**Lab Guide**

IBM Decision Manger Open Edition

Nigel Crowther – ncrowther@uk.ibm.com

Hands-on Guide

DMN Beyond the Basics



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 2020.

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**

[Introduction 4](#_Toc130900564)

[1 Prerequisites 5](#_Toc130900565)

[2 What is DMN 6](#_Toc130900566)

[3 Using Data Types 7](#_Toc130900567)

[3.1 Why use Data Types? 7](#_Toc130900568)

[Figure 1: Applying a Data Model to Unstructured Data 7](#_Toc130900569)

[3.2 Lab 1 – Structure the Input Data 8](#_Toc130900570)

[Figure 8: PlanetInfo Data Type 9](#_Toc130900571)

[Figure 9: Planet DMN Input defined as a PlanetInfo Data Type 10](#_Toc130900572)

[Figure 10: Decision Table referencing PlanetInfo variable 10](#_Toc130900573)

[Figure 6: Test Result for Uninhabitable Planet 11](#_Toc130900574)

[4 Hit Policies and Decision Table Analysis 12](#_Toc130900575)

[4.1 Why use Hit Policies? 12](#_Toc130900576)

[4.2 Lab 14](#_Toc130900577)

[4.2.1 Unique Policy 14](#_Toc130900578)

[4.2.2 First Policy 14](#_Toc130900579)

[4.2.3 String Collection policy 14](#_Toc130900580)

[4.2.4 Numeric Collection policy 14](#_Toc130900581)

[5 Going beyond basics of Decision Requirements diagrams 15](#_Toc130900582)

[5.1 Lab 15](#_Toc130900583)

[6 Advanced DMN using FEEL 16](#_Toc130900584)

[https://access.redhat.com/documentation/en-us/red\_hat\_process\_automation\_manager/7.12/html/developing\_decision\_services\_in\_red\_hat\_process\_automation\_manager/dmn-con\_dmn-models 16](#_Toc130900585)

[6.1 Lab 16](#_Toc130900586)

[6.1.1 Import DMN project using Kie Sandbox 16](#_Toc130900587)

[6.1.2 Understand the service 17](#_Toc130900588)

[6.1.2.1 Data Types 19](#_Toc130900589)

[6.1.3 Decisions 20](#_Toc130900590)

[6.1.4 Functions 21](#_Toc130900591)

[6.1.5 Feel Functions 22](#_Toc130900592)

[6.1.6 Run the service 24](#_Toc130900593)

[6.1.7 Extend the diagram 28](#_Toc130900594)

[7 Conclusion 29](#_Toc130900595)

# Introduction

In this guide we go beyond basics of DMN to build real-world DMN.

The following topics are presented:

* **What is DMN**
* **Data Types**
* **Decision Table Hit Policies**
* **Advanced Feel**

By the end, you will be able to apply these real-world techniques to your DMN projects.

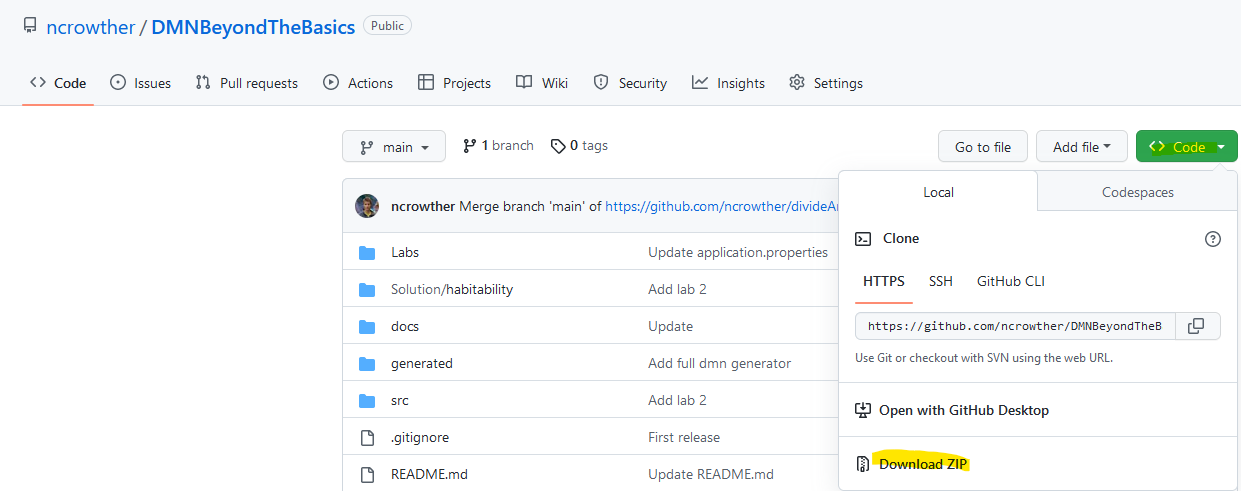
# Prerequisites

For this lab you need **Kie Sandbox** with *Kie Sandbox Extended Services* running. If you have not already done so, download and install the *Kie Sandbox Extended Services*.

