Address Book

HTML5 Coding Exercise

Contents

- Overview

- Goal

- Setup Instructions

- Coding Instructions

- Backend API Specification

- User Interface Design

Overview

- This is a practical exercise to assess your HTML-5 coding skills.

- It is meant to simulate the type of tasks you will be given as an front-end engineer with Aerohive Networks.

- The exercise provides a Node JS HTTP server for a simple hypothetical backend API as well as a means to serve the content you will write.

- This document includes details on the provided API as well as a simple User Interface design.

- You are expected to provide the required HTML, CSS and JavaScript to build the hypothetical application.

- Feel free to contact your interviewer with questions as they arise.

Goal

- The goal is to build the HTML5 (HTML, JavaScript and CSS) code for a simple Address Book web app (or as much as you can complete within one week).

- Your Address Book web app should follow the provided design mockups and should provide the following functions

  - Display a list of contacts in a sidebar with ability to select one at a time from the list to display its details

  - Ability to create & delete contacts

  - Ability to edit existing contacts by changing given name, surname, phone numbers and addresses

  - Ability to create and delete new phone numbers for contacts

  - Ability to create and delete new addresses for contacts

Setup Instructions

- Download and install Node.js for your platform from https://nodejs.org/en/download/

  - This will give you the "node" and "npm" executables

- Extract the AddressBookServer.zip ZIP file in a working directory. This will create a directory called "AddressBookServer" (hereafter referred to as <abroot>)

- Open a command-line terminal/shell and change your current directory to the <abroot> directory where you extracted AddressBookServer.zip

- Execute "node AddressBookServer.js" on your command-line. This will start the HTTP server (use CTRL-C to quit the server)

- Open a Web browser and go to http://localhost:18080

- **NOTE:** Restarting the server will revert to its initial state. No changes to data are permanently saved.

Coding Instructions

- You will edit the <abroot>/webroot/index.html file and provide other files as needed in <abroot>/webroot to complete the exercise.

- You may write all code from scratch or incorporate third-party toolkits such as Dojo, AngularJS, Bootstrap, ReactJS, BackboneJS, jQuery, etc…. — either by embedding them in the <abroot>/webroot directory or using a CDN.

- Do **NOT** use proprietary code from your current employer.

- Some icon image files are provided in <abroot>/webroot/images, but you may create your own or use stylized unicode characters to achieve the same affect — it is your call.

- Create as many HTML, CSS and JavaScript files as you see fit.

- It is okay to take shortcuts in the interest of keeping this simple, but if you would like to point out an alternative "best practice", just put a comment in your code indicating how you might normally implement the feature.

- Do as much as you can within one week. Incomplete functions should simply pop open a JavaScript alert indicating that they are "under construction".

- When complete or after one week, simply ZIP up the <abroot> and email it back.

Backend API Specification

Backend API Overview

- The backend API is a REST/JSON API and **is provided for you** by the AddressBookServer.js file

- It provides CRUD (Create, Read, Update, Delete) functions for Contacts, Phone Numbers and Addresses

- The REST API is broken down into URL paths that identify specific resources and HTTP methods for acting on them

- Each API function is described in terms of

  - HTTP Method

  - URL path pattern identifying the target resource

  - Short functional description

  - Request Document (body content)

  - Expected HTTP Response Codes

  - Response Document (body content)

- Also provided is a basic overview of the JSON format used to describe the relevant input and output documents for each resource.

Contact API Specification

- **GET /contacts**   *(get all contacts)*

  - Request Document: ***none***

  - Response Codes: 200 *(if no contacts, then an empty array)*

  - Response Document: Array of JSON Contact

- **GET /contacts/[contact-id]**   *(get a specific contact)*

  - Request Document: ***none***

  - Response Codes: 200 if found, 404 if not found

  - Response Document: JSON Contact

- **POST /contacts**   *(create a new contact)*

  - Request Document: JSON Contact

  - Response Codes: 200 if successful

  - Response Document: JSON Contact *(including identifier for contact)*

- **PUT /contacts/[contact-id]**   *(update a specific contact)*

  - Request Document: JSON Contact

  - Response Codes: 200 if found, 404 if not found

  - Response Document: JSON Contact

- **DELETE /contacts/[contact-id]** *(delete a specific contact)*

  - Request Document: ***none***

  - Response Codes: 200 *(if not found, still 200 since DELETE is idempotent)*

  - Response Document: ***none***

Address API Specification

- **GET /address-types** *(get all address types)*

  - Request Document: ***none***

  - Response Codes: 200

  - Response Document: JSON Array of Unique String Address Types

- **GET /contacts/[contact-id]/addresses** *(get addresses for a specific contact)*

