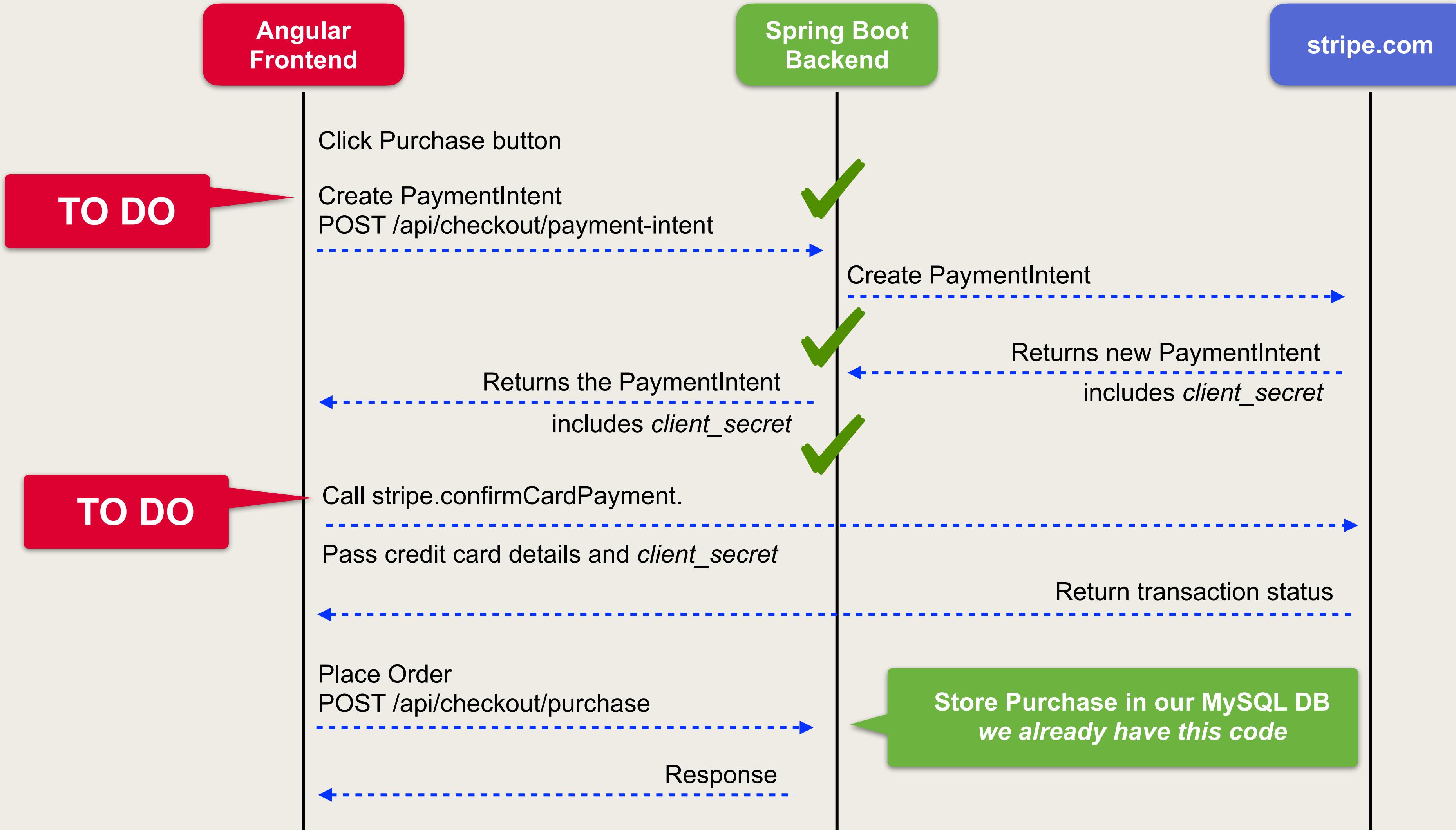


Payment Processing with Stripe

Frontend Development Process



Stripe Payment Processing Flow



Stripe Integration: Custom Code Integration

- Stripe provides **Stripe Elements**
 - Prebuilt UI components for collecting credit card info
 - Components are hosted on the Stripe servers
 - Can easily integrate into your existing / custom checkout form
 - Provide seamless integration
 - Supports PCI ... the credit card data is never sent to your servers

Development Process - Frontend

Step-By-Step

1. Install Stripe API
2. Create a TypeScript declaration file
3. Add Stripe Publishable key
4. Create custom PaymentInfo DTO
5. Update CheckoutService to create a PaymentIntent
6. Load Stripe JavaScript library in index.html
7. Add Stripe Elements to Checkout form

Step 1: Install Stripe API

- Stripe API provides a library for JavaScript development

```
$ npm install stripe
```

- Our Angular app can access the library from TypeScript

Step 2: Create a TypeScript declaration file

- TypeScript declaration file provides type information for JavaScript libraries
- Helps IDEs and compilers to leverage type information of JavaScript libraries

src/typings.d.ts

```
declare var Stripe: any;  
declare var elements: any;
```

<https://angular.io/guide/typescript-configuration>

Step 3: Add Stripe Publishable key

- API keys are available on Stripe Developer Dashboard
- **Publishable key** is used to identify your account with Stripe
- The key isn't secret
- You can safely publish the key in your frontend application

Step 3: Add Stripe Publishable key

environment.ts

```
export const environment = {
  production: false,
  luv2shopApiUrl: "https://localhost:8443/api",
  stripePublishableKey: "pk_test_xxyyzz112233..."
};
```

The screenshot shows the Stripe API keys page. A red arrow points from the 'stripePublishableKey' line in the code above to the 'Publishable key' row in the table below. Another red arrow points from the 'Publishable key' row to the 'TOKEN' column, which contains a long string of characters. This string is also circled with a red dotted line.

NAME	TOKEN	LAST USED	CREATED
Publishable key	pk_test_51IisA8EBpPFwN9Nb1XI0Z63rxW75DAqik dirt3hPCULGfwGvfJVsEt18ChH2xhAdkLn0591XZF cxCfuC3VvTAr300gMJIUMxA	Oct 9	Apr 21
Secret key	Reveal test key	Oct 9	Apr 21

Step 4: Create custom PaymentInfo DTO

- We'll create a Data Transfer Object (DTO)
- Contains info regarding payment: amount and currency

payment-info.ts

```
export class PaymentInfo {  
    amount: number;  
    currency: string;  
}
```

Step 5: Update CheckoutService to create a PaymentIntent

checkout.service.ts

```
export class CheckoutService {  
  
    private paymentIntentUrl = environment.luv2shopApiUrl + '/checkout/payment-intent';  
  
    constructor(private httpClient: HttpClient) { }  
  
    createPaymentIntent(paymentInfo: PaymentInfo): Observable<any> {  
        return this.httpClient.post<PaymentInfo>(this.paymentIntentUrl, paymentInfo);  
    }  
    ...  
}
```

Our custom DTO

Make REST API call

Step 6: Load Stripe JavaScript library in index.html

index.html

```
<!doctype html>
<html lang="en">

<head>
  ...
  <script src="https://js.stripe.com/v3" async></script>
</head>
...
</html>
```

*Script is downloaded in parallel
to parsing the page*