You will also need a local copy of the following Git repo.

<https://github.com/ncrowther/DMNBeyondTheBasics>

Click on the link above and then click the *Code* button and then *Download ZIP*:



Unpack the zip to a local directory and note the location.

# What is DMN

DMN is a decision model based on a notation standard defined by the Object Management Group (OMG)

DMN models consist of the following five elements:

* **Decisions:** Nodes in the model where one or several inputs determine an output based on decision logic.
* **Input data:** Information necessary to determine a decision. This information usually includes business-level concepts or objects relevant to the business, such as a restaurant’s peak business hours and staff availability.
* **Business knowledge models:** Reusable pieces of decision logic. Decisions that have the same logic but depend on different sub-inputs or sub-decisions use business knowledge models to determine which procedure to follow.
* **Knowledge sources:** External regulations, documents, committees, policies, and so on that shape decision logic. Knowledge sources are references to real-world factors rather than executable business rules.
* **Decision service:** A decision service is a top-level decision, with well-defined inputs, that is published as a service for invocation. In the diagram it is represented by an overlay rectangle with round corners. The decision service can be invoked from an external application or business process (BPMN). For more information, see page 36 of the DMN specification document.

# Using Data Types

## Why use Data Types?

One of the easiest ways to improve DMN readability is to replace primitive inputs with data structures. In this lab you will learn how to import a data type and apply it to a diagram using primitive attributes. See below:

