Benutzungs-oberflächen-Labor

Prof. Dr.-Ing. Holger Vogelsang

Winterssemester 2015/2016

Don’t panic

Inhaltsverzeichnis

1

Übersicht

Dieses Kapitel gibt Ihnen einen kleinen Überblick über die Lernziele der Veranstaltung und stellt die Use-Cases für die Aufgabenstellungen vor.

1.1 Lernziele

Es soll der Bau einer Web-Oberfläche für eine verteilte Anwendung in Gruppenarbeit geübt werden. Dabei soll sich die Oberfläche sowohl auf Desktop-Browsern als auch Smartphones sauber darstellen und bedienen lassen (Stichwort „Responsive Design“).

1.2 Organisation

Bitte bearbeiten Sie die Aufgabe in Gruppen mit jeweils ca. drei Mitgliedern. Jede Gruppe erstellt unabhängig von den anderen Gruppen eine eigene Lösung.

2

Grundgerüst der Anwendung

In dieser Aufgabe erstellen Sie einen Webauftritt für den Zugriff auf die Praxisbörse. Die Börse ist über eine REST-Schnittstelle zugänglich. Die Dokumentation der Schnittstelle finden Sie unter [Intr] im Abschnitt 3.7. Als Technologien verwenden Sie AngularJS und Bootstrap oder Angular Material.

2.1 Use-Cases

Folgende Use-Cases werden umgesetzt:

1. Anwender können ihre Anmeldedaten eingeben.
2. Ein Anwender kann sich alle Angebote eines Typs wie Praxissemester usw. anzeigen lassen. Weiterhin darf er einen Text eingeben, anhand dessen die Treffer serverseitig gefiltert werden. Laden Sie im Falle mobiler Clients immer nur jeweils 10 Angebote herunter und erlauben Sie dem Anwender das Blättern zu den Folgeangeboten. Wird die Anwendung im Browser aufgeführt, dann laden Sie stets alle Angebote herunter. Ermitteln Sie die Angebotstypen dynamisch über die REST-Schnittstelle. Ein festes „Einprogrammieren“ ist keine Lösung.
3. Beim Klick auf den Firmennamen wird eine Detail-Information zur Firma z.B. in einem Popup-Dialog angezeigt.
4. Beim Klick auf ein Angebot wird eine Detail-Information zum Angebot z.B. in einem Popup-Dialog angezeigt.
5. Erlauben Sie dem Anwender auch das Filtern nach Ländern, in denen die Arbeit durchgeführt werden soll. Da das die REST-Schnittstelle nicht unterstützt, filtern Sie clientseitig.
6. Anwender können Angebote auf ihren eigenen Merkzettel setzen.
7. Anwender können den Merkzettel einsehen, Angebote davon entfernen oder sich Details der Angebote anzeigen lassen.

2.2 Design

Für das Design können Sie beispielsweise Bootstrap [BStr], [AnB] oder Angular Material [AngMat] verwenden. Bootstrap basiert auf einer Sammlung von CSS- und Javascript-Dateien, die ein „Responsive Design“ erlauben. Damit ist gemeint, dass Sie recht einfach Web-Anwendungen bauen können, die sich automatisch der Display-Größe eines Geräts anpassen. Twitter hat die Bibliothek zur freien Verwendung gestellt. zur Verfügung . Verwenden Sie diese Bibliothek, damit Ihre Anwendung sowohl auf Smartphones als auch auf Desktop-Browsern eine optimale Darstellung erhält. Alternativ können Sie auch das sehr schöne Angular Material verwenden. Achten Sie bei jeder Entwurfsentscheidung darauf, dass die Anwendung sowohl auf Desktop-Browsern als auch auf Smartphones gut aussehen und sich problemlos bedienen lässt.

2.3 Konzept

Ihre Anwendung soll diese Merkmale besitzen:

- Sie erstellen eine sogenannte „Single-Page-Application“ sein. Es finden also keine Seitenwechsel statt.
- Sie Nutzen die Modularisierung von AngularJS, indem Sie für die tabellarische Übersicht der Suchergebnisse Templates verwenden: Ein Firmeneintrag und ein Angebot werden jeweils durch ein Template beschrieben. Erstellen Sie ebenfalls Templates für die Detailansichten.
- Wenn Sie Angebote auf Ihrem Merkzettel speichern oder wieder vom Merkzettel entfernen, dann soll das ohne ein explizites Speichern möglich sein: Klickt der Anwender also auf ein Symbol am Angebot für das Hinzufügen zum Merkzettel, so wird diese Einstellung sofort auf dem Server gespeichert.