  - Request Document: ***none***

  - Response Codes: 200 if found, 404 if not found

  - Response Document: Array of JSON Address

- **POST /contacts/[contact-id]/addresses** *(create a new address for a contact)*

  - Request Document: JSON Address

  - Response Codes: 200 if successful, 400 if bad input, 404 if contact not found

  - Response Document: JSON Address *(including identifier for the new address)*

- **PUT /contacts/[contact-id]/addresses/[address-id]** *(update a specific address)*

  - Request Document: JSON Address

  - Response Codes: 200 if found, 404 if contact or address not found, 400 if bad input

  - Response Document: ***none***

- **DELETE /contacts/[contact-id]/addresses/[address-id]** *(delete a specific address for a contact)*

  - Request Document: ***none***

  - Response Codes: 200 *(if not found, still 200 since DELETE is idempotent)*

  - Response Document: ***none***

Phone API Specification

- **GET /phone-types**   *(get all phone types)*

  - Request Document: ***none***

  - Response Codes: 200

  - Response Document: JSON Array of Unique String Phone Types

- **GET /contacts/[contact-id]/phones**   *(get phone numbers for a specific contact)*

  - Request Document: ***none***

  - Response Codes: 200 if found, 404 if not found

  - Response Document: Array of JSON Phone Number

- **POST /contacts/[contact-id]/phones**   *(create a new phone number for a contact)*

  - Request Document: JSON Phone Number

  - Response Codes: 200 if successful, 400 if bad input, 404 if contact not found

  - Response Document: JSON Phone Number *(including identifier for the new phone number)*

- **PUT /contacts/[contact-id]/phones/[phone-id]**   *(update a specific phone number)*

  - Request Document: JSON Phone Number

  - Response Codes: 200 if found, 404 if contact or address not found, 400 if bad input

  - Response Document: ***none***

- **DELETE /contacts/[contact-id]/phones/[phone-id]** *(delete a specific phone number for a contact)*

  - Request Document: ***none***

  - Response Codes: 200 *(if not found, still 200 since DELETE is idempotent)*

  - Response Document: ***none***

JSON Specification

JSON Contact

```
{

    "contactID": <integer>,  (optional*)

    "givenName": <string>,

    "surname": <string>

}
```

JSON Address

```
{

    "addressID": <integer>, (optional*)

    "addressType": <string>,

    "street": <string>,

    "city": <string>,

    "state": <string>,

    "postalCode": <string>

}
```

JSON Phone Number

```
{

    "phoneID": <integer>,  (optional*)

    "phoneType": <string>,

    "phoneNumber": <string>

}
```

*Identifiers are omitted (or ignored) when a the JSON object is input for a new record or when updating a record (i.e.: the record identifier is in the URL when updating).

User Interface Design

UI Design Overview

- The User Interface Design is provided with a series of mockups that show various use cases as you step through them in sequence.

- Many features of the UI can be inferred. For example, clicking on ⊕ Add new contact is intended to create a new Contact record. Clicking on the Delete button when editing a Contact should delete it.

- You need to create HTML, JavaScript and CSS that leverages the provided backend API to make the user interface functions come to life. You may optionally use a third-party toolkit or library.

- If you end up using a third-party toolkit that provides specific CSS styling, feel free to leverage that default styling for your implementation rather than adhering perfectly to the mockups.

- Remember, do as much as you can and if you cannot complete a task simply stub it out with a JavaScript alert indicating that it is "under construction".