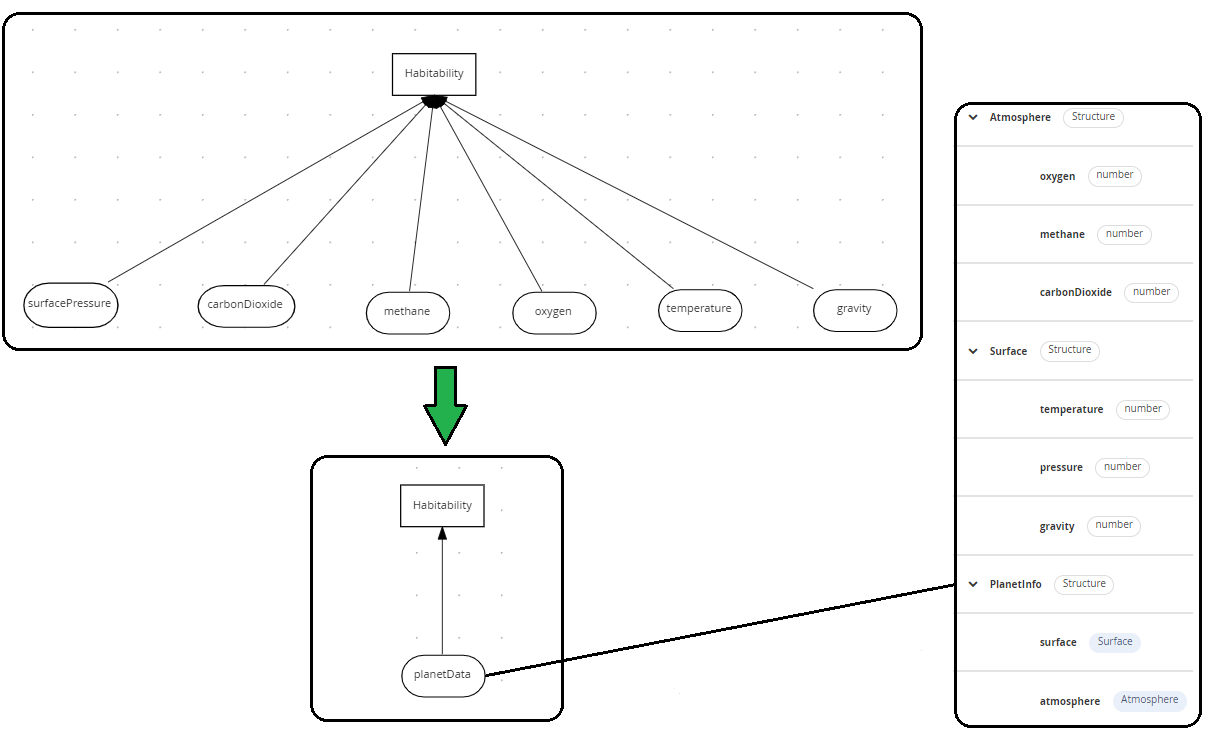


Figure 1: Applying a Data Model to Unstructured Data

## Lab 1 – Structure the Input Data

1. In browsers Chrome or Safari open the web site <https://sandbox.kie.org/>  
     
   Graphical user interface, application, website

   Description automatically generated

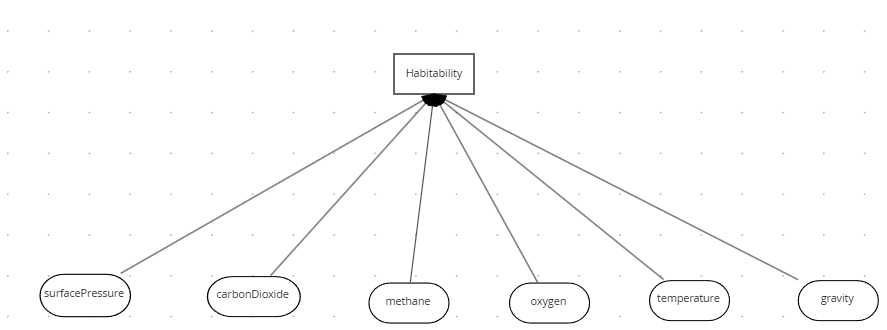
Click on **New Decision**.

1. In the top left corner rename the decision service from *Untitled* to Habitability.

An empty canvas opens. Click *New file* and then *upload…*

Select the DMN file: *lab01*/*HabitabilityStart.dmn*.

You should see this:



The inputs are all simple data types and there are lots of them. This is an antipattern! We will now apply a data type to this DMN to make it more readable and maintainable.

Click *New file* and upload *PlanetDataType.dmn*

Click on the *Data Types* tab. You should see the structure below. This is a data structure containing the attributes created in the DMN diagram above. See how it is structured into sub classes of *surface* and *atmosphere*.

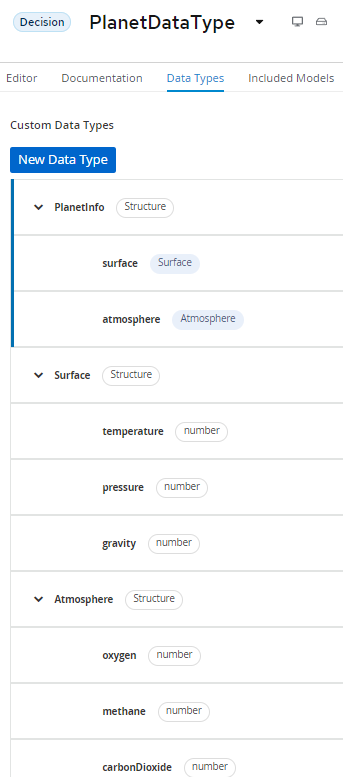
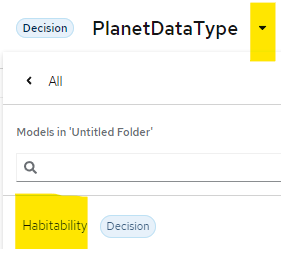


Figure 8: Planet Data Type

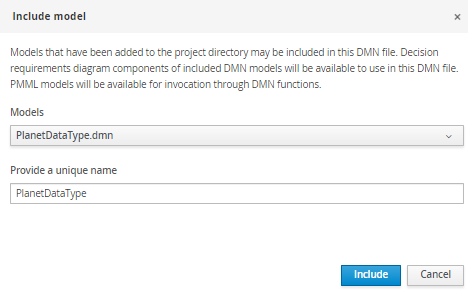
Go back to *Habitability.dmn* by clicking the drop down arrow next to *PlanetDataType*:



Select the **Included Models tab**, and then click **Include Model**



Add the *PlanetDataType* model and give it the same name of *PlanetDataType*



Click *Include*

1. Switch to the **Editor** tab.

Back in the DMN Diagram, replace all the input data with just one DMN input called *planet* of type PlanetDataType.*PlanetInfo*:

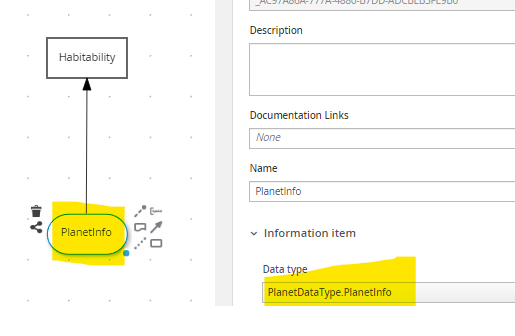


Figure 9: Planet DMN Input defined as a PlanetInfo Data Type

Edit the Habitability decision table to reference the *PlanetInfo* variable:

