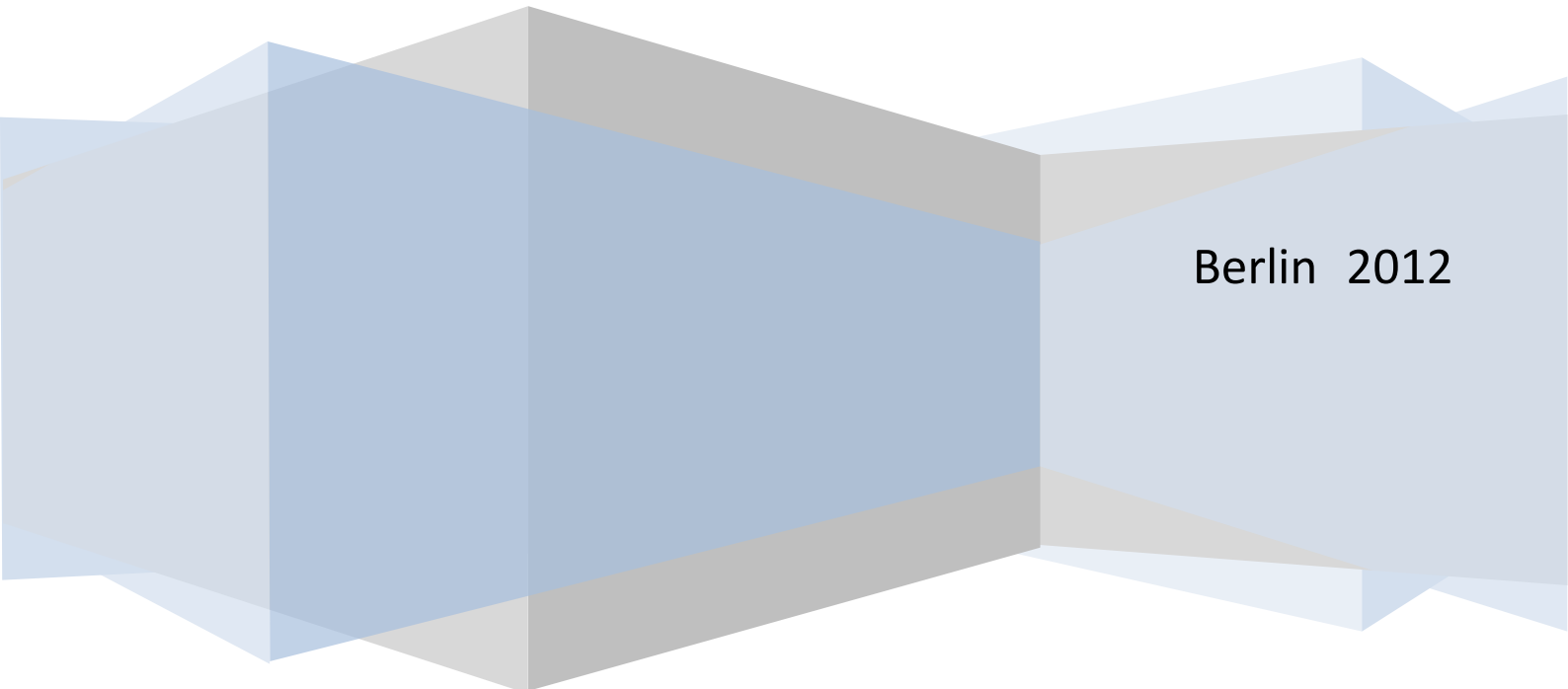


Windows Azure Community Deutschland

Windows Azure Storage Analytics

Developer Guide CTP

Oliver Michalski (Senior Software Engineer .NET)



Berlin 2012

1.0 Einführung

Im August 2011 wurde die Windows Azure Storage APIs um einen neuen Bereich erweitert: „Windows Azure Storage Analytics“. Windows Azure Storage Analytics besteht aus einer Logging API und einer Metrics API. Beide APIs stehen nicht generell zur Verfügung, sondern müssen in den zu überwachenden Bereichen eingerichtet werden. Die Auswertung erfolgt über die Windows Azure Storage Services REST API oder einer Windows Azure Storage Client Library.

