
- service responses
 - scoreProviderDetails elements, 52
 - scoringServiceDetails objects, 51
- standard deviations
 - in scoring, 17
- state element
 - for configurationReference objects, 30
- statusCode element
 - for configurationStatus elements, 32
- support vector machine models
 - scoring, 17
- supportedMimeTypes element
 - for scoreProviderDetails elements, 52
- table element
 - for metadataContextTable elements, 36
- trademarks, 55
- tree models
 - scoring, 17
- type element
 - for contextColumn elements, 36
 - for metadataInputField elements, 35
 - for metadataOutputField elements, 36
 - for returnedDPDOutputValue elements, 51
 - for returnedRequestInputValue elements, 48
- unit element
 - for metricItem objects, 37
- uptime
 - for scoring configurations, 24
- URL
 - for configuration resources, 10
 - for metadata resources, 13
 - for metric item resources, 25
 - for metric value resources, 28
 - for requests, 1
 - for score resources, 19
 - for service resources, 8
- value element
 - for input elements, 42
 - for metricValue objects, 38
 - for returnedDPDOutputValue elements, 51
 - for returnedRequestInputValue elements, 48
 - for value elements, 44, 47
- value elements, 43–44, 46
- version element
 - for scoreProviderDetails elements, 52
 - for scoringServiceDetails objects, 52