



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

LV-Titel: Wissenschaftliches Projek

LV-Leiter: Prof. Dr. Stefan Edlich

Semester: Wintersemester 2012/13

Studiengang: Online Medieninformatik Master

Projektdokumentation:

Google Fusion Tables:

Daten-Cockpit für Berliner Weihnachtsmärkte 2012

Projektteam/Verfasser:

Patrick Böhm (783318), E-Mail: patty.boehm@gmail.com

Manuel Wisniewski (786598), E-Mail: manuel.wisniewski@gmail.com

letzte Änderung: 25.01.2013

DOKUMENT-HISTORIE

Version	Datum	Änderungen	Bearbeiter
0.1	13.11.2012	Projektbeschreibung in GoogleDocs angelegt	Patrick Böhm, Manuel Wisniewski
1.0	02.12.2012	Dokumentation begonnen	Patrick Böhm, Manuel Wisniewski
1.1	15.12.2012	Inhalte Strukturiert	Patrick Böhm
1.2	16.12.2012	Formatierung angepasst	Manuel Wisniewski
1.2	18.12.2012	Projektbeschreibung und Informationen zu Kapitel 2 ergänzt	Patrick Böhm
1.3	6.1.2013	Kapitel 4.2 <i>Umsetzung</i> hinzugefügt	Manuel Wisniewski
1.3	11.01.2013	Google Fusion Table Web Apps und API-Beschreibung, Erfahrung dokumentiert.	Patrick Böhm
1.x	24.01.2013	Ergänzungen Kapitel 2 „Google Fusion Tables“	Manuel Wisniewski
1.x	25.01.2013	Anpassung Struktur	Patrick Böhm, Manuel Wisniewski

ZEITPLANUNG

Meilenstein	Plan	Ist
Einarbeitung	01.11.2012 - 13.11.2012	
Spezifikation / Lastenheft	14.11.2012 - 30.11.2012	
Vorbereitung Datenbeschaffung	01.12.2012 - 13.12.2012	
Umsetzung	14.12.2012 - 31.12.2012	
Dokumentation der verwendeten Technologien	01.01.2013 - 14.01.2013	
Abschlusspräsentation	28.01.2013	

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

INHALTSVERZEICHNIS

1	Einleitung.....	5
2	Google Fusion Tables.....	6
2.1	Überblick.....	6
2.2	Systemarchitektur.....	6
2.2.1	<i>Speicher-Stack.....</i>	<i>7</i>
2.2.2	<i>Anwendungsdatenspeicher.....</i>	<i>7</i>
2.2.3	<i>Datenabfrage.....</i>	<i>8</i>
2.2.4	<i>Transaktionsmanagement.....</i>	<i>8</i>
2.3	Datenvisualisierung.....	9
2.3.1	<i>Client-seitige Visualisierung von Diagrammen.....</i>	<i>9</i>
2.3.2	<i>Server-seitige Visualisierung von geografischen Features.....</i>	<i>9</i>
2.4	Entwicklungsumgebung und API.....	10
2.4.1	<i>Google Fusion Tables Web App.....</i>	<i>11</i>
2.4.2	<i>Google Fusion Tables API 1.0.....</i>	<i>12</i>
2.4.3	<i>Funktionsweise der API.....</i>	<i>13</i>
3	Anwendungsspezifikation.....	17
3.1	Motivation.....	17
3.2	Suchfunktionalität.....	17
3.3	Infrastruktur.....	19
3.4	Ziele.....	19
4	Technische Umsetzung.....	20
4.1	Schematischer Entwurf.....	20
4.2	Datenerhebung.....	21
4.3	Datenimport und -aufbereitung.....	23
4.3.1	<i>Probleme beim Importieren der Daten in Tabellenkalkulationsprogramm:.....</i>	<i>23</i>
4.4	Konfigurierung der Visualisierungen.....	24
4.5	HTML-Prototyp.....	24
4.6	Interaktive Anzeigesteuerung.....	24
4.7	Einschränkungen/Ausblick.....	24
5	Zusammenfassung.....	25

1 EINLEITUNG

Fusion Tables ist Googles umfangreiches Toolkit zur Speicherung und Visualisierung (alpha)numerischer Daten. Durch die Verknüpfung mit anderen Google Services – allen voran Google Maps – ergeben sich interessante Anwendungsmöglichkeiten, insbesondere in der geospatialen Projektion statistischer Daten. Das Ergebnis sind visuell aufbereitete, interaktive Datenbilder bzw. -karten, die neue und aufschlußreiche Einblicke in die Datenbasis ermöglichen.

Das vorliegende Fallbeispiel dokumentiert die Konzeption und Entwicklung einer Fusion Tables-Anwendung in Form einer Informationsplattform für Berliner Weihnachtsmärkte. Dabei ermöglicht die Anwendung das Anzeigen von Weihnachtsmärkten auf einer Karte und erlaubt die Einschränkung bzw. Filterung der Zielmenge anhand mehrerer Kriterien, z.B. nach Bezirk, Eintrittspreis, etc. Die Umsetzung des Projektes begann mit der theoretischen Einarbeitung in die Google Fusion Tables-Infrastruktur (Kapitel 2) sowie der Spezifikation der Datencockpit-Anwendung (Kapitel 3).

Im Rahmen der Umsetzung mussten zunächst die notwendigen Daten und insbesondere die Geokoordinaten beschafft und in Fusion Tables importiert werden. Anschließend wurden die Datentabellen mit Google Maps kombiniert und in einen HTML-Prototypen integriert (Kapitel 4). Betrachtungen zu den Stärken und Schwächen von Fusion Tables sowie reflektierende Überlegungen zur Arbeit mit der Architektur schließen das Projekt ab (Kapitel 5).

Zu den vollständigen Projektergebnissen gehört neben der Dokumentation der Quelltext des angesprochenen HTML-Prototypen sowie die verwendeten Fusion-Tables-Tabellen. Die Verweise und Links zu den eingesetzten Ressourcen sind im Anhang vollständig aufgelistet.

2 GOOGLE FUSION TABLES

2.1 Überblick

Google Fusion Tables ist ein 2009 gestarteter cloud-basierter Webdienst zur Verwaltung, Integration und insbesondere zur Visualisierung tabellenbasierter Daten. Der Service unterstützt Anwender wie Unternehmen bei der internen wie externen Bereitstellung von Daten und zielt darauf ab, die Zusammenarbeit von Usern aus verschiedenen Organisationseinheiten zu vereinfachen. Für die grafische Aufbereitung der Werte ist dabei neben herkömmlichen Diagrammen auch die geospatiale Projektion der Daten in das Kartenmaterial der Geodatendienste Google Maps und Google Earth möglich.

Google Fusion Tables erlaubt den Upload verschiedener Datenformate (csv, kml, etc.) über den Service Google Drive (bis zu 250 MB). Für eine möglichst benutzerfreundliche Auswertung hält das Framework Aggregationsfunktionalitäten bereit und schlägt auf Basis der hinterlegten Daten automatisch passende Visualisierungsformen vor. User können ihre Daten für die Zusammenarbeit mit anderen Anwendern (und für Suchmaschinen) freigeben und dank einer Diskussionsfunktion sogar bis auf Zeilen-, Spalten- und Zellenebene detailliert kommentieren. Neben der Integration verschiedener Datenquellen können so zu Auswertungszwecken auch Datentabellen verschiedener Nutzer verbunden werden.

Die Entwicklung von Fusion Tables folgt drei zentralen Leitlinien:

1. Vereinfachung der Datenverwaltung für Anwender ohne Erfahrung im Umgang mit Datenbanksystemen,
2. Erhöhung der Attraktivität von Data-Sharing und Informationsintegration durch aussagekräftige Visualisierungen sowie
3. Unterstützung von cloud-basierter Kollaboration, die über reine Datenabfrage hinausgeht, z.B. durch Diskussions- und Abstimmungsfeatures.