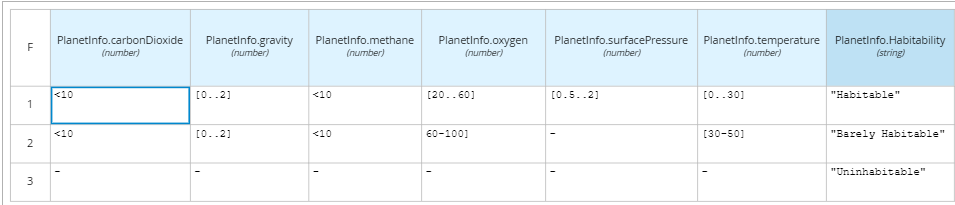


Figure 10: Decision Table referencing PlanetInfo variable

Test the model by pressing *Run*. If you have not installed the *Kie Server Extended Services*, now is the time to do so.

1. Enter zero values for all attributes and the expected result is *Uninhabitable :*

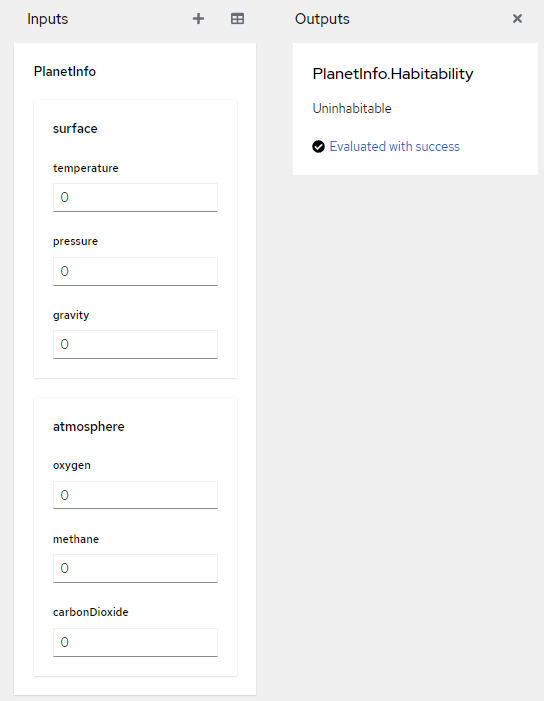


Figure 6: Test Result for Uninhabitable Planet

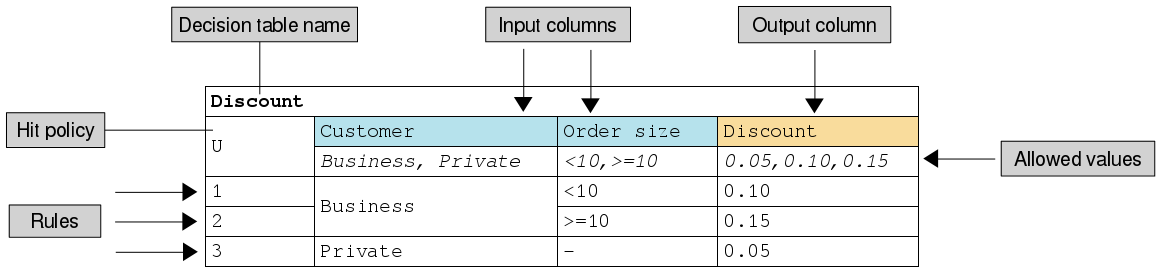
# Hit Policies and Decision Table Analysis

Hit policies change the evaluation order of a decision table. Each hit policy requires different ways of reasoning over the table. Therefore, it is important to be able to select the right hit policy for your application.

## Why use Hit Policies?

Not all decisions are the same! Some count outcomes, some require a go / no go decision and others require exact reasoning. You need a different hit policy for each of these.

In DMN the hit policy is specified in the top left of the decision table. See below:



The commonly used hit policies are:

|  |  |  |
| --- | --- | --- |
| Hit Policy | Description | When to use |
| ****Unique (U)**** | Permits only one rule to match. Any overlap raises an error. | For detailed reasoning. Ensures your rules cover all cases and are complete |
| First (F) | Rules are evaluated from top to bottom. Rules may overlap, but only the first match counts. | For concise decision tables where a go / no go decision is needed rather than complete reasoning. |
| ****Collect (C)**** | Aggregates values in an arbitrary list. | For multiple row satisfaction.  **String** aggregator:   * <None> - matching instances returned as list * Count – Matching instances counted   **Number** aggregator:   * SUM – matching instances added * Count – matching instances counted * Min – minimum value * Max – maximum value |

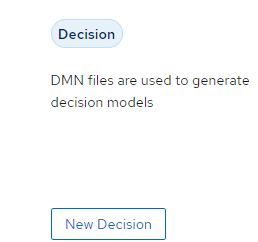
Other less commonly used hit policies are **Any** which has similar behaviour to **Unique**, and **Priority**, **Rule Order** and **Output Order** which have similar behaviour to **Collect.**

## Lab

Open **Kie Sandbox** in Chrome:

<https://sandbox.kie.org/#/>

Create a new decision:



Click *New File,* thenimport the following DMN as a URL. For each hit policy, run the rules and examine the results. Which algorithm is most appropriate for this use case?

