

# View Order History



# View Order History - Front End

- We have the Spring Boot back end developed
- Now let's focus on the Angular front end

The screenshot shows the luv2shop application interface. At the top, there is a blue header bar with the luv2shop logo, a search bar, a 'Search' button, a welcome message 'Welcome back Demo Darby!', and three buttons: 'Logout', 'Member', and 'Orders'. The 'Orders' button is highlighted with a red dashed circle and has a red dashed arrow pointing to it from the left. To the right of the 'Orders' button is a shopping cart icon with '\$0.00' and '0' items. Below the header, on the left, there is a sidebar with links: 'Books', 'Coffee Mugs', 'Mouse Pads', and 'Luggage Tags'. The main content area is titled 'Your Orders' and contains a table with two rows of order data.

Order Tracking Number	Total Price	Total Quantity	Date
51542a04-e16d-4e3e-9549-7e3394558f8d	\$36.98	2	Feb 13, 2021, 10:39:57 AM
0e8014f2-0cd9-4ce4-bb5a-e451154ee9f0	\$17.99	1	Feb 13, 2021, 11:59:58 AM

# Development Process - Angular

Step-By-Step

1. Keep track of logged in user's email with web browser storage
2. Create OrderHistory class
3. Develop OrderHistory service
4. Generate order-history component
5. Update template text in HTML page
6. Add "Orders" button to login-status component
7. Define protected route for order-history component

# Step 1: Keep track of logged in user's email with web browser storage

- We know that we will need to retrieve orders by user's email address
- Once they log in, let's keep track of the user's email address
  - Store the email address in the web browser storage
  - Other Angular components can retrieve it at a later time
- Use Browser **session** storage
  - Data is only available during the current web browser session

# Step 1: Keep track of logged in user's email with web browser storage

File: login-status.component.ts

```
export class LoginStatusComponent implements OnInit {  
  ...  
  storage: Storage = sessionStorage;  
  ...  
  
  getUserDetails() {  
    if (this.isAuthenticated) {  
  
      // Fetch the logged in user details (user's claims)  
      this.oktaAuthService.getUser().then(  
        (res) => {  
          this.userFullName = res.name;  
  
          // retrieve the user's email from authentication response  
          const theEmail = res.email;  
  
          // now store the email in browser storage  
          this.storage.setItem('userEmail', JSON.stringify(theEmail));  
        }  
      );  
    }  
  }  
}
```

Reference to web browser's session storage

Store user's email in web browser's session storage

Key

Value

# Step 2: Create OrderHistory class

```
$ ng generate class common/OrderHistory
```

File: order-history.ts

```
export class OrderHistory {  
    id: string;  
    orderTrackingNumber: string;  
    totalPrice: number;  
    totalQuantity: number;  
    dateCreated: Date;  
}
```

Your Orders			
Order Tracking Number	Total Price	Total Quantity	Date
51542a04-e16d-4e3e-9549-7e3394558f8d	\$36.98	2	Feb 13, 2021, 10:39:57 AM
0e8014f2-0cd9-4ce4-bb5a-e451154ee9f0	\$17.99	1	Feb 13, 2021, 11:59:58 AM

# Step 3: Develop OrderHistory service

```
$ ng generate service services/OrderHistory
```

# Step 3: Develop OrderHistory service

File: order-history.service.ts

```
export class OrderHistoryService {  
  
    private orderUrl = 'http://localhost:8080/api/orders';  
  
    constructor(private httpClient: HttpClient) { }  
  
    getOrderHistory(theEmail: string): Observable<GetResponseOrderHistory> {  
  
        // need to build URL based on the customer email  
        const orderHistoryUrl = `${this.orderUrl}/search/findByCustomerEmail?email=${theEmail}`;  
  
        return this.httpClient.get<GetResponseOrderHistory>(orderHistoryUrl);  
    }  
}  
  
interface GetResponseOrderHistory {  
    _embedded: {  
        orders: OrderHistory[];  
    }  
}
```

