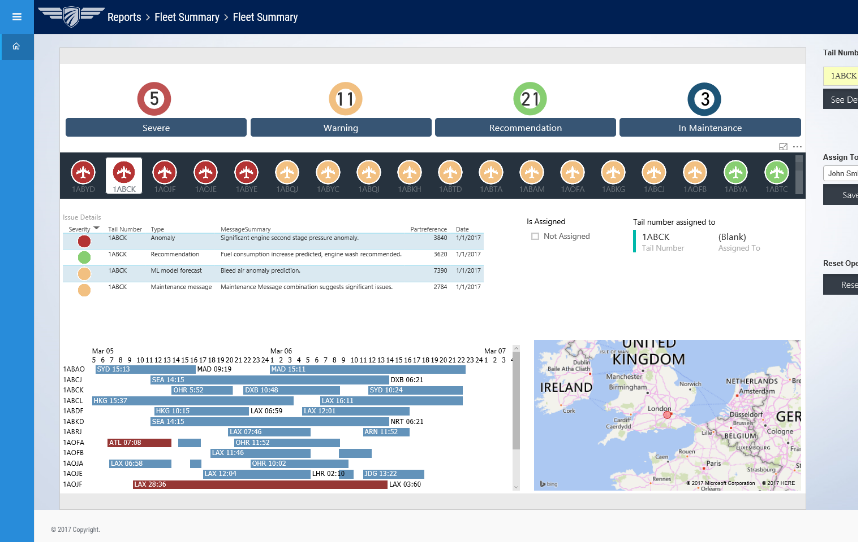


**Power BI Embedded**

Demo Script



March 2017

©2016 Microsoft Corporation. All rights reserved.

By using this demo/lab, you agree to the following terms:

The technology/functionality described in this demo/lab is provided by Microsoft Corporation for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the demo/lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this demo/lab or any portion thereof.

COPYING OR REPRODUCTION OF THE DEMO/LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION IS EXPRESSLY PROHIBITED.

THIS DEMO/LAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS DEMO/LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCITONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

**FEEDBACK**. If you give feedback about the technology features, functionality and/or concepts described in this demo/lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You may not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE DEMO/LAB, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF DEMO/ LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE DEMO/LAB FOR ANY PURPOSE.

**DISCLAIMER**

This demo/lab contains only a portion of new features and enhancements in Microsoft Power BI. Some of the features might change in future releases of the product. In this demo/lab, you will learn about some, but not all, new features.

# Contents

[Contents 3](#_Toc476657192)

[Goals of the demo 4](#_Toc476657193)

[Products and Technologies showcased 4](#_Toc476657194)

[Scenario of the Power BI Embedded Airline Maintenance Demo 4](#_Toc476657195)

[Part I: Power BI Embedded for the Operations Fleet Manager 6](#_Toc476657196)

[Part II: Row Level Security for the Maintenance Engineer 9](#_Toc476657197)

[Part III: PowerApp for the Maintenance Engineer (optional) 15](#_Toc476657198)

[Conclusion 20](#_Toc476657199)

# Goals of the demo

With Power BI, you can now integrate stunning, fully-interactive reports into customer facing applications, mobile or web, bringing the power of analytics everywhere decisions are made. The goal of this demonstration is to showcase the new Power BI Embedded functionality, an Azure service that embeds reports. This service makes it possible not only to deliver high-fidelity data visualizations across any device anywhere, but also reach massive scale, reliably and securely.

# Products and Technologies showcased

This demo showcases the following Power BI Embedded features:

* Power BI Embedded – reports embedded in a web app and mobile app.
* Power BI JavaScript API - provides bidirectional communication between Power BI reports and your app
* Row level security – put the right data in front of the right user, every time
* Microsoft Azure web application
* Power Apps and mobile (optional)

# Scenario of the Power BI Embedded Airline Maintenance Demo

The demo is based on a jet engine manufacturer’s maintenance workflow scenario. In this demo, the company was in an interesting position. Their business was in transformation from just selling jet engines to leasing jet engines. Along with leasing jet engines, they were also offering maintenance and repair of jet engines. This manufacturer moved from a business model where they used to get paid upfront for a jet engine, to a model where they get paid for time that a jet engine is in the air. So, the decision about when to take an engine out for maintenance is a critical one for the business.