### Unique Policy

<https://github.com/ncrowther/divideAndConquerDMN/blob/main/Labs/Lab01/src/main/resources/org/acme/habitability/Lab01/HabitabilityLab00-Unique.dmn>

### First Policy

<https://github.com/ncrowther/divideAndConquerDMN/blob/main/Labs/Lab01/src/main/resources/org/acme/habitability/Lab01/HabitabilityLab00-First.dmn>

### String Collection policy

<https://github.com/ncrowther/divideAndConquerDMN/blob/main/Labs/Lab01/src/main/resources/org/acme/habitability/Lab01/HabitabilityLab00-Collect.dmn>

### Numeric Collection policy

<https://github.com/ncrowther/divideAndConquerDMN/blob/main/Labs/Lab01/src/main/resources/org/acme/habitability/Lab01/HabitabilityLab00-Count.dmn>

# Going beyond basics of Decision Requirements diagrams

In the previous lab, we build the DMN diagram with just two components:

|  |  |  |
| --- | --- | --- |
| Description | Notation | |
| Decision | Node where one or more input elements determine an output based on defined decision logic. |  |
| Input data | Information used in a decision node or a business knowledge model. Input data usually includes business-level concepts or objects relevant to the business, such as loan applicant data used in a lending strategy. |  |

These two components are all you need to build simple models. But if you want full power, consider the following components:

|  |  |  |
| --- | --- | --- |
| Description | Notation | |
| Business knowledge model | Reusable function. Use this when decisions refer to common decisions or functions. |  |
| Decision service | Top-level decision containing a set of reusable decisions published as a service for invocation. A decision service can be invoked from an external application or a BPMN business process. |  |
| Text annotation | Explanatory note associated with an input data node, decision node, business knowledge model, or knowledge source. |  |
| Knowledge source | External authorities, documents, committees, or policies that regulate a decision. This component is not executable. |  |

Let’s explore these components in a lab

## Lab

# Advanced DMN using FEEL

In the final lab we put together what we have learnt so far and examine an advanced DMN example. The example has additional documentation here:

## <https://access.redhat.com/documentation/en-us/red_hat_process_automation_manager/7.12/html/developing_decision_services_in_red_hat_process_automation_manager/dmn-con_dmn-models>

## Lab

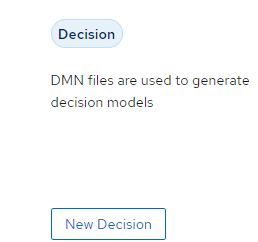
In this lab we will examine an advanced DMN decision service that reschedules flights.

### Import DMN project using Kie Sandbox

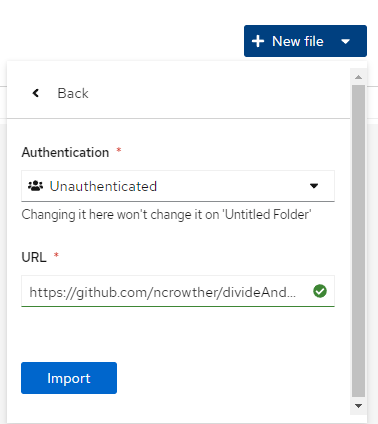
1. Open **Kie Sandbox** in Chrome:

<https://sandbox.kie.org/#/>

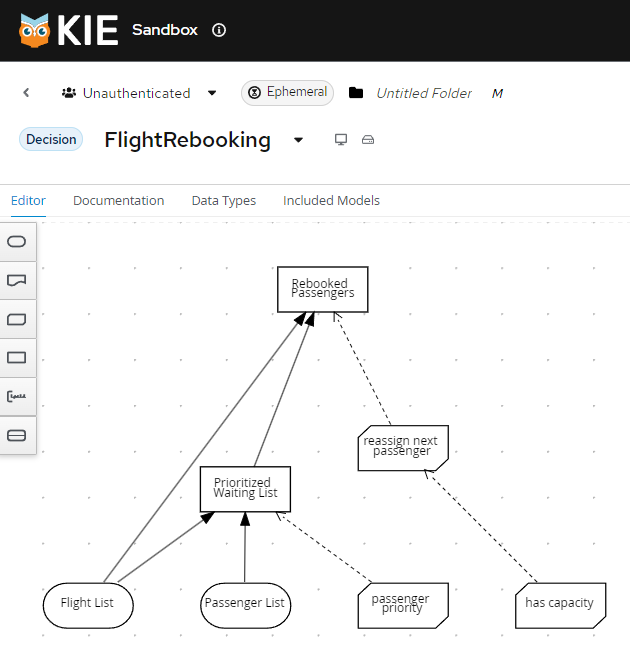
1. Create a new decision:



