How Klarna Checkout Component Works

The Klarna Checkout component makes use of the aforementioned front end services made available by upscale-provided dependencies; **upscale-web-storefront-sdk** and **caas-service-client-angular**, as well as some provided by **angular** itself, to execute the following:

|  |  |
| --- | --- |
| **Action** | **Relevant services, members** |
| Map the component class to the string matching the value of the custom component's component identifier. This should be done in the constructor of the module for the component library. | |  | | --- | | RegistrationService  register(id: string, type: ClassRef): void | |
| Get the locale currently being used by the PWA. | |  | | --- | | ApplicationLocaleService  get(): Observable<SupportedLocale> | |
| Get the current draft order of the app.  If cart does not exist, navigate to cart. | |  | | --- | | ShoppingCartService  draftOrder: Observable<void | Order<AggregatedOrderLine>>  Router  navigate(commands: any[], extras?: NavigationExtras): Promise<boolean> | |
| Get the application configuration. | |  | | --- | | AppConfigurationService  getAppConfiguration(): Observable<SupportedLocale> | |
| Get the current global consents which have been set by the user. | |  | | --- | | ConsentService  consentsStatus: Observable<CustomerConsentInfo[]> | |
| Patch the order with the current consent configuration. | |  | | --- | | OrdersService  updateOrder(  orderId: string,  request: UpdateOrderRequest,  opts?: AsyncOrderLineView  ): Observable<Order<UnityOrderLine>> | |
| Get the active payment account configurations for the current division, and use the first config with a **gatewayProviderName** which matches the one set on the configuration above. | |  | | --- | | PaymentService  getActiveConfigurationV2(  opts?: {  active?: boolean;  divisionId?: string;  }  ): Observable<ActiveConfiguration[]> | |
| Call initiate with the order id, payment accountId, and information from the current window to initiate the transaction and return the dynamic payment script | |  | | --- | | PaymentService  opfInitiate(initRequest: InitOpenPaymentRequest): Observable<InitOpenPaymentResponse> | |
| Load the JS/CSS resources needed to render the payment gateway and render the dynamic script to the page | |  | | --- | | OpenPaymentService  loadResources(jsResources?: ResourceUrl[], cssResources?: ResourceUrl[]): Observable<void>;  renderHtml(htmlElement: ElementRef<any> | HTMLElement, htmlString?: string): void; | |