



Was wir vorhersagen, soll auch eintreffen!



Realtime Daten-Rückschreibung in Tableau mit der Extensions API //

Pascal Muth



Zusammenfassung

In diesem Whitepaper wird die Tableau Extensions API von Tableau für Dashboards vorgestellt. Diese ermöglicht, den Funktionsumfang von Tableau beliebig zu erweitern, indem externe Web-Applikationen in bestehende Tableau Dashboards integriert werden. Durch die Integration ist eine Kommunikation mit den Inhalten des Dashboards möglich und es können spezielle Anwendungsfälle implementiert werden, angefangen von der Integration von 3rd Party Visualisierungen (z.B. d3.js Diagramme), über die Integration von Business Prozessen bis hin zu Write-Back Funktionen. Letzteres wird im Folgenden anhand eines Szenarios für Realtime Interaktionen mit Tableau Dashboards veranschaulicht. Durch die Entwicklung und Einbettung von Erweiterungen mit Hilfe der Extensions API können die limitierenden Faktoren von Tableau umgangen werden, ohne dabei den Anwender aus dem Software-Ökosystem Tableau loten zu müssen.

Schlüsselwörter

Tableau, Daten-Rückschreibung, Realtime, Write-Back, Dashboard, Realtime Visualisierung, Extensions API

Einleitung

Tableau stellt ein außerordentlich gutes Werkzeug für die Datenvisualisierung und -exploration dar. Auf einfache und schnelle Weise ist es möglich, bestehende Datenquellen anzubinden, zu transformieren und mit geeigneten Diagrammen aufzubereiten. Tableau bietet hierzu die essentiellen Mittel und Möglichkeiten, um die Daten entsprechend der Best Practices zu analysieren.

Es werden zum einen genau die Funktionalitäten geboten, welche der Analyst benötigt, um seine Daten effizient aufzubereiten zu können, ohne durch unnötige Funktionen im Erstellungsprozess aufgehalten zu werden. Auf der anderen Seite kann das aber dazu führen, dass nicht alle Anwendungsfälle und Business Prozesse damit gelöst werden können. Dieses Problem haben alle standardisierten Software-Lösungen zu bewältigen.

Um trotzdem Spezial-Fälle abzubilden, wurde die Extensions API eingeführt. Diese, per [GitHub](#) verwaltete open-source Javascript Library, erlaubt es den Funktionsumfang und die Möglichkeiten von Tableau fast beliebig zu erweitern. Die Erweiterungen selbst stellen externe Web-Applikationen dar, welche sich die Funktionen der Javascript Library zu Nutze machen, um mit den Daten und Elementen (Filter, Parameter) innerhalb des Tableau Dashboards zu kommunizieren und interagieren.

Im Folgenden wird auf den grundlegenden Aufbau der Tableau Erweiterungen eingegangen und die Funktionalität anhand eines Szenarios für Daten-Rückschreibung mit Realtime Visualisierung demonstriert.

Tableau Extensions API

Schon vor der Einführung der Extensions API gab es über die reguläre Tableau Javascript API die Möglichkeit, mit Tableau Visualisierungen zu interagieren. Die beiden APIs sind sich sehr ähnlich. Der Hauptunterschied liegt darin, dass für die Verwendung der Javascript API die Visualisierung über Tableau Online oder den Tableau Server gehostet werden muss. Die Extensions API erlaubt es nun, entsprechende Erweiterungen direkt in Tableau Desktop einzubetten und erfordert daher verschiedene Vorgehensweisen für die beiden Schnittstellen. Während bei der Javascript API das Tableau Dashboard in eine bestehende Web-Umgebung oder Unternehmensportal integriert wird, nimmt die Extensions API den umgekehrten Weg: Das bestehende Tableau Dashboard wird in eine externe Applikation eingebunden. Der Vorteil hierbei ist, dass die Erweiterungen flexibel geteilt und über das Tableau Objekt „Erweiterung“ zu den Dashboards hinzugefügt werden können. Eine Tableau Server Komponente ist somit nicht zwingend erforderlich, um die Erweiterungen zu nutzen, wird aber ebenso wie die Desktop Version unterstützt.

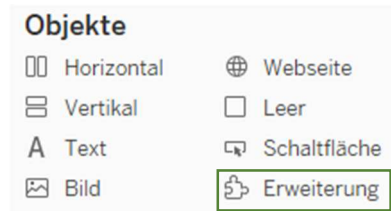


Abbildung 1: Tableau Objekt "Erweiterung"

Für die Entwicklung von Tableau Erweiterungen werden folgende Elemente benötigt:

- Eine erweiterungsfähige Tableau Desktop Version
- Eine Konfigurationsdatei (.trex-Datei)
- Eine gehostete Web-Applikation, welche die Extensions API nutzt und die entsprechende Business Logik enthält

Die Applikation für das Szenario wurde Client-seitig mit JQuery und reinem Javascript, sowie auf der Server-Seite mit den Javascript Frameworks NodeJs und ExpressJs implementiert. Als Datenbank dient die In-Memory Datenbank Exasol.

```
tableau.extensions.initializeAsync().then(function () {  
    console.log(">>>> Tableau Extension API initialized successfully");  
    // Add Mark Listener to "Forecast Overview" Worksheet  
    addMarkEventListener("Forecast Overview");  
}, function(error){  
    console.log(error.toString());  
});
```

Abbildung 2: Code-Beispiel zur Initialisierung der Extensions API

Use-Case

Als Szenario für die Applikation dient die Forecast-Planung für ausgewählte Einzelhandels-Produkte. Der Business Anwender, der das Tableau Dashboard verwendet, möchte seine Forecast Werte für den Absatz der Produkte planen. Für eine möglichst genaue Planung, werden verschiedene Simulationen durchspielt. Hierfür müssen die Parameter im Dashboard direkt verändert und in die Datenbank geschrieben werden, damit die Änderungen direkt ersichtlich werden.