2.4 Authentifizierung mit AngularJS

Weil die API auf einem reinen REST-Ansatz basiert, ist der Server also aus Sicht des Clients zustandslos. Alle Zugriffe basieren auf dem Ressourcen-Gedanken. Ein daraus sich ergebendes Problem ist, dass es jetzt keinen klassischen Login-Vorgang mehr gibt, weil bei einem Login der Server ja eine Sitzung anlegt, die entweder bis zum Abmelden oder einem Timeout bestehen bleibt. Solch ein Vorgehen würde bedeuten, dass der Server nicht mehr zustandslos ist. Deshalb wird hier für die Aufgabe ein Ansatz gewählt, bei dem der Client für jeden Server-Zugriff seine Authentisierungsinformationen mitliefert. Diese sollen aus Benutzernamen und Passwort bestehen. Das hat bei einer HTTP-Verbindung mit der Basic-Authentifizierung den großen Nachteil, dass gerade das Passwort lediglich Base64-kodiert und damit praktisch im Klartext verschickt wird. Verwenden Sie deshalb nur HTTPS-Verbindungen. Sie greifen bei dieser Aufgabe auf die „echte“ Praxisbörse im Produktivbetrieb zu. Ihre Anmeldedaten entsprechen daher Ihrem IZ-Account.
Ihre Lösung wird nicht von dem Server heruntergeladen, an den Sie die REST-Zugriffe schicken. Das führt zum Problem, dass der Browser solche Zugriffe eigentlich blockiert, weil sie auf eine andere Domäne führen. Mit Hilfe von CORS (Cross-Origin Resource Sharing) lässt sich das Problem umgehen, wobei Sie einige Punkte client-seitig beachten müssen. Die folgenden Code-Abschnitte zeigen Ihnen exemplarisch, wie Sie einen Request auf eine URL in einer anderen Domäne absetzen können, wenn Authentifizierungsinformationen erforderlich sind. Der Server ist bereits dafür konfiguriert.

2.4.1 Globale Einstellungen

Setzen Sie global an Ihrem Modul (im Beispiel heißt es `app`) die folgenden beiden Attribute:

```
app.config(function($httpProvider) {
  // Cross-Domain-Aufrufe erlauben
  $httpProvider.defaults.useXDomain = true;
  // Das Mitsenden von Authentifizierungsinformationen erlauben
  $httpProvider.defaults.withCredentials = true;
});
```

2.4.2 REST-Aufruf

Das folgenden Beispiel zeigt einen REST-Aufruf („POST“), der eine Authentifizierung verlangt:

```
$http.defaults.headers.common.Authorization =
  'Basic ' + $base64.encode(user + ':' + password);

$http.post(url, 1234);
```

AngularJS beherrscht nicht direkt die Base64-Kodierung, die für die Authentifizierung erfolderlich ist. Sie können aber das AngularJS-Modul `angular-base64` verwenden: https://github.com/ninjatronic/angular-base64.

3

Erweiterung durch Kartendarstellung

Erweitern Sie Ihre Detailansicht der Firmen um eine Kartendarstellung. Damit zeigen Sie auf einer Karte von Google oder OpenStreetMap den Firmensitz an. Für die Google-API [GoMaAp] benötigen Sie einen kostenlosen API-Key (siehe API-Dokumentation), für OpenStreetMap ist das nicht erforderlich. Sie können sich überlegen, eine Bibliothek wie OpenLayers [OpLay] einzusetzen. Aus Zeitgründen können Sie die Kartendarstellung auf Desktop-Browser beschränken.

4

Literaturverzeichnis

[AngMat] Angular Material: https://material.angularjs.org/

[BStr] Bootstrap: http://getbootstrap.com/

[AnAuth] Authentifizierung mit AngularJS: http://jasonwatmore.com/post/2014/05/26/AngularJS-Basic-HTTP-Authentication-Example.aspx

[AnB] Bootstrap-Komponenten zusammen mit AngularJS: http://angular-ui.github.io/bootstrap/

[GoMaAp] Google Maps API: https://developers.google.com/maps/

[Intr] REST-Schnittstelle zum Intranet: http://www.iwi.hs-karlsruhe.de/Intranetaccess/public/doc/REST-Schnittstelle.pdf

[OpLay] OpenLayers: http://openlayers.org/

[TB14] Tarasiewicz, Böhm: AngularJS – Eine praktische Einführung in das JavaScript-Framework, dpunkt-Verlag