IBM Cloud Pak for Business Automation Demos and Labs 2024

IBM Automation Decision Services
Manage Decisions and infuse Machine Learning

V 1.5 for (CP4BA 23.0.2)

Decisions
dba_decisions_l3_services@wwpdl.vnet.ibm.com

NOTICES

This information was developed for products and services offered in the USA.

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, MD-NC119
Armonk, NY 10504-1785
United States of America

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 CORPORATION 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

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.

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.

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at 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.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

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.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. 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.

© Copyright International Business Machines Corporation 2024.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

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

Table of Contents

1 Introduction 4
1.1 IBM Automation Decision Services 4
1.2 Lab Overview 5
1.3 Lab Setup Instructions 5
1.3.1 Prerequisites 5
1.3.2 Log in to the environment 6
2 Exercise 1: Modeling Decisions 7
2.1 Introduction 7
2.2 Exercise Instructions 7
2.2.1 Creating a project and importing a decision service 7
2.2.2 Exploring the decision service: data model and decision model 9
2.2.3 Updating decision logic: adding a node, a rule, editing a decision table 16
2.2.4 Validating the decision service 23
2.3 Summary 24
3 Exercise 2: Adding machine learning in the decision model 25
3.1 Introduction 25
3.2 Exercise instructions 25
3.2.1 Log in to your project 25
3.2.2 Creating a project and importing a Decision service 25
3.2.3 Testing the model before changes 27
3.2.4 Creating a Predictive Model and Map Initial Data 29
3.2.5 Add Additional Data Mapping 33
3.2.6 Validating the predictive model 39
3.2.7 Use the predictive model in the scoreboard decision model 42
3.2.8 Validating the final decision service 43
3.3 Summary 44
4 Exercise 3: Sharing and publishing decision services 45
4.1 Introduction 45
4.2 Exercise instructions 45
4.2.1 Log in to your project 45
4.2.2 Creating a project and importing a decision service 45
4.2.3 Exploring a decision operation 47
4.2.4 Creating a Git repository 47
4.2.5 Connecting your project to the Git repository 48
4.2.6 Sharing your decision service 49
4.2.7 Creating a version 52
4.2.8 Deploying your project 53
4.2.9 Executing your decision through the ADS runtime 54
4.2.10 Publishing your decision service through Business Automation Studio 54
4.3 Summary 56
5 Additional information 57
5.1 Explore the Samples 57
5.2 Consult Documentation and Communities 57

1 Introduction

In this Lab, you build business operational decisions to implement services evaluation and client risk assessment for a Client Onboarding solution. You capture and automate these decisions using IBM Automation Decision Services.

It includes three exercises. Each exercise can be done separately.

Duration: About 3 hours (each exercise is about 1 hour).

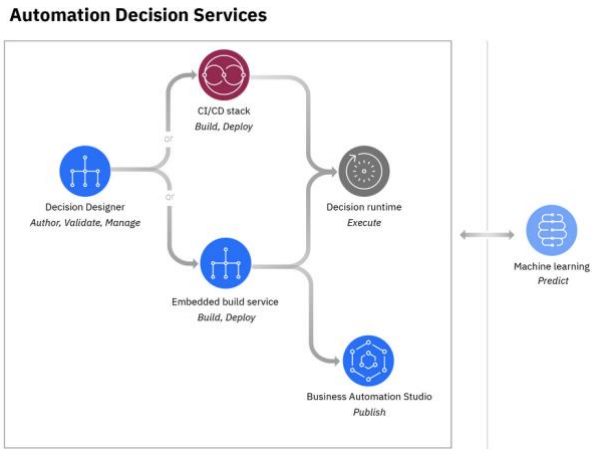
Audience: Anyone who wants to learn how to use Automation Decision Services.

1.1 IBM Automation Decision Services

IBM Automation Decision Services provides a comprehensive environment for authoring, managing, and running decision services. Operational decisions apply business policies, which are often influenced by numerous factors that can be both internal and external to an organization.

Part of the IBM Cloud Pak® for Business Automation platform, Automation Decision Services delivers advanced decision automation capabilities. Using Decision Designer in Business Automation Studio, business experts can model, author, and validate decisions in a low-code development environment. They can also infuse intelligence into business decisions by combining decision models and predictive models into unified decision services. They can collaborate with others in their organization by easily sharing and collaborating with projects in Business Automation Studio, backed by a central Git repository.

Automation Decision Services integrates with a continuous integration and delivery (CI/CD) stack. You can build and deploy decision services directly from Decision Designer. Deployed decision services can then either be published as automation services in Business Automation Studio, or invoked through the decision runtime.



For more information, see the IBM Documentation:
[IBM Automation Decision Services]
[What is Automation Decision Services]

1.2 Lab Overview

The end-to-end Client Onboarding solution explores a generic use case for onboarding new services to an existing client. Automation Decisions Services provides two decisions services that are integrated in the solution.

The first decision is to define the fees for the services the client is being onboarded to, and to suggest additional services that the client could be interested in, based on the information gathered. It is consumed in the end-to-end solution via an automation service from a Business Automation Application in the front office intake app.

The second decision provides input to determine the onboarding approval, the scoreboard. It implements a client risk evaluation based on multiple information sources such as the client profile, industry, and a client classification based on the client revenue. This decision combines descriptive and predictive models and is consumed in the end-to-end solution as an automated service in Workflow, during the approval step.

In this Lab you learn how to:

- Manage business decisions based on multiple data sources.
- Infuse intelligence into business decisions by adding a predictive model into the scorecard decision model to deliver the scoreboard of the client.
- Collaborate by sharing projects through a Git repository, build and deploy decision services from Decision Designer. Learn how to publish automation services in Business Automation Studio.

Integrating the automated services in other applications is not covered in this Lab. Look at the Workflow and Business Automation Application labs for this topic.

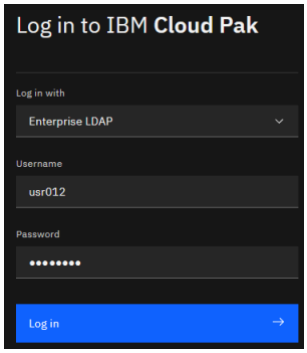
1.3 Lab Setup Instructions

1.3.1 Prerequisites

1. If you are performing this Lab as a part of an IBM event, access the document that lists the available systems and URLs along with login instructions. For this lab, you need to access:
    - **IBM Business Automation Studio** to access Decision Designer.
2. Download the file **ClientOnboardingLab.zip** from the **Lab Data** folder onto your computer.
3. Download the file **clientDefaultPaymentRS.xml** from the **Lab Data** folder onto your computer.
4. A GitHub account is needed for Exercise 3. You use a GitHub account to create a repository. If you do not have an account, create one in GitHub. Your work will be saved in your repository.

1.3.2 Log in to the environment

1. Launch **Business Automation Studio** (URL listed in the Lab access document, see prerequisites).
2. Select your authentication type: **Enterprise LDAP**
3. Enter your *Username* and *Password* then click on **Log in**



2 Exercise 1: Modeling Decisions

2.1 Introduction

This exercise is an introduction to the authoring environment of Automation Decision Services. You navigate in Decision Designer, which is the development environment for creating decision services. You discover the main concepts of Automation Decision Services by exploring and modifying a given decision service. The scenario of this decision service is to define the fees for services that the client is being onboarded to and to suggest additional services the client could be interested in, based on the information gathered.

2.2 Exercise Instructions

In this exercise you import a decision service and explore it:

- Create a **project** and import a **decision service**
- Explore the **data model** and the **decision model**
- Add a business rule and edit a decision table to update the **logic** used in the model
- Validate the decision service

Prerequisites: you need to download the provided file **ClientOnboardingLab.zip** from the Lab data folder.

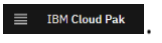
2.2.1 Creating a project and importing a decision service

In this step, you create a project in Business Automation Studio. You import a decision service from the zip prepared for this exercise.

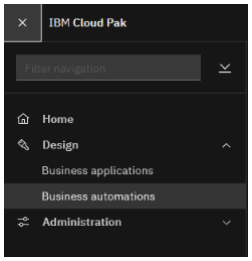
A project is a set of artifacts that share the same lifecycle and are grouped to solve a particular business problem. In this Lab you work with a decision automation project. It provides decision capabilities to help business experts to model and automate repeatable decisions.

If you are already connected to Automation Decision Services and have your project opened following a previous exercise, you can skip this section and continue in section Exploring the decision service: data model and decision model.

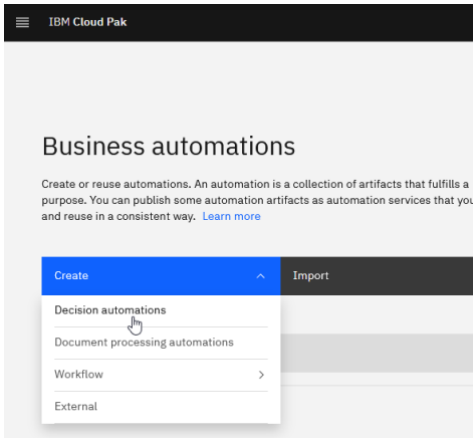
If you are starting the Lab with this exercise, do the following steps:

1. Log in to Business Automation Studio.
2. Click on the Navigation Menu on the top left corner 

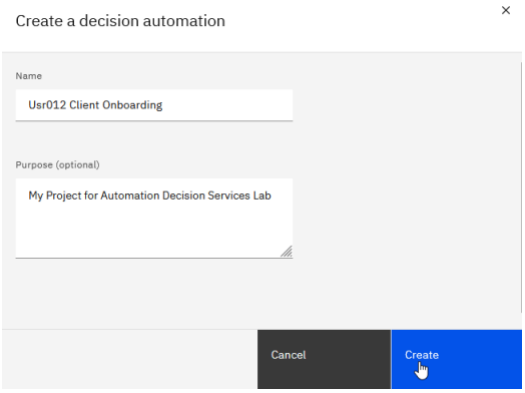
.
3. Expand **Design**, and then click on **Business automations**.



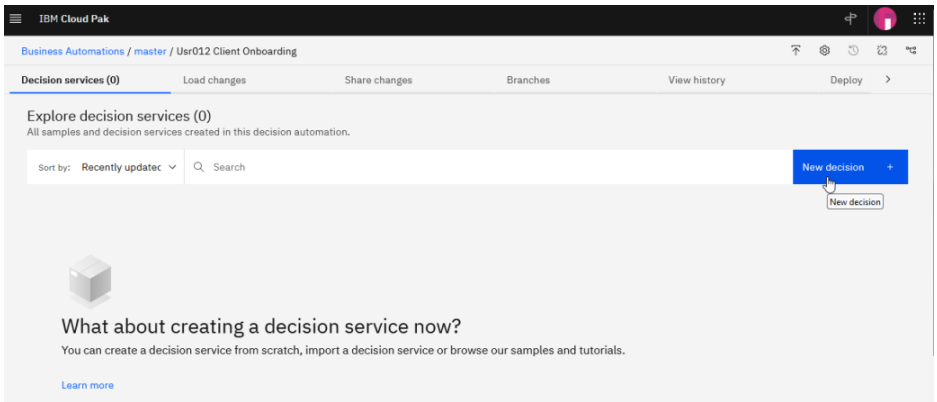
4. Click Create and select Decision automations.



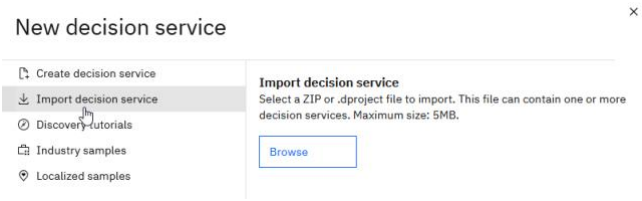
5. Enter a name for your project. Enter **UsrNNN Client Onboarding Decision** where *UsrNNN* is your assigned username (this avoids conflicts with other projects if you are sharing the cluster).
6. Click Create



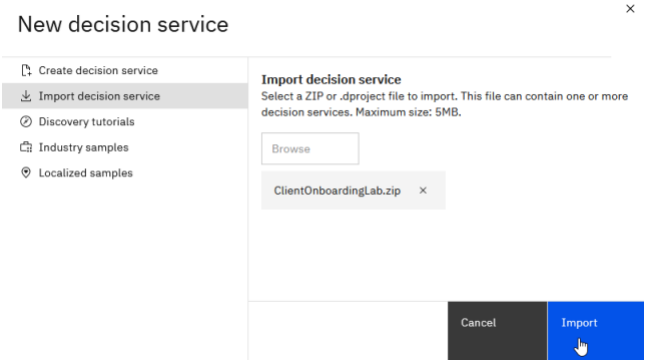
7. Your new project opens in Decision Designer. You can optionally follow the guided popup if it shows or you can click **Maybe Later** to continue.
8. Click New decision +.



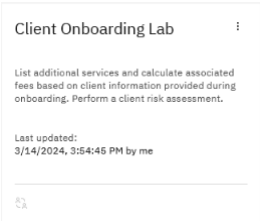
9. Select Import decision service on the left to import the decision service provided by the Lab team.



10. Browse to select the project prepared for the Lab **ClientOnboardingLab.zip** and click **Import**.



11. A tile of a decision service named **Client Onboarding Lab** appears on the project page.



A decision project can be composed of several decision services. You create them or import them from the samples or from a zip file. A project can also be shared with other users and connected to a Git repository. These features will be covered in Exercise 3.

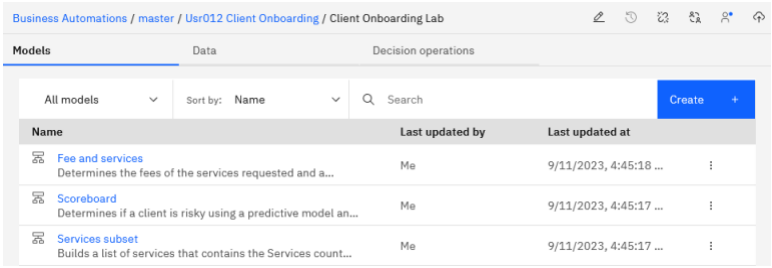
2.2.2 Exploring the decision service: data model and decision model

In this step, you explore the decision service that you imported. A decision service uses decision artifacts to define the business decision.

- **Decision models**: Represent a diagram that expresses the business logic. You can reuse the output of a decision model in another decision model.
- **Predictive models**: Apply data from a machine learning model to make a prediction.
- **Data model**: Represents the data structure used by the business logic. You can use the data model vocabulary in your decision models and predictive models.
- **External libraries and data sources**: Extend the data models using Java code or contain data models and functions that you can use in decision models.
- **Decision operations**: Define entry points for decision service execution.
- **Task models**: Allows business experts to define the decision logic outside model diagram (Note: task models are not covered in this lab, but you can see them in the samples proposed in Additional information).

For more information, see IBM Documentation Building decision models.

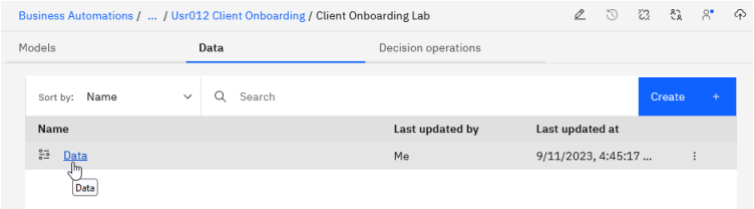
1. Click on **Client Onboarding Lab** to open your decision service.
2. Your decision service is displayed.



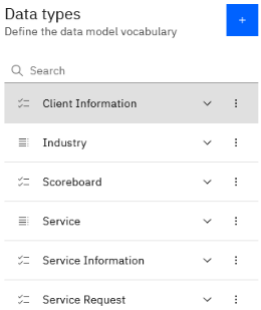
2.2.2.1 Exploring the data model

A data model defines the data that is needed to make your decisions. In a data model, you use predefined and custom data types to create a vocabulary that you can use to populate decisions. Basic data types, including string and integer, are predefined by default in Automation Decision Services. You create a data model to define custom data types to match specific needs for your service. These types are defined in a data model and can be used in one or more decision models in a service.

1. Open the **Data** tab and click on **Data**.



The data model is a collection of Data types that represents the data needed to make the decision. This collection is used by all the decision models defined in the decision service.
The model for Client Onboarding contains 6 Data types: Client Information, Industry, Scoreboard, Service, Service Information, and Service Request.



In the decision model for this exercise, 5 of these data types are used: Client Information, Industry, Service, Service Information, and Service Request.

Two categories of data types can be defined:

- The **enumeration type**, such as Industry and Service, which contain a list of possible values.
- The **composite type**, such as Client Information, Service Information, and Service Request, which contain a set of attributes with a name and a type. The type can be a basic type predefined in Automation Decision Services such as string or integer or a custom data type.

The types defined as input of the model for this exercise (Fee and services) are:

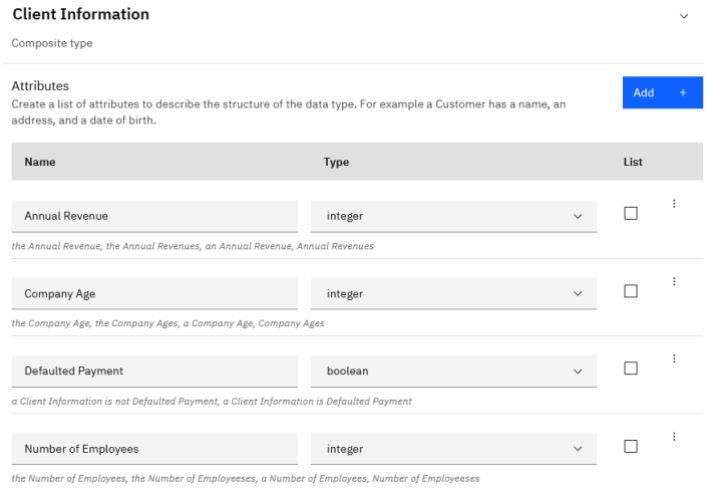
- The enumeration **Industry**: lists the available industries.
- The enumeration **Service**: lists the available services.
- The composite type **Service request**: describes in which industry the client is and what are the services requested.
- The composite type **Client Information**: describes the characteristics of the client including annual revenue, number of employees, company age, defaulted payment.