1.1 Logging API

Die Logging API soll die folgenden Szenarien abdecken:

- Anfragen an die Storage-Konten überwachen
- Performance einzelner Requests ermitteln und verstehen
- Analyse der Nutzung von spezifischen Containern und Blobs
- Debug - API auf Request Ebene

Die folgenden authentifizierten und anonymen Zugriffe werden protokolliert:

Authentifizierte Requests

- 1.Erfolgreiche Requests
- 2.Fehlgeschlagene Requests
- 3.SAS (Shared Access Signature) Requests

Anonyme Requests

- 1.Erfolgreiche Requests
- 2.Fehlgeschlagene Requests

Die Logging API unterscheidet die drei Request-Typen:

- Read
- Write
- Delete

Die Logging API kann für jeden Request-Typ individuell eingestellt werden und ermöglicht dem Entwickler z.B. die gezielte Überwachung von Schreibaktionen.

1.2 Metrics API

Die Metrics API liefert folgende Informationen:

1. Speicherkapazität des Blob Services - täglich aktualisiert
2. Allgemeine Informationen – stündlich aktualisiert
 - Gesamtzahl der Zugriffe
 - Eingang
 - Ausgang
 - Durchschnittliche Latenzzeit
 - Gesamtzahl der Ausfälle

Die allgemeinen Informationen stehen jeweils auf Service- und API-Ebene zur Verfügung.

2.0 Welche Informationen liefert die Logging API?

2.1 Der Logging Datensatz

Bei jeder Logging - Aktion wird folgender aus zurzeit 30 Feldern bestehender Datensatz erfasst:

Nr.:	Name:	Typ:	Beschreibung:
01.	Log Version	String	Hier ist als Wert momentan nur „1.0“ möglich
02.	Transaction Start Time	Timestamp (UTC-Time)	Dieser Zeitstempel erfasst den Zeitpunkt an dem die Anfrage den Service erreicht.
03.	REST Operation Type	String	In diesem Wert steht die betroffene API (Vgl. Annex 1 für gültige Werte)
04.	Request Status	String	(Vgl. Annex 2 für gültige Werte)
05.	HTTP Status Code	String	
06.	E2E Latency	Time	Dieser Wert gibt die Gesamtzeit in Millisekunden an, den die Anfrage benötigt um vollständig verarbeitet zu werden.

07.	Server Latency	Time	Dieser Wert gibt die Gesamtzeit in Millisekunden an, den die Anfrage benötigt um innerhalb des Azure Storage Services verarbeitet zu werden
08.	Authentication Type	String	Authentifiziert, SAS (Shared Access Signature) oder Anonym
09.	Requestor Account Name	String	Bezeichnet das Konto des Anfragenden. Für anonyme oder SAS Anfragen bleibt dieses Feld leer.
10.	Owner Account Name	String	Bezeichnet das Konto des Objektbesitzers
11.	Service Type	String	Welche Art von Storage (Table, Blob oder Queue) wird angesprochen?
12.	Request URL	String	Die vollständige Anfrage URL
13.	Object Key	String	Dies ist der Schlüssel des Objektes
14.	Request ID	GUID	Dies ist die Anfrage ID (wird durch den Dienst vergeben)
15.	Operation Number	Int	Dies ist eine eindeutige Nummer für jede Aktion innerhalb einer Anforderung. Dieser Wert kann mehrfach vergeben werden (z.B. der Operation Type CopyBlob

			besteht aus den Aktionen Copy (= 0), Lesen des Blobs (= 1) und Schreiben des Blobs am Ziel (= 2))
16.	Client IP	String	Die Client IP der Anforderung
17.	Request Version	String	
18.	Request Header Size	Long	Die Größe des Anfrage Headers
19.	Request Package Size	Long	Die Größe der Anfrage (Server Nutzlast)
20.	Response Header Size	Long	Die Größe des Antwort Headers
21.	Response Package Size	Long	Die Größe der Antwort (Payload)
22.	Request Content Length	Long	Länge des Content-Length- Header Der Wert sollte außer im Fehlerfall der Request Package Size entsprechen.
23.	Request MD5	String	MD5 Wert des in der Anforderung übertragenden Content
24.	Server MD5	String	MD5 Wert auf dem Server ausgewertet.
25.	ETag	String	Wert des ETag (Entity Tag) wenn vorhanden. Sollten mehrere Rückgabe-

