カート機能に伴う新API群の追加について

本ドキュメントは、カート機能導入に伴うAPIへの影響にて「今後追加予定」としていた以下のAPIの仕様を記載するものです。

本ドキュメントに記載のインターフェース仕様は現時点での設計に基づくものであり、実際に提供するAPIとは一部差異が生じる可能性があります。正式な仕様は提供開始時のAPIスキーマおよびドキュメントをご確認ください。

1. テスト注文用API

1.1 概要

現在提供しているdebugCreateOrder APIは単一商品×1個の注文のみ作成可能です。カート機能の導入に伴い、複数商品・複数個数の注文にも対応した新しいテスト注文APIを追加します。

1.2 変更点

項目	旧API (debugCreateOrder)	新API (debugCreateOrderTransaction)
商品の指定	productId, variantId を各1つ指定	products 配列で複数商品・複数個数を指定
数量	1個固定	quantity フィールドで任意の数量を指定可能
戻り値	Order（旧リソース）	OrderTransaction（新リソース）

既存のdebugCreateOrderはdebugCreateOrderTransactionの提供開始後、一定の移行期間を経て廃止予定です。

1.3 インターフェース

mutation

```
None
mutation {
  debugCreateOrderTransaction(input: DebugCreateOrderTransactionInput!):
DebugCreateOrderTransactionPayload!
}
```

DebugCreateOrderTransactionInput

フィールド名	型	必須	説明
products	[DebugCreateOrderTransactionProductInput!]!	Yes	注文する商品のリスト（1つ以上）
creditCardPaymentMethod	PaymentMethodCreditCardInput	No	クレジットカード支払い情報。Debug APIでは入力不要です
balancePaymentMethod	PaymentMethodBalanceInput	No	残高・ポイント支払い情報

DebugCreateOrderTransactionProductInput

フィールド名	型	必須	説明
productId	String!	Yes	商品ID
variantId	String!	Yes	バリエーションID
quantity	Int!	Yes	数量（1以上）

DebugCreateOrderTransactionPayload

フィールド名	型	説明
orderTransaction	OrderTransaction!	作成されたOrderTransaction

送料割引の適用

ショップに送料割引設定（4. 送料割引設定用APIを参照）がされている場合、注文作成時に割引条件を満たすかが自動的に判定され、条件を満たす場合は割引後の送料が注文に適用されます。

リクエスト例

```
None
mutation {
  debugCreateOrderTransaction(
    input: {
      products: [
        { productId: "product_1", variantId: "variant_1", quantity: 3 }
        { productId: "product_2", variantId: "variant_2", quantity: 1 }
      ]
    }
  ) {
    orderTransaction {
      id
      status
      totalPrice
      products {
        productId
        purchasedQuantity
        unshippedQuantity
      }
    }
  }
}
```

2. OrderTransactionへのフィールド追加

送料割引機能の導入に伴い、OrderTransactionに以下のフィールドが追加されます。

フィールド名	型	説明
unifiedShippingFee	Int!	送料割引が適用された後の送料合計額。注文作成時に確定し、以降変更されません。送料割引が適用されなかった場合や、割引の結果送料が無料になった場合は0となります
refundableUnifiedShippingFee	Int!	現時点で返金可能な送料割引額。初期値はunifiedShippingFeeと同じ値で、割引された送料の返金が発生すると減少します
isPartialCancelable	Boolean!	部分キャンセルが可能かどうか。部分キャンセル可能な場合はtrueになります。詳細は3.3 制約事項を参照

unifiedShippingFeeおよびrefundableUnifiedShippingFeeは、送料割引が適用された場合（4. 送料割引設定用APIのパターンBに該当する場合）にのみ値が設定されます。送料割引が適用されていない場合（パターンA）はいずれも0となり、送料の部分返金はできません。

2.1 フィールドの変化の例

商品A（送料500円）×3個の注文で、送料割引により合計送料が1,000円に割引された場合:

段階	**unifiedShippingFee**	**refundableUnifiedShippingFee**
注文作成時	1,000	1,000
一部商品をキャンセル・送料500円分を返金	1,000	500
さらに送料500円分を返金	1,000	0

unifiedShippingFeeは常に注文作成時の値のまま変わりません。refundableUnifiedShippingFeeは返金のたびに減少し、現時点でさらに返金可能な送料割引額を表します。

なお、cancelOrderTransactionで注文全体をキャンセルした場合は、refundableUnifiedShippingFeeの残額が全て返金されます。