The type defined as output of the model for this exercise (Fee and services) is:

- **Service Information**: a list of additional services suggested to the client and the fee for the services requested.

Automation Decision Services allows you to import sets of values extracted from an Excel file to populate enumeration data types. This is not covered in this Lab. For more information, see IBM Documentation Extracting values from an external data source.

2. Click on **Client Information** to explore the data types. It is a Composite type composed of 4 attributes. Attributes and values allow you to define the characteristics of a data type.



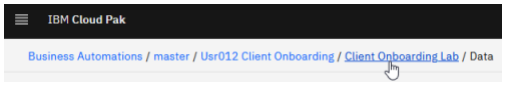
(If you see long instead of integer as type, note that these are equivalent and you can continue.)

The **verbalization** defines how a data type is referenced in business rules. It depends on the type of element. Each attribute comes with a set of automatically generated expression and action phrases.
For more information, see IBM Documentation Default verbalization.

3. Explore each **data type** and **attributes** to learn more about the data types.

   For more information, see IBM Documentation Creating a data model.

4. Click on **Client Onboarding Lab** in the breadcrumb to navigate back to your decision service.



Working with external libraries to enrich your data model is not covered in the Lab. For more information see IBM Documentation Working with external libraries, or the External library tutorial available on GitHub.

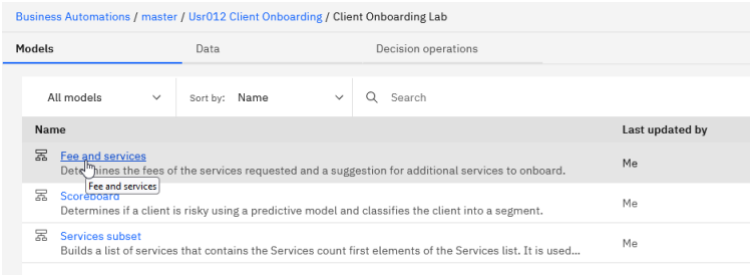
2.2.2.2 Exploring the decision model

You define the structure of a decision in a model diagram. The primary part of the diagram are nodes:

- **Decision nodes**: represent the decision and contain logic that defines how each decision is made.
- **Input data nodes:** represent the data used to determine the decision output and are associated with a data type.
- **Function nodes:** represent values that are computed from other decision models.
- **Prediction nodes**: represent values that are computed in machine learning predictive models (prediction nodes are not used in this exercise, they are used in Exercise 2).
- **Links** represent the relationship between the decisions, input data, and the invocation of a function or prediction.

The rules define the logic of the decision and are expressed in business rules or decision tables. The rules determine the output of the model.

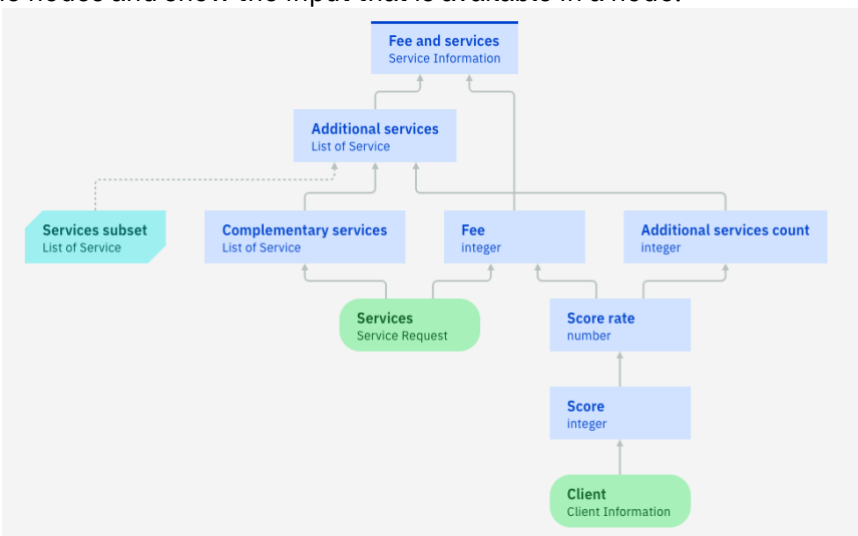
1. In the Models tab, click on **Fee and services.**



It opens the decision model Fee and services.

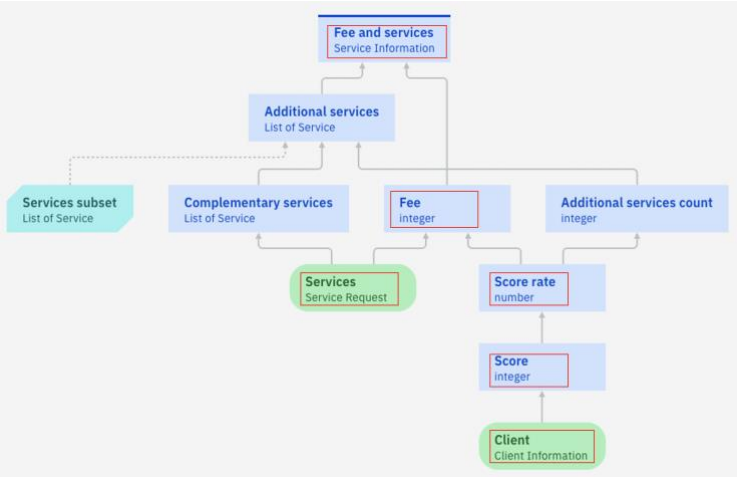
In Automation Decision Services, you implement your decision by building a decision model diagram. The diagram shows how the decision depends on several sub-decisions.
The modeling diagram is inspired by the Decision Model and Notation (DMN) standard.

The diagram below shows the decision model for **Fee and services.** As explained in Exploring the decision mode introduction, the diagram is composed of nodes and links. The decision nodes are blue, input nodes are green, and function nodes cyan. The links represent the relationship between the nodes and show the input that is available in a node.

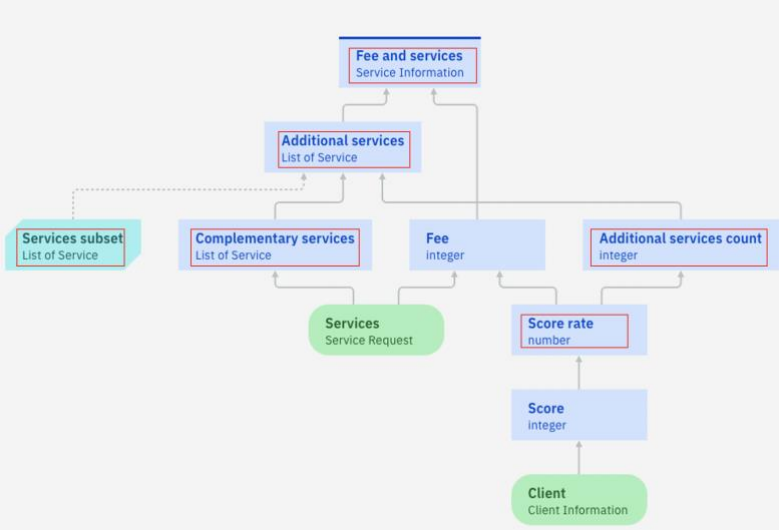


In this exercise, the top decision **Fee and services** is the final decision. It calculates the fees for the services requested by the client and suggests additional services to offer.
It is based on the **Client** information and the **Services** requested as input. To provide the outcome, it depends on intermediate steps, sub-decisions.
The final decision node **Fee and services** depends on two sub-decisions: **Fee** and **Additional services.**

Fee
The fee is computed by the **Fee** decision node, based on the service list represented by the input node **Services** and the **Score rate** decision node.
The **Score rate** decision node computes a rate to apply to the default fee for the client based on a score computed by the **Score** decision node.
The **Score** decision node computes the client score from its different characteristics described by the **Client** input node.

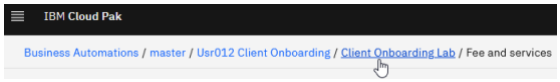


Additional services
The additional services list is computed by the **Additional services** decision node. It takes a subset of the list computed by the **Complementary services** node.
To extract the subset of services it depends on the function node **Services subset** which is computed by taking the count of elements from the list. In the exercise, this computation is arbitrarily simplified. The count is computed by **Additional service count** decision node and the **Score rate** decision node.

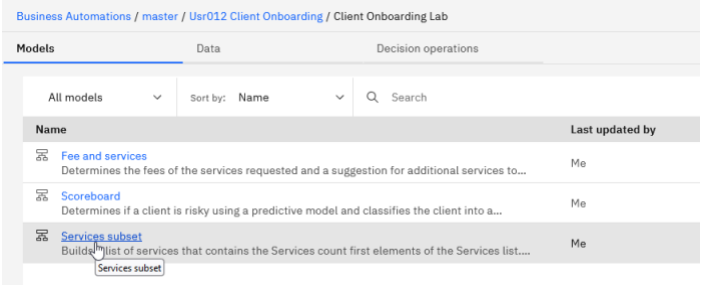


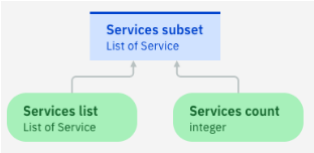
Note that the Services subset is a function node. It is computed from another decision model named Services subset.

2. Click on **Client Onboarding Lab** in the breadcrumb to navigate back to the **Models** tab.



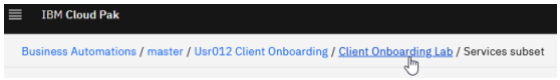
3. To see its diagram, click **Services subset** in the Models list:



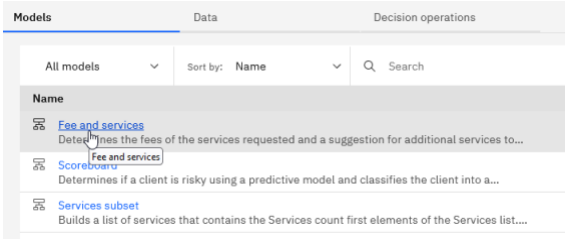


For more information, see IBM Documentation [Creating a decision diagram]

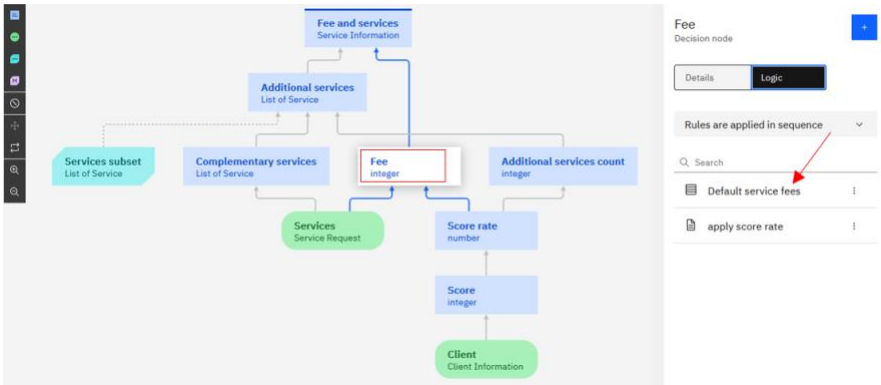
4. Return to the Decision model Fee and services by clicking **Client Onboarding Lab** in the breadcrumb to navigate back to the **Models** tab.



5. In the **Models** tab, click on **Fee and services**.



6. Click on the **Fee** node, and then select the **Logic** tab to explore the decision logic. It displays the decision table **Default services fees** and the rule **apply score rate** that implement the business logic of the fee calculation. In Automation Decision Services, you express decision logic with a set of business rules and/or decision tables.



7. Click on **Default services fees** in the **Logic** tab to open the decision table.

Default service fees

	Services count	Fees
1	0	0
2	1	15,000
3	2	25,000
4	3	38,000
5	4	50,000
6		

Each row represents a single rule, where the **Services count** column represents a condition parameter and the **Fees** column the value of the action when the condition is met.

8. Click on **apply score rate** in the **Logic** tab to open the business rule.

apply score rate

Type your rule using the list below as reference

```
set decision to round ( decision * 'Score rate', 0 );
```

Inputs (2) Output (1)

Services	Service Request
Score rate	number

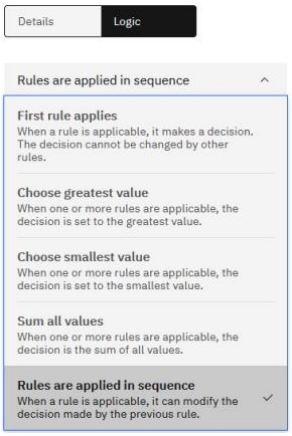
It is a simple text rule statement that determines a rate to apply to the fee calculation.

9. Click on **Inputs** and then **Output** at the bottom of the screen.

The rule uses the vocabulary of the input data **Services** and **Score rate** and sets a value for the output of the decision **Fee**. In this decision logic, the rules are applied in sequence. A fee is assigned according to the number of services. Then, a rate is applied to calculate the final fee for the requested services.

To define how the rules interact with each other, you select a **rule interaction policy**.

10. On the Logic tab, expand the **Rules are applied in sequence** drop down.



(Note: You might see less entries but you can continue.)
The interaction policy choices differ according to decision node type (number, list, integer...). In this exercise, the decision node Fee has an output type of integer, so the options above are displayed for the integer type.
For more information, see the IBM Documentation [Choosing an interaction policy.]

2.2.3 Updating decision logic: adding a node, a rule, editing a decision table

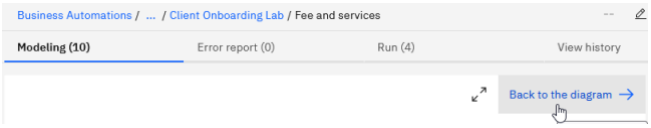
You will now change the decision logic for the **Score** computation to take into account the number of services requested by the client. A higher number of services leads to a higher score. The change in the decision logic implies:

- Knowing the number of requested services from the **Score** node. This requires adding a link from the **Services** input and the **Score** node.
- Adding the computation for the number of services. This requires adding a rule to compute the new score.
- Adapting the score rate computation to the new score range.

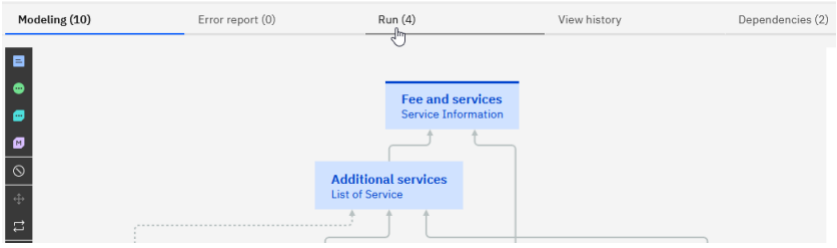
Before modifying the decision logic, you will check the current behavior of the rule execution.

2.2.3.1 Test the decision logic

1. Click on **Back to the diagram** to return to the decision diagram.

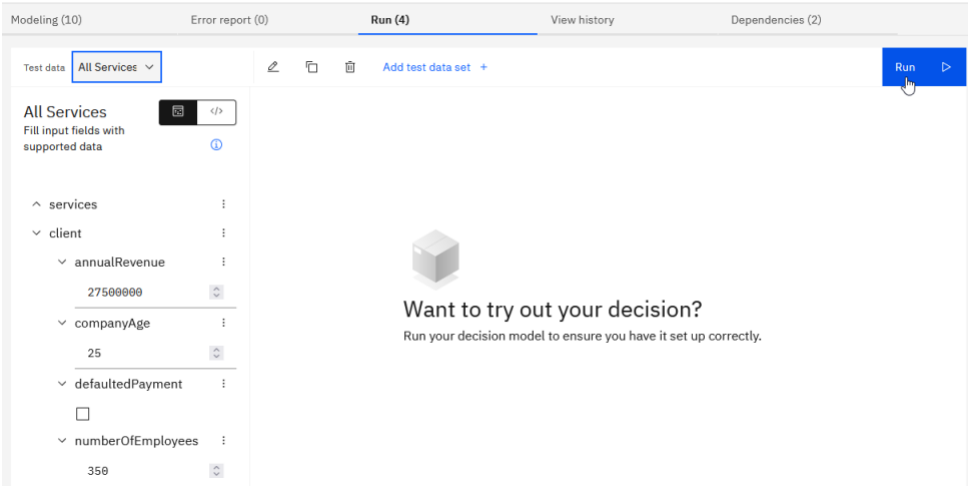


2. Click on the **Run** tab.

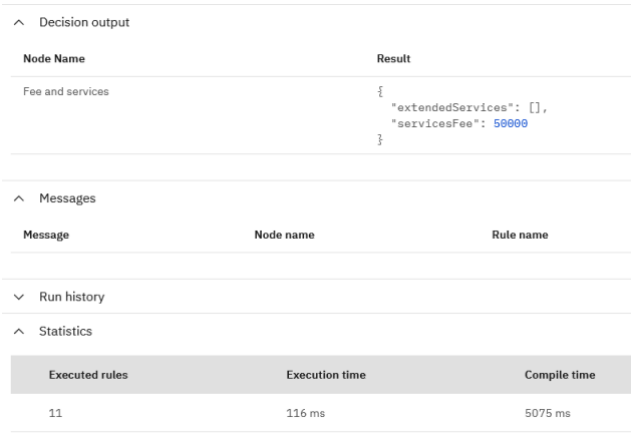


On the left, the **Test data** pane lets you select the set of data to submit to execute the rules. In this Lab, 4 sets of data are pre-defined: All Services, Federal 2, Federal 3, Telecom 1.