2.2 Systemarchitektur

Die Systemarchitektur von Google Fusion Tables wird in drei Schichten eingeteilt: Frontend Dispatcher, Query Processing Module und Backend. Anfragen (Requests) können grundsätzlich von verschiedenen Clients gestellt werden (Google Fusion Tables-Website, Stand-Alone-Anwendungen, eingebettete Visualisierungen, etc.). Der Frontend Dispatcher übersetzt die Anfragen und leitet die standardisierten Queries an das Query Processing Module weiter. Das Processing Module erstellt einen Query Plan und übergibt ihn ans Backend. Im Backend werden die Queries abgearbeitet und die Ergebnisse an das Query Processing Module und schlussendlich an das Frontend zurückge-

liefert. Als Persistenzschicht dienen dabei synchron replizierte Bigtable-Server (siehe 2.2.1). Die größte Herausforderung der Datenbankschicht besteht in der Bearbeitung hunderttausender Tabellen mit unterschiedlichen Schemata, Größen und Anfragelast-Statistiken.

2.2.1 Speicher-Stack

Die Fusion Tables-Architektur operiert auf zwei Ebenen des Google Speicher Stacks: Bigtable und Megastore:

1. *Bigtable* ist Googles proprietäres Hochleistungs-DBMS und speichert in erster Linie Tabellen als komplexe und nicht zwingend normalisierte (key, value)-Tupel. Diese werden nach Schlüsselwerten sortiert über mehrere Server verteilt. Jede Tabelle ist mehrdimensional, wobei jeder Eintrag mithilfe eines Timestamps versioniert wird.
2. *Megastore* ist ein auf Bigtable aufbauendes Speichersystem, das die Vorteile von NoSQL-Datenbanken (in erster Linie die Skalierbarkeit) mit den Vorzügen traditioneller relationaler DBMS verbindet. Megastore bietet u.a. konsistente Sekundärindizes, mehrzeilige Transaktionen und konsistente Replikation.

Eine ausführliche Beschreibung der Bigtable und Megastore-Architektur liefern XXX.

2.2.2 Anwendungsdatenspeicher

Der Anwendungsdatenspeicher der Fusion Tables-Infrastruktur unterteilt sich in vier Bereiche: Zeilenspeicher (*Row Store*), Schema-Speicher (*Schema Store*), Sichten-Speicher (*View Store*) und Kommentar-Speicher (*Comment Store*).

1. Zeilenspeicher:

Die Zeilen aller Datentabellen der Anwender werden gemeinsam in einer einzigen Bigtable *Rows* gespeichert. Dabei repräsentiert jede Zeile genau eine Zeile in einer Nutzertabelle. Der Primärschlüssel einer Zeile wird intern generiert und aus der individuellen Tabellen-ID und der jeweiligen Zeile zusammengesetzt. Dank der guten Skalierbarkeit der Bigtable-Architektur bleibt der Zugriff auf die Rows-Tabelle auch bei Millionen von Anwendertabellen performant, da die Tabelle in einzelne Untertabellen aufgeteilt wird, die wiederum auf verschiedenen Servern lagern.

Eine Tabellenzeile besteht aus dem Zeilen-Schlüssel sowie je einem Satz indizierter und nicht-indizierter Properties. Ein Property setzt sich aus Property-Name und Property-Wert zusammen. Da folglich jeder Property-Name redundant in der Rows-Tabelle hinterlegt wird, können in den einzelnen Zeilen unterschiedliche Property-Sets (d.h. unterschiedliche Tabellenschemata) gespeichert werden. Standardmäßig werden Property-Werte als Strings gespeichert. Ein Annotationsservice identifiziert automatisch Datentypen wie Zahlen, Datum/Zeit oder Geodaten. Je nach Datentyp schlägt das Fusion Tables-Framework in

weiterer Folge passende Visualisierungsformen vor. Zum effizienteren Zugriff werden ausgewählte Properties indiziert.

2. Schema-Speicher:

Tabellenschemata von Benutzertabellen werden ebenfalls gemeinsam in einer zentralen Bigtable gespeichert. Dabei dient die Tabellen-ID als Primärschlüssel. Für jede Tabelle werden Informationen zu Spalten und Berechtigungen abgelegt, dazu gehören Spaltenname, Datentyp und Access Control Lists (ACLs). Dabei werden für jede Tabelle separate Listen für Nutzer mit Lese- (Viewer) und/oder Schreibberechtigung (Collaborator) geführt. Öffentliche Tabellen haben einen speziellen Vermerk, dass sie für jeden sichtbar sind. Tabellenschemata können verändert, Spalten können hinzugefügt oder entfernt und automatisch identifizierte Datentypen können nach Belieben angepasst werden.

3. View-Speicher:

Individuelle Tabellen (base tables) verschiedener Benutzer können in Fusion Tables zusammengeführt werden. Zusammengesetzte Tabellen werden Views genannt und nicht materiell repliziert. Lediglich ihre Zusammensetzung und Zugriffsrechte werden gespeichert, wobei die gleichen Anwenderrollen gelten wie bei Tabellen, erweitert nur um die Contributor-Rolle für Nutzer, die die View-Definition editieren können.

4. Kommentar-Speicher:

Nutzerkommentare zu Datensätzen werden ebenfalls in einer einzigen Bigtable zusammengefasst. Als Schlüssel dient hierbei das 3er-Tupel (Tabelle, Zeile, Spalte). Gespeichert werden der Kommentartext sowie Autor und Datum des Kommentars.

2.2.3 Datenabfrage

Der Tabellenzugriff in Fusion Tables via SQL erfolgt über klassische Abfragen (SELECT), Aggregat-Funktionen (GROUP BY) und Verknüpfungen (JOIN). Aus Performancegründen werden die SQL-Statements auf Standardoperationen von Bigtable (Key Lookup, Prefix Scan und Range Scan) gemappt. Dabei kommt ein Property-Index zum Einsatz, der in einer eigenen Tabelle alle Schlüssel zusammenfasst. Eine Überblicksdarstellung der gängigen Zugriffsmodi liefert **XXX**.

2.2.4 Transaktionsmanagement

Da Fusion Tables in erster Linie als Kollaborations- und Visualisierungsplattform eingesetzt wird und nicht auf großen Datendurchsatz ausgelegt ist, kommt dem Transaktionsmanagement eine niederrangige Rolle zu. Im Rahmen von Megastore erfolgt Transaktionskontrolle daher lediglich auf Tabellenebene mittels Write Ahead Logging (WAL) und optimistischem Locking (Optimistic Concurrency Control) zur Gewährleistung von Atomarität und Dauerhaftigkeit. Für eine detaillierter Darstellung der Transaktionskontrolle sei auf **XXX** verwiesen.

2.3 Datenvisualisierung

Eine zentrale Funktionalität von Fusion Tables ist die Datenvisualisierung. Verfügbare Visualisierungsmethoden werden auf Grundlage der verwendeten Datentypen automatisiert berechnet. So sind beispielsweise Streudiagramme nur verfügbar, wenn mindestens zwei Spalten mit numerischen Werten vorhanden sind. Analog dazu ist die Kartenvisualisierung nur auf Basis von Geo-Daten möglich. Bei der Visualisierung werden zwei Konzepte unterschieden: Client-seitige Visualisierung für die Darstellung von Diagrammen und Server-seitige Visualisierung für das Rendern von geografischen Darstellungen in Google Maps.

2.3.1 Client-seitige Visualisierung von Diagrammen

Die Client-seitige Visualisierung wird mittels Google Visualization API realisiert (siehe Kapitel X). Das Rendern der Visualisierung erfolgt mittels Javascript oder Flash direkt im Browser. Über das Fusion Tables-Framework können dabei Datentabellen abgefragt und Ergebnismengen direkt an die Visualisierungsinfrastruktur übergeben werden. Zudem können Anwender ihre Visualisierungen individuell konfigurieren, wobei das Framework anhand der zur Verfügung stehenden Daten automatisch Vorschläge generiert. Erstellte Datendiagramme können durch Einsetzen weniger Zeilen JavaScript-Code in beliebige Webseiten eingebunden werden. Die Visualisierungen sind dabei direkt mit der Datenquelle verbunden und werden bei Änderungen im Datensatz automatisch aktualisiert.

