Checkout Form
Populate Credit Card Expiration Dates

Credit Card Expiration Dates

© luv2code LLC

Credit Card Expiration Dates

© luv2code LLC

Credit Card Expiration Dates

© luv2code LLC

Credit Card Expiration Dates

© luv2code LLC

Populating Drop-Down Lists

© luv2code LLC

Populating Drop-Down Lists

- We could just hard code the values ... but not ideal :-(

© luv2code LLC

Populating Drop-Down Lists

- We could just hard code the values ... but not ideal :-(
- Years

© luv2code LLC

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  - Years will become outdated ...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???

© luv2code LLC

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  - Years will become outdated ...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

© luv2code LLC

Development Process

Step-By-Step

© luv2code LLC

Development Process

Step-By-Step

1. Generate our form service: `Luv2ShopFormService`

© luv2code LLC

Development Process

Step-By-Step

1. Generate our form service: `Luv2ShopFormService`
2. Add methods to the form service for months and years

© luv2code LLC

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

© luv2code LLC

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

© luv2code LLC

Step 1: Generate our form service

© luv2code LLC

Step 1: Generate our form service

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


© luv2code LLC

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

File: luv2-shop-form.service.ts

© luv2code LLC

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';

...
```

rxjs: Reactive JavaScript

© luv2code LLC

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() { }
```


© luv2code LLC

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[]> {
```

Return an Observable array

© luv2code LLC

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[] = [];
```

Return an Observable array

© luv2code LLC

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
    //
```

Return an Observable array

© luv2code LLC

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 an Observable array

© luv2code LLC

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

© luv2code LLC

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

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

© luv2code LLC

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

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

© luv2code LLC

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

© luv2code LLC

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

© luv2code LLC

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[]> {









  }

}
```


© luv2code LLC

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[]> {









  }

}
```

Add similar method for Years

© luv2code LLC

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;





  }

}
```

Add similar method for Years

© luv2code LLC

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;





  }

}
```

Add similar method for Years

Get the current year

© luv2code LLC

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;





  }

}
```

Add similar method for Years

Get the current year

Works as expected:

If current year is 2020 will return 2020

© luv2code LLC

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

© luv2code LLC

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

© luv2code LLC

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

© luv2code LLC

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

© luv2code LLC

Step 3: Update checkout component to retrieve months and years

File: checkout.component.ts

© luv2code LLC

Step 3: Update checkout component to retrieve months and years

File: checkout.component.ts

```
export class CheckoutComponent implements OnInit {

  ...
```


© luv2code LLC

Step 3: Update checkout component to retrieve months and years

File: checkout.component.ts

```
export class CheckoutComponent implements OnInit {

  ...
  creditCardYears: number[] = [];
  creditCardMonths: number[] = [];
```


© luv2code LLC

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) { }
```

Inject our form service

© luv2code LLC

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 {
    ...
```

Inject our form service

© luv2code LLC

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);
```

Inject our form service

Get the current month

© luv2code LLC

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);
```

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)

© luv2code LLC

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);
```

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

© luv2code LLC

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 => {
        console.log("Retrieved credit card months: " + JSON.stringify(data));
        this.creditCardMonths = data;
      }
    );
 }

}
```

Inject our form service

Get the current month

Get the credit card months

© luv2code LLC

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









 }

}
```


© luv2code LLC

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

© luv2code LLC

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">



    </select>
</div>
```


© luv2code LLC

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">
        {{ month }}
      </option>
    </select>
</div>
```


© luv2code LLC

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">
        {{ month }}
      </option>
    </select>
</div>
```

Loop over array

File: checkout.component.ts

```
export class CheckoutComponent implements OnInit {

  ...
  creditCardMonths: number[] = [];
```


© luv2code LLC

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">



    </select>
</div>
```


© luv2code LLC

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">
        {{ year }}
      </option>
    </select>
</div>
```


© luv2code LLC

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">
        {{ year }}
      </option>
    </select>
</div>
```

Similar code for years

File: checkout.component.ts

```
export class CheckoutComponent implements OnInit {

  ...
  creditCardYears: number[] = [];
```


© luv2code LLC