John Doe	
Jane Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Given Name	Jane
Surname	Doe
Home Phone	415-555-1414
Mobile Phone	415-555-1515
Home Address	101 Main Street
	Nowheresville, CA 91000

John Doe

Jane Doe

John Smith

Jill Smith

Bill Rogers

⊕ *Add new contact*

Edit

Given Name   Jane

Surname   Doe

Home Phone   415-555-1414

Mobile Phone   415-555-1515

Home Address   101 Main Street

                    Nowheresville, CA 91000

John Doe	
Jane Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Let's see what happens when we click the sidebar to change the selection

Edit

Given Name	Jane
Surname	Doe
Home Phone	415-555-1414
Mobile Phone	415-555-1515
Home Address	101 Main Street
	Nowheresville, CA 91000

John Doe	
Jane Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Given Name   John

Surname   Smith

Home Phone   415-555-1212

Mobile Phone   415-555-1313

Home Address   200 First Street

               Somewheresville, CA 91002

Notice that the content in this area changed. No CSS3 transition was used in the design, but feel free to incorporate some sort of visual cue to help the user see the change occur (e.g..: slide or flip the content in or fade the background color in and out to call attention to the change)

John Doe	
John Smith	
Jane Doe	
Jill Smith	
Bill Rogers	

⊕ *Add new contact*

Given Name	John
Surname	Smith
Home Phone	415-555-1212
Mobile Phone	415-555-1313
Home Address	200 First Street
	Somewheresville, CA 91002

| John Doe |
| Jane Doe |
| **John Smith** |
| Jill Smith |
| Bill Rogers |
| ⊕ *Add new contact* |

Edit

Given Name	John
Surname	Smith
Home Phone	415-555-1212
Mobile Phone	415-555-1313
Home Address	200 First Street
	Somewheresville, CA 91002

Now let's try to edit this record by clicking the edit button.

3. Notice that the "Delete" button is now available. This should delete the contact.

| John Doe |
| Jane Doe |
| **John Smith** |
| Jill Smith |
| Bill Rogers |
| ⊕ *Add new contact* |

Delete   Done

Given Name | John

Surname | Smith

2. Click "Done" to conclude editing and return to previous view.

⊖ Home Phone ▼ | 415-555-1212

⊖ Mobile Phone ▼ | 415-555-1313

⊕ *Add new phone*

⊖ Home Address ▼ | 200 First Street
                    Street Address
                    Somewheresville | CA | 91002
                    City   State Postal Code

⊕ *Add new address*

4. Adding a new contact while editing another should behave as if you clicked "Done" first and then clicked to add the new contact.

5. You can delete an existing phone number or address.

6. You can add a new phone number or address. **NOTE:** The API requires a valid phone type or address type to create the record.

1. Our content area has changed to reveal the ability to edit the fields that were previously read-only. You may want to consider a CSS3 transition here to show off your skills.
NOTE: No "Save" button is shown. You are free to save with the "onchange" event of each field OR you can queue up changes and wait until the user clicks "Done", but that will be more complex.

John Doe
Jane Doe
John Smith
Jill Smith
Bill Rogers
⊕ *Add new contact*

Delete   Done

Given Name | John

Surname | Smith

⊖ Home Phone   ▼ | 415-555-1212

⊖ Mobile Phone ▼ | 415-555-1313

⊕ Add new phone

⊖ Home Address ▼ | 200 First Street
                      Street Address
                      Somewheresville   CA   91002
                      City   State   Postal Code

⊕ Add new address

| John Doe |
| Jane Doe |
| **John Smith** |
| Jill Smith |
| Bill Rogers |
| ➕ *Add new contact* |

Delete   Done

Given Name | John |

Surname | Smith |

➖ Home Phone ▼ | 415-555-1212 |

➖ Mobile Phone ▼ | 415-555-1313 |

➕ *Add new phone*

➖ Home Address ▼ | 200 First Street |
Street Address

| Somewheresville | CA | 91002 |
City   State   Postal Code

➕ *Add new address*

Let's start off by adding a new address to this record. **REMEMBER:** The API requires a valid address type to create the record.