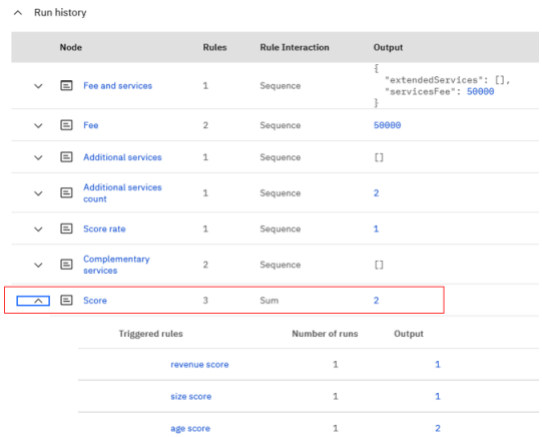
3. Select **All Services** to the left and then click on **Run** to the right.



It generates a report and displays the input and output of each node in the decision model. The final decision is displayed in the top of the report: for Fee and services the result is `"servicesFee": 50000`.



4. Expand **Run history**, and then expand **Score** node.



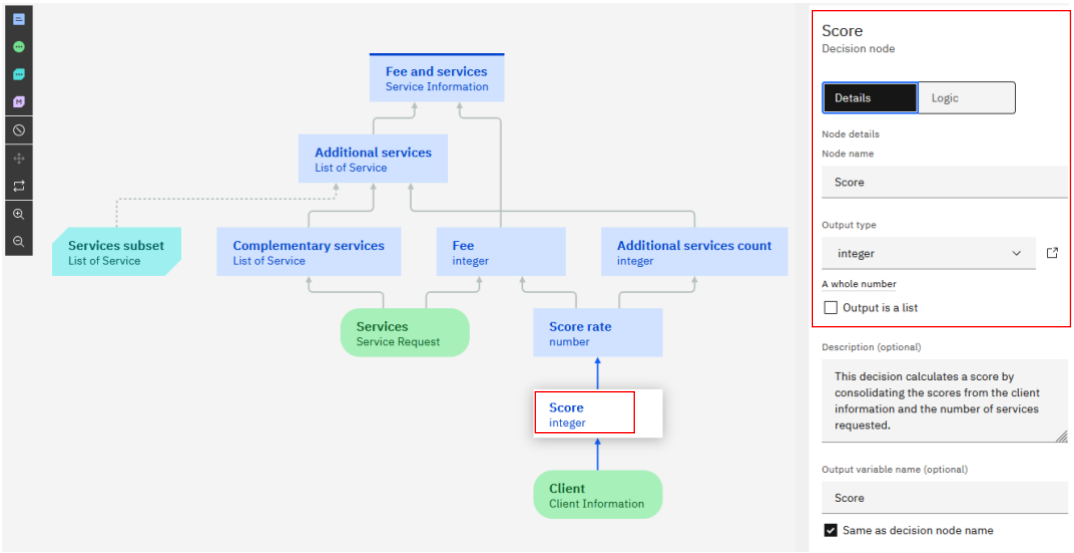
The output displays the result for the decision node Score, which is a consolidated result. In this example, the Score decision node output is 2:
- Revenue score output is 1,
- Then added to size score (size score is 0) it remains 1,
- Then added to age score result is 2 (age score is 1).

5. Click on the **Modeling** tab to return to your decision model.

2.2.3.2 Exploring the current logic

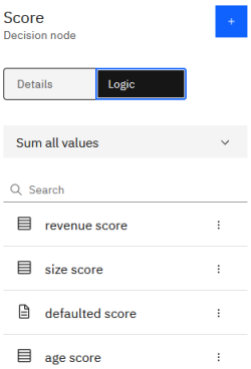
The Score node returns an integer that is used by the Score rate node to compute a rate from the score.

1. Click on the **Score** node, and then the **Details** tab.



It returns an integer computed by adding all the values returned by its defined rules and decision tables.

2. Click on the **Logic** tab to explore the logic of the business rules.



The logic contains one rule (defaulted score) and 3 decision tables (revenue score, size score, age score).

3. Click on **defaulted score** to explore the logic.

The defaulted score rule adds -1 to the decision in case of a defaulted payment.

4. Click on the **revenue score** decision table.

   The revenue score decision table computes a score from the annual revenue.

revenue score

	Annual Revenue		Score
	min	max	
1	0	1,000,000	0
2	1,000,000	50,000,000	1
3		≥ 50,000,000	2
4			

5. Click on the **size score** decision table.

   The size score decision table computes a score from the number of employees.

   size score

	Number of Employees		Score
	min	max	
1	100	3,000	0
2	3,000	8,000	1
3		≥ 8,000	2

6. Click on the **age score** decision table.

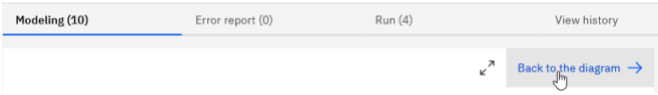
   The age score decision table computes a score from the company age.

   age score

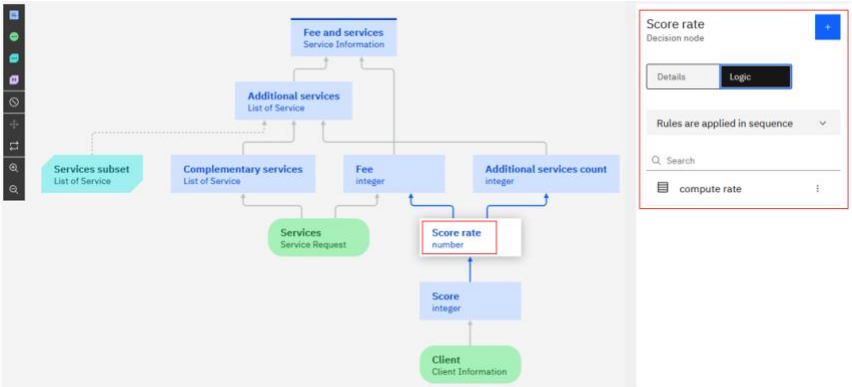
	Company Age		Score
	min	max	
1	2	15	0
2	15	30	1
3		≥ 30	2
4			

   The Score output is used by the Score rate node to compute a rate from the score.

7. Click on **Back to the diagram** to return to the decision diagram.



8. Click on the **Score rate** node, and then on the **Logic** tab.



9. Open the **compute rate** decision table.

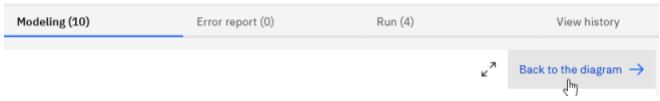
   You see the score values.

compute rate

	Score	Score rate
1	≤ -1	1.7
2	0	1.4
3	1	1.2
4	2	1
5	3	0.95
6	4	0.9
7	5	0.8
8	≥ 6	0.7
9		

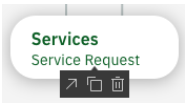
In the next step, you change the decision logic.

10. Click on Back to the diagram.

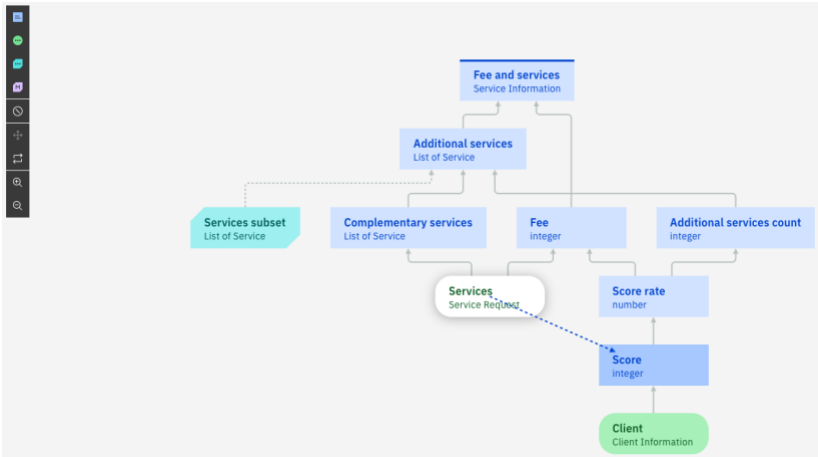


2.2.3.3 Adding a business rule

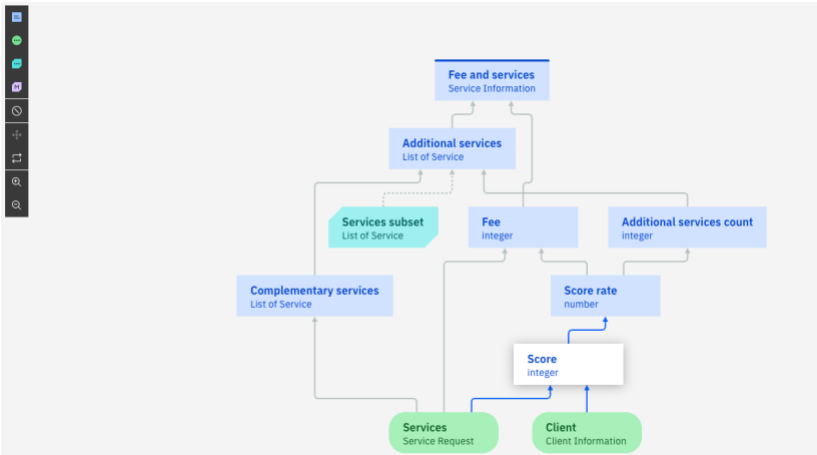
The score of the client is calculated based on 3 decision tables: revenue score, size score, and age score, and 1 rule: defaulted score. The score should also be evaluated based on the number of services that the customer requested to be onboarded. You modify the business rule that calculates the score to add the number of services to its logic. A higher number of services results in a higher score.
To add the number of services as an input to the score computation, you update the decision node **Score** to add **Services** as input and add a new rule to determine the score.

1. Click on the **Services** input node, then hover over the icon 

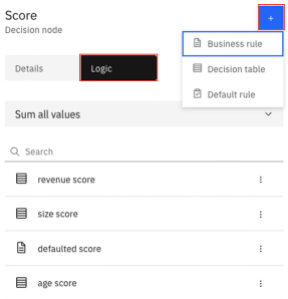
, and then single click on **Connect to another node** (arrow) to add a link to connect to another node. Move your mouse to the **Score** decision node and click again.



This creates a link and the decision diagram is updated to reflect the new dependency:

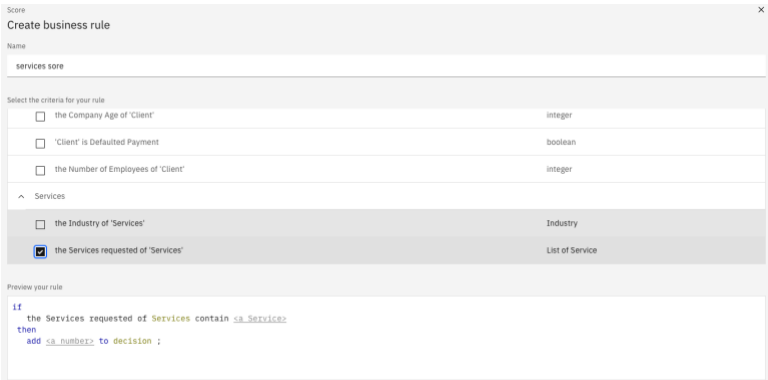


2. Click on the **Score** decision node, then select the **Logic** tab in the right pane, and then click on the plus (+) and click **Business rule**.



This opens a wizard to help you select the criteria of your rule.

3. Enter **services score** for the name for the rule, and then scroll down to select the criteria **the Services requested of 'Services'**.



4. Click on **Create**. You see a red icon that indicates an error. Hover over the icon to see the details.

5. In the rule editor, update the rule to contain the text shown below. You may wish to delete the text that is proposed first.

```
if
  the number of elements in the Services requested of Services is more than 3
then
  add 1 to decision ;
```

While you are typing, the auto-completion guides you through writing your rule. You can also trigger the auto-completion by typing Ctrl-Space in the editor.

Once finished the rule appears as follows:

services score

Type your rule using the list below as reference

```
1 if
2     the number of elements in the Services requested of Services is more than 3
3 then
4     add 1 to decision ;
```

In the decision logic, the **Score** node is now the sum of 5 rules: revenue score, size score, defaulted score, age score, and services score.

Score
Decision node

Details | Logic

Sum all values

Search

- revenue score
- size score
- defaulted score
- age score
- services score

Click on Back to the diagram.

2.2.3.4 Updating a decision table

You update the decision table for the score rate because the values change following the update of the rule to compute the score.

1. Click on the **Score rate** node. Then, in the **Logic** tab, click on the **compute rate** decision table.

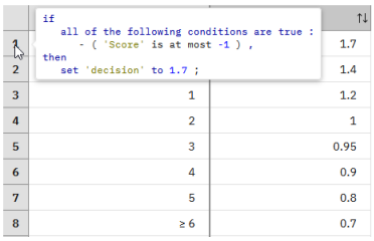
   It displays the decision table that contains the decisions to determine the score rate.

compute rate

	Score	Score rate
1	≤ -1	1.7
2	0	1.4
3	1	1.2
4	2	1
5	3	0.95
6	4	0.9
7	5	0.8
8	≥ 6	0.7
9		

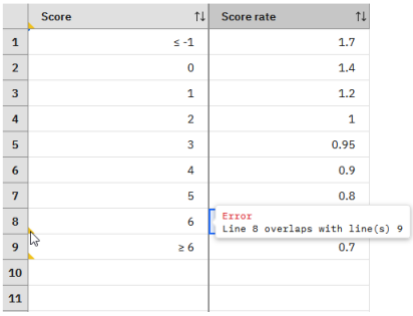
In this table, each row represents a single rule. The Score represents a condition parameter and the Score rate the value for the action.

2. Hover over the **1** in the first row to display the following rule:



1		1.7
2		1.4
3	1	1.2
4	2	1
5	3	0.95
6	4	0.9
7	5	0.8
8	≥ 6	0.7

3. Select **row 8** and then right-click to open the contextual menu. Select **Insert row -> Above**. Enter 6 in Score column and 0.7 in Score rate column.



	Score	Score rate
1	≤ -1	1.7
2	0	1.4
3	1	1.2
4	2	1
5	3	0.95
6	4	0.9
7	5	0.8
8	6	
9	≥ 6	0.7
10		
11		

The warning icon appears in lines 8 and 9 to automatically notify you of an issue. The warning indicates an overlap of rows 8 and 9 so you must change the values in row 9.

4. Change to 7 for Score column and 0.6 for Score rate column.

compute rate

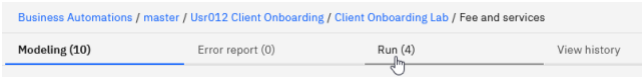
	Score	Score rate
1	≤ -1	1.7
2	0	1.4
3	1	1.2
4	2	1
5	3	0.95
6	4	0.9
7	5	0.8
8	6	0.7
9	≥ 7	0.6

5. Click on Back to the diagram.

2.2.4 Validating the decision service

Now that you have modified the decision model and updated the decision logic, you want to test the decision service before deploying to production. Automation Decision Services integrates a test environment allowing you to fine tune your rules and verify their behavior.

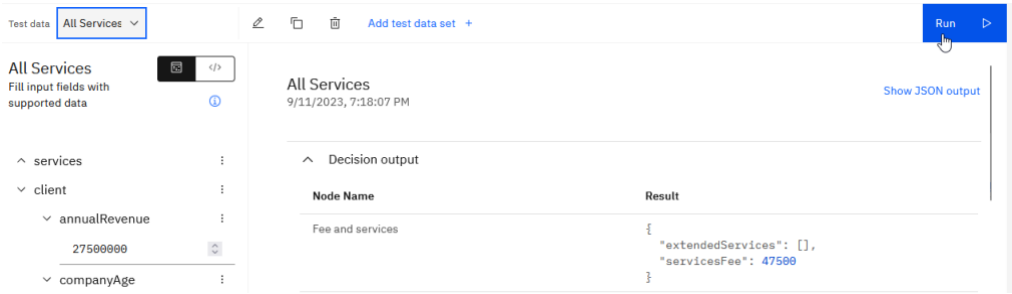
1. Select the **Run** tab.



On the left side, the **Test data** pane lets you select the data to submit to execute the rules.

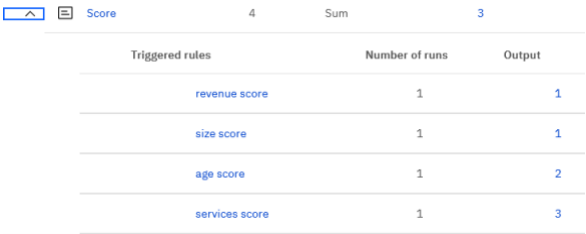
2. To see the behavior after the changes, select the same data set as you did before the changes: **All Services** and click on **Run**.

The decision output for servicesFee is now 47500.



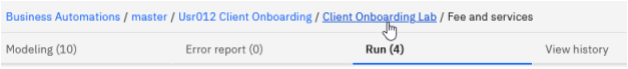
3. Expand **Run history**, and then expand **Score**.

   It shows that the rule you added **services score** has been triggered. The result for the decision node **Score** is a consolidated result. It is the sum of the rules that triggered the result, applied in sequence.
   Each displayed output is the sum, including the previous output.



   In this example, the output of the **Score** decision is 3:
   - Revenue score output is 1
   - Then added to size score (size score is 0) it remains 1
   - Then added to age score the result is 2 (age score is 1)
   - Then added to services score it is 3 (services score is 1).

4. Click on **Client Onboarding Lab** in the breadcrumbs to navigate back to the Models tab.



For more information, see IBM Documentation
[Building decision models]
[Creating a decision diagram]

2.3 Summary

You have completed Exercise 1 - Modeling Decisions. You created a project and a decision service to define the fee and services for a client being onboarded to a set of services.

- You explored the decision model diagram and its elements.
- You updated the business logic.
- You created a rule.
- You added a link in the decision diagram.
- You edited a decision table.
- You tested and validated the decision service.