1. Click *New File* **
2. Select *From URL*.
3. Paste the text below into the URL: <https://github.com/ncrowther/divideAndConquerDMN/blob/main/Labs/Lab00/src/main/resources/org/acme/habitability/Lab00/FlightRebooking.dmn>
4. Click Import 



You should see the following:

­

### Understand the service

**Data Types**

A list of flights

A list of passengers

**Decisions**

Prioritizing the passengers who will get seats on a new flight

Determining which flights those passengers will be offered

**Business knowledge models**

The company process for determining passenger priority

Any flights that have space available

Company rules for determining how best to reassign inconvenienced customers

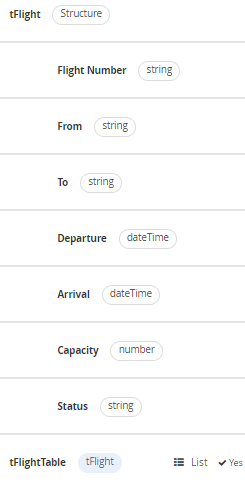
Then, the airline uses the DMN standard to model its decision process in a decision requirements diagram (DRD), and creates the following diagram for determining the best rebooking solution:

**Feel Functions**

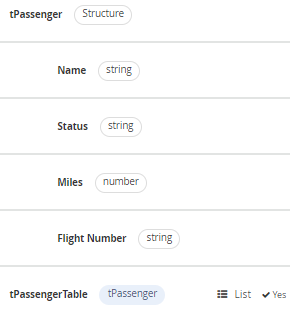
* Sort
* List contains

Let’s examine each of these components in detail, starting with data types.