			werte möglich sein, wird dieses Feld nicht gefüllt.
26.	Last Modified Time	DateTime	Datum der letzten Änderung. Sollten mehrere Rückgabewerte möglich sein, wird dieses Feld nicht gefüllt.
27.	ConditionsUsed	String	Filterbedingungen (Mögliche Werte: If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match)
28.	User Agent	String	
29.	Referrer	String	
30.	Client Request ID	String	

Technischer Hinweis: Im Falle eines fehlgeschlagenen Request oder einer fehlgeschlagenen Response sind die Felder 18 bis 21 leer.

3.0 Welche Informationen liefert die Metrics API?

Metric Daten werden in den zwei Kategorien Capacity Data und Transaction Data erfasst und in den folgenden Tabellen des Table Storage Services niedergeschrieben:

- · \$MetricsCapacityBlob
- · \$MetricsTransactionsBlob
- · \$MetricsTransactionsTable
- · \$MetricsTransactionsQueue

Anmerkung: In Version 1.0 der Storage Analytics API gibt es die Kategorie Capacity Data nur für die Storage Blob Services. Eine Datenerfassung für die Storage Table Services und die Storage Queue Services erfolgt in einer späteren Version.

3.1 Capacity Data

Beginnen möchte ich mit den Capacity Data (Kapazitätsdaten). Diese werden einmal täglich erfasst und umfassen zwei Entitäten (= Datensätze im Table Storage) und zwar einen für die Kapazität des Storage Accounts und einen für die Kapazität der \$logs Daten (= Daten der Storage Analytics API).

Die Entität ist jeweils wie folgt definiert:

Nr.:	Name:	Typ:	Beschreibung:
01.	PartitionKey	String	Zeitstempel (UTC-Time) für den Beginn der Metricdatenerfassung
02.	RowKey	String	Gibt an ob die Metric für Nutzdaten (Wert: Data) oder Analysedaten (Wert: Analytics) gilt
03.	Capacity	Long	Benutzter Speicherplatz des Storage Accounts in Bytes

04.	ContainerCount	Long	Anzahl der Container im Storage Account
05.	ObjectCount	Long	Anzahl der Block - und Page - Blobs im Storage Account

3.2 Transaction Data

Die Metriken für Transaction Daten werden einmal pro Stunde erfasst und zwar einmal auf Service Level (Blob, Table, Queues) und je einmal für jede spezifizierte API.

Die Entität ist jeweils wie folgt definiert:

Nr.:	Name:	Typ:	Beschreibung:
01.	Time als PartitionKey	String	Zeitstempel (UTC-Time) für den Beginn der Metricdatenerfassung
02.	AccessType; TransactionType als RowKey	String	<p>Der RowKey ist eine Kombination aus jeweils einen Wert für AccessType (Gültige Werte: „user“ für Nutzdaten - oder „system“ für Analysedaten - Requests) und TransactionType (einer API entsprechend Annex 1) getrennt durch ein Semikolon.</p> <p>Sollte für TransactionType der Wert Unknown stehen, ist der Request gescheitert bevor eine API erkannt wurde.</p>
03.	TotalIngress	Long	Der gesamte Input in Bytes der durch den Dienst (bzw. die API) genutzt wurde.
04.	TotalEgress	Long	<p>Der gesamte Output in Bytes der durch den Dienst (bzw. die API) genutzt wurde.</p> <p>Dies ist nicht der abrechenbare Output. Kunden zahlen z.B. nicht für Datentransfer innerhalb des gleichen Rechenzentrums</p>