John Doe
Jane Doe
John Smith
Jill Smith
Bill Rogers
➕ *Add new contact*

Delete   Done

Given Name | John |

Surname | Smith |

➖ Home Phone ▼ | 415-555-1212 |

➖ Mobile Phone ▼ | 415-555-1313 |

➕ *Add new phone*

➖ Home Address ▼ | 200 First Street |
Street Address
| Somewheresville | CA | 91002 |
City   State   Postal Code

➖ Home Address ▼ | |
Street Address
| | | |
City   State   Postal Code

➕ *Add new address*

Notice how the "Add new address" button slides down to make room for the new "Address" record (which starts out blank except for selecting a default address type).

Optionally, animate this transition with CSS3 transitions to help the user understand the change that occurred on the screen.

John Doe
Jane Doe
John Smith
Jill Smith
Bill Rogers
⊕ *Add new contact*

Delete   Done

Given Name | John |

Surname | Smith |

⊖ Home Phone ▼ | 415-555-1212 |

⊖ Mobile Phone ▼ | 415-555-1313 |

⊕ *Add new phone*

⊖ Home Address ▼ | 200 First Street |

Street Address

| Somewheresville | CA | 91002 |

City   State   Postal Code

⊖ Home Address ▼ | |

Street Address

| | | |

City   State   Postal Code

⊕ *Add new address*

John Doe	Delete		Done

John Doe
Jane Doe
John Smith
Jill Smith
Bill Rogers
⊕ *Add new contact*

Given Name

John

Surname

Smith

⊖ Home Phone ▼ 415-555-1212

⊖ Mobile Phone ▼ 415-555-1313

⊕ *Add new phone*

⊖ Home Address ▼ 200 First Street

Street Address

Somewheresville   CA   91002

City   State   Postal Code

⊖ Home Address ▼

Click to change the address type. The list of possible address types comes from the backend API. Check the API reference.

Street Address

City   State   Postal Code

⊕ *Add new address*

John Doe	
Jane Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Done

Given Name   John

Surname   Smith

⊖ Home Phone ▼   415-555-1212

⊖ Mobile Phone ▼   415-555-1313

⊕ *Add new phone*

⊖ Home Address ▼   200 First Street
                    Street Address
                    Somewheresville   CA   91002
                    City   State   Postal Code

⊖ Home Address

Home Address
Mailing Address
Office Address
Work Address
Other Address

Street Address

City   State   Postal Code

John Doe	Delete			Done
Jane Doe				
John Smith				
Jill Smith				
Bill Rogers				
⊕ Add new contact				

Given Name: John

Surname: Smith

⊖ Home Phone ▼ 415-555-1212

⊖ Mobile Phone ▼ 415-555-1313

⊕ Add new phone

⊖ Home Address ▼ 200 First Street
Street Address
Somewheresville | CA | 91002
City | State | Postal Code

⊖ Home Address
Mailing Address
Office Address
Work Address
⊕ Other Address

Street Address

City | State | Postal Code

John Doe
Jane Doe
John Smith
Jill Smith
Bill Rogers
⊕ *Add new contact*

Delete   Done

Given Name   John

Surname   Smith

⊖ Home Phone ▼   415-555-1212

⊖ Mobile Phone ▼ 415-555-1313

⊕ *Add new phone*

⊖ Home Address ▼ 200 First Street
                  Street Address
                  Somewheresville   CA   91002
                  City   State   Postal Code

⊖ Work Address ▼
                  Street Address

                  City   State   Postal Code

⊕ *Add new address*

John Doe	

Jane Doe	

John Smith

Jill Smith

Bill Rogers

⊕ *Add new contact*

Delete   Done

Given Name | John |

Surname | Smith |

⊖ Home Phone ▼ | 415-555-1212 |

⊖ Mobile Phone ▼ | 415-555-1313 |

⊕ Add new phone

⊖ Home Address ▼ | 200 First Street |
Street Address
| Somewheresville | CA | 91002 |
City   State   Postal Code

⊖ Work Address ▼ | 100 Second Ave |
Street Address
| Somewheresville | CA | 91003 |
City   State   Postal Code

⊕ Add new address

John Doe
Jane Doe
John Smith
Jill Smith
Bill Rogers
⊕ *Add new contact*

Delete   Done

Given Name | John

Surname | Smith

⊖ Home Phone   ▼ | 415-555-1212

Complete editing by clicking the "Done" button.