2.3.2 Server-seitige Visualisierung von geografischen Features

Für geografische Visualisierungen über den Service Google Maps wird hingegen die Server-seitige Berechnung der Darstellung angewandt. Dabei werden geokodierbare Daten (Adressen, Punkte, Linien, Polygone) auf sogenannte Map Layers gerendert. Die Repräsentation von Daten in Google Maps erfolgt stets in Form von Layern, die sich aus einzelnen „Bildkacheln“ (Tiles) zusammensetzen. Die Entscheidung für eine Server-seitige Berechnung liegt im erheblichen Datenaufwand begründet, der ohne Kenntnis über die aktuellen Anzeigeeinstellungen zur Visualisierung an den Client übertragen werden müsste, sowie in der begrenzten Leistungsfähigkeit der Browser. Wenn nun der Anwender die Anzeige einer Map (Karte) anfragt, wird ein entsprechender Request zum Backend-Server mit Informationen über die aktuell sichtbaren Layer, die Geokoordinaten und den gegenwärtigen Zoom Level gesendet. Das Backend setzt nun aus allen Layer-Informationen einzelne Bild-Tiles zusammen und sendet diese zurück an den Browser.

In Fusion Tables werden Geodaten in einen Raum-Index (Spatial Index) eingetragen. Für die Arbeit mit Fusion Tables bedeute dies, dass zwischen strukturierten (structured) und räumlichen (spatial) Datenabfragen unterschieden wird. Bei Abfragen wie z.B. „alle Weihnachtsmärkte in Berlin (spatial query) ohne Eintritt (structured query)“ werden beide Anfragen parallel verarbeitet und die Ergebnismengen verschränkt. Structured queries werden klassisch mittels SQL beantwortet während

spatial queries mithilfe des spatial index erfolgen. Für Details zum räumlichen Mapping von Informationen auf einzelne Tiles und zum genauen Verfahren der spatial queries sei an dieser Stelle auf **xxx** verwiesen.

Um schnelle Antwortzeiten der Kartenvisualisierung zu ermöglichen, ist die Anzahl der darstellbaren Features auf einer Tile begrenzt. Im Streben um bessere Übersichtlichkeit trifft das Visualisierungs-frame-work eigenständig eine Auswahl darzustellender Details in Abhängigkeit vom aktuellen Zoom Level. Fusion Tables erlaubt auch das unkomplizierte Rendern von Heat Maps, wobei unterschiedliche Kartenbereiche in Abhängigkeit von statistischen Daten unterschiedlich stark eingefärbt werden. Genau wie bei der Client-Visualisierung können gerenderte Karten auch auf individuellen Webseiten veröffentlicht werden. Durch Einsetzen weniger Zeilen JavaScript können Karten angezeigt werden, die live mit Tabellendaten verlinkt sind.

2.4 Entwicklungsumgebung und API

Google bietet für das Arbeiten mit den Google Fusion Tables zum einen die sogenannte *Web App* als browsergesteuerte Anwendung und zum Anderen den Zugriff über APIs. **Fusions Tables sollten jedoch nicht als „kostenfreier Unendlichspeicher“ angesehen werden, da sie auch Beschränkungen und Obergrenzen haben, die jedoch für die meisten Webanwendungen keine Relevanz haben sollten¹:**

- **Import-Dateien dürfen maximal 100 MB sein**
- **Requests über die Google API dürfen maximal 1 MB groß sein**
- **es dürfen nur 5 Requests je Sekunden über die Google Fusion API abgesetzt werden**
- **das „mergen“ (Vermischen) von Tabellen erfolgt ausschließlich über die „left outer join“, so dass die erste Tabelle links angeordnet wird und die zweite Tabelle nur bei passenden Treffern rechts angehängen wird**

In den folgenden Kapiteln werden die Web App und Google Fusion Tables API genauer beschrieben.

¹ URL: <http://webkompetenz.wikidot.com/blog:83> [11.01.2013]



Abbildung 1: Nutzungswege von Google Fusion Tables

2.4.1 Google Fusion Tables Web App

Innerhalb der Google Web App, welche in Google Drive integriert ist können Tabellen direkt im Browser erstellt, bearbeitet und befüllt werden. Google Drive ist eine von Google angebotene Webanwendung, welche ein verteiltes Dateisystem für die Synchronisation von Dateien zwischen verschiedenen Benutzern bereitstellt. Neben der Google Fusion Table gibt es weitere Dokumententypen wie Tabellenkalkulation, Textverarbeitung, Formular, Bildschirmpräsentation oder Zeichnungen.

Es gibt die Möglichkeiten eine Excel-Tabelle oder sogar Geokoordination im kml-Format zu importieren, ein Google Spreadsheet zu verwenden oder eine leere Google Fusion Table zu erzeugen.

Dem ersten Anschein nach wirkt eine leere Google Fusion Table wie ein „abgespecktes“ Tabellenkalkulationsdokument. Relativ schnell fällt jedoch auf, dass die Tabellenköpfe selbst keine Überschriften darstellen, sondern eigene Datentypen, für die Werte in der jeweiligen Spalte repräsentieren. Neben dem Namen gibt es somit die Möglichkeit aus vordefinierten Datentypen zu wählen. Zur Auswahl stehen dabei Date / Time, Text, Number und Location. Die Auszeichnung als ein spezieller Typ hilft Google dabei, die Daten entsprechend weiter zu verarbeiten bzw. anderen zur Verfügung zu stellen.

In dem Typ „Location“ liegt auch einer der Stärken von Google Fusion. Besitzt eine Spalte den Typ „Location“ ist es notwendig, dass die in den Zeilen hinterlegten Werte eine Basis für die Generierung von Geokoordinaten bieten. Das können neben den Geokoordination Longitude und Latitude auch vollständige Adressdaten wie eine Anschrift inklusive Postleitzahl, Ort und Land sein. Außerdem ist es möglich vollständige Polygon-Netze bzw. Umrisse im kml-Format zu hinterlegen (zum Beispiel Bezirksgrenzen). In dem Reiter „Map“, welche per Default bei der jeder Fusion Table vor-

handen ist, interpretiert Google anschließend die als „Location“ hinterlegten Daten und stellt sie direkt auf der Karte dar. Des Weiteren können ausgewählte Tabellendaten wie in Bekannten Tabellenkalkulationsprogrammen visualisiert (Google Maps oder Google Chart) oder in Funktionen wie „min“, „max“ und „average“ auf „Knopfdruck“ repräsentiert werden.

Mit Hilfe von Google Chart lassen sich verschiedene Arten von interaktiven Diagrammen auf Basis einer Tabelle, z.B. Google Fusion Table visualisieren. Neben den genannten Funktionen bietet Google Fusion Tabelle natürlich auch die Möglichkeit, die Tabelle selbst zu teilen oder zu publizieren.

2.4.2 Google Fusion Tables API 1.0

Um eine breitere Masse an Anwendungen und demzufolge Anwendern zu erreichen, die mit Google Fusion Tables arbeiten, bietet google eine eigenständige Google Fusion Tables API in der Version 1.0. Die damals noch eigenständige SQL API wird seit Juni 2012 als „deprecated“ bezeichnet und ist seit dem fester Bestandteil der Google Fusion Tables API 1.0.