#### Data Types

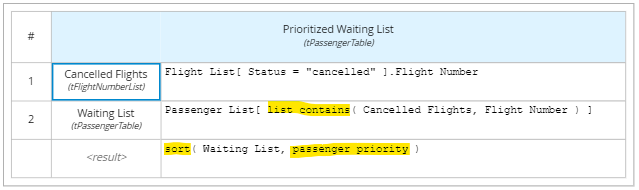


Flight

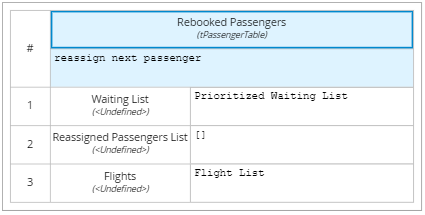


Passenger

### Decisions

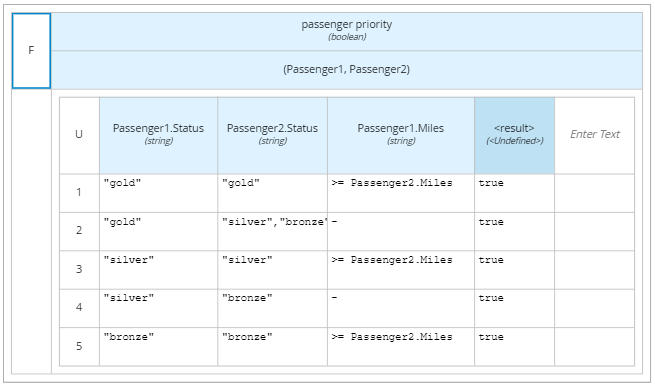


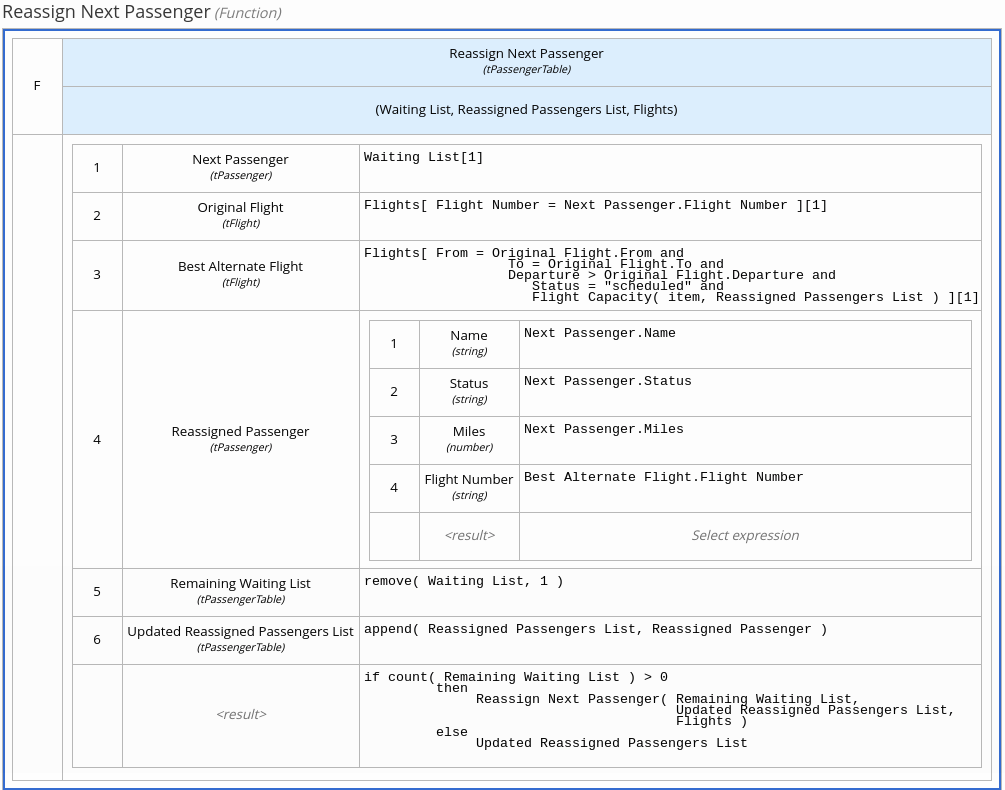
Prioritized Waiting List



Rebooked Passengers

### Functions





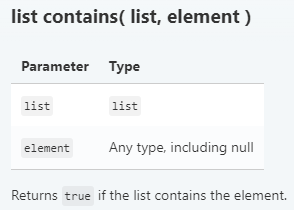
Reassign Next Passenger



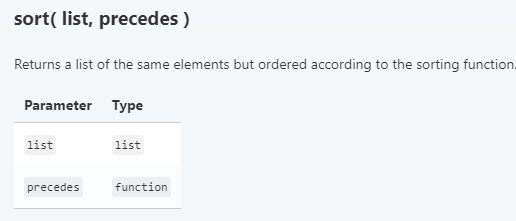
Has Capacity

### Feel Functions

<https://kiegroup.github.io/dmn-feel-handbook/#sort-functions>



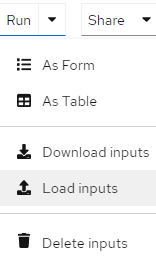
List Contains Function



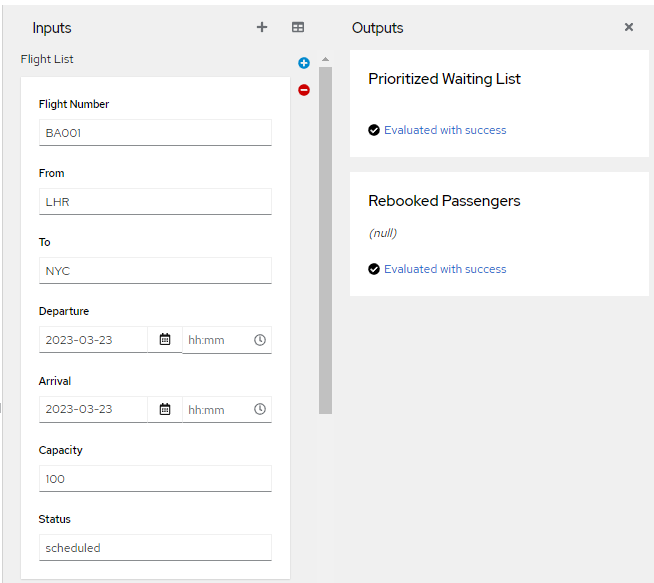
Sort Function

### Run the service

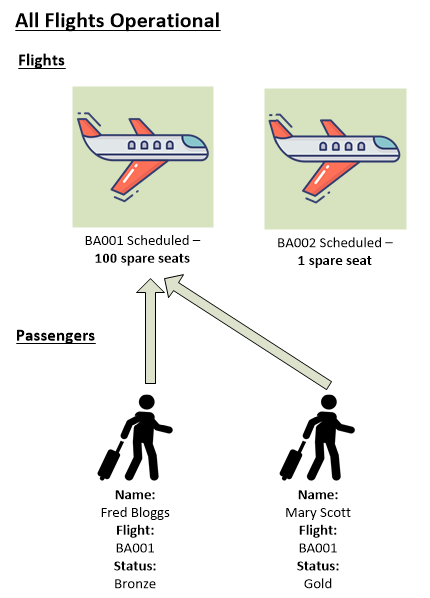
1. Now add some test data. Click *Run->Load Inputs*