3 Exercise 2: Adding machine learning in the decision model

3.1 Introduction

In this exercise you leverage a machine learning model to improve the intelligence of the overall decision. This model evaluates risk based on information provided during the request. Adding a predictive model allows you to benefit from the experience of previous customers to estimate the risk level.
This model provides a prediction for the client risk based on the client information (annual revenue, company age, number of employees, and industry). If the client onboarding is risky, the prediction returns 1. If it is not, it returns a value of 0. It also provides the prediction confidence.

3.2 Exercise instructions

In this exercise, to use the machine learning model deployment into Automation Decision Services you:

- Create a **predictive model** connected to the machine learning deployment.
- Add this **predictive model** into a **decision service**.
- Validate a decision service.

3.2.1 Log in to your project

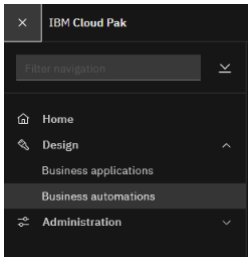
If you are already connected to Automation Decision Services and have your project open you can skip this section and continue in the next section Testing the model before changes.

If you are starting the Lab from this exercise, follow the steps below.

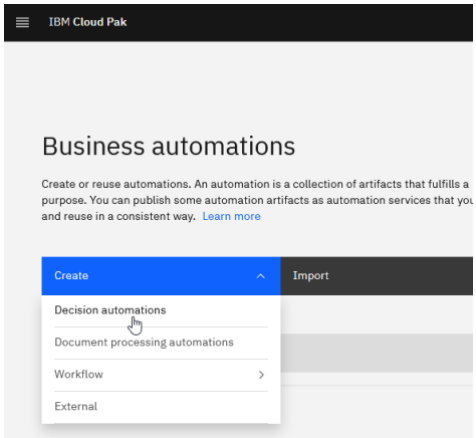
3.2.2 Creating a project and importing a Decision service

1. Log in to Business Automation Studio.
2. Click on the Navigation Menu on the top left corner

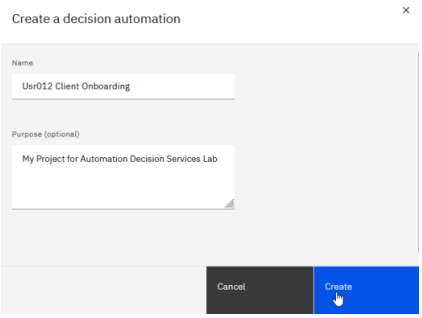
.
3. Expand **Design**, and then click on **Business automations**.



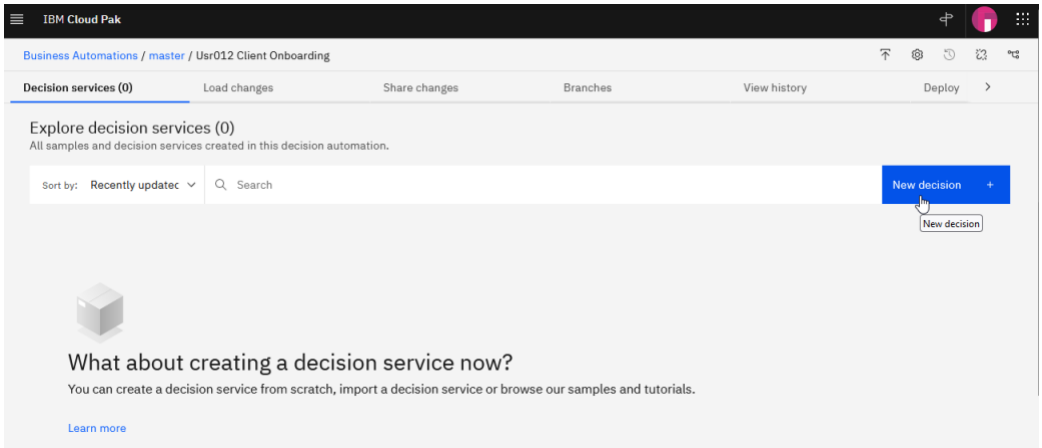
4. Click Create and select Decision automations.



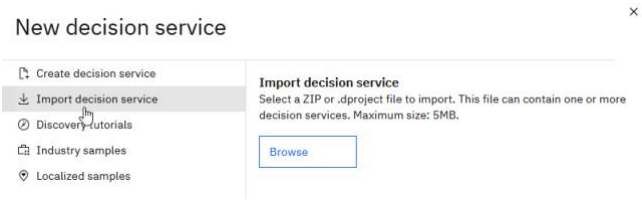
5. Enter a name for your project. Enter **UsrNNN Client Onboarding Decision** where *UsrNNN* is your assigned username (this avoids conflicts with other projects if you are sharing the cluster).
6. Click Create



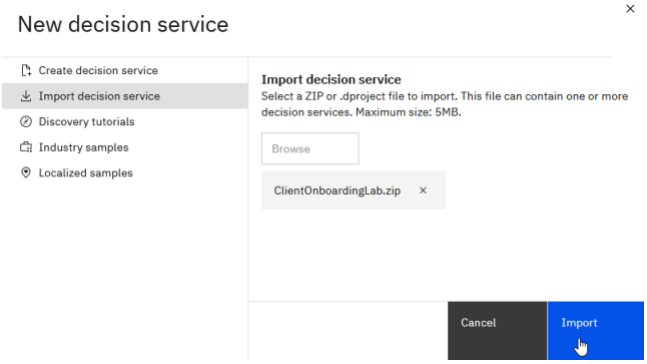
7. Your new project opens in Decision Designer. You can optionally follow the guided popup if it shows or click **Maybe Later** to continue.
8. Click New decision +.



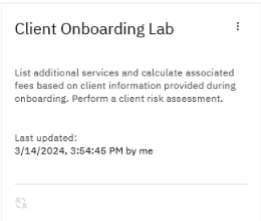
9. Select Import decision service on the left to import the decision service provided by the Lab team.



10. Browse to select the project prepared for the Lab **ClientOnboardingLab.zip** and click **Import**.



11. A tile of a decision service named **Client Onboarding Lab** appears on the project page.



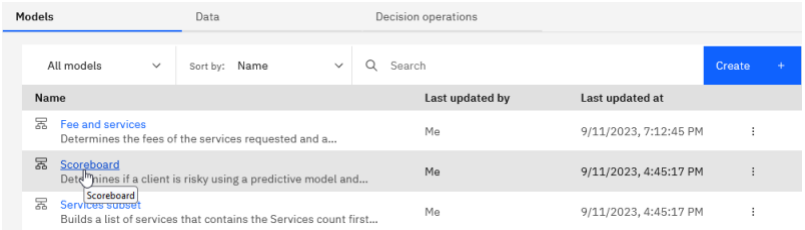
12. Click on **Client Onboarding Lab** to open your decision service.

3.2.3 Testing the model before changes

The Client Onboarding Lab decision service contains several decision models. In this exercise, you work on the Scoreboard decision model.

If you are not familiar with the data model, you can check [Exercise 1 Exploring the decision service: data model and decision model].

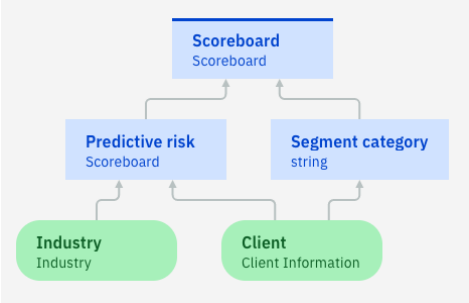
1. In the **Models** tab, click on **Scoreboard**.



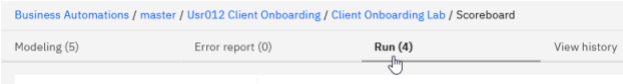
2. It opens the decision model **Scoreboard**.

   The decision model diagram is a dependency diagram that shows the steps to make the final decision. The top decision **Scoreboard** is the final decision. It calculates a client risk and categorizes the client in a business segment. The segment is computed by the decision node **Segment category** from the input node **Client**. The risk is computed from the **Predictive risk** node

with the input nodes **Industry** and **Client**. This model is not complete because the Predictive risk node has no business logic defined. You will complete it in this exercise.

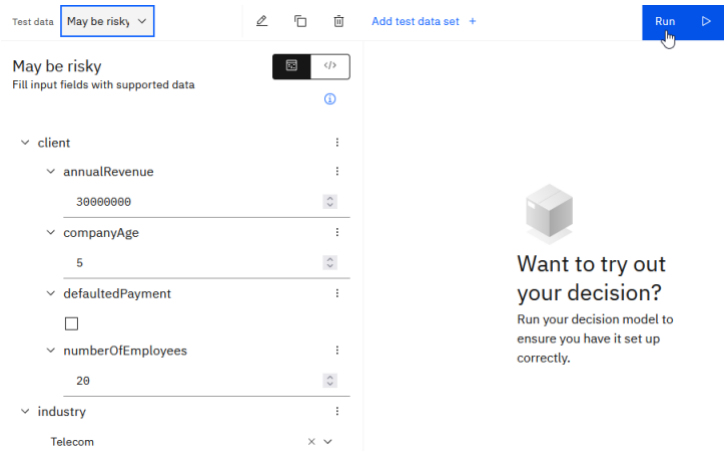


3. Select the **Run** tab to test the model.



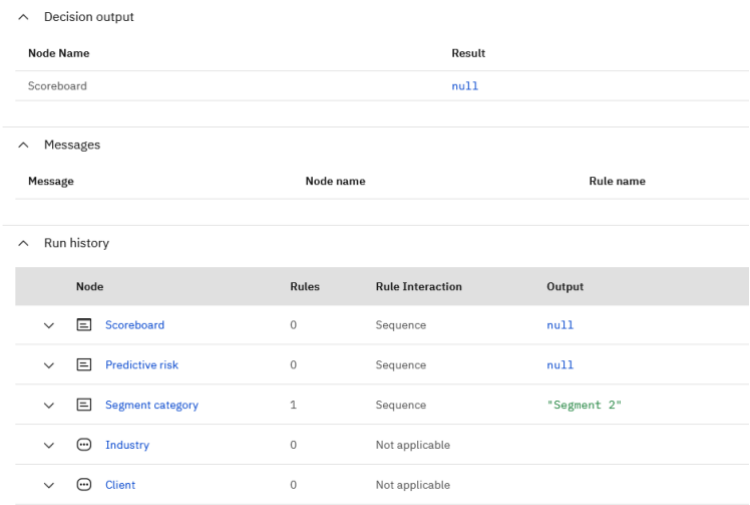
On the left side, the **Test data** pane lets you select which set of data to submit to execute the rules. In this Lab, 4 sets of data are pre-defined: May be risky, Not so risky, Risky, Safe.

4. Select the **May be risky** data set and click on **Run** to test the model.



A report displays the input and output of each node in the decision model.

5. Expand **Decision output** and **Run history** sections if not already expanded.



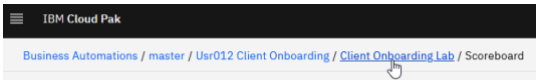
The final decision is displayed in the top of the report: for Scoreboard the result is "null".
The decision result provides the segment 2 category based on the annual revenue specified in the data set. If the revenue is below 50000000, it categorizes the client in segment 2. If the revenue is greater, the client is set in segment 1. Here the risk is not yet computed. You need to add the logic in the decision node **Predictive risk**.

3.2.4 Creating a Predictive Model and Map Initial Data

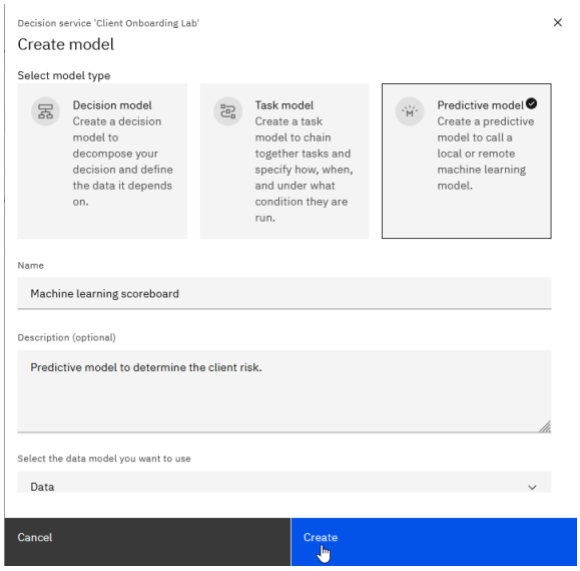
You will now create a predictive model to encapsulate a machine learning model created elsewhere. You connect it by using the wizard to import a **local machine learning model**. You then define the input and the output of this predictive model.

Note: Automation Decision Services also supports connecting to a **remote machine learning model** deployment. A machine learning provider gives access to Machine learning deployment. Two types of providers are currently supported in Automation Decision Services: IBM Watson® Machine Learning and IBM Open Prediction Service. For more information about how to use machine learning models in Automation Decision Services see Integrating machine learning in the IBM Knowledge Center.

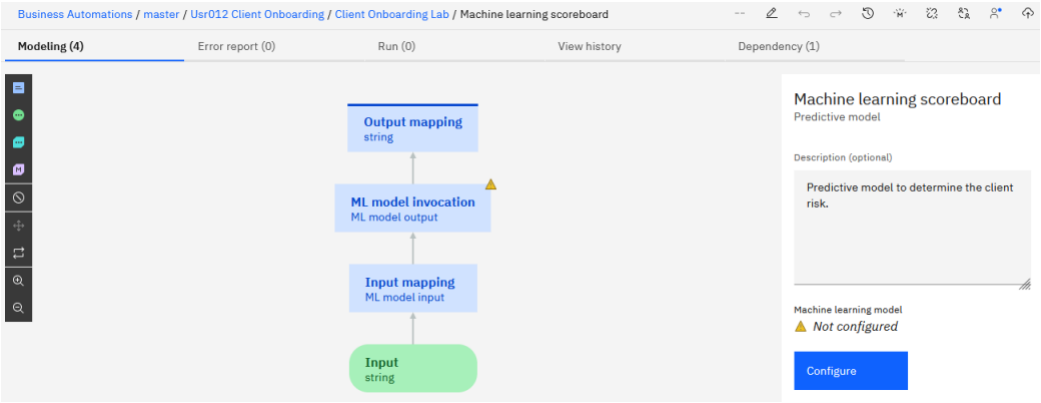
1. Click **Client Onboarding Lab** in the breadcrumbs to navigate back to the list of models.



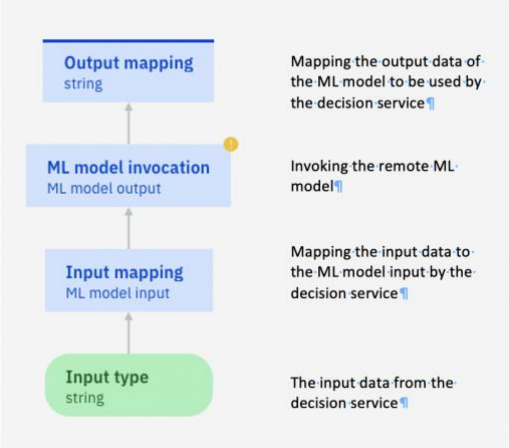
2. On the **Models** tab, click on **Create** and select **Predictive model**.
3. Enter the name **Machine learning scoreboard** and an optional description, and click on **Create**.



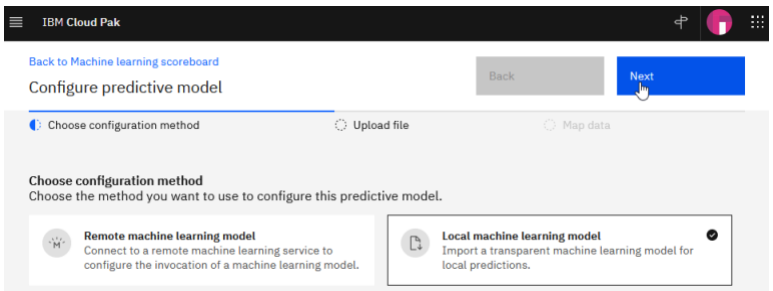
4. The Predictive model **Machine learning scoreboard** is opened.



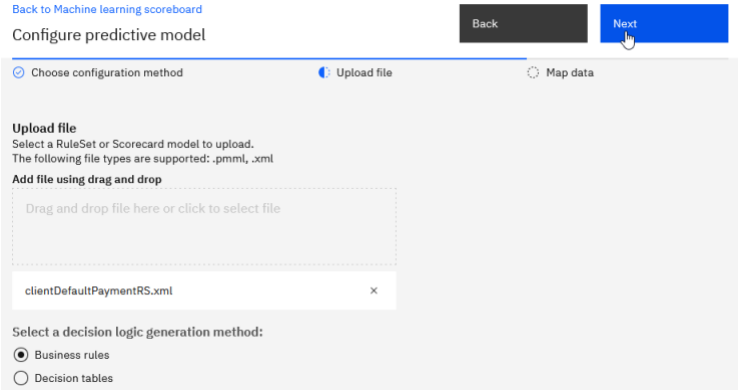
A predictive model is a simple Decision Model diagram. It has one or multiple input nodes, two mapping nodes, and one node that invokes the Machine Learning model.