Die Google Fusion Tables API ist letztlich ein umfangreicher Webservice von Google auf Basis des Representational State Transfer Verfahrens, kurz RESTful. RESTful oder auch nur REST – ist ein beliebtes Verfahren zur Realisierung von Webservices. Historisch ist REST auf Roy Fielding zurückzuführen, einen der Hauptautoren der ursprünglichen HTTP-Spezifikation, der in seiner Dissertation (2000) REST als eine Schlüsselarchitektur für das World Wide Web beschrieb. Als netzwerkorientiertes Konzept zielt REST im Sinne einer optimalen Skalierbarkeit auf die Minimierung der Latenz und des Kommunikationsaufwandes im Netzwerk ab, während gleichzeitig die Unabhängigkeit aller beteiligten Komponenten gewährleistet werden soll. Zentrale Funktionsmerkmale von REST sind dabei das Zwischenspeichern (Caching) und Wiederverwenden von Interaktionen sowie das auf schnelle Ersetzbarkeit fokussierte Design der Komponenten.

Im Zentrum des REST-Modellansatzes steht dabei die Atomarisierung von Anwendungen und Präsentationen in ihrer Einzelbestandteile, sogenannte Ressourcen, und die Adressierung einer jeden Ressource mithilfe eines einheitlichen Bezeichners, des Unified Resource Identifiers (URI). Dieser Identifikator ist eine konkrete Zeichenfolge, mit deren Hilfe abstrakte oder physische Ressourcen eindeutig identifiziert und angesprochen werden können. Eine weitläufig bekannte Unterart der URIs sind die URLs, Uniform Resource Locators, die zur Identifikation von Ressourcen wie etwa Webseiten über das jeweils eingesetzte Netzwerkprotokoll (HTTP, FTP, SMTP, etc.) und der eindeutigen Lokalisierung an einem Ort des Netzwerks dienen. Mittels URLs können so Dokumente in komplexen Netzwerken wie dem WWW eindeutig beschrieben und adressiert werden.

Mit Hilfe der durch die Google Fusion Tables API zur Verfügung gestellten URLs können folgende Funktionen abgedeckt werden:

- Erstellen und Löschen von Tabellen
- Lesen und Bearbeiten der Metadaten (Tabellen- und Spaltennamen sowie Spaltentypen)

- Hinzufügen, Aktualisieren und Löschen von Zeilen innerhalb der Tabelle
- Erstellen, Aktualisieren und Löschen von Konfigurationen für die Visualisierung von Daten
- gezielte Abfragen auf die Zeilen innerhalb einer Tabelle

Das Ergebnis der einzelnen Anfragen von Daten bzw. ganzen Tabellen erfolgt dann je nach Bedarf im CSV- oder im JSON-Format.

Mit Hilfe der zur Verfügung gestellten Methoden und gängigen Formate ist die API universell und programmiersprachenunabhängig einsetzbar. Methoden für das Absetzen eines HTTP-Requests so wie die Verarbeitung aus JSON sind mittlerweile Standard in allen Programmiersprachen im Bereich der Webentwicklung. Nach aktuellen Recherchen findet Google Fusion Tables vor allem Anwendung in Browsergesteuerten Anwendungen und weniger in komplexen Software-Anwendungen.

Die Auslagerung von tabellarischen Daten, also verschiedenen Anwendungsdaten in das Internet zu Google, bringt auch Nachteile mit sich. Die Daten sind nur Online verfügbar und können nicht entkoppelt, lokal in eine Anwendung integriert werden. Ein weiterer Nachteil ist das „Teilen“ mit Google. Es gibt verschiedene Befürworter als auch Widersacher was das Thema Datenschutz in Verbindung mit Google anbelangt.

Ersteres erfolgt natürlich bewusst und zählt zur Strategie von Google, wie auch das folgende Zitat zeigt²:

„The overarching goal of structured-data research at Google is to build tools that enable a rich ecosystem of structured data on the Web. To fuel such an ecosystem we need to build tools for discovery, extraction, annotation, sharing, querying, integration, visualization and publishing of structured data sets.“

Mit Hilfe von Google Fusion Tables soll letztlich ein Netzwerk an tabellarischen Daten entstehen, die im Idealfall größtenteils miteinander verknüpft sind. Auf diese Weise kann genutzt werden was bereits existiert und muss nicht mühselig von jedem Benutzer erstellt werden, z.B. Berliner Weihnachtsmärkte. Um passende Tabelle für einen „Merge“ zu finden, hat Google im Oktober 2012 die Funktion „Find a table to merge with...“ unter dem Menüpunkt „File“ in Google Fusion Tables integriert³.

Eine Auslagerung von Datenbeständen für eigenen Anwendungen in Google Fusion Tables wird voraussichtlich dennoch keine Gefahr für existierende Datenbankmanagementsysteme (DBMS).

2.4.3 Funktionsweise der API

Dieses Kapitel soll einen kurzen Einblick in die Verwendung der API auf Basis von PHP, aber auch Javascript geben: Tbc, Code-Snippets

² URL: <http://research.google.com/pubs/DataManagement.html> [11.01.2013]

³ URL: <https://groups.google.com/forum/#!topic/fusion-tables-announce/SMTNjTZve64> [11.01.2013]

Die Google Fusion Tables API 1.0 ist eine REST-API, die Anwendern ermöglicht, Fusion Tables-Resourcen wie Tabellen, Zeilen, Spalten, Templates und Styles anzulegen und zu verwalten.

Basic Concepts:

- **Table:** Tables consist of data arranged in rows and columns, similar to a spreadsheet. There are three types of tables: base, view, and merged.
 - **base:** A table created by uploading data from a repository or through inserting rows with SQL SELECT statements. This is the table from which you create views.
 - **view:** A table created by selecting a subset of columns from a base table with or without filters on the row data. A view is a virtual table, with its own metadata, providing a window into the underlying base table. A view may have a filter applied so that it includes only a subset of rows and columns. Currently, you can only programmatically create a view with the SQL API, not with Fusion Tables API.
 - **merged:** A table created by combining two or more tables, each of which must have a column containing values that match a key column in the other table(s). Merged tables inherit the column types, data, and edit permissions on data from the base table. You can merge tables owned by different people, but you cannot insert new rows into a merged table.
- **Table ID:** The table identifier is an encrypted string value such as 1e7y6mt-qv8911111111_aaaaaaaa_CvWhg9gc.

There are several ways to get a table's ID:

- Choose the File > About menu command in the web application.
 - Use the Google Fusion Tables API to retrieve a list of all tables where the authenticated user is the owner or editor.
 - Use the Google Docs List API with the table's name (See the code sample and the Google Docs List API documentation for details).
- **Column:** A column within a table. Each column has an ID, a name, and an assigned data type: "STRING", "NUMBER", "DATETIME", or "LOCATION". The column ID is a string value. See Working with columns for details on how the ID is parsed.
 - **Template:** A layout template that defines the content and appearance of the row data that is displayed when a user clicks on a marker, line, or polygon in a map visualization. According to the Google Maps API documentation, which calls this information an info window, "The info window looks a little like a comic-book word balloon; it has a content area and a tapered stem, where the tip of the stem is at a specified location on the map." You can customize the template by selecting which columns to display and by specifying a custom

layout using HTML code.

- **Style:** A style definition for map features. Features displayed on a map can be assigned a single style, styled per a column containing style definitions, or styled per a column containing numerical data. You can configure the styles in a map.
- **Row:** Fusion Table data is stored in rows, each of which has an internally generated ROWID that identifies it uniquely within a table. To locate a specific row of data, issue an SQL query statement with criteria to match that row. The response includes the row's ROWID, which you can now use to update or delete that row.

Basic Operations:

- **REST: Table, Column, Template, Style**

For Table, Column, Template, and Style Resources, you can use HTTP requests as described in the following table:

Operation	Description	HTTP mappings
list	Lists all resources of this type.	GET on a resource list URI.
get	Gets a specific resource.	GET on a resource URI.
insert	Inserts a new resource (create a new resource).	POST on a resource list URI, where you pass in data for a new resource.
update	Updates a specific resource.	PUT on a resource URI, where you pass in data for the updated resource.
delete	Deletes a specific resource.	DELETE on a resource URI.

