PLChessQL - Schach mit APEX, Node.js und Twitter

Moritz Schenkel
Syntegris information solutions GmbH
Neu-Isenburg

Schlüsselworte

Apex, JavaScript, NodeJS, Oracle Jet, PLSQL, SQL

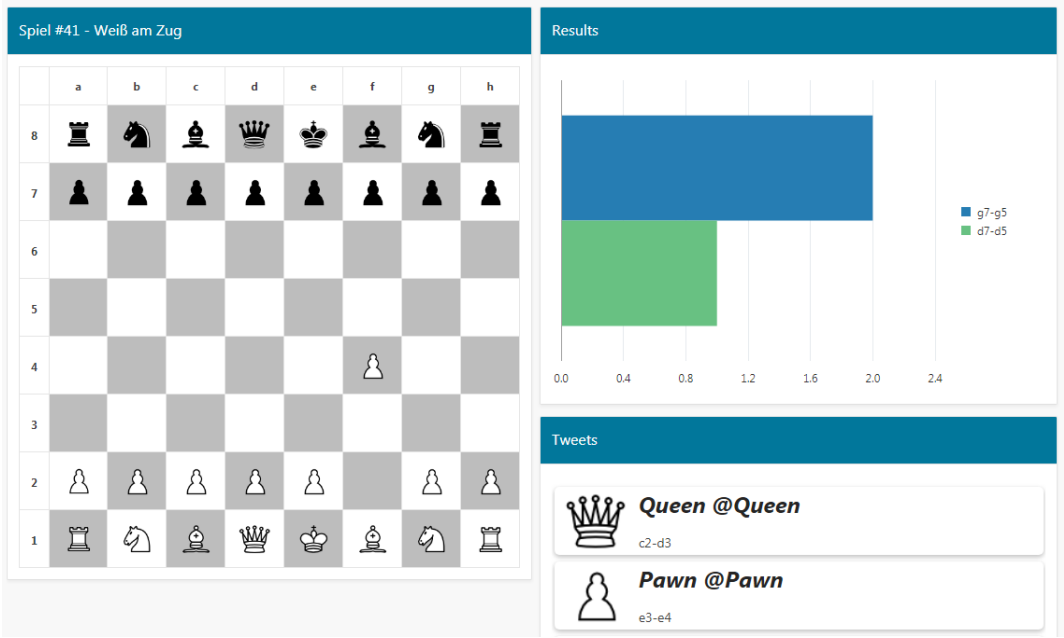
Einleitung

Apex wird in Unternehmen vorrangig als Frontend zur Verwaltung von Stammdaten verwendet, mit dem Zweck Geschäftsprozesse zu unterstützen. Dabei wird sich meist der Hausmittel bedient, die Apex in seinem Grundumfang anbietet. Dies umfasst vorwiegend simple Prozesse und gipfelt in Dynamic Actions. Geht man einen Schritt weiter, können Daten aktualisieren werden, indem AJAX Callbacks genutzt werden. Damit wird eine Interaktion zwischen Datenbank und Benutzer geschaffen, welche einen Page Refresh unnötig macht und durch die Apex API unterstützt wird. Will man dies jedoch noch weitertreiben und eine direkte Interaktion zwischen Benutzern ermöglichen, stößt Apex an seine Grenzen. Hier bietet NodeJS mit seinem extrem umfangreichen Package Manager, eine optimale Plattform, um darauf weiter zu entwickeln. Viele Funktionen, die sonst selbst entwickelt werden müssen, lassen sich so durch die Nutzung eines gut getesteten Packages schneller und einfacher implementieren.

Der Vortrag soll hier anhand einer beispielhaften Schachanwendung aufzeigen in welchen Schritten die Entwicklung stattfinden kann und wie man die zahlreichen Komponenten verknüpft.

Die Apex Anwendung

Apex dient in dieser Anwendung vorwiegend der Präsentation des Spielfeldes. Zusätzlich wurden aber auch Funktionen implementiert, die es ermöglichen sollen, Züge selbst durchzuführen. So kann direkt gegen einen anderen Spieler angetreten werden.



Um die Partie zwischen zwei Spieler auszuweiten, kann aber auch der Gegner durch die Twitter-Community ersetzt werden. Dies wird durch einen NodeJS Server ermöglicht, der via Websockets angesprochen wird. Zur Präsentation der Twitter Ergebnisse wird Oracle Jet verwendet. Jet nutzt die ViewModels von Knockout.js und erzeugt damit sich automatisch aktualisierende Diagramme.

NodeJS

Der NodeJS Server nimmt die zentrale Rolle in jeglicher Kommunikation zwischen den Spielern ein. Sei es ein Client an der Apex Anwendung oder ein Twitter User, der sich am Spiel beteiligt – alle Verbindungen werden über den Node Server gesteuert. Dazu wird das NPM Package websocket in der Version 1.0.24 verwendet. Mit nur wenigen Zeilen Code lassen sich so Daten zwischen den Teilnehmern austauschen.

Twitter hingegen wird nicht über Websockets angebunden. Hierfür findet sich ein weiteres Package im npm - twit. Twit ist einer von vielen Clients, der offiziellen Twitter REST-API in npm. Auch hier sind wieder nur wenige Codezeilen erforderlich, um das Package einzubinden und nutzbar zu machen. Folgende Codezeilen als Beispiel:

```
var Twit = require('twit')

var T = new Twit({
  consumer_key:         '...',
  consumer_secret:      '...',
  access_token:         '...',
  access_token_secret:  '...',
  timeout_ms:           60*1000,  // optional HTTP request timeout to apply to all requests.
})

//  tweet 'hello world!'
T.post('statuses/update', { status: 'hello world!' }, function(err, data, response) {
  console.log(data)
})
```

Ein weiterer zentraler Punkt des NodeJS Servers der Applikation ist das oracledb Package. Da die Logik in PLSQL Packages realisiert wurde, ist dieses unabdinglich, um eine Kommunikation von NodeJS und Oracle zu ermöglichen. Spätestens bei dessen Einbindung fällt die Asynchronität des JS Codes auf, da eine Initialisierung einer Connection diese nicht unbedingt sofort verfügbar macht. Stattdessen sind Callback Functions oder Promises nötig, um einen korrekten Ablauf zu gewährleisten. Vornehmlich werden Prozeduren und Funktionen innerhalb des Packages aufgerufen und deren Ergebnisse an die Benutzer der Apex Anwendung weitergeleitet.

PLSQL

Als namensgebende Komponente für den Vortrag ist hierauf natürlich auch einzugehen. Sämtliche Spiellogik wurde in PLSQL implementiert - das Schachbrett, alle Züge sowie die aktuelle Brettsituation in Tabellen persistiert.

Fazit

NodeJS ist eine mächtige Erweiterung für Apex Umgebungen um Anwendungen noch interaktiver machen zu können. Es ermöglicht vor allem die Event bedingte Entwicklung von Applikationen. Eine

weitere angenehme Erweiterung stellt Oracle Jet dar, dass einen großen Komfortgewinn für die dynamische Aktualisierung von Elementen innerhalb der Anwendung darstellt.

Sämtlicher Code der Anwendung wird nach dem DOAG Vortrag öffentlich zugänglich gemacht.

Kontaktadresse

Moritz Schenkel

Syntegris information solutions GmbH

Herrmannstraße 54

63263 Neu-Isenburg

Telefon:	+49 (0) 6102 - 29 86 68
Fax:	+49 (0) 6102 - 55 88 06
E-Mail	moritz.schenkel@syntegris.de
Internet:	www.syntegris.de