5. In the right-hand pane click **Configure**.
6. Select Local machine learning model and click Next.

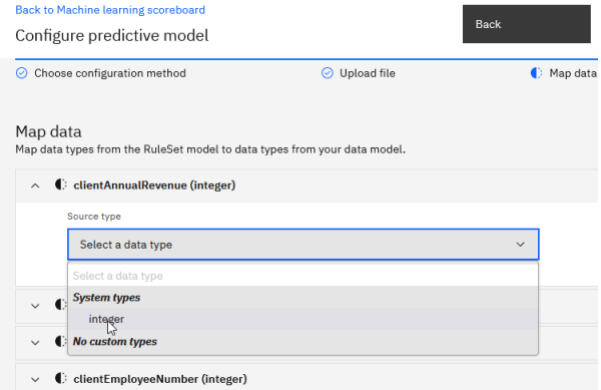


7. Upload the file clientDefaultPaymentRS.xml and click Next.

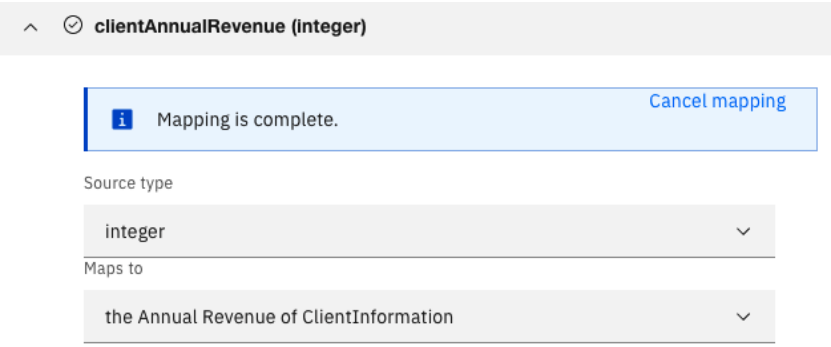


8. Under Map data, expand the first input to the predictive model, **clientAnnualRevenue**, and select **integer** for the **Source type**.

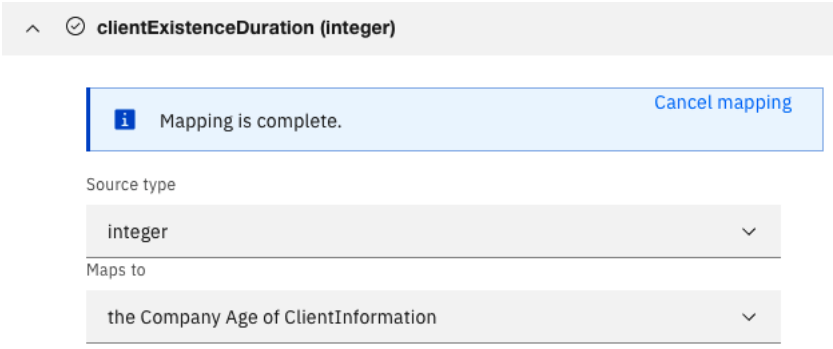
   Note: If you are presented with **long** instead of **integer** as type, note that these are equivalent. Select **long** and continue.



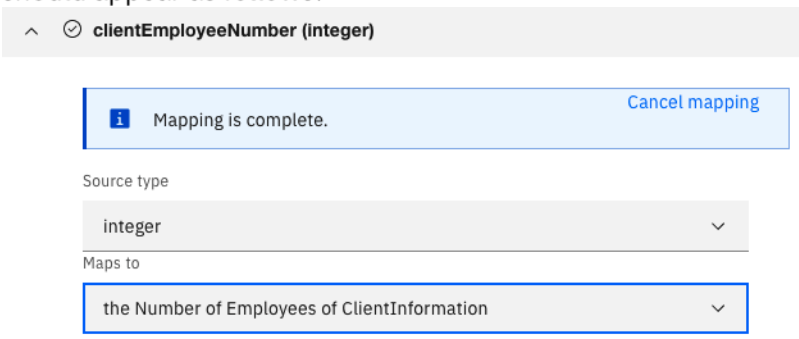
9. Once the **Maps to** loads, select the corresponding data attribute from the Data model of the decision service, **the Annual Revenue of ClientInformation**. The resulting mapping should appear as follows.



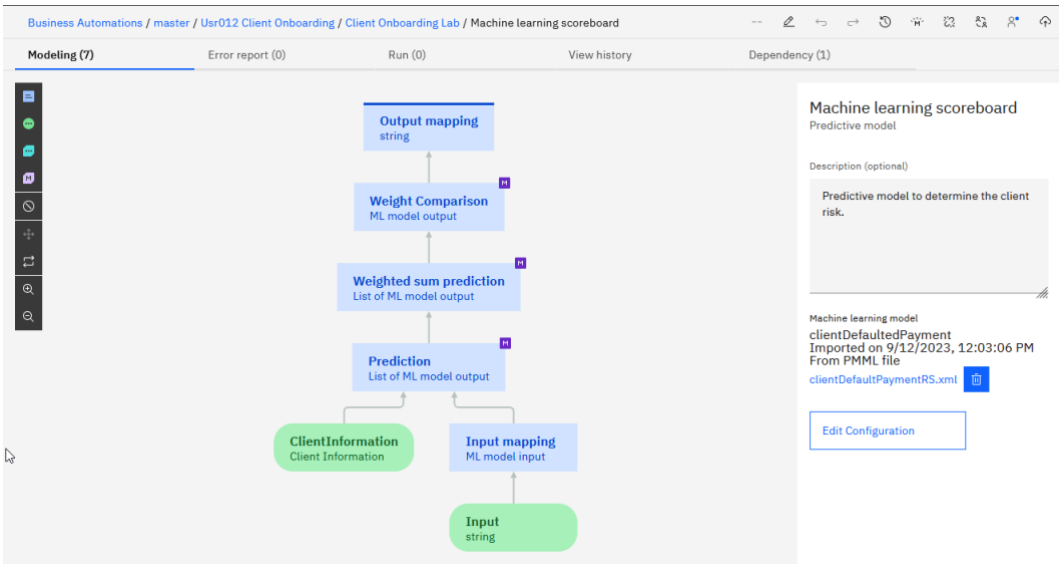
10. Repeat these steps for the **clientExistanceDuration** input to the predictive model. The result should appear as follows.



11. Skip the clientIndustry input, which will be mapped later.
12. Repeat the steps again for the **clientEmployeeNumber** input to the predictive model. The result should appear as follows.

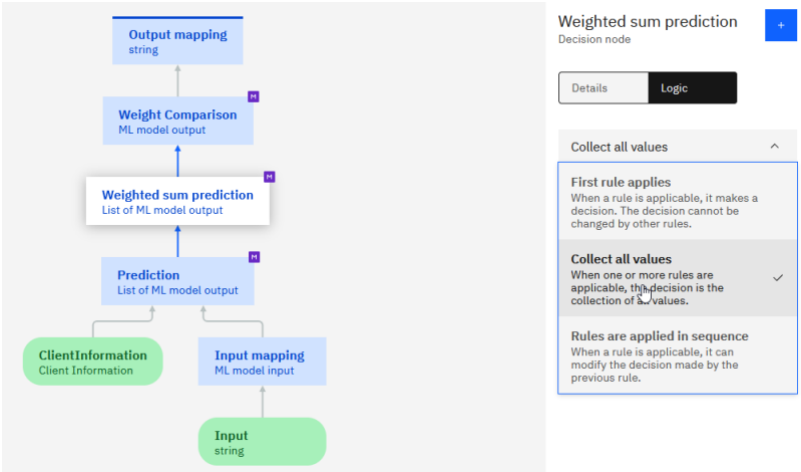


13. When three out of four mappings are complete, click **Apply** in the upper right. The predictive model appears as follows:



14. Select the **Weighted sum prediction** node, and then click the Logic tab. Make sure that the rule

interaction policy is **Collect all values**. Otherwise select it.



3.2.5 Add Additional Data Mapping

You mapped three of the input data variables from the machine learning model to the data types of the Decision Service data model using the import wizard. Now you will finish the mapping.

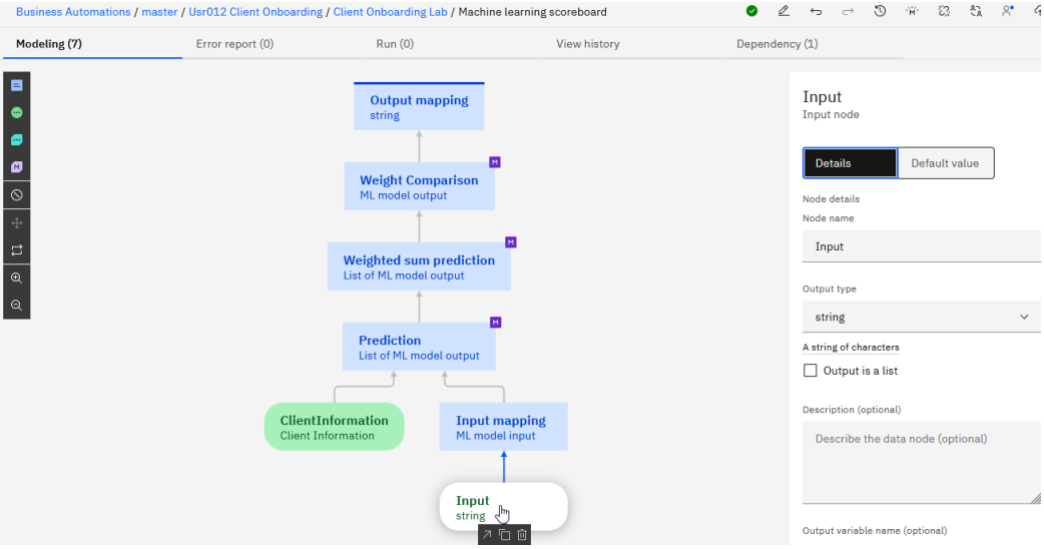
3.2.5.1 Defining the input mapping

In the previous section you saw the input mapping of the ML model, which requires:
- clientAnnualRevenue as a number
- clientExistenceDuration as a number
- clientEmployeeNumber as a number
- clientIndustry as a number

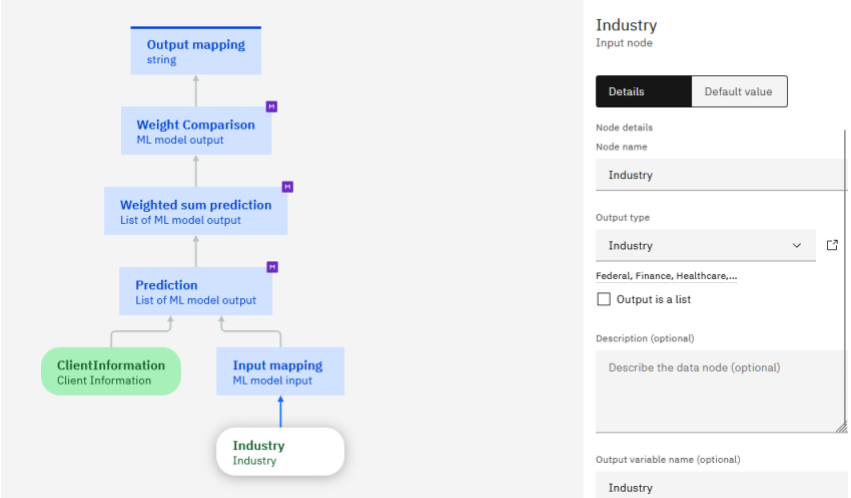
Three of the values are mapped from the Client information type and the last value is from the Industry enumeration type. You must map each value to a number. Since Industry is not a number in the decision model, you must now:
- Define an input node for **Industry**
- Map the Industry input node to the value expected by the ML model using a decision table to map the enumeration type to an integer.

You update the model to map your data to the required input:

1. Select the **Input** node in the diagram.



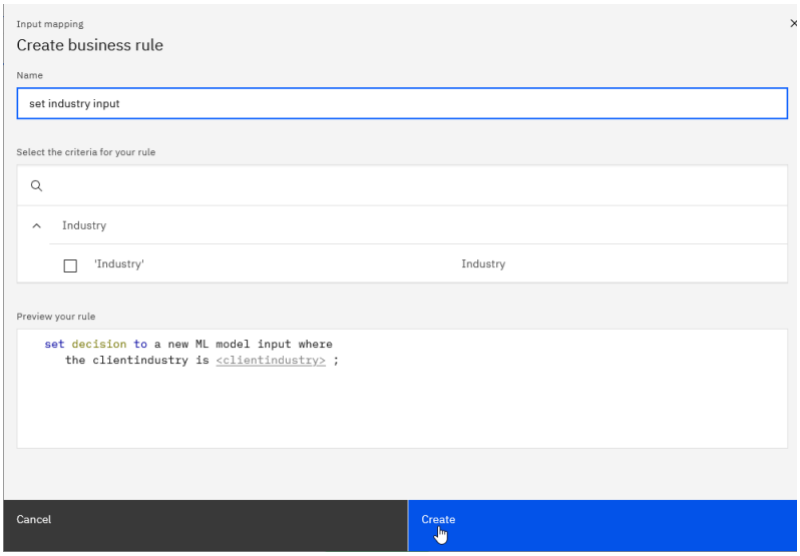
2. In the right **Details** pane, enter **Industry** as the **Node name** and expand **Output type** to select **Industry** as the type. The result should be as follows:



Now, you map the **Industry** input to the Decision Service data model.

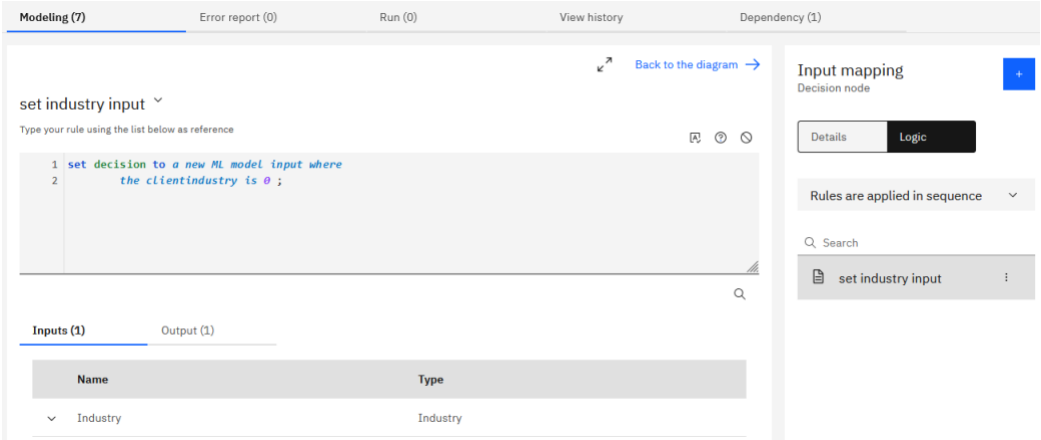
3. In the Predictive model diagram, select the **Input mapping** node. Then, click on the **Logic** tab, select the plus (+) sign, and add a **Business rule**.
4. In the wizard, notice the preview of the generated rule. It matches the input of the Machine

Learning Service (Industry). Enter the rule name **set industry input** and click on **Create**.



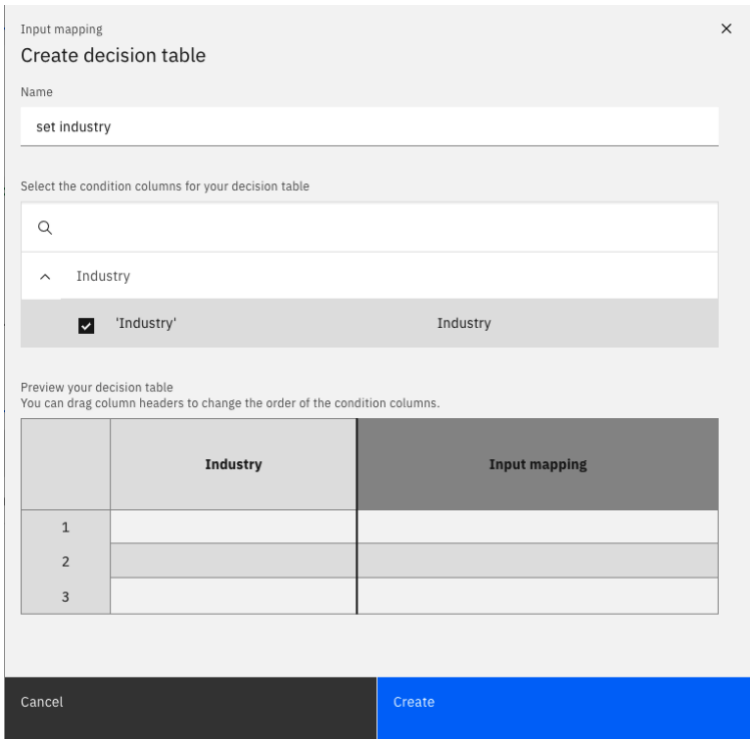
5. Edit the rule with the following statement:

```
set decision to a new ML model input where
      the clientindustry is 0 ;
```



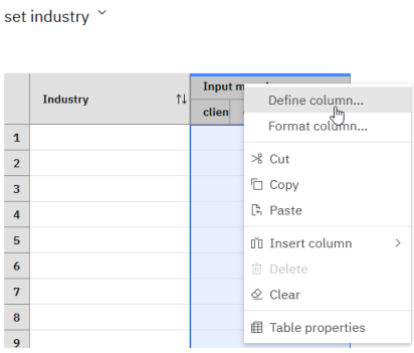
This rule always sets the client industry to 0. Now, you add a decision table to update the industry to the value in the industry provided as input.

6. In the **Logic** tab, click the plus (+) sign and select **Decision table**.
7. Enter **set industry** for the Name and select Industry in the **condition columns** list. Click on **Create**.



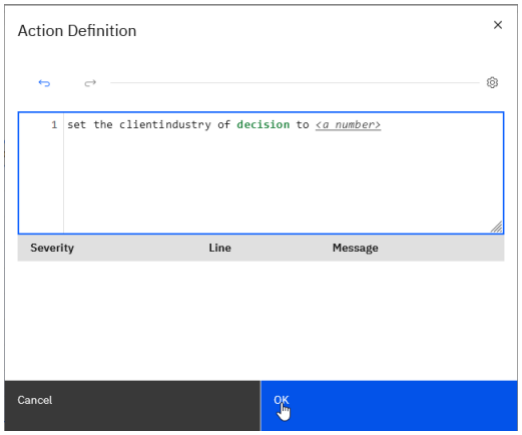
8. Right-click on the **Input mapping** column and select **Define column**.

   You need to update the column to set the industry only. The other attributes were set in an earlier step by the rule set client input.