Unwraps the JSON from  
Spring Data REST  
\_embedded entry

Call REST API

# Step 4: Generate order-history component

```
$ ng generate component components/OrderHistory
```

# Step 4: Generate order-history component

File:order-history.component.ts

```
export class OrderHistoryComponent implements OnInit {  
  
    orderHistoryList: OrderHistory[] = [];  
    storage: Storage = sessionStorage; Reference to web browser's session storage  
  
    constructor(private orderHistoryService: OrderHistoryService) { }  
  
    ngOnInit(): void {  
        this.handleOrderHistory();  
    } Inject OrderHistoryService  
  
    handleOrderHistory() {  
  
        // read the user's email address from browser storage  
        const theEmail = JSON.parse(this.storage.getItem('userEmail'));  
  
        // retrieve data from the service  
        this.orderHistoryService.get0rderHistory(theEmail).subscribe(  
            data => {  
                this.orderHistoryList = data._embedded.orders;  
            }  
        );  
    } assign data  
}
```

# Step 5: Update template text in HTML page

File: order-history.component.html

```
<h3>Your Orders</h3>

<div *ngIf="orderHistoryList.length > 0">
  <table class="table table-bordered">
    <tr>
      <th width="20%">Order Tracking Number</th>
      <th width="10%">Total Price</th>
      <th width="10%">Total Quantity</th>
      <th width="10%">Date</th>
    </tr>
    <tr *ngFor="let tempOrderHistory of orderHistoryList">
      <td>
        {{ tempOrderHistory.orderTrackingNumber }}
      </td>
      <td>
        {{ tempOrderHistory.totalPrice | currency: 'USD' }}
      </td>
      <td>
        {{ tempOrderHistory.totalQuantity }}
      </td>
      <td>
        {{ tempOrderHistory.dateCreated | date: 'medium' }}
      </td>
    </tr>
  </table>
</div>
```

Your Orders			
Order Tracking Number	Total Price	Total Quantity	Date
51542a04-e16d-4e3e-9549-7e3394558f8d	\$36.98	2	Feb 13, 2021, 10:39:57 AM
0e8014f2-0cd9-4ce4-bb5a-e451154ee9f0	\$17.99	1	Feb 13, 2021, 11:59:58 AM

Loop over  
orderHistoryList

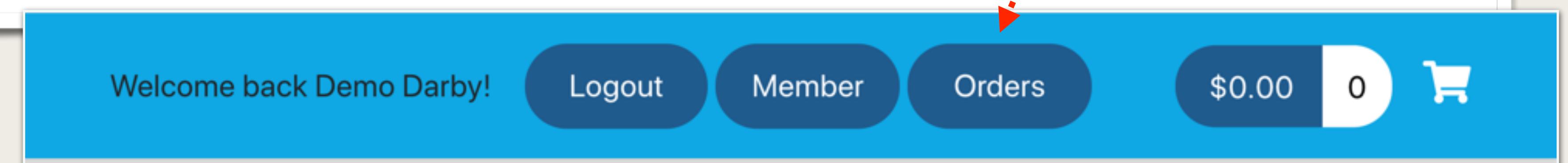
Other date formats:  
short, medium, long, full ...

<https://angular.io/api/common/DatePipe>

# Step 6: Add "Orders" button to login-status component

File: login-status.component.html

```
<div class="login-status-header">  
  
  <div *ngIf="isAuthenticated && userFullName" class="login-status-user-info">  
    Welcome back {{ userFullName }}!  
  </div>  
  ...  
  
  <button *ngIf="isAuthenticated" routerLink="/members" class="security-btn">Member</button>  
  <button *ngIf="isAuthenticated" routerLink="/order-history" class="security-btn">Orders</button>  
</div>
```



# Step 7: Define protected route for order-history component

File: app.module.ts

```
const routes: Routes = [
  {path: 'order-history', component: OrderHistoryComponent, canActivate: [ OktaAuthGuard ]},
  ...
];
```

Route Guard