- **SQL: Rows**

For rows, you can use SQL statements with HTTP requests, as described in the following table. SQL queries are sent to the following URL endpoint as the value of an "sql" parameter: <https://www.googleapis.com/fusiontables/v1/query?sql=>. See [Sending requests for examples](#).

Operation	Description	Query format of the "sql" parameter
list	Lists all rows within a table.	GET with a specific table ID: SELECT ROWID FROM <table_id>
get	Gets a specific row.	GET with a specific table ID and criteria:

		SELECT ROWID FROM <table_id> WHERE <your filter>
insert	Inserts a new row into a table.	POST with a specific ROWID, where you pass in data for a new row: INSERT INTO <table_id> (<column_name> {, <column_name>}*) VALUES (<value> {, <value>}*)
update	Updates a specific row.	POST with a specific ROWID, where you pass in data for the updated row: UPDATE <table_id> SET <column_name> = <value> {, <column_name> = <value>}* WHERE ROWID = <row_id>
delete	Deletes a specific row.	POST with a specific ROWID: DELETE FROM <table_id> {WHERE ROWID = <row_id>}

Authentication: (https://developers.google.com/fusiontables/docs/v1/getting_started?hl=de)

3 ANWENDUNGSSPEZIFIKATION

Ziel der Spezifikation beinhaltet die Definition der Anforderungen an das zu entwickelnde Datencockpit für die Berliner Weihnachtsmärkte auf Basis von Google Fusion Tables. Neben den Anforderungen an die Applikation sind die Meilensteine, Ziel sowie auch ein einfaches Design / Wireframing enthalten. Die Umsetzung der Applikation "Datencockpit" erfolgt auf Basis dieser Spezifikation.

3.1 Motivation

Sobald die Weihnachtszeit vor der Tür beginnt für viele eine ganz besondere Zeit des Jahres - Weihnachtsmarktzeit. Ende November, mit dem offiziellen Start der Saison schießen sie dann wie Plize aus dem Boden. Es gibt Weihnachtsmärkte in verschiedensten Kategorien, Größen und Ausprägungen, für Kinder, für Jugendliche oder Erwachsene die nur etwas schlendern wollen. Das Überangebot ist dabei sehr unübersichtlich und intransparent. Die Informationen müssen über einzelne Weihnachtsmärkte mühselig zusammen gesucht werden. Interessant sind hierbei vor allem Informationen wie Öffnungszeiten, Anfahrt und ggf. Eintrittspreise. Mehr als diese Informationen sind darüber hinaus meist nicht bekannt.

Das zu entwickelnde Datencockpit soll als zentrale Anlaufstelle für die Suche nach Weihnachtsmärkten dienen. Um die volle Funktionsweise innerhalb des zur Verfügunggestellten Zeitrahmens im Wissenschaftlichen Projekt darstellen zu können, werden nur Berliner Weihnachtsmärkte betrachtet. Allein in Berlin gibt ca. 100 Weihnachtsmärkte. Neben der Bezeichnung "Weihnachtsmärkte" sind Advents-, Winter-, Kiez- aber auch Weihnachtsflohmärkte teil der Analyse und Auswertung im Datencockpit.

Die Funktionalität des Datencockpits unterteilt sich in zwei Hauptbereiche: die Suche und **das Einreichen eines neuen** Weihnachtsmarktes.

3.2 Suchfunktionalität

Die Suche ist keine gewöhnliche Volltextsuche sondern letztlich eine Filter, welcher die Ergebnisse auf Basis einer Datenquelle filtert. Die Datenbasis ist hierbei nicht in einer Datenbank abgelegt sondern in Form einer Google Fusion Table.

Das Datencockpit, was letztlich eine Micro-Website ist, stellt die notwendigen Filtermöglichkeiten zur Verfügung. Nach der Auswahl eines oder mehrerer Kriterien, erfolgt die Ausgabe der gefilterten Ergebnisse. Die einzelnen Kriterien werden dabei mit einem logischen UND verknüpft. Umso mehr Kriterien gleichzeitig ausgewählt werden, desto weniger Ergebnisse werden letztlich zur Verfügung stehen. Die zur Verfügung stehenden Kriterien sollen hierbei ebenfalls in Abhängigkeit angepasst werden, so dass nur mögliche Kriterienkombinationen für die Filterung verwendet werden

können.

Aktuelle Filtermöglichkeiten (abhängig von den zu ermittelnden Daten)

- Bezirk
- Öffnungszeiten
- Eintrittspreis
- Sehenswürdigkeiten in der Nähe
- Verfügbare Weihnachtsmärkte an einem bestimmten Tag
- Fahrgeschäfte
- Programme (Kinderprogramm, Live-Musik)
- Sortiment / Kunst

Eine der Stärken von Google Fusion Tables ist unter anderem die Verknüpfung verschiedener Google Fusion Tables. Das heisst, dass die eigentlich Datenbasis aus verschiedenen Quellen bestehen kann. Auf diese Weise können vorhandene Informationen verwendet werden und müssen nicht noch einmal in eine eigene Tabelle überführt werden. Für die sinnvolle Verknüpfung benötigt es jedoch einen gemeinsamen Schlüssel. Je nach Kriterium wäre eine Google Fusion Table mit Berliner Sehenswürdigkeiten denkbar. Sofern die PLZ eines Weihnachtsmarktes und einer Sehenswürdigkeit identisch ist, können diese Daten verknüpft werden und dem Benutzer letztlich hilfreiche Informationen zur Sehenswürdigkeiten in der Nähe zur Verfügung gestellt werden.

Die Ausgabe der gefilterten Ergebnisse erfolgt neben der klassischen Listendarstellung auch auf Basis einer interaktiven Karte mit Hilfe von Google Maps.

Eine der Stärken von Google Fusion Tables ist generieren eines eigenen Overlays über einer Google Map. Durch die Angabe von einzelnen Geokoordinaten können gar ganze Bezirksgrenzen bzw. Weihnachtsmärkte nachgezogen und als eine separate Schicht über der Google Map angezeigt werden. Die Google Map aktualisiert sich dabei nach jeder weiteren Kriterienauswahl. Gewöhnlicher Weise können auf der Google Map einzelne Marker positioniert werden, beispielsweise der Mittelpunkt eines Weihnachtsmarktes. Mit Hilfe von Google Fusion Tables ist jedoch möglich, den genauen Rahmen des jeweiligen Weihnachtsmarktes zu visualisieren. Auf diese Weise erschließt gleich der gesamte Weihnachtsmarkt und auch mögliche Zugänge über benachbarte Straßen.

Um die Weihnachtsmärkte auf einer Karte zu platzieren, müssen die entsprechenden Daten für die Geokoordination ebenfalls vorliegen. Google Maps bietet hierzu verschiedene Möglichkeiten. Das Ziel sollte hierbei sein Weihnachtsmärkte anhand ihrer öffentlichen Adressen direkt auf der Karte platzieren, ohne den Umweg der gewohnten Geokoordination Longitude und Latitude.

Nachdem sich der Benutzer für einen Weihnachtsmarkt entschieden hat ist Verknüpfung an diverse Anfahrtsberechnungssysteme wie Google Maps (Route) oder die Berliner Verkehrsbetriebe (BVG) denkbar.

Neben der Funktion, Benutzer bei der Suche nach bestimmten Weihnachtsmärkten behilflich zu sein, können anhand der Daten anschließend auch verschiedene Diagramme generiert werden. Beispielsweise die Anzahl der Weihnachtsmärkte je Bezirk oder eine Gegenüberstellung der Weihnachtsmärkte mit Fahrgeschäften oder ohne.