05.	TotalRequest	Long	<p>Gesamtzahl aller Requests an den Dienst (bzw. die API).</p> <p>Alle fehlgeschlagenen Requests werden mitgezählt.</p>
06.	TotalBillableRequest	Long	<p>Gesamtzahl aller abrechenbarer Request.</p> <p>Nicht mitgerechnet werden anonyme Ausfälle (außer Network Errors), Drosslungen (Throttling Errors), Server Timeouts und jeder unbekannte Fehler.</p>
07.	Availability	Double	<p>Dieser Wert spiegelt die durchschnittliche Verfügbarkeit des Dienstes (bzw. der API) pro Stunde wieder</p> <p>Achtung!: Dieser Wert ist nicht identisch mit dem Wert Verfügbarkeit aus dem Windows Azure SLA</p> <p>Berechnung:</p> $\text{TotalBillableRequest} / (\text{TotalRequest} - (\sum \text{Throttling Errors})) * 100$
08.	AverageE2ELatency	Double	<p>Durchschnittliche Bearbeitungszeit eines Request</p> <ul style="list-style-type: none"> • Lesens des Requests • Beantworten • Warten auf Bestätigung der Antwort
09.	AverageServerLatency	Double	<p>Durchschnittliche Bearbeitungszeit eines Request (Serverseitig) ohne fehlgeschlagene Anfragen.</p>

10.	PercentSucess	Double	Anzahl erfolgreicher Requests in Prozent
11.	PercentThrottlingError	Double	Anzahl fehlgeschlagener Requests (Throttling Errors) in Prozent
12.	PercentTimeoutError	Double	Anzahl fehlgeschlagener Requests (Timeout Errors) in Prozent
13.	PercentServerOtherError	Double	Anzahl fehlgeschlagener Requests (Server Other Errors - HTTP Code 500) in Prozent
14.	PercentClientOtherError	Double	Anzahl fehlgeschlagener Requests (Client Other Errors - HTTP Code 3xx und 4xx) in Prozent
15.	PercentAuthorizationError	Double	Anzahl fehlgeschlagener Requests (Authorization Errors) in Prozent
16.	PercentNetworkError	Double	Anzahl fehlgeschlagener Requests (Network Errors) in Prozent
17.	Success	Long	Gesamtzahl erfolgreicher Requests
18.	AnonymousSuccesses	Long	Gesamtzahl erfolgreicher Requests (anonymer Zugriff)
19.	SASSuccess	Long	Gesamtzahl erfolgreicher Requests (SAS Zugriff)
20.	ThrottlingError	Long	Gesamtzahl Throttling Errors
21.	AnonymousThrottlingError	Long	Gesamtzahl Throttling Errors (anonymer Zugriff)
22.	SASThrottlingError	Long	Gesamtzahl Throttling Errors (SAS Zugriff)
23.	ClientTimeoutError	Long	Gesamtzahl Client Timeout Errors
24.	AnonymousClientTimeoutError	Long	Gesamtzahl Client Timeout Errors (anonymer

			Zugriff)
25.	SASClientTimeoutError	Long	Gesamtzahl Client Timeout Errors (SAS Zugriff)
26.	ServerTimeoutError	Long	Gesamtzahl Server Timeout Errors
27.	AnonymousServerTimeoutError	Long	Gesamtzahl Server Timeout Errors (anonymer Zugriff)
28.	SASServerTimeoutError	Long	Gesamtzahl Server Timeout Errors (SAS Zugriff)
29.	ClientOtherError	Long	Gesamtzahl Client Other Errors
30.	AnonymousClientOtherError	Long	Gesamtzahl Client Other Errors (anonymer Zugriff)
31.	SASClientOtherError	Long	Gesamtzahl Client Other Errors (SAS Zugriff)
32.	ServerOtherError	Long	Gesamtzahl Server Other Errors
33.	AnonymousServerOtherError	Long	Gesamtzahl Server Other Errors (anonymer Zugriff)
34.	SASServerOtherError	Long	Gesamtzahl Server Other Errors (SAS Zugriff)
35.	AuthorizationError	Long	Gesamtzahl Authorization Errors
36.	AnonymousAuthorizationError	Long	Gesamtzahl Authorization Errors (anonymer Zugriff - Authentifizierung erforderlich)
37.	SASAuthorizationError	Long	Gesamtzahl Authorization Errors (SAS Zugriff - Höhere Rechte erforderlich)

38.	NetworkError	Long	Gesamtzahl Network Errors
39.	AnonymousNetworkError	Long	Gesamtzahl Network Errors (anonymer Zugriff)
40.	SASNetworkError	Long	Gesamtzahl Network Errors (SAS Zugriff)