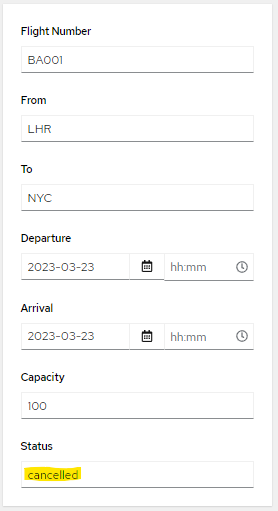
1. Open */Labs/Lab00/src/test/resources/FlightRebooking.json*
2. Click 
3. The service will run using the test data. You should see the test data in the *Inputs* panel and the results in the *Outputs* panel:



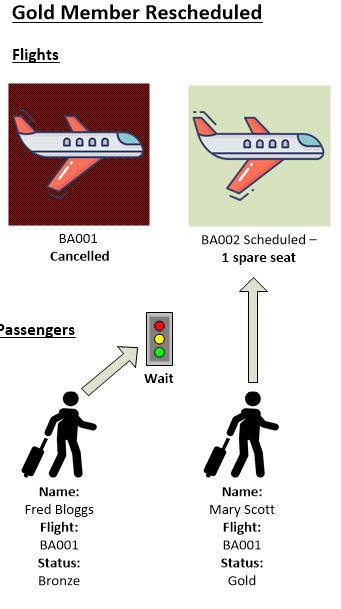
The output contains no waiting list or rebooked passengers all flights are operational. This is represented by the diagram below:



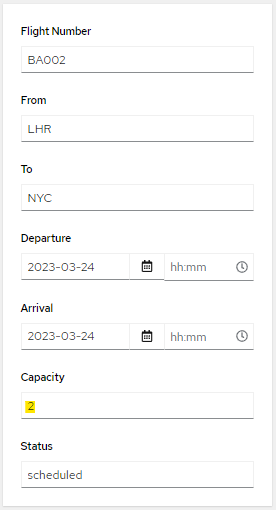
Now change the status of flight BA001 to *cancelled* (all lower case):



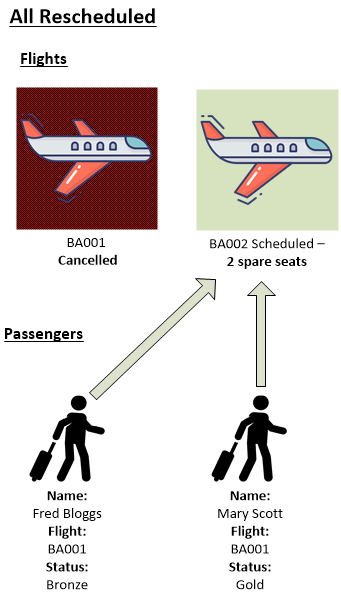
Check that *Mary Scott* is rebooked onto flight *BA002*. *Fred Bloggs* must wait as there is only one spare seat and Mary is a gold member.



Now change the spare seat capacity on flight *BA002* to *2*:



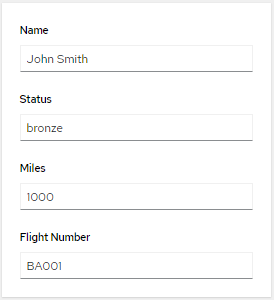
You should see that both Mary and Fred are scheduled to fly on *BA002*:



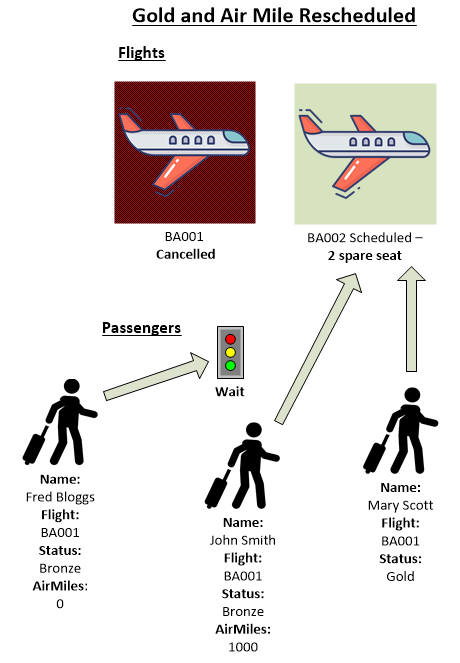
Add another passenger, *John Smith*. To do this, click on the plus button next to *Passenger List* in the *Inputs* Panel:



Add the following data:



After you enter the data, the test should automatically run. Verify that *Fred Bloggs* is put on hold again as *John Smith* has 1000 air miles and *Fred* Bloggs has 0



### Extend the diagram

Modify the DRG so that is formatted as shown in Figure 1. (to be expanded)

Modify the service (to be expanded)

# Conclusion

In this lab we went beyond the basics of DMN to give ideas and techniques for building real world DMN projects.