3.3 Infrastruktur

Neben den Anforderungen an die finale Anwendung müssen auch verschiedene technische Anforderungen erfüllt werden:

- Google Drive Benutzer-Accounts für die Google Fusion Tables
- gewöhnlicher Apache Webserver mit PHP
- Integration der Google Fusion API
- GIT-Repository für die gleichzeitige Entwicklung der Anwendung

3.4 Ziele

- Ziele (Patrick):
Wireframing / Design für das Datencockpit (Patrick)
- Berliner Sehenswürdigkeiten:
<https://www.google.com/fusiontables/DataSource?docid=1HDqaECa6vdWYiqsXNxpjdg-z7A1XLiDyq0MbxMRo>
- Antiquitätenhändler:
<https://www.google.com/fusiontables/DataSource?docid=1W9Euxyje8GcwT1FTwFfFCNhzbA1bqZfPGAzFIEz4>

4 TECHNISCHE UMSETZUNG

Abseits der thematischen wie technischen Einarbeitung und der Spezifikation des Entwicklungsvorhabens und Projektziels gliedert sich die Umsetzung und Implementierung des Weihnachtsmarktcockpits in die folgenden Arbeitsschritte:

1. Inhaltlicher und schematischer Entwurf des Datencockpits
2. Datenerhebung bzw. -recherche,
3. Importieren und Aufbereiten der Daten,
4. Konfigurieren der Datenvisualisierungen,
5. Erstellung der Cockpit-Website und
6. Implementierung der interaktiven Anzeigesteuerung.

4.1 Schematischer Entwurf

Gemäß der oben angeführten Spezifikation soll das Weihnachtsmarkt-Datencockpit bequem und benutzerfreundlich Informationen zu Berliner Weihnachtsmärkten auf Grundlage des Kartenmaterials von Google Maps aufbereiten und zur Verfügung stellen. Dabei soll Anwendern die Möglichkeit gegeben werden, das Weihnachtsmarktangebot gezielt nach relevanten Kriterien zu filtern und zu durchsuchen. Zentrale Suchkriterien sind dabei die Öffnungszeiten der Märkte, die Lage (unterschieden nach Bezirk), das Marktsortiment (unterschieden nach traditionell, Mittelalter, kinderfreundlich, etc.) sowie die Frage nach eventuellen Eintrittskosten und nach der Verfügbarkeit von Fahrgeschäften. Zudem soll die Anwendung Usern erlauben, über die Eingabe der eigenen Position alle verfügbaren Weihnachtsmärkte in der Nähe aufgezeigt zu bekommen. Als besonderes Feature können zusätzlich Weihnachtsmarktstatistiken auf dem Kartenmaterial und in beigefügten Diagrammen abgefragt werden, etwa über die anteilmäßige Verteilung der Märkte auf die Berliner Stadtbezirke oder etwa durchschnittliche Glühweinpreise.

In einem ersten Entwurf für die Vorgabe und Umsetzung des Datencockpits wurde folgender Wireframe erstellt, der die eben genannten Funktionsmerkmale berücksichtigt:

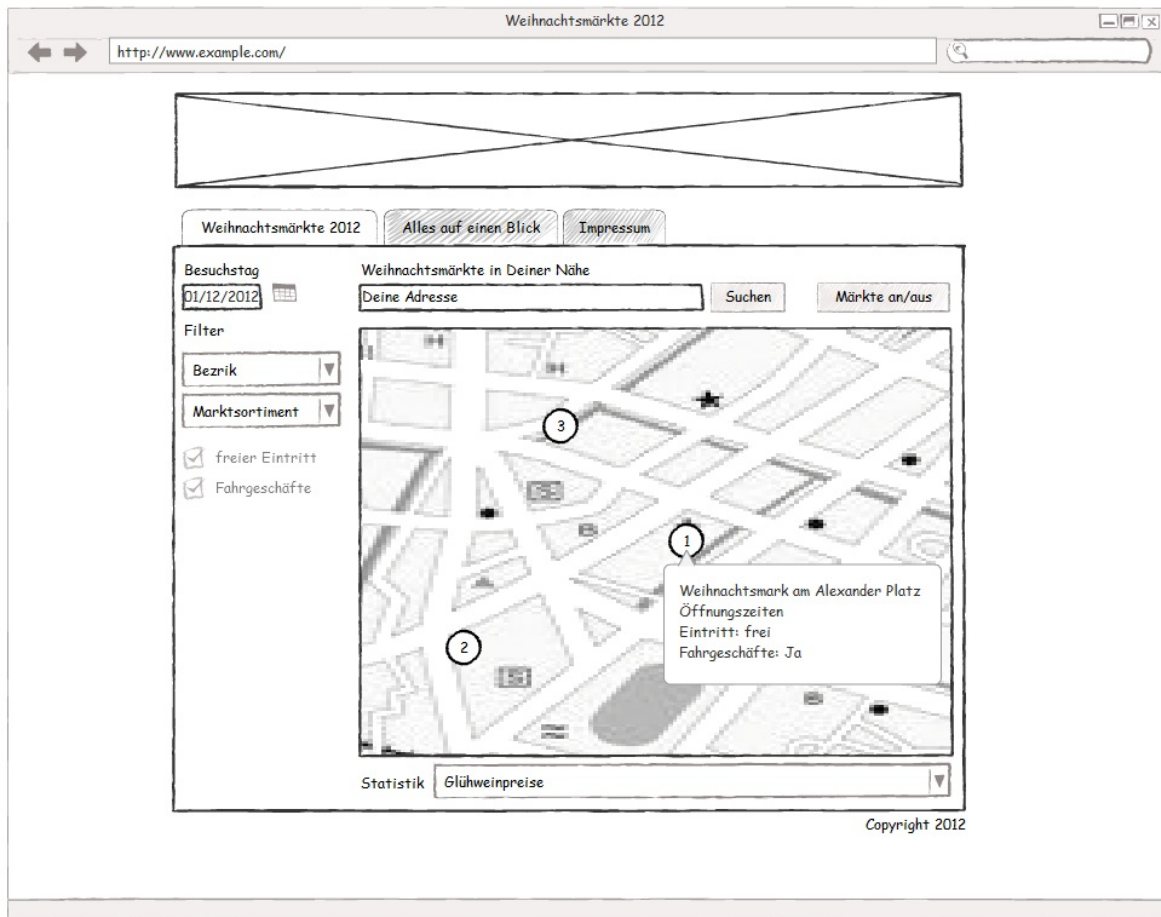


Abb. X: Wireframe des Weihnachtsmarkt-Datencockpits

4.2 Datenerhebung

Die Daten für die Berliner Weihnachtsmärkte 2012 sind abrufbar über das *Berlin Open Data*-Portal der Senatsverwaltung für Wirtschaft Technologie und Forschung des Landes Berlin unter <http://daten.berlin.de/datensaetze/berliner-weihnachtsm%C3%A4rkte-2012>:

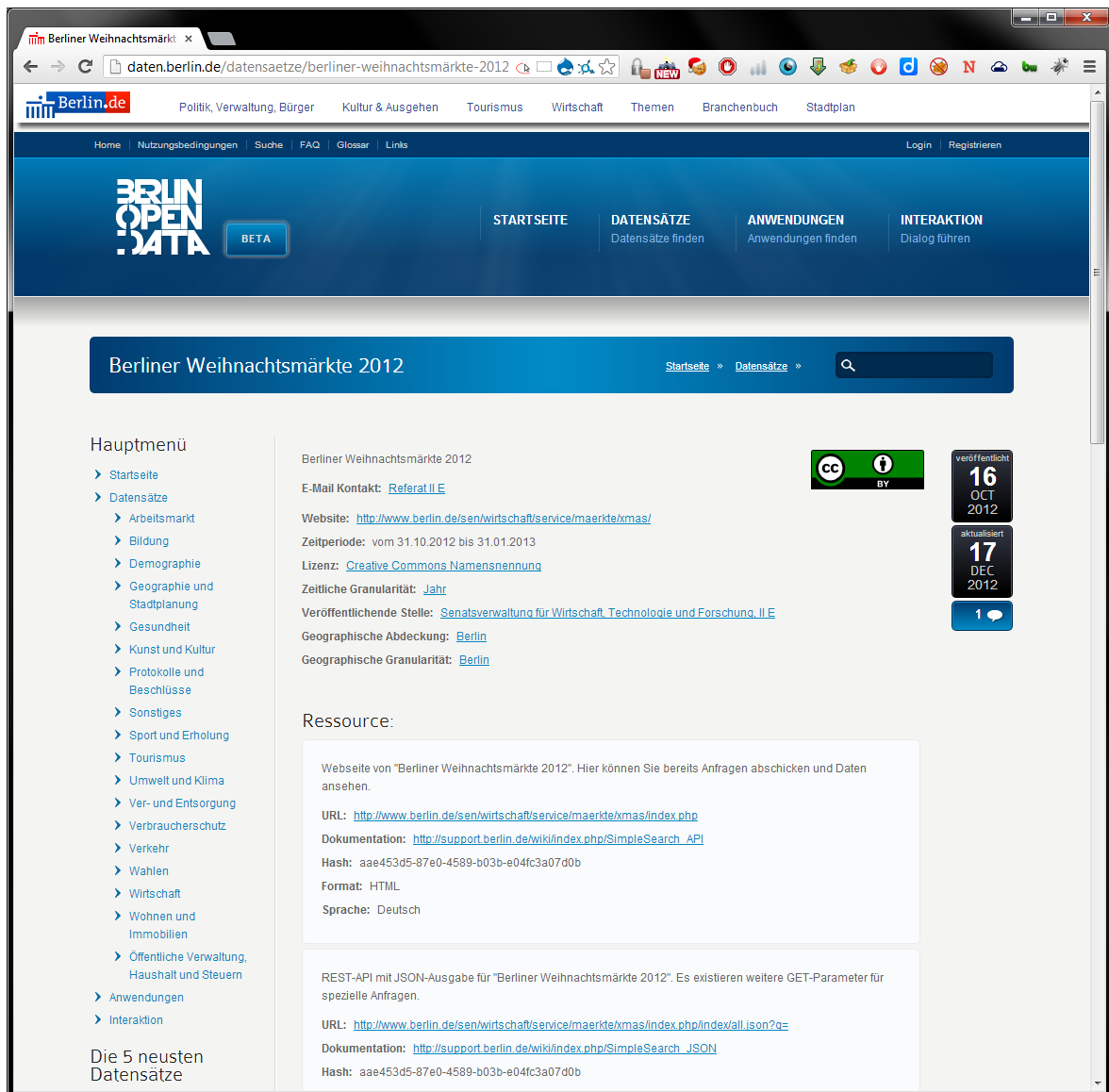


Abb. X: Berlin Open Data-Portal der Senatsverwaltung Berlin

Das Portal stellt mehrere Möglichkeiten zur Abfrage der Daten in den Formaten HTML, JSON, XML, XLS und CSV bereit. Nach dem Herunterladen der Daten müssen diese für den Einsatz mit Google Fusion Tables optimiert werden. Dazu gehört u.a. die Verkettung der Adressdaten in ein von Google geocodierbares bzw. von Google Maps als Geodaten interpretierbares Format. Dieses Format ist landesspezifisch und für die Berliner Anschriften gilt folgendes Schema:

[spez. Ortsbezeichnung]⁴, [Straßenname] [Hausnummer], [Postleitzahl] Berlin, Deutschland-Lorem...

4 optional

4.3 Datenimport und -aufbereitung

Fusion Tables anlegen, Views und Merges

4.3.1 Probleme beim Importieren der Daten in Tabellenkalkulationsprogramm:

Sowohl in Microsoft Excel 2010 als auch in OpenOffice Calc traten beim Einlesen der Daten zur weiteren Nachbearbeitung Fehler auf. So wurde beispielsweise beim Standardmäßigen Öffnen der csv-Datei in Excel die UTF-8-Kodierung nicht erkannt und somit alle Umlaute verworfen. Der Versuch, die csv-Datei über die Datenimportfunktion mit richtiger Kodierung zu importieren scheiterte daran, dass einige Zelleninhalte Zeilenumbrüche enthalten. Dieser wird standardmäßig von Excel als Zeilenende und damit Ende des Datensatzes interpretiert. Der Einfachheit halber wurde im vorliegenden Fall die csv-Datei klassisch geöffnet und die Umlaute kurzerhand mittels Suchen und Ersetzen ausgebessert.

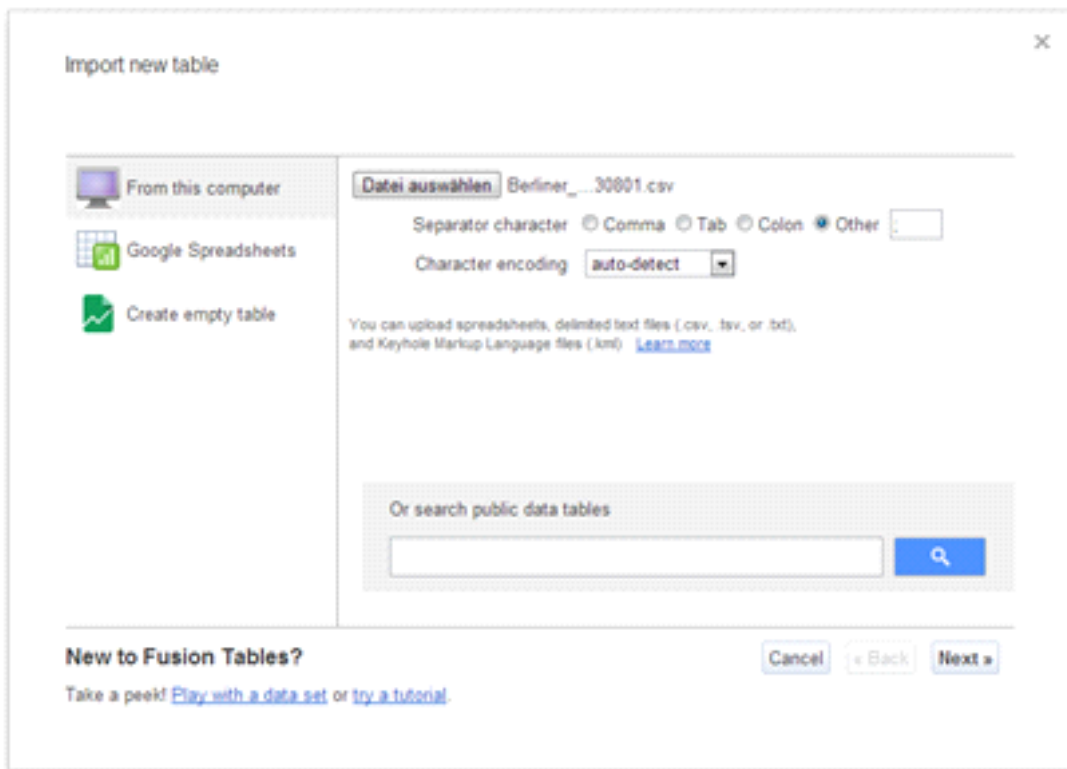


Abb. X: Neue Tabelle importieren (1)

Beim Import in Google Fusion Tables ist es dann wichtig, als Trennsymbol das von Excel für csv-Dateien verwendete Semikolon („;“) unter „Other“ einzustellen und das Character Encoding die Anweisung „auto-detect“. Excel speichert die Dateien standardmäßig in ANSI-Kodierung ab. Die von der API generierten csv-Files haben eine „ANSI as UTF-8“-Kodierung und werden von Google nicht unterstützt.