Annex 1: Windows Azure Storage APIs

Die in der folgenden Liste verzeichneten Windows Azure Storage APIs, werden sowohl von der Logging API (als sog. REST Operation Type) als auch von der Metrics API (als Transaction Type im RowKey der Transaction Data) als gültige Werte benutzt:

Blob Service API

- AcquireLease
- BreakLease
- ClearPage
- CopyBlob
- CopyBlobSource
- CopyBlobDestination
- CreateContainer
- DeleteBlob
- DeleteContainer
- GetBlob
- GetBlobMetadata
- GetBlobProperties
- GetBlockList
- GetContainerACL
- GetContainerMetadata
- GetContainerProperties
- GetLeaseInfo
- GetPageRegions
- LeaseBlob
- ListBlobs
- ListContainers
- PutBlob
- PutBlockList
- PutBlock
- PutPage
- ReleaseLease
- RenewLease
- SetBlobMetadata
- SetBlobProperties
- SetContainerACL
- SetContainerMetadata
- SnapshotBlob
- SetBlobServiceProperties

- GetBlobServiceProperties

Queue Service API

- ClearMessages
- CreateQueue
- DeleteQueue
- DeleteMessage
- GetQueueMetadata
- GetQueue
- GetMessage
- GetMessages
- ListQueues
- PeekMessage
- PeekMessages
- PutMessage
- SetQueueMetadata
- SetQueueServiceProperties
- GetQueueServiceProperties

Table Service API

- EntityGroupTransaction
- CreateTable
- DeleteTable
- DeleteEntity
- InsertEntity
- QueryEntity
- QueryEntities
- QueryTable
- QueryTables
- UpdateEntity
- MergeEntity
- SetTableServiceProperties
- GetTableServiceProperties

Annex 2: Request (Transaction) Status

Folgende Zustandsbeschreibungen (Transaction Status Values) sind möglich:

Nr.:	Name:	Beschreibung:
01.	Success	Erfolgreiche Abfrage
02.	AnonymousSuccess	Erfolgreiche Abfrage vom Typ Anonym
03.	SASSuccess	Erfolgreiche Abfrage vom Typ SAS
04.	ThrottlingError	Abfrage wird vom Server gedrosselt mit HTTP- Code 503 ServerBusy
05.	AnonymousThrottlingError	ThrottlingError vom Typ Anonym
06.	SASThrottlingError	ThrottlingError vom Typ SAS
07.	ClientTimeoutError	Abfrage endet mit HTTP-Code 500 Timeout. Netzwerk IO auf dem Client ist zu langsam.
08.	AnonymousClientTimeoutError	ClientTimeoutError vom Typ Anonym)
09.	SASClientTimeoutError	ClientTimeoutError vom Typ SAS
10.	ServerTimeoutError	Abfrage endet mit HTTP-Code 500 Timeout
11.	AnonymousServerTimeoutError	ServerTimeoutError vom Typ

		Anonym
12.	SASServerTimeoutError	ServerTimeoutError vom Typ SAS
13.	ClientOtherError	Abfrage endet mit unbekannten Fehler
14.	AnonymousClientOtherError	ClientOtherError vom Typ Anonym
15.	SASClientOtherError	ClientOtherError vom Typ SAS
16.	ServerOtherError	Abfrage endet mit HTTP-Code 500 UnknownError
17.	AnonymousServerOtherError	ServerOtherError vom Typ Anonym
18.	SASServerOtherError	ServerOtherError vom Typ SAS
19.	AuthorizationError	Eine authentifizierte Abfrage schlägt fehl, da für den abgefragten Bereich höhere Rechte erforderlich sind z.B. Schreibzugriffe auf Logs unter \$logs
20.	SASAuthorizationError	Eine SAS Abfrage schlägt fehl, da für den Abfragetyp höhere Rechte erforderlich sind
21.	NetworkError	Abfrage endet, da die Verbindung vorzeitig unterbrochen wurde
22.	AnonymousNetworkError	NetworkError vom Typ

		Anonym
23.	SASNetworkError	NetworkError vom Typ SAS