



Recurring PSP API Integration Checklist

PSP name: _____

Merchant Serial Number: _____

Name of your solution/platform: _____

Endpoints to integrate	Your Reference	Date of request
Merchant		
Create merchant – POST: /merchant-management/psp/v1/merchants Reference: merchantSerialNumber		
Delete merchant: PATCH: /merchant-management/psp/v1/merchants/{merchantSerialNumber} Reference: merchantSerialNumber		
Agreement		
Create an agreement POST:/recurring/v3/agreements		
Create an agreement with initial charge POST:/recurring/v3/agreements		
Create an agreement with price campaign <i>Must be filled out if feature is used</i> POST:/recurring/v3/agreements		
Create an agreement with period campaign <i>Must be filled out if feature is used</i> POST:/recurring/v3/agreements		
Create an agreement with event campaign <i>Must be filled out if feature is used</i> POST:/recurring/v3/agreements		
Retrieve an agreement GET:/recurring/v3/agreements/{agreementId}		
Update agreement details PATCH:/recurring/v3/agreements/{agreementId}		
Stop an agreement PATCH:/recurring/v3/agreements/{agreementId}		
List charges for an agreement GET:/recurring/v3/agreements/{agreementId}/charges		
Charges		
Create Recurring Charge POST:/recurring/v4/agreements/charges		
Retrieve a charge GET:/recurring/v3/agreements/{agreementId}/charges/{chargeId}		
Get payment info GET:/recurring/v4/agreements/{agreementId}/charges/{chargeId}/payment-info		





Report a charge result POST:/recurring/v4/agreements/{agreementId}/charges/{chargeId}/result		
Cancel a charge DELETE:/recurring/v3/agreements/{agreementId}/charges/{chargeId}		
Refund a charge POST:/recurring/v3/agreements/{agreementId}/charges/{chargeId}/refund		
Capture a reserved charge POST:/recurring/v3/agreements/{agreementId}/charges/{chargeId}/capture		

Quality assurance	Your comments
Proper logging All integrations should log error responses to be used for later reference in troubleshooting. Logs must include endpoint, headers, request body, error code and message. <i>Please supply an example from your logs</i>	
Include standard HTTP headers Send the HTTP headers in all API requests for better tracking and troubleshooting (mandatory for partners and platforms, who must send these headers as part of the checklist approval). <i>Please insert the value for the Vipps system headers</i>	
Handle redirects The merchant <i>must not</i> rely on Redirect from Vipps MobilePay app alone, but must rely on either webhook or poll GET:/recurring/v3/agreements/{agreementId} to check the status of the agreement. The merchant must handle that the returnUrl URL is opened in the default browser on the phone, and not in a specific browser, in a specific tab, in an embedded browser, requiring a session token, etc. Follow our recommendations regarding handling redirects . <i>Please acknowledge that your system does not rely on redirects or session tokens etc.</i>	
Handle errors Make sure to handle all errors . All integrations should display errors in a way that the users (customers and merchant employees/administrators) can see and understand them. <i>Please give examples on how your solution display errors to users</i>	
Charge status All charges must be updated with capture or cancel status, as well as refund if this is performed.	☐ We acknowledge that we must update all payments with correct status
Monitor agreement and charge statuses Merchants must listen to Webhooks to track critical events such as: agreement cancelations, activations of agreements and charge-related updates (e.g. successful or failed charges). This ensures merchants do not request payments on canceled agreements and avoid errors caused by outdated agreement statuses. <i>Please describe your webhook implementation</i>	





Quality assurance	Your comments
Track status with Webhooks and Polling The merchant <i>must not</i> rely on Redirect alone, but must implement the Webhooks API to receive real-time updates on agreement or charge status change. Polling GET:/recurring/v3/agreements/{agreementId} can be used as a fallback for verification. <i>Please describe your redirect implementation</i>	
Provide subscription tools The merchant must have a way for the end-user to manage and stop the subscription in merchantAgreementUrl in their agreement. This should result in a timely update of the Vipps MobilePay Agreement. <i>Please describe the agreement management available for end-users</i>	
Use at least two retryDays The success rate of charges increases significantly when retryDays is set to two or more. <i>Please explain your retry logic. What options do the merchants have to define retryDays</i>	
Merchant onboarding All merchants must be onboarded as individual merchants and not super merchants. All merchants must be onboarded with valid MCC All merchants must use their own merchant/webshop name and logo See PSP Merchant for more details.	☐ We acknowledge that we must onboarding individual merchants ☐ We acknowledge that all merchants have a valid MCC ☐ We acknowledge that all merchants must use their own name and logo
Follow the design guidelines The Vipps MobilePay branding must be according to the design guidelines . <i>Please acknowledge that you have read and used the design guidelines</i>	☐ We acknowledge that we have read and are using the design guidelines
Sign up for operational updates Subscribe to the Vipps MobilePay Operational Status Page for real-time updates on service availability, incidents, and maintenance.	☐ We have registered for the operational updates

Partners, please send your checklists to developer@vippsmobilepay.com. Include example reference IDs from the test environment and a description of the implemented solution.

We will verify the integration and contact you. After the checklist is approved, you'll receive all necessary information from partner@vippsmobilepay.com.