Das Dashboard liefert gleichzeitig Indikatoren, für welche Produkte und Zeiträume ein Forecast Wert bereits vorhanden ist sowie ein Performance-Vergleich der Güte der eingegebenen Forecast Werte im Vergleich zu den tatsächlichen Absatzmengen. Somit ist eine ad-hoc Anpassung der Forecast Werte möglich, ohne dass das Software-Ökosystem Tableau verlassen werden muss. Der Business Anwender kann somit flexibel die Forecast Planung anpassen, ohne dabei auf die Unterstützung von z.B. ETL-Prozessen oder der IT-Fachabteilung angewiesen zu sein. Forecast Werte können angepasst oder hinzugefügt werden, indem der Anwender die Datenpunkte im Bereich „Forecast Year Overview“ auswählt und diese im „Data Entry Tool“ (die integrierte Web-Applikation, welche automatisch die ausgewählten Datenpunkte übernimmt) bearbeitet. Durch die Live-Verbindung zur Datenbank stehen die Änderungen umgehend zur



Verfügung. Mithilfe der Integration der Erweiterung konnte das Dashboard von einer reinen Datenvisualisierungs-Plattform in eine Daten Management Plattform transformiert werden.

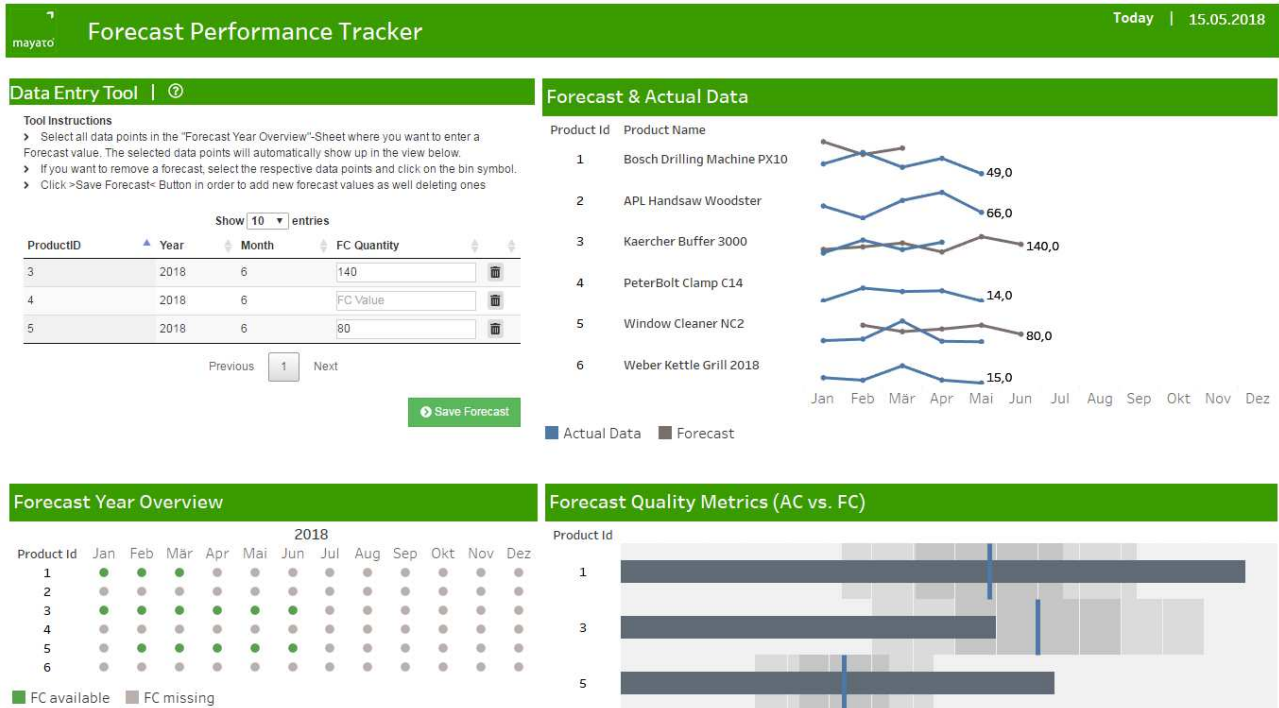


Abbildung 3: Tableau Extensions API Demo "Forecast Performance Tracker"



Kontaktieren Sie uns //

mayato GmbH
Am Borsigturm 9
13507 Berlin

info@mayato.com

+49 / 30 4174 4270 10