For this business scenario, the jet engine manufacturer used a web application. The manufacturer would like to integrate stunning, fully-interactive reports into customer-facing applications, bringing the power of analytics to the operations fleet managers and maintenance engineers. The reports were created using Power BI Desktop with many insightful graphics and data points to help the users make business decisions. The next steps were to make the web application and reports work together.

The important takeaways of the scenario are:

1. No need to redesign your existing web applications
2. Seamlessly embed reports into existing web application
3. Row level data security
4. Ability to respond and react to user interaction within the report [Power BI - Bidirectional data flow]

To better explain the scenario, the example uses two personas – the **Operations Fleet Manager** (Emily Hart) and the **Maintenance Engineer** (John Smith). The critical aspect of the existing web application was that based on the logged in user, the application presented different data. Integration of reports with the existing application needed to take these factors into account and the application needed to show the right dashboards and data to a given user. In this demo, we are showing a web application hosted in Azure by the jet engine manufacturer for the airline employees including the Operations Fleet Manager and the Maintenance Engineer.

[**Part I**](#_Part_I:_Bi-directional): When Emily Hart logs in as the Operations Fleet Manager, she is responsible for the overall health of the fleet. Her home page shows the status of all the engines, and she can assign engines needing attention to the Maintenance Engineers. She can see that 5 planes have been flagged in critical status, 7 have been flagged with warnings, and 14 have been flagged with recommendations. As the Fleet Manager, she can click on any of the flights to see the airplane details, current status etc. If she sees any flights that need attention, she can assign them to the Maintenance Engineer for analysis. She assigns the airplane 1ABCK to John Smith who is a Maintenance Engineer.

[**Part II**](#_Part_II:_Row)**:** When John Smith logs in as the Maintenance Engineer, he can see the fleet summary of the engines assigned to him and detailed diagnostics of the engines that need maintenance. He can see the flights that are in critical state, warning state, or healthy state. He sees that 2 jet engines which are in critical state are assigned to him. He can click on the plane with the tail number 1ABCK to navigate to a page with more details about sensor 17. He can further drill into a diagnostics report to see the engine from two different perspectives – a frontal, three-fourth, as well as a lateral view of the same engine. He can see the number of faults for various engine parts.

Optionally: The Maintenance Engineer uses his mobile phone with a Power App to take a picture and capture his notes for the next steps.