3. 部分キャンセル・返金API

3.1 概要

複数商品の注文において、一部の商品のみをキャンセル・返金できるAPIを追加します。既存のcancelOrderTransactionはキャンセル可能な商品・送料を全てキャンセルするAPIですが、本APIでは商品単位・数量単位でのキャンセルが可能になります。部分キャンセル後に残りの商品をcancelOrderTransactionでまとめてキャンセルすることも可能です。

3.2 インターフェース

mutation

```
None
mutation {
  cancelOrderProducts(input: CancelOrderProductsInput!): CancelOrderProductsPayload!
}
```

CancelOrderProductsInput

フィールド名	型	必須	説明
orderTransactionId	ID!	Yes	キャンセル対象のOrderTransactionのID
idempotencyKey	String!	Yes	重複処理防止のためのキー。OrderTransaction内でユニークである必要があります。リクエストが失敗した場合は同じキーでリトライしてください。リトライ時に異なるキーを指定すると意図せず過剰なキャンセル・返金が発生する可能性があります
cancelReasonType	CancelReasonType!	Yes	キャンセル理由。cancelOrderTransactionと共通のEnum
unifiedShippingFeeRefundAmount	Int!	Yes	送料割引分の返金額。送料割引が適用されている場合に、返金する送料割引額を指定します。送料割引がない場合や返金しない場合は0を指定してください
products	[CancelOrderProductInput!]!	Yes	キャンセル対象の商品リスト（1つ以上）

CancelOrderProductInput

フィールド名	型	必須	説明
productId	String!	Yes	商品ID
variantId	String!	Yes	バリエーションID
quantity	Int!	Yes	キャンセルする数量（1以上）
orderShippingId	ID	Yes	発送済み商品をキャンセルする場合に、対象のOrderShippingのIDを指定します。未発送商品の場合は不要

CancelOrderProductsPayload

フィールド名	型	説明
orderTransaction	OrderTransaction!	キャンセル処理後のOrderTransaction

リクエスト例（未発送商品の部分キャンセル）

```
None
mutation {
  cancelOrderProducts(
    input: {
      orderTransactionId: "order_transaction_1"
      idempotencyKey: "cancel_unshipped_001"
      cancelReasonType: BY_BUYER
      unifiedShippingFeeRefundAmount: 0
      products: [{ productId: "product_1", variantId: "variant_1", quantity: 2 }]
    }
  ) {
    orderTransaction {
      id
      status
      refundableUnifiedShippingFee
      products {
        productId
        purchasedQuantity
        unshippedQuantity
        unshippedCancelingQuantity
      }
    }
  }
}
```

リクエスト例（発送済み商品のキャンセル + 送料返金）

```
None
mutation {
  cancelOrderProducts(
    input: {
      orderTransactionId: "order_transaction_1"
      idempotencyKey: "cancel_shipped_001"
      cancelReasonType: DEFECTED
      unifiedShippingFeeRefundAmount: 500
      products: [
        {
          productId: "product_1"
          variantId: "variant_1"
          quantity: 1
          orderShippingId: "order_shipping_1"
        }
      ]
    }
  ) {
    orderTransaction {
      id
      status
      refundableUnifiedShippingFee
      products {
        productId
        shippedCancelingQuantity
      }
    }
  }
}
```

3.3 制約事項

以下の条件に該当する注文は部分キャンセルができません（OrderTransactionのisPartialCancelableがfalseになります）。この場合、cancelOrderProductsは実行できず、cancelOrderTransactionによる注文全体のキャンセルのみ可能です。

条件	説明
支払い待ち状態	OrderTransactionのステータスがWAITING_FOR_PAYMENTの場合
キャリア決済による購入	キャリア決済で支払われた注文の場合
メルカリ発行クーポンの利用	メルカリが発行したクーポンが使用されている注文の場合
ショップ発行クーポンの部分適用	ショップ発行のクーポンが注文内の一部の商品にのみ適用されている場合（例: 5個購入のうち3個だけクーポンで値引きされている）

4. 送料割引設定用API

4.1 概要

カート購入時に送料割引を適用するための設定APIを追加します。送料割引を設定しない場合、デフォルトで割引はされません。

割引の対象となるのは購入者負担の送料のみです。メルカリ便・**Biz**配送による販売者負担の送料には影響しません。

提供環境について: 本APIは最初はSandbox環境のみで利用可能です。本番環境で実行した場合はエラーとなります。カート機能のリリース後、本番環境でも利用可能になります。

4.2 送料割引の仕組み

送料割引は 送料計算方法（calculationStrategy）と 割引設定（discountStrategy）の2つの要