⊖ Mobile Phone ▼ | 415-555-1313

⊕ *Add new phone*

⊖ Home Address ▼ | 200 First Street
Street Address
| Somewheresville | CA | 91002 |
City | State | Postal Code

⊖ Work Address ▼ | 100 Second Ave
Street Address
| Somewheresville | CA | 91003 |
City | State | Postal Code

⊕ *Add new address*

John Doe
Jane Doe
John Smith
Jill Smith
Bill Rogers
⊕ *Add new contact*

Edit

Given Name   John

Surname   Smith

Home Phone   415-555-1212

Mobile Phone   415-555-1313

Home Address   200 First Street

                  Somewheresville, CA 91002

Work Address   100 Second Ave

                  Somewheresville, CA 91003

Our content area once again transforms, but this time back to read-only fields with the new data we just added shown. Optionally, edit this transition with CSS3 transitions.

John Doe	
Jane Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Given Name	John
Surname	Smith
Home Phone	415-555-1212
Mobile Phone	415-555-1313
Home Address	200 First Street
	Somewheresville, CA 91002
Work Address	100 Second Ave
	Somewheresville, CA 91003

John Doe		Edit
Jane Doe		
John Smith		
Jill Smith		
Bill Rogers		
⊕ Add new contact		

Let's switch back to the "Jane Doe" record so we can delete it.

Given Name   John

Surname   Smith

Home Phone   415-555-1212

Mobile Phone   415-555-1313

Home Address   200 First Street

                    Somewheresville, CA 91002

Work Address   100 Second Ave

                    Somewheresville, CA 91003

John Doe	
Jane Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Edit

Given Name   Jane

Surname   Doe

Home Phone   415-555-1414

Mobile Phone   415-555-1515

Home Address   101 Main Street

                    Nowheresville, CA 91000

John Doe	
Jane Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Edit

Given Name   Jane

Surname   Doe

Home Phone   415-555-1414

Mobile Phone   415-555-1515

Home Address   101 Main Street

                    Nowheresville, CA 91000

Our content area now shows the record for "Jane Doe", but no option to delete it.

John Doe	Edit
Jane Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ Add new contact	

Given Name   Jane

Surname   Doe

Home Phone   415-555-1414

Mobile Phone   415-555-1515

Home Address   101 Main Street

                  Nowheresville, CA 91000

Click "Edit" to reveal the "Delete" button for this record.

| John Doe | Delete | | Done |

Jane Doe

John Smith

Jill Smith

Bill Rogers

⊕ Add new contact

| | Given Name | | Jane |

| | Surname | | Doe |

⊖ Home Phone ▼ | 415-555-1414

⊖ Mobile Phone ▼ | 415-555-1515

⊕ Add new phone

⊖ Home Address ▼ | 101 Main Street

Street Address

| Nowheresville | CA | 91000 |

City | State | Postal Code

⊕ Add new address

John Doe	Delete	Done
Jane Doe		
John Smith		
Jill Smith		
Bill Rogers		

⊕ Add new contact

Given Name   Jane

Surname   Doe

⊖ Home Phone ▼   415-555-1414

⊖ Mobile Phone ▼   415-555-1515

⊕ Add new phone

⊖ Home Address ▼   101 Main Street
Street Address

Nowheresville   CA   91000
City   State   Postal Code

⊕ Add new address

John Doe

Jane Doe

John Smith

Jill Smith

Bill Rogers

➕ *Add new contact*

Delete   Done

Given Name   Jane

Surname   Doe

➖ Home Phone   ▼   415-555-1414

➖

➕

➖

Are you sure you want to delete
the selected Contact?

Yes (Delete It)   No (Keep It)

Nowheresville   CA   91000

City   State   Postal Code

ddress

A "modal" dialog opens to confirm the delete. This
dialog prevents access to other functions on the
screen behind it. Clicking on "No" will close the
dialog with no change. Clicking on "Yes" will delete
the selected record.