# Part I: Power BI Embedded for the Operations Fleet Manager

| Narrative | Click Step | Screenshot |
| --- | --- | --- |
| For more details about the scenario & personas, please see [Scenario](#_Scenario_of_the) section of this document.  For this portion of the demo, let me step into the role of the Operations Fleet Manager Emily Hart. I am responsible for the overall health of the fleet.  To start, I will log into my fleet operations web application. | 1. Open Microsoft Edge (or Internet Explorer) on your machine.   Navigate to the URL: <http://airlinedemo-embedded.azurewebsites.net/>   1. Login with the credentials:   **User**: emily.hart  **Password**: Password@123 |  |
| **Note:** Before proceeding with the demo let us reset thefleet assignment to maintenance engineer by clicking the reset button in the home page. | 1. Click on Reset button to reset the operator assignment. |  |
| The home page shows the status of all the engines, including the ones needing attention.  **Note**: The interactions of Emily Hart with the data the in Power BI Embedded reports are tracked and these events (or clicks) are communicated to the container web application.  Since I was authenticated as Emily Hart, I can see a summary of the various flights with the count of flights in critical (red), warning (yellow), and healthy (green) state. These are shown with Power BI custom visualizations.  The report shows that 5 planes have been flagged as severe, 11 have been flagged with warnings, and 21 have been flagged with recommendations.  As a Fleet Manager, I can click on any flight to see the details of the flight and the current status.  For the flights with engines that needs attention, I can assign them to the Maintenance Engineer for analysis. |  |  |
| I notice that a particular engine with tail number *1ABCK* is in critical state and needs attention.  I can select it, and the report will refresh with selected tail number as a filter.  I notice that the selected data point is also highlighted on the right-hand side of the web application.  I will now assign the plane with tail number *1ABCK* to John Smith, the Maintenance Engineer.  **Note**: Reports can be seamlessly embedded within a web app. The user interactions with the report visuals are also available to the container app, effectively removing the boundaries between the container and embedded report.  This shows bi-directional communication that is enabled between Power BI and the web app.  For more information about John Smith personal, please refer to [Scenario](#_Scenario_of_the) section. | 1. Click on tail number **1ABCK** in the report. 2. Select **John** **Smith** from the **Operator** dropdown list. 3. Click **Assign**. 4. Click the **Logout** button to exit the application. |  |

# Part II: Row Level Security for the Maintenance Engineer

| Narrative | Click Step | Screenshot |
| --- | --- | --- |
| For more details about scenarios and personas, please see [Scenario](#_Scenario_of_the) section of this document.  In this portion of the demo, I will step into the role of John Smith, the Maintenance Engineer.  Emily Hart has previously assigned the engine on the airplane with tail number *1ABCK* to me.  The reports and visuals that I can see are controlled and based on my role and differ from the date and visuals that Emily Hart can see.  This is enabled through row level security in Power BI which will be showcased in this portion of the demo. | N/A |  |
| To start, I will log into my maintenance web application as John Smith.  On my home page, I can see the details of the engines that are assigned to me. | 1. Navigate to the URL:   <http://airlinedemo-embedded.azurewebsites.net/>   1. Login with the credentials:   **User**: john.smith  **Password**: Password@123 |  |
| I can drill down into the details of the tail numbers that are in critical state. Based on the details, I can decide where and when the plane needs to be taken out for maintenance.  I can see three tail numbers that are flagged as critical. | 1. Highlight the red/yellow/green indicators on the home page. 2. Click on the red indicator with the number **3**. |  |
| Let me take a closer look at the tail number *1ABCK* and decide when to take it out for maintenance.  When we select *1ABCK*, I notice that a number of issues have been flagged for this plane. I also notice that the map has zoomed into London, UK where this plane currently is located. I can see the flight schedule as well. This helps me to quickly get a sense of where the plane is expected to be so I can better decide where we want to take the plane out for maintenance. | 1. Click on the **Severe** status selector at the top. 2. Highlight the three tail numbers which are shown. 3. Click on the **1ABCK** tail number. 4. Highlight the issue details and severity color codes filtered data. 5. Highlight the location of **1ABCK** in London, UK. 6. Highlight the schedule for tail number **1ABCK**. |  |
| As I review the set of issues that have been flagged, I notice there is one issue flagged as red - a significant engine second stage pressure anomaly.  Right here in this list, I can click to see a more detailed page where Power BI provides me with all the information about why this particular engine was flagged for requiring maintenance. | 1. In the **Issue Details** list, highlight the first row with the red indicator and the message summary. 2. Click on the **See** **Details** button, to the left of the red severity indicator. |  |
| In this report, I see that Power BI has flagged this engine as having an 81% chance of requiring maintenance right away.  This recommendation was driven by four sensors - sensors 17, 19, and the two sensors we see at the bottom, reaching significant thresholds. In the past, this combination has resulted in the engine requiring maintenance. | 1. The **Anomaly Detection** report will be shown. 2. Highlight the gauge chart for **Confidence of prediction**. 3. Highlight the area charts for the four engine sensors. |  |
| The very first sensor, which is sensor 17, was the biggest contributor to this overall prediction. | 1. Highlight the **Sensor 17** column in the **Determination by feature & rank** chart. |  |
| Let me click on engine diagnostics to see some of the details that are available. | 1. Click on the **engine Diagnostics** button. This will open engine diagnostics report. 2. Click on the **Faults by Engine Parts** visual. |  |
| In the engine diagnostics report, I can see the engine from two different perspectives: a frontal, three-fourth view as well as a lateral view of the same engine.  I notice that the parts of the engine have been color coded.  On the right-hand side, I see the faults categorized by engine parts that occurred over the last 12 months. The graph entries are color coded based on the part it represents.  For example, if I click on this part of the engine here, the cone, I notice on the right-hand side that the cone has been the part with the most defects – 65 over the last 12 months. | 1. Highlight the two different views of the engine. 2. Highlight the data in the bar chart, representing the engine parts. 3. In the **3/4** view of the engine, click on the blue cone. 4. Highlight the top bar for **Cone** in the **Engine Part by Faults** bar chart. |  |
| If I click on the second biggest contributor, the bleed air channel, you see the bleed air channel highlighted in both perspectives of the engine.  This is really interesting because this is an example of how custom visuals can bring data to life in a customer-specific context.  In this case, what made sense to the manufacturer, is to be able to see the faults overlaid on the schematic of the engine so that they can understand how the faults map to the different engine parts.  This is a great example of how embedded Power BI enhances the web app experience and allows the user to interactively find insights. | 1. Click on the bar for the **Bleed** **air channel** part in the bar chart. 2. In the engine visuals, highlight the location of the part and its color coding. |  |