9. Update the rule in the editor with the following rule statement, then click **OK**.

```
set the clientindustry of decision to <a number>
```



10. Double click in the **Industry** column of **Row 1** and select **Federal** in the drop-down list. Add **0** for Input mapping.
11. Repeat for rows 2 to 5 as shown below:

set industry

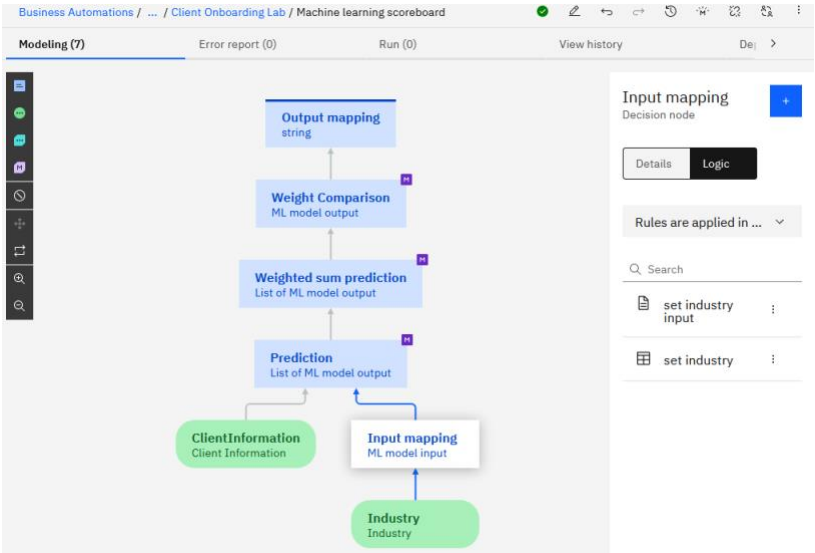
	Industry	Input mapping
1	Federal	0
2	Finance	1
3	Healthcare	2
4	Insurance	3
5	Telecom	4
6		

3.2.5.2 Defining the output mapping

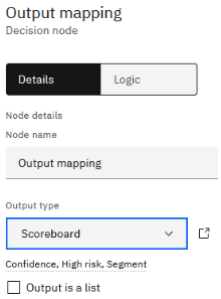
Now you must map the output of the machine learning model to the output of your predictive model. Your predictive model will return a scoreboard in which it predicts the score and gives the confidence. To do this you:

- Change the output type of the Output mapping node of the predictive model to scoreboard.
- Write a rule to map the output of the machine learning model to the scoreboard. The segment remains Unknown since it is not computed by the machine learning model.

1. Click **Back to the diagram** to return to the Predictive model diagram.

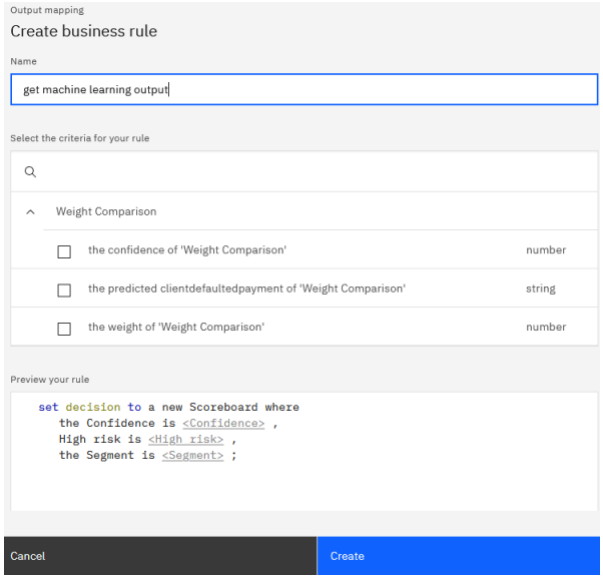


2. In the Predictive model diagram, select the **Output mapping** node. Then, in the **Details** tab, expand **Output type** and select **Scoreboard** in the list.



You define the business logic by adding a rule to map the machine learning model output.

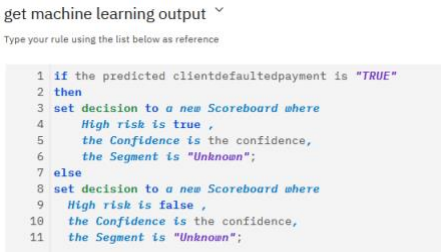
3. Select the **Logic** tab, then select the plus (+) sign, and add a **Business rule**.
4. Enter the name **get machine learning output**. Click on **Create** to edit your rule.



The rule is incomplete as you can see from the icon in line 1.

5. Edit the rule with the following statement:

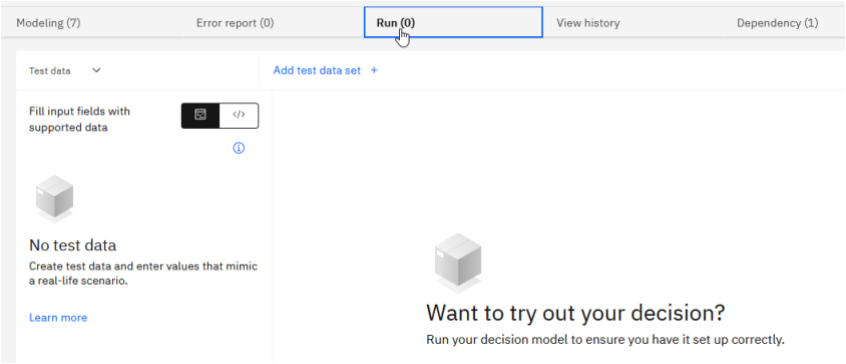
```
if the predicted clientdefaultedpayment is "TRUE"
then
set decision to a new Scoreboard where
    High risk is true ,
    the Confidence is the confidence,
    the Segment is "Unknown";
else
set decision to a new Scoreboard where
  High risk is false ,
  the Confidence is the confidence,
  the Segment is "Unknown";
```



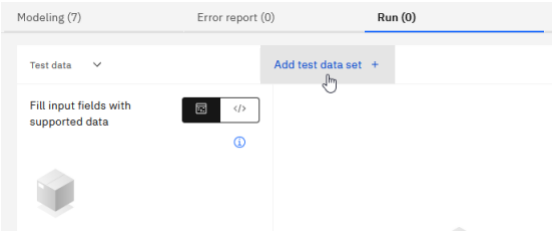
3.2.6 Validating the predictive model

Now that you have created your Predictive model and defined the predictive decision logic, you test it to verify. IBM Automation Decision Services integrates a test environment allowing you to fine-tune your rules and verify their behavior against the test data.
You add datasets to define the data that you want to test.

1. Select the **Run** tab.



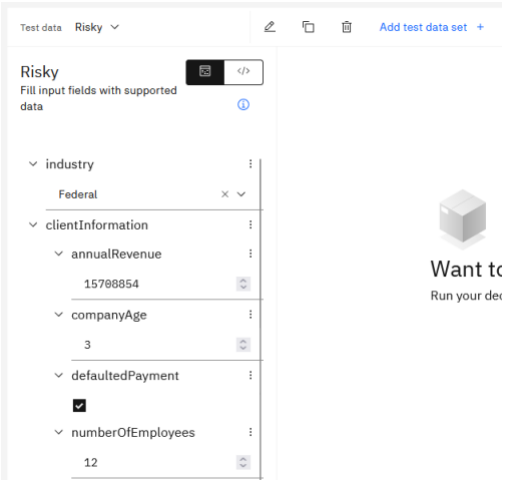
2. In the Test data pane, click on Add test data set +.



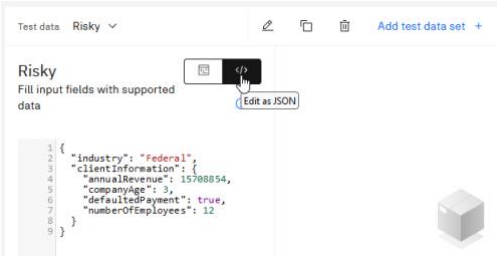
Now you add four data sets: Risky, May be risky, Not so Risky, Safe.
To add test data, you enter a name and define the values.

3. Click on the **Rename test data set** (pencil) icon next to **dataset** and enter **Risky**. Expand **Industry** and **Client Information** to enter the test data

   Name: **Risky**
   Industry: Federal
   Annual Revenue: 15708854
   Company Age: 3
   Defaulted Payment: true (checked)
   Number of Employees: 12



4. Click the Edit as JSON button (</>) to see the JSON content:




```
{
  "industry": "Federal",
  "clientInformation": {
    "annualRevenue": 15708854,
    "companyAge": 3,
    "defaultedPayment": true,
    "numberOfEmployees": 12
  }
}
```

You can edit datasets as FORM by entering the data in the form as you did in this step or as JSON. It can be easier to copy/paste using the JSON option.

5. (Optional) Repeat the steps to add 3 additional datasets. Copy/Paste the JSON content:

Name: **Not so risky**

```
{
  "industry": "Healthcare",
  "clientInformation": {
    "annualRevenue": 61399457,
    "companyAge": 4,
    "defaultedPayment": true,
    "numberOfEmployees": 10
  }
}
```

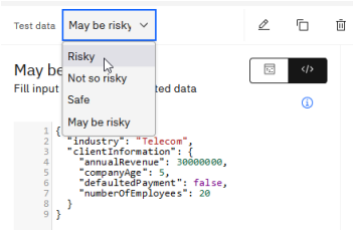
Name: **May be risky**

```
{
  "industry": "Telecom",
  "clientInformation": {
    "annualRevenue": 30000000,
    "companyAge": 5,
    "defaultedPayment": false,
    "numberOfEmployees": 20
  }
}
```

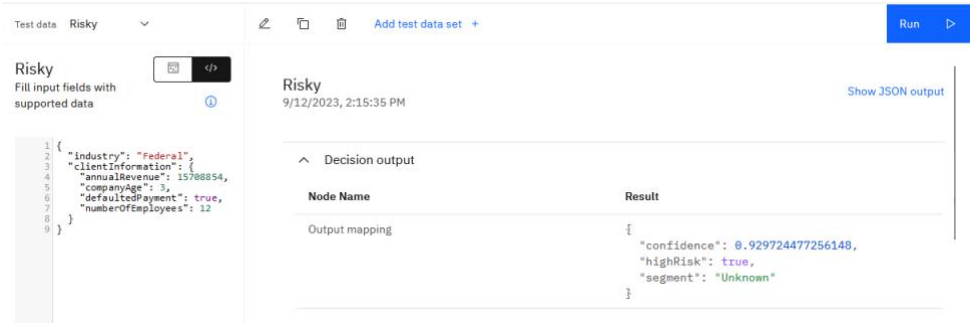
Name: **Safe**

```
{
  "industry": "Healthcare",
  "clientInformation": {
    "annualRevenue": 103314927,
    "companyAge": 26,
    "defaultedPayment": true,
    "numberOfEmployees": 67
  }
}
```

You now have 4 data sets available.



6. Select a data set of your choice and click on **Run**.
7. See the Decision output for the data set **Risky** below.



8. You can explore the output in JSON format, by clicking on **Show JSON output**

Risky
9/12/2023, 2:15:35 PM

Show formatted output

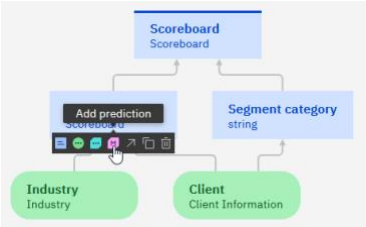

```
{
  "type": "Just",
  "value": {
    "a": "Risky",
    "b": {
      "tag": "Ok",
      "value": {
        "success": {
          "payload": "{\"confidence\":0.929724477256148,\"highRisk\":true,\"segment\":\"Unknown\"}",
          "trace": "",
          "infos": [
            {
              "nodeName": "Output mapping",
              "nodeKind": "Decision",
              "result": "{\"confidence\":0.929724477256148,\"highRisk\":true,\"segment\":\"Unknown\"}",
              "executedRules": [
                {
                  "ruleName": "get machine learning output",
                  "result": "{\"confidence\":0.929724477256148,\"highRisk\":true,\"segment\":\"Unknown\"}",
                  "executionCount": 1,
                  "isDefault": false
                }
              ]
```


3.2.7 Use the predictive model in the scoreboard decision model

You will start by adding a prediction node to the scoreboard decision model and connect it to this predictive model. You will also connect this prediction node to the predictive risk input node and create a rule to define the risk.

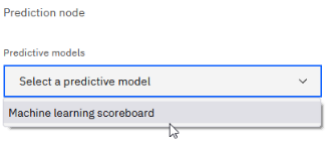
3.2.7.1 Add a prediction node

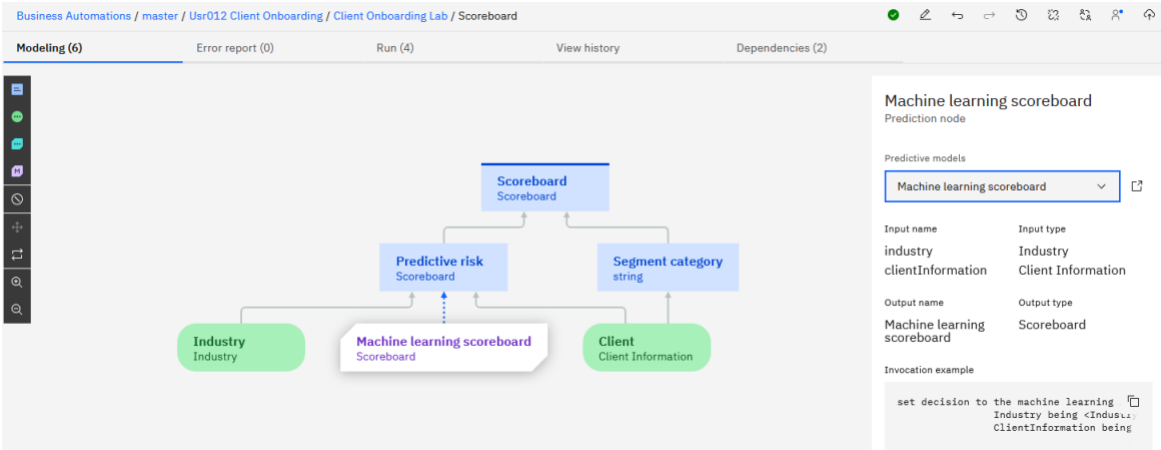
1. Go back to your Scoreboard decision model by clicking **Client Onboarding Lab** in the breadcrumbs and then on **Scoreboard**.
2. It shows the decision model **Scoreboard**. Hover over the **Predictive risk** node and click on the **Add prediction** icon.



The prediction node shows an error because it needs to be connected to a predictive model.

3. Click on your new **Prediction node** if not already selected, and in the right-side pane select the predictive model you created, **Machine learning scoreboard**.





3.2.7.2 Editing the logic definition of the predictive risk

You update the decision logic by adding a predictive rule. You add a rule calling the predictive model with the appropriate input, to define the risk and the confidence of the Scoreboard decision node.

1. Select the decision node **Predictive risk**. In the **Logic** tab, click the plus (+) sign and create a **Business rule**.
2. Enter a name for the rule: **predictive risk**.
3. Click on **Create**.
4. Edit the rule statement as follows:

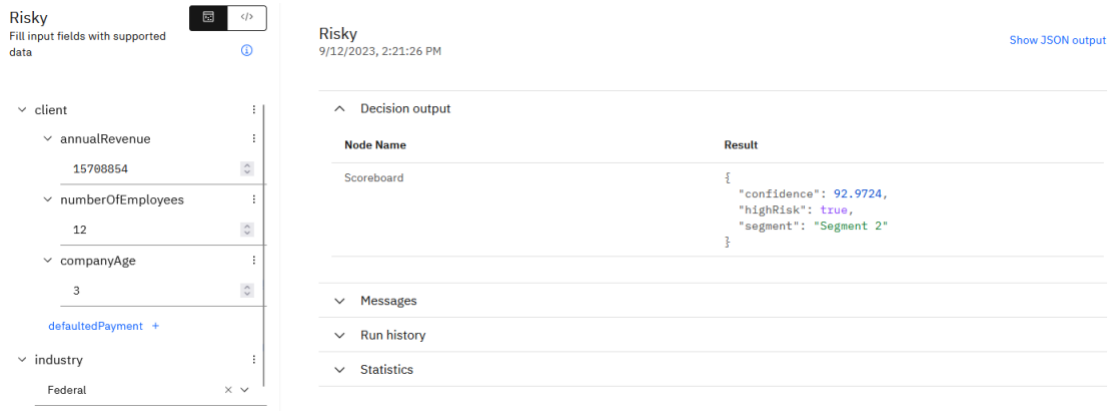
```
set decision to the machine learning scoreboard computed from
     ClientInformation being Client ,
     Industry being Industry ;
```

3.2.8 Validating the final decision service

Now that you created the predictive model and updated the decision model Scoreboard to integrate the prediction to compute the risk for the client onboarding, you can validate the changes. You use the four datasets previously created to validate: Risky, May be risky, Not so Risky, Safe

1. Select the **Run** tab.

2. In the **Test data** tab, select the **Risky** dataset. Then click **Run** and check the results.



The run history shows that the rule **Predictive risk** ran and computed the **confidence** level 92.97% and **high risk** value (true) for the **Scoreboard** decision node.

Run history

Node	Rules	Rule Interaction	Output
Scoreboard	1	Sequence	{ "confidence": 92.9724, "highRisk": true, "segment": "Segment 2" }
Predictive risk	1	Sequence	{ "confidence": 0.929724477256148, "highRisk": true, "segment": "Unknown" }
Segment category	1	Sequence	"Segment 2"
Industry	1	Not applicable	"Federal"
Client	1	Not applicable	{"annualRevenue":15708854,"companyAge":3,"defaultedPayment":false Show more

3. Click on **<user> Client Onboarding** in the breadcrumbs to move on to the next exercise.

3.3 Summary

You completed Exercise 2 - Adding machine learning in the decision model.

