

Checkout Form

Populate Credit Card Expiration Dates



Credit Card Expiration Dates

Credit Card

Card Type

Name on Card

Card Number

Security Code

Expiration Month

Expiration Year

Populate Months

Populate Years

The form is a light gray rectangle with a thin black border. It contains seven text input fields and two dropdown menus. The fields are labeled from top to bottom: 'Card Type', 'Name on Card', 'Card Number', 'Security Code', 'Expiration Month', and 'Expiration Year'. The 'Expiration Month' and 'Expiration Year' fields are currently set to 'TO DO'. Two callout bubbles point to these fields: a purple one labeled 'Populate Months' points to the month field, and a pink one labeled 'Populate Years' points to the year field.

Populating Drop-Down Lists

- We could just hard code the values ... but not ideal :-(
- Years
 - Years will become outdated ... if we are in year 2022, why show 2020???
 - Dynamically populate current year up to next 10 years

Development Process

Step-By-Step

1. Generate our form service: Luv2ShopFormService
2. Add methods to the form service for months and years
3. Update checkout component to retrieve the months and years from service
4. Update HTML template to populate drop-down lists for months and years

Step 1: Generate our form service

```
> ng generate service services/Luv2ShopForm
```

Step 2: Add methods to form service for months and years

File: luv2-shop-form.service.ts

```
import { Observable } from 'rxjs';
import { of } from 'rxjs';
...
export class Luv2ShopFormService {
  constructor() { }

  getCreditCardMonths(startMonth: number): Observable<number[]> {
    let data: number[] = [];

    // build an array for "Month" dropdown list
    // - start at desired startMonth and loop until 12
    //
    for (let theMonth = startMonth; theMonth <= 12; theMonth++) {
      data.push(theMonth);
    }
    return of(data);
  }
}
```

Return an
Observable array

rxjs: Reactive JavaScript
<https://angular.io/guide/rx-library>

The "of" operator from rxjs,
will wrap an object as an Observable

Step 2: Add methods to form service for months and years

File: luv2-shop-form.service.ts

```
import { Observable } from 'rxjs';
import { of } from 'rxjs';

export class Luv2ShopFormService {

    ...

    getCreditCardYears(): Observable<number[]> {

        let data: number[] = [];

        // build an array for "Year" dropdown list
        // - start at current year and loop for next 10
        //
        const startYear: number = new Date().getFullYear();
        const endYear: number = startYear + 10;

        for (let theYear = startYear; theYear <= endYear; theYear++) {
            data.push(theYear);
        }

        return of(data);
    }
}
```

Add similar method for Years

Get the current year

The "of" operator from rxjs,
will wrap an object as an Observable

Step 3: Update checkout component to retrieve months and years

File: checkout.component.ts

```
export class CheckoutComponent implements OnInit {  
  ...  
  creditCardYears: number[] = [];  
  creditCardMonths: number[] = [];  
  
  constructor(private formBuilder: FormBuilder,  
              private luv2ShopFormService: Luv2ShopFormService) { }  
  
  ngOnInit(): void {  
    ...  
  
    // populate credit card months  
    const startMonth: number = new Date().getMonth() + 1;  
    console.log("startMonth: " + startMonth);  
  
    this.luv2ShopFormService.getCreditCardMonths(startMonth).subscribe(  
      data => {  
        this.creditCardMonths = data;  
      }  
    );  
  }  
}
```

Inject our form service

Get the current month

JavaScript Date object
the months are zero-based
(0 ... 11)

So we add 1 to get
(1 ... 12)

www.luv2code.com/javascript-date

Step 3: Update checkout component to retrieve months and years

File: checkout.component.ts

```
export class CheckoutComponent implements OnInit {  
  ...  
  creditCardYears: number[] = [];  
  creditCardMonths: number[] = [];  
  
  constructor(private formBuilder: FormBuilder,  
             private luv2ShopFormService: Luv2ShopFormService) { }  
  
  ngOnInit(): void {  
    ...  
  
    // populate credit card years  
    this.luv2ShopFormService.getCreditCardYears().subscribe(  
      data => {  
        console.log("Retrieved credit card years: " + JSON.stringify(data));  
        this.creditCardYears = data;  
      }  
    );  
  }  
}
```

Get the
credit card years

Step 4: Update HTML template to populate drop-down lists

File: checkout.component.html

```
<div class="col-md-2"> <label>Expiration Month</label></div>

<div class="input-space">
  <select formControlName="expirationMonth">
    <option *ngFor="let month of creditCardMonths"
      value={{ month }}>
      {{ month }}
    </option>
  </select>
</div>
```

Loop over array

File: checkout.component.ts

```
export class CheckoutComponent implements OnInit {
  ...
  creditCardMonths: number[] = [];

  constructor(private formBuilder: FormBuilder) { }
```

Step 4: Update HTML template to populate drop-down lists

File: checkout.component.html

```
<div class="col-md-2"> <label>Expiration Year</label></div>

<div class="input-space">
  <select formControlName="expirationYear">
    <option *ngFor="let year of creditCardYears"
      value={{ year }}>
      {{ year }}
    </option>
  </select>
</div>
```

Similar code for
years

File: checkout.component.ts

```
export class CheckoutComponent implements OnInit {
  ...
  creditCardYears: number[] = [];
  ...
}
```