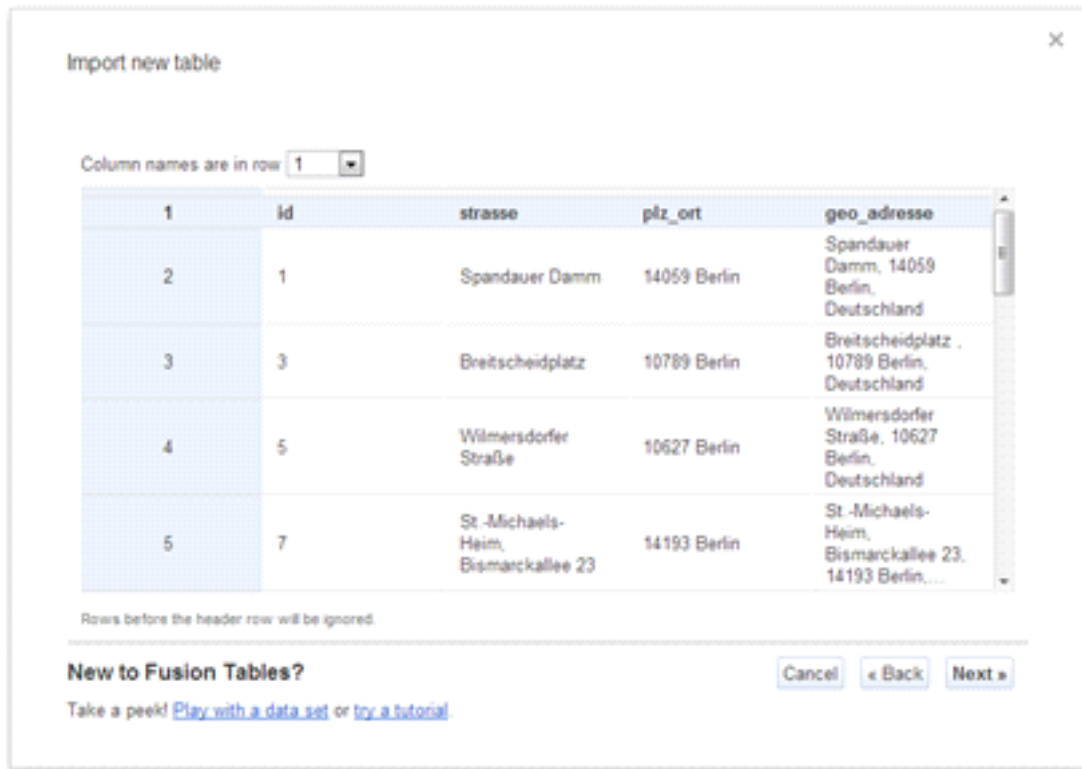


Abb. X: Neue Tabelle importieren (2)

Lorem...

4.4 Konfiguration der Visualisierungen

Visualisierungen & Styles

4.5 HTML-Prototyp

HTML-Prototyp und Einbettung der Map

4.6 Interaktive Anzeigesteuerung

Initialisierung, MapView, Style, ActionListeners

4.7 Einschränkungen/Ausblick

Was könnte man noch machen? Welche Abfragen wären noch denkbar? Kann aber eigentlich auch in die Zusammenfassung, ggf. nur noch ein Schlusssatz zur Umsetzung hier.

5 ZUSAMMENFASSUNG

Lorem ipsum...

Schwierigkeiten, Herausforderungen, etc.

Abschlusspräsentation

Peka Kucha Stil (vielleicht)

LITERATUR/QUELLEN

- <http://www.berlin.de/orte/weihnachtsmaerkte/>
- <http://www.mediummagazin.de/magazin-plus/technikipp-so-funktioniert-google-fusion/>
- Berliner Geo-Tags (Bezirksmitten):
https://maps.google.com/maps?q=http://toolserver.org/~para/cgi-bin/kmlexport%3Fproject%3Dde%26article%3DListe_der_Bezirke_und_Ortsteile_Berlins
- Aktuelle Fusion Tables API:
https://developers.google.com/fusiontables/docs/v1/getting_started?hl=de
- deprecated Fusion Tables API:
https://developers.google.com/fusiontables/docs/developers_guide?hl=de-DE
- Berliner Geo-Tags:
<http://ags.misterboo.de/#x=13.4246&y=52.5071&z=11>
- Berlin Pankow-Versuch:
<https://maps.google.de/maps/ms?authuser=0&ie=UTF8&hl=de&oe=UTF8&msa=0&msid=209178059779463884773.0004ce8f590a5735236ce&output=kml>

Conference Papers & Publications⁵

- [Big Data Storytelling through Interactive Maps.](#) Jayant Madhavan, Sreeram Balakrishnan, Kathryn Brisbin, Hector Gonzalez, Nitin Gupta, Alon Halevy, Karen Jacqmin-Adams, Heidi Lam, Anno Langen, Hongrae Lee, Rod McChesney, Rebecca Shapley, Warren Shen. Google Inc. [Bulletin of the Technical Committee on Data Engineering, IEEE. June 2012, Vol. 53 No. 2.](#)
- Field Information Management Systems for DNA Barcoding. Deck John, Gross Joyce, Stones-Havas Steven, Davies Neil, Shapley Rebecca, and Meyer Christopher. Chapter 12 in: W. John Kress and David L. Erickson (eds.), DNA Barcodes: Methods and Protocols, Methods in Molecular Biology, vol. 858
- [Google Earth and Google Fusion Tables in support of time-critical collaboration: Mapping the deepwater horizon oil spill with the AVIRIS airborne spectrometer.](#) Eliza S. Bradley, Dar A. Roberts, Philip E. Dennison, Robert O. Green, Michael Eastwood, Sarah R. Lundeen, Ian B. McCubbin and Ira Leifer. Earth Science Informatics, 17 September 2011.
- [Fusion Tables: new ways to collaborate on structured data.](#) Jonathan Goldberg Kidon, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science, 2010. 61 pages.
- [Google Fusion Tables: Web-Centered Data Management and Collaboration.](#) Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley,

⁵ URL: <https://sites.google.com/site/fusiontablestalks/talks>

Warren Shen, Jonathan Goldberg-Kidon. Proceedings of SIGMOD 2010 (Industrial Track), Indianapolis, Indiana.

- [Google Fusion Tables: Data Management, Integration and Collaboration in the Cloud](#). Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen. Proceedings of the Symposium on Cloud Computing, 2010 (Industrial Track), Indianapolis, Indiana.

Presentations & workshops by the Fusion Tables team⁶

- Data Boot Camp using Google tools, LA Hackathon on Immigration, 8-9 Dec 2012
- Data sharing, visualization and app building with Fusion Tables, Open Developer Week Dec 2012
- Find data and Merge faster, a free training over Google Hangouts, October 2012.
- TechRaking: CIR and Google, April 12, 2012, "Tips and Tricks: Fusion Tables and more"
- Investigative Reporters & Editors CAR conference, "Getting the most out of Fusion Tables" demo and "Advanced Fusion Tables" hands-on workshop, Feb 2012
- A data tools and visualization workshop in January 2012 in Nairobi Kenya
- Silicon Valley Code Camp, October 2011
- Online News Association, September 2011
- Managing and Visualizing your geospatial data with Fusion Tables, Google I/O video, May 2011
- Data 2.0 conference workshop April 2011
- Silicon Valley GTUG, April 2011
- Where 2.0 workshop, April 2011
- Investigative Reporters & Editors CAR conference, NICAR demo & workshops February 2011
- Association of American Geographers, "Visualize gobs of data with Fusion Tables" April 2010
- Google Earth Outreach, Train the Trainers event 2.5 hour training and 1 hour training, The International Livestock Research Institute , Nairobi, Kenya Nov 2009
- Google Geo-Public sector partnership event, UN, New York, Nov, 2009
- Google Geo-Public sector partnership event, Washington DC, Oct. 2009

⁶ URL: <https://sites.google.com/site/fusiontablestalks/talks>