Optionally use CSS3 transitions when displaying this
dialog.

John Doe	Delete	Done

Jane Doe

John Smith

Jill Smith

Bill Rogers

➕ *Add new contact*

Given Name | Jane

Surname | Doe

➖ Home Phone ▼ | 415-555-1414

➖

Are you sure you want to delete
the selected Contact?

➕

➖

Yes (Delete It)   No (Keep It)

Nowheresville	CA	91000
City	State	Postal Code

➕ *Add new address*

John Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Edit

Given Name   John

Surname   Smith

Home Phone   415-555-1212

Mobile Phone   415-555-1313

Home Address   200 First Street

                    Somewheresville, CA 91002

Notice that the "Jane Doe" record has been removed from our list on the left hand side and that the next record is automatically selected. If we have no "next" record, then the previous record should be selected.

If we are left with no records then only the "Add new contact" button is displayed with a blank details area on the right.

Once the record is deleted we default to showing the newly selected record in read-only mode.

John Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Given Name	John
Surname	Smith
Home Phone	415-555-1212
Mobile Phone	415-555-1313
Home Address	200 First Street
	Somewheresville, CA 91002

John Doe	
John Smith	
Jill Smith	
Bill Rogers	
⊕ *Add new contact*	

Edit

Given Name	John
Surname	Smith
Home Phone	415-555-1212
Mobile Phone	415-555-1313
Home Address	200 First Street
	Somewheresville, CA 91002

Finally, we will click to add a new record.

NOTE: The API **will allow** for an empty given name and surname when creating the contact.

John Doe

John Smith

Jill Smith

Bill Rogers

[No Name]

⊕ *Add new contact*

Delete   Done

Given Name

Surname

⊕ Add new phone

⊕ Add new address

Notice that since there is no given or surname for the record that we default to showing "[No Name]" (in italics) to indicate that no name has been provided.

If the user clicks "Done" without providing a name then it will continue to display "[No Name]"

The new record has no phone numbers or addresses to display and the name fields are blank.

NOTE: That this record is **AUTOMATICALLY** put into "Edit" mode without having to click the "Edit" button. This is because creating a new contact likely means that the user wants to immediately name it and edit it.

John Doe	Delete	Done

John Smith

Jill Smith

Bill Rogers

[No Name]

⊕ *Add new contact*

Given Name

Surname

⊕ *Add new phone*

⊕ *Add new address*

| John Doe | Delete | Done |

John Doe

John Smith

Jill Smith

Bill Rogers

[No Name]

⊕ *Add new contact*

Delete   Done

Given Name [   ]

Surname   [   ]

⊕ *Add new phone*

Click to provide a given name.

⊕ *Add new address*

John Doe
John Smith
Jill Smith
Bill Rogers
Wal
➕ Add new contact

Delete   Done

Given Name | Wal |

Surname | |

➕ *Add new phone*

➕ *Add new address*

As we type (or at least after an on-change event), the name on the left should update to match what has been provided so far.

John Doe	
John Smith	
Jill Smith	
Bill Rogers	
Walter Wil	
⊕ Add new contact	

Delete   Done

Given Name   Walter

Surname   Wil

⊕ Add new phone

⊕ Add new address

As we type (or at least after an on-change event), the name on the left should update to match what has been provided so far.

John Doe	Delete	Done
John Smith		
Jill Smith	Given Name	Walter
Bill Rogers	Surname	Williams
Walter Williams		
⊕ *Add new contact*	⊕ *Add new phone*	
	⊕ *Add new address*	

John Doe
John Smith
Jill Smith
Bill Rogers
Walter Williams
⊕ *Add new contact*

Delete   Done

Given Name   Walter

As before, click "Done" to
leave "Edit" mode.

Surname   Williams

⊕ *Add new phone*

⊕ *Add new address*

John Doe	
John Smith	
Jill Smith	
Bill Rogers	
Walter Williams	

⊕ *Add new contact*

Edit

Given Name   Walter

Surname   Williams