# Part III: PowerApp for the Maintenance Engineer (optional)

| Narrative | Click Step | Screenshot |
| --- | --- | --- |
| [The demo starts with the Engine Maintenance page below]. | 1. Launch **PowerApps** on your mobile phone. 2. **Sign in** to **PowerApps** with your work account. 3. Tap on the **Airline Engine Inspection** app to launch it. |  |
| We saw in the previous portion of the demo, that the plane with the tail number *1ABCK* needed maintenance.  Now let’s assume that I am the Maintenance Engineer. I bring up this application on my iPhone that has been built using PowerApps.  PowerApps is a rapid application development platform, that helps business users and business analysts build applications without any coding whatsoever.  In this case, I am using this app on my iPhone, and you can see, as a maintenance engineer it has my schedule for the day. These are the planes with the engines that I need to inspect today.  You can see that the second item on my list is the plane with the tail number 1ABCK. | 1. Navigate to the **Engine Maintenance** schedule. 2. Highlight information provided – scheduled time, location, plane model and tail number, engine ID. 3. Tap on **Bay-C26** which has the plane with the tail number **1ABCK**. |  |
| Once I perform my inspection, I can recommend the action, let’s say in this case I’m going to recommend a repair.  I can also specify a date for when I want the repair to be performed, let’s say the 30th of the month, and click OK. | 1. Click the **Repair** button. 2. Click on the calendar icon to pick a date. 3. Click on **Photo Attachment**. |  |
| I can even take a picture and attach the photo for my inspection. | 1. Tap on **Capture** to take a picture of the engine parts. 2. Tap on the check mark when done. |  |
| Then I submit this inspection. | 1. Tap on the check mark to complete the inspection details for the **High Pressure Compressor** part**.** |  |
| You can see that the app acknowledges that this particular plane has a repair needed on a specific date. | 1. Navigate back to the **Engine Maintenance** page. 2. Highlight the check mark and **Repair Needed** status for tail number **1ABCK**. |  |

# Conclusion

With the new Power BI Embedded capability, application developers can integrate Power BI reports seamlessly into their apps, such as a mobile app or web app. There is no need to redesign the existing web applications. The web application can respond and react to user interactions within the report.

Thank you, and we hope you enjoyed this presentation!