- You modified the scoreboard for the client onboarding by combining descriptive rules and predictive rules into a unified decision. By adding a predictive model to your project decision, you infused machine learning to evaluate the client risk based on a trained model.
- You added data sets to verify that your model is running correctly against the business rule policies defined for the Client Onboarding scenario.

4 Exercise 3: Sharing and publishing decision services

4.1 Introduction

In this exercise, you learn how to collaborate on your decision services and how to make them ready to execute by the other components of the platform. You connect your project to a Git repository to be able to build and deploy a decision service as an archive. You can easily collaborate by sharing your decision service and giving appropriate access. You then publish the decision service archive as an automation service in Business Automation Studio.
This exercise is dedicated to integrators and anyone who wants to understand how to execute a decision service.

Integrating the automation services in other applications is not covered in this Lab. Look at the Workflow and Business Automation Application Labs for this.

For this exercise, you can either work with the project you created or with the decision services prepared by the Lab team.

4.2 Exercise instructions

In this exercise you prepare a decision service for collaboration:

- **Connect** a decision service to a Git repository
- **Share changes** in a decision service to make them visible to collaborators
- **Deploy** the decision service as **an automation service**
- **Execute** the decision service

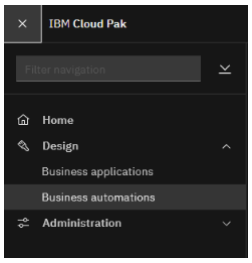
4.2.1 Log in to your project

If you did the previous exercises and have your decision service you can continue with your project. In Decision designer, open your project. Then move to the step Exploring a decision operation.

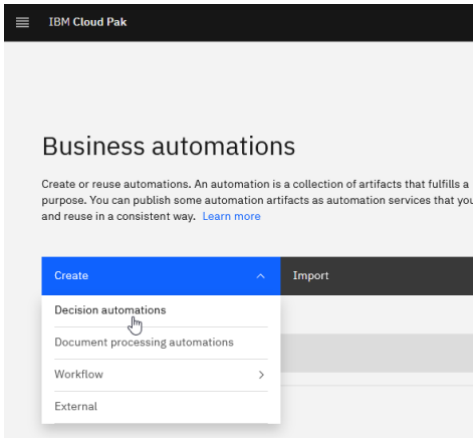
If you are starting the Lab with this exercise, follow the steps below.

4.2.2 Creating a project and importing a decision service

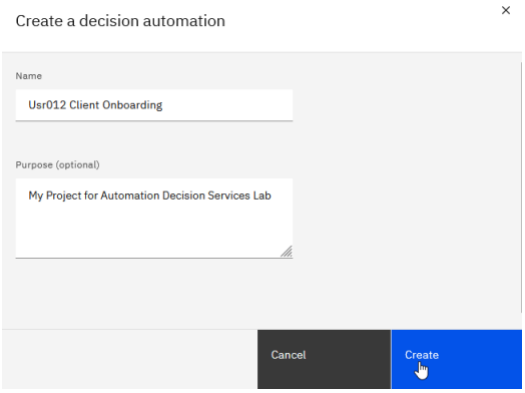
1. Log in to Business Automation Studio.
2. Click on the Navigation Menu on the top left corner IBM Cloud Pak.
3. Expand **Design**, and then click on **Business automations**.



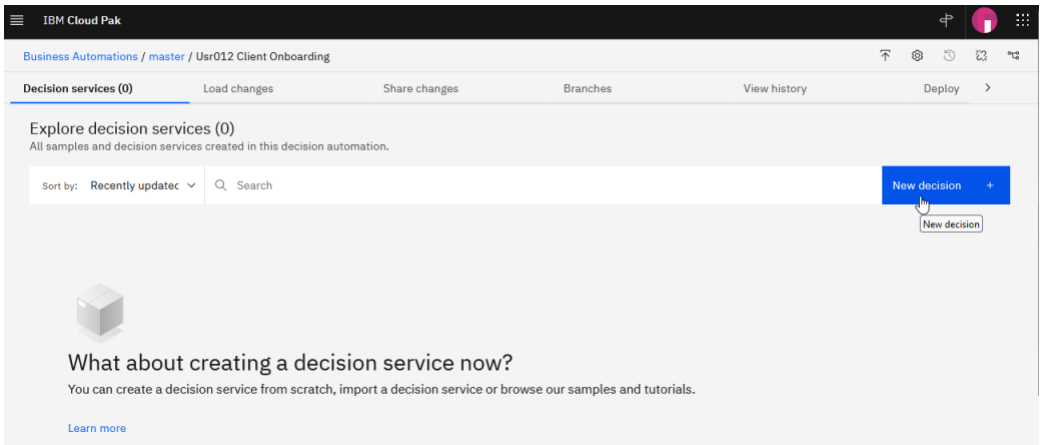
4. Click Create and select Decision automations.



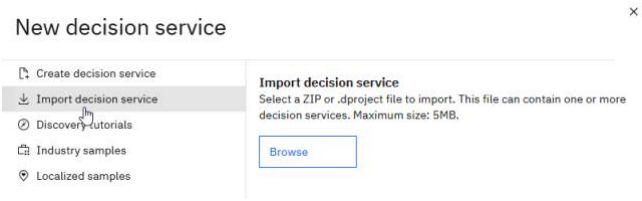
5. Enter a name for your project. Enter **UsrNNN Client Onboarding Decision** where *UsrNNN* is your assigned username (this avoids conflicts with other projects if you are sharing the cluster).
6. Click Create.



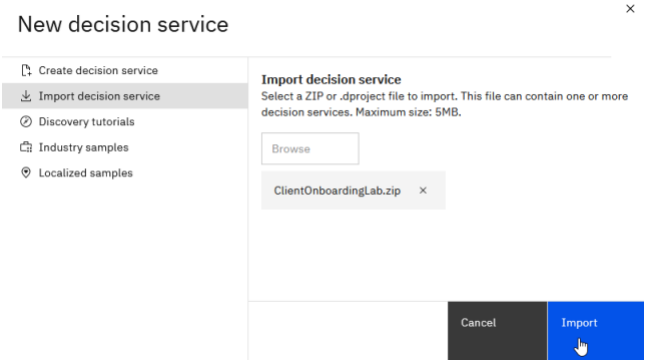
7. Your new project opens in Decision Designer. You can optionally follow the guided popup if it shows or you can click **Maybe Later** to continue.
8. Click New decision +.



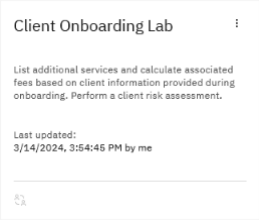
9. Select Import decision service to import the decision service provided by the Lab team.



10. Browse to select the project prepared for the Lab **ClientOnboardingLab.zip** and click **Import**.



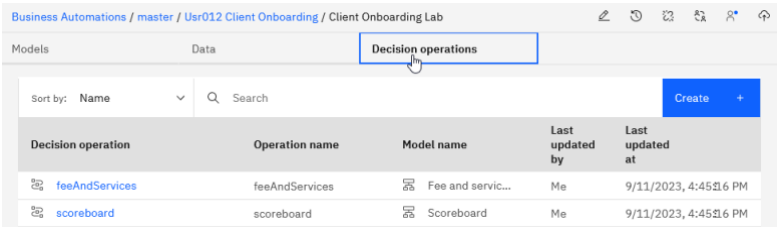
11. A tile of a decision service named **Client Onboarding Lab** appears on the project page.



4.2.3 Exploring a decision operation

To deploy your decision service, you need to define an operation that is used to call the service. The operation includes the name and a reference to a decision model. A decision service must contain at least one decision operation to be deployed and executed. The decision service prepared for the Lab already contains two decision operations **feeAndServices** and **scoreboard**.

1. Click on the tile of the decision service Client Onboarding Lab.
2. Open the **Decision operations** tab to explore the Decision operations:



3. Click **<user> Client Onboarding** in the breadcrumbs to return to the project.

For more information, see the IBM Documentation [Creating decision operations](#).

4.2.4 Creating a Git repository

As a prerequisite you must have a GitHub account to do this step.

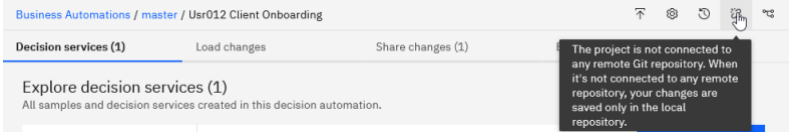
4. Open GitHub in your browser, and sign-in with your GitHub credentials.
5. Click the plus (+) button at the upper-right part of the page and select **New repository** to create an empty repository.
6. Give the repository a unique name, and add the following description:

   Git repository for the Automation Decision Services Lab
7. Select Private and Click on **Create repository**. (The repository must **not** contain a readme, .gitignore, or license file.)
8. To the right side of the HTTPS URI field, click the copy button, then paste the value in a safe location for use in the next step. The URI has the following format: https://github.com/<yourAccountName>/<yourRepoName>.git
9. Open the drop-down list for your GitHub account in the upper right corner of the page.
10. Click **Settings** and then **Developer settings** > **Personal access tokens**.
11. You can choose either **Fine-grained tokens** or **Tokens (classic)** based on your preference although **Tokens (classic)** is easier.
12. Click Generate new token, enter a name, and make sure it has access to the **repo** scope permissions to give full control of the repository you just created to Automation Decision Services.
13. Click **Generate token** at the bottom of the page. Copy the generated access token before closing this page, then paste the value in a safe location for use in the next step.

4.2.5 Connecting your project to the Git repository

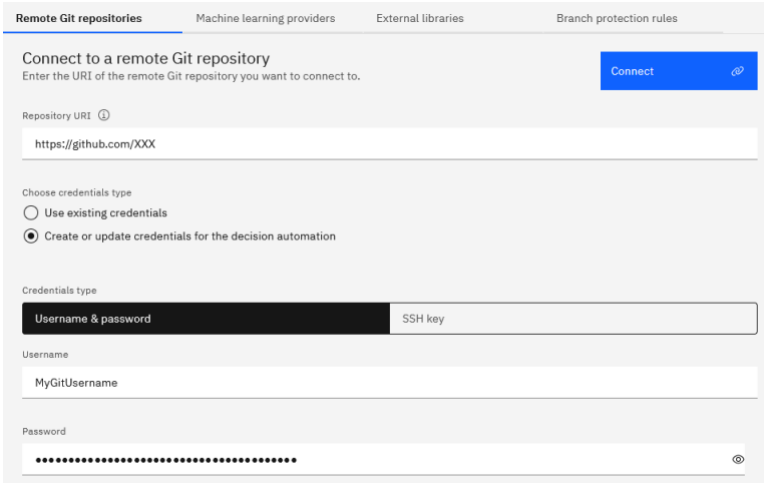
You now manage your project on a Git repository where you will get the history of all changes from Automation Decision Services. Connecting to a Git repository is required to be able to build and deploy archives.

1. Check the status of **Remote Git repository** in the upper right corner of Decision Designer. It shows that the project is not connected. If you do not see this icon, click **<user> Client Onboarding** in the breadcrumbs to return to the project.

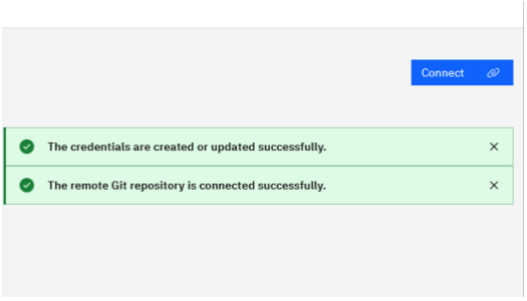


2. Click on this Git connection button to make a connection.
3. Enter the URI you previously saved, select **Username & password**, and enter your Github username and the personal access token you previously saved for the password. Then click on

Connect.



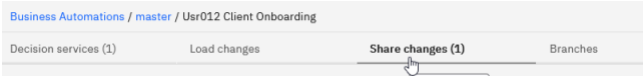
Upon successful connection, Decision Designer displays the following messages and updates the status of the Remote Git repository.



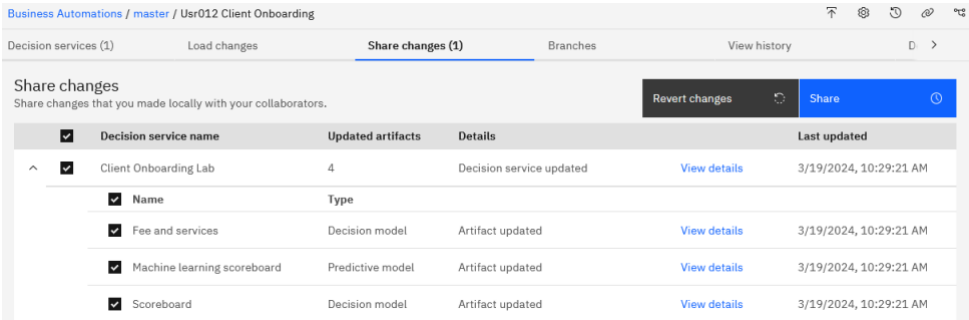
4.2.6 Sharing your decision service

The changes made on a decision service become visible when you share them. You give the access rights in Business Automation Studio to your collaborators.

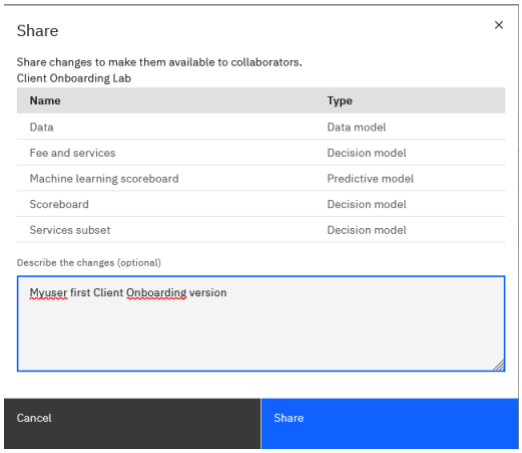
1. Return to <user> Client Onboarding and click on the Share changes tab.



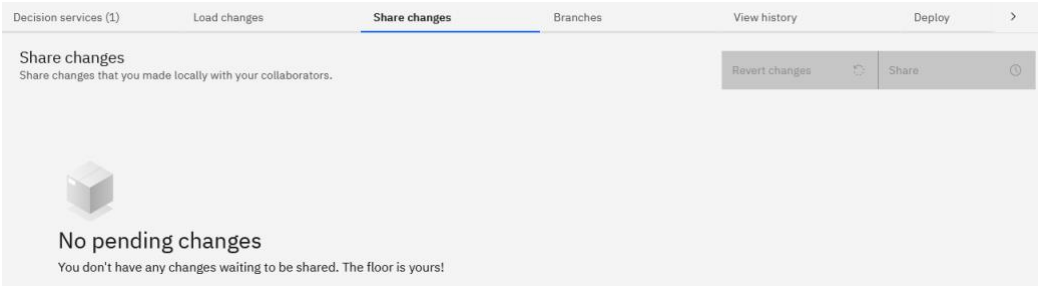
It displays the changes you made to the Decision service and shows the number of changes. You see the artifacts in your decision service and not for your project (the number of changes may be slightly different).



2. Ensure **Client Onboarding Lab** is selected and click **Share**.
3. Enter a comment to describe the changes (`<user> first Client Onboarding version`) and then click **Share**.



4. After the changes are shared, there are no pending changes.

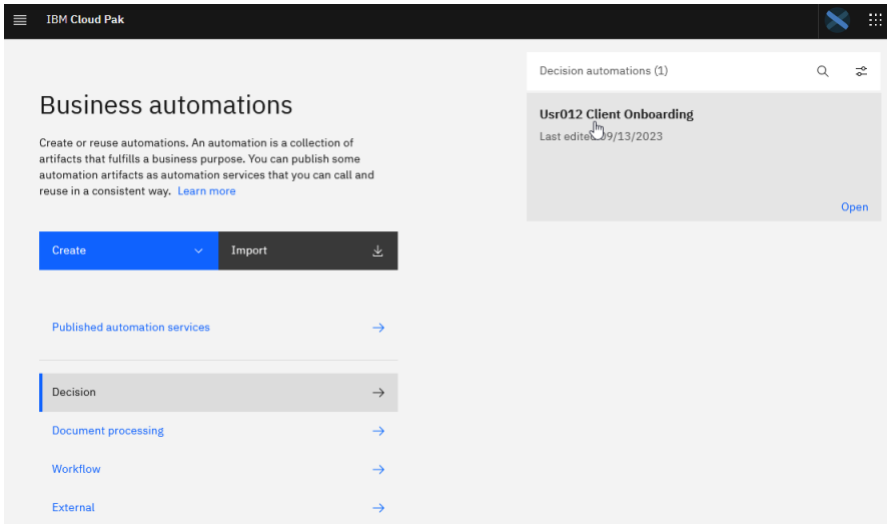


Sharing changes means that updates done locally are published and visible to other users provided that you give them access.

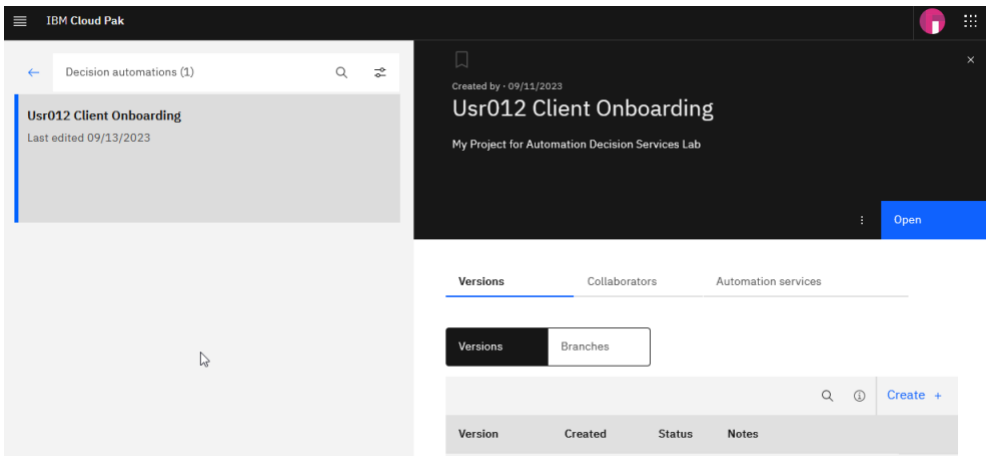
In this Lab, adding other users is not required. However below is the procedure you would follow.

To share a project with other users, click **Business Automations** in the breadcrumbs or go to **Business automations** from the upper left menu, then select **Decision**.

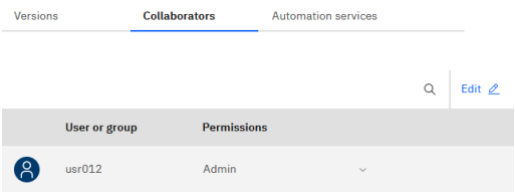
1. In Business automations you select your project by clicking on your project name. Search if you cannot find it easily.



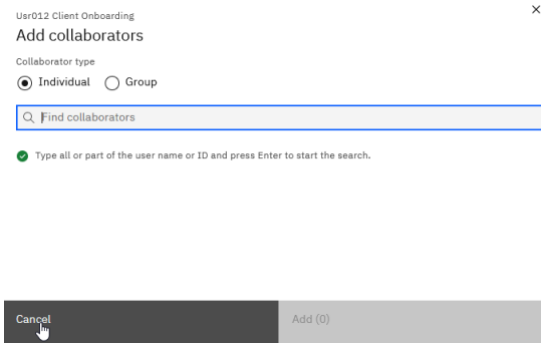
5. The right-side panel opens with your project details.



6. Select the **Collaborators** tab. You have Admin permissions on your project by default. To add collaborators, click **Edit**, enter the names of the users you want to add and select the permissions (Admin, Edit or Read).



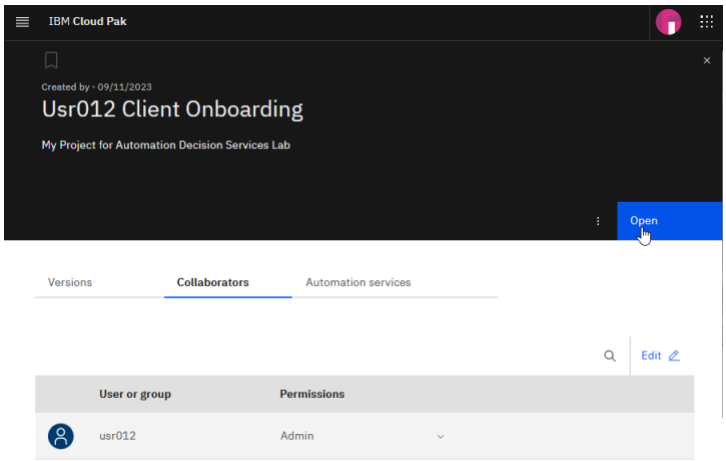
7. Click **Cancel** (adding users is not covered in this lab).



4.2.7 Creating a version

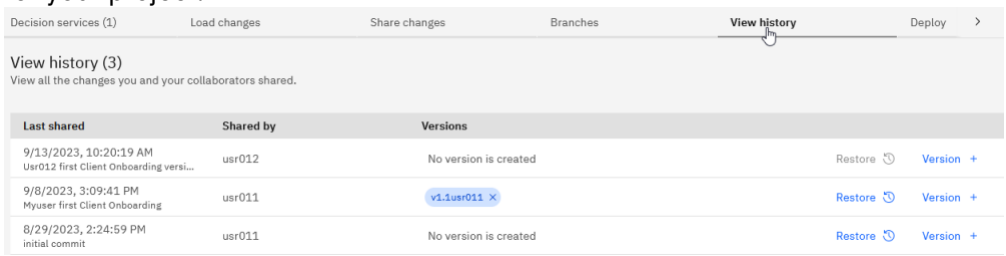
To deploy your decision service, you need to create a version of the current content. Versions correspond to tags in Git. A version is a snapshot of the project and records a point-in-time of the decision services within the project.
Creating versions requires one of the following permissions in the project:
- **Admin** - Administrative privileges
- **Edit** - Write permission

1. Return to your project by clicking on **Open** if you see the following screen. Otherwise navigate from the upper left menu.

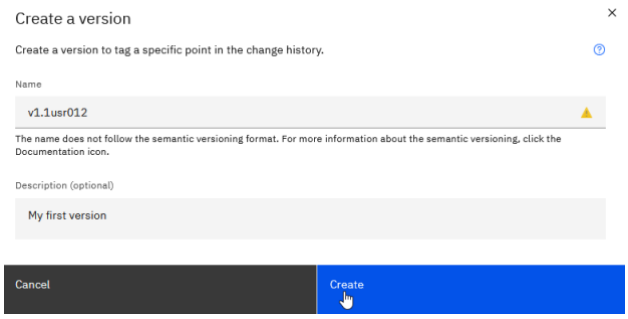


2. Open the **View history** tab.

   It shows all the changes made to the project. You may have a slightly different amount of history for your project.

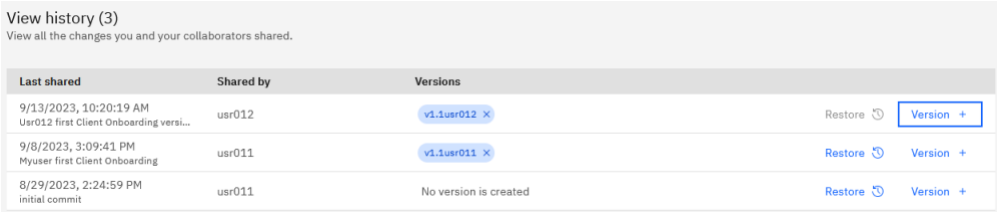


3. In the row for the most recent version, click the **Version +** button at the right.
4. Enter a name for this version, it is recommended to add your user number to the name such as v1.1**usrNNN** where **usrNNN** is your username) and a description.



5. Click **Create**.

   A new version is created:



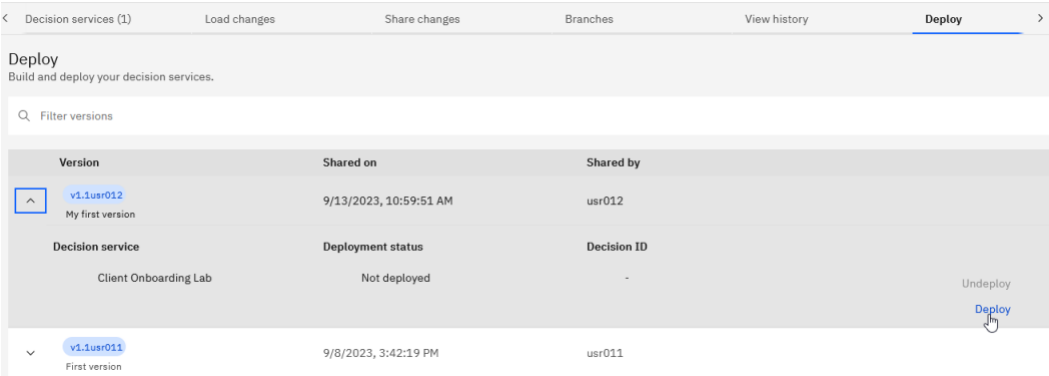
   You are now ready to deploy your version.

4.2.8 Deploying your project

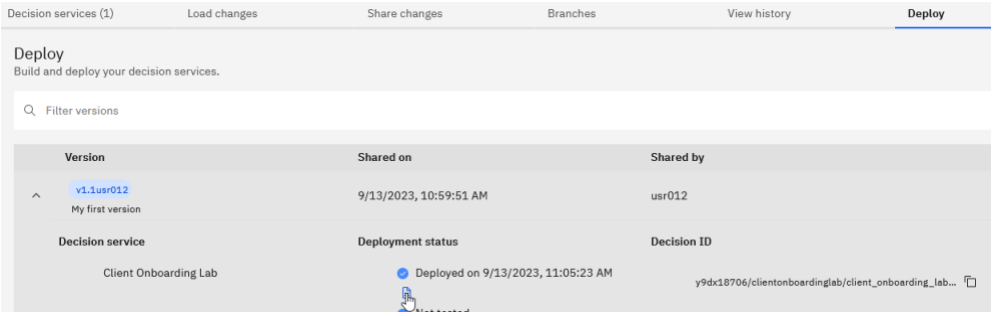
You build and deploy your decision service directly in Automation Decision Services. A decision service archive is built from your decision service and is deployed to the decision runtime in the deployment space ID named 'embedded'. It is then ready to be published as an automation service.

1. Click the **Deploy** tab and expand the version your created in the previous section.

   Note: in this example the name of the version is v1.1usr012 but in your exercise you added your user number as the name of the version.



2. Click **Deploy** and **Deploy** again in the confirmation window. This triggers a build and deployment through the embedded repository for runtime archives. Wait for the deployment to finish (it could take a few minutes).
3. Once completed, you can check the logs by clicking the **View logs button** in the Deployment status.

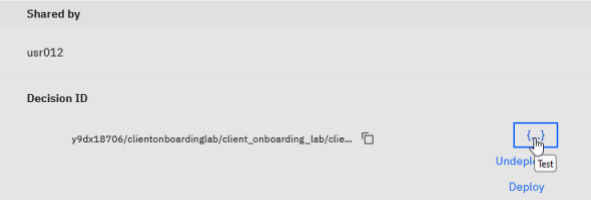


   Note that a **Decision ID** was added. The decisionID parameter is required to call the decision service in the runtime service. It provides the decision path to the generated decision service archive.

4.2.9 Executing your decision through the ADS runtime

To directly execute your decision in the ADS runtime:

1. Click **{...}** next to the decision id. This opens the Swagger UI tool dedicated to the REST API generated for your decision service archive.



2. Expand `POST /feeAndServices/execute`, click **Try it out**, and enter the following JSON:

```
{
 "client": {
   "annualRevenue": 27500000,
   "companyAge": 25,
   "defaultedPayment": false,
   "numberOfEmployees": 350
 },
 "services": {
   "servicesRequested": [
     "Employee Benefits Plan",
     "Mental Health Care",
     "Onsite Medical Testing",
     "Virtual Medical Assistance"
   ],
   "Industry": "Healthcare"
   }
}
```

3. Click **Execute**.
4. Scroll down to **Server response** and see the **Code** is **200** and the **Response body**:

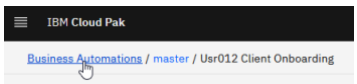
```
{
 "extendedServices": [],
 "servicesFee": 47500
}
```

Note: You may get a different result if you did not do Exercise 1.

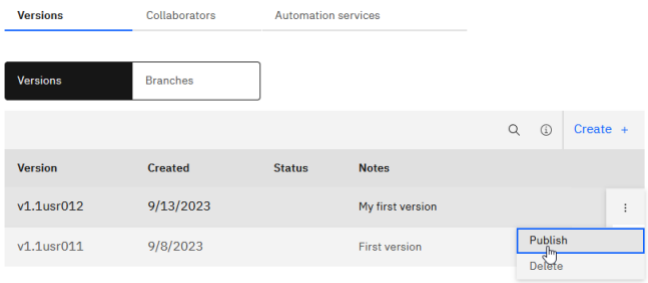
4.2.10 Publishing your decision service through Business Automation Studio

To publish the deployed version of your decision service into the catalog of Automation Services, follow the below procedure.

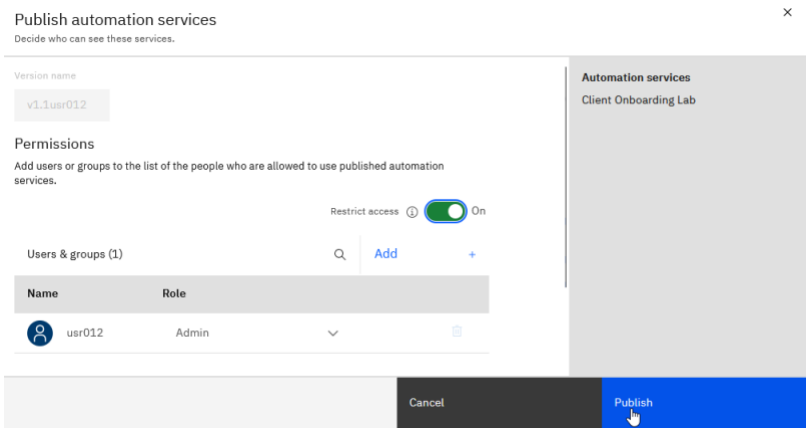
1. Close the Swagger UI browser window.
2. Click on **Business Automation** in the breadcrumbs.



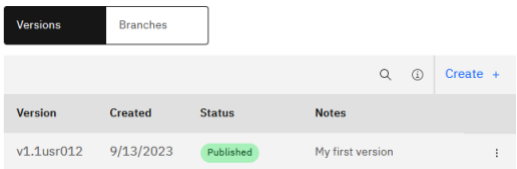
3. Select the **Versions** tab, click the 3 dots menu, and then click **Publish.**



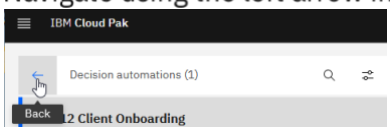
4. The next dialog lets you set users or group permissions to use or administer the automation service once published. Set **Restrict access** to On to prevent exposing your service to all other users on the cluster, and then click **Publish.**



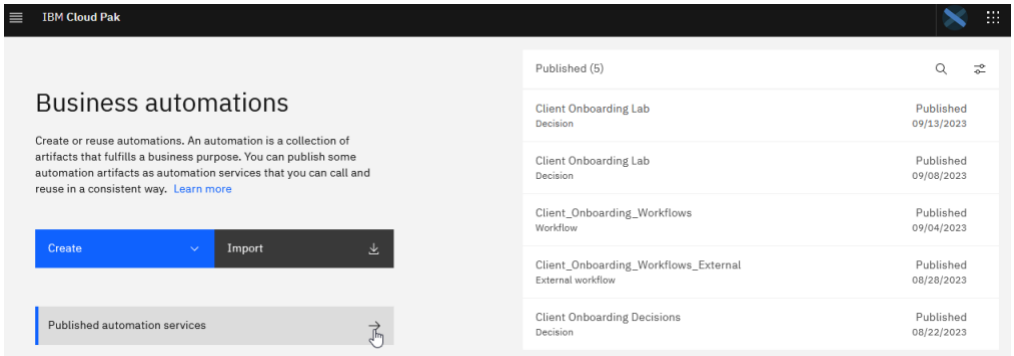
After a few moments, the **Status** is updated to Published.



5. Navigate using the left arrow in the upper left of the Business Automation Studio



6. Expand the **Published automation services** list to see your published service.



Once the archive is published as an automation service, you can execute it in other Cloud Pak for Business Automation capabilities. Look at the Workflow and Business Automation Application labs to learn how to work with published decisions.

4.3 Summary

You completed Exercise 3 - Sharing and publishing decision service.

- You made updates to your decision services visible for other collaborators by sharing the changes.
- You connected your decision project to a Git repository.
- You created a version and explored the procedure to deploy and publish a decision service.

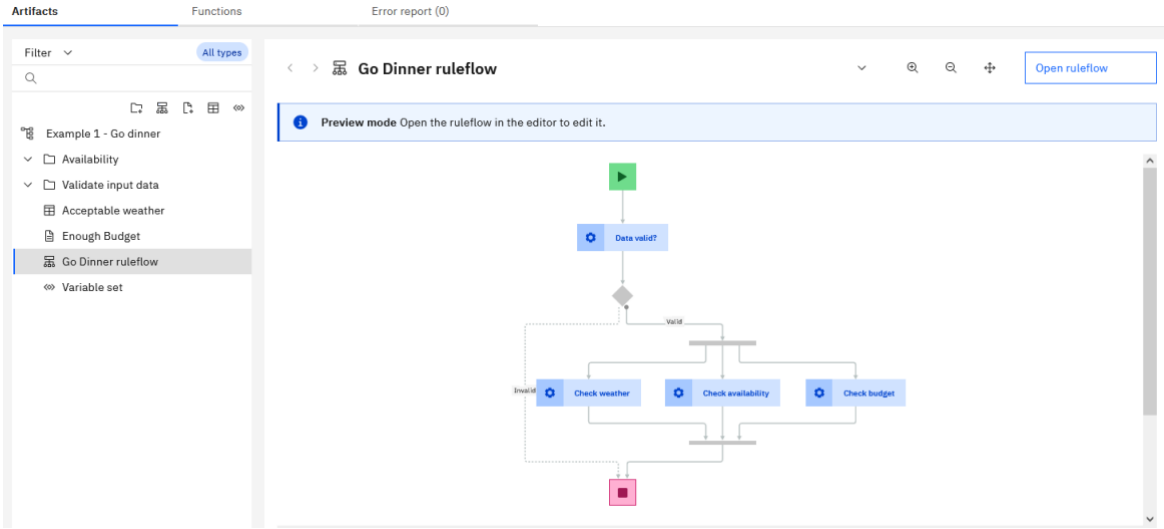
Congratulations on completing the lab!

5 Additional information

5.1 Explore the Samples

Samples and tutorials for Automation Decision Services
[Samples and tutorials in GitHub]

The repository of decision services includes the Training sample, which has several decision services to introduce the main features in Automation Decision Services: diagrams, business rules, decision tables, rule policies, and data and task models. The sample includes a series of decision models that gradually increase in complexity in defining a decision service. The following image shows a task model from the Training sample:



5.2 Consult Documentation and Communities

IBM Documentation
[IBM Automation Decision Services]
[What is Automation Decision Services]

IBM Business Automation Community
[Decision Management]

DMN
[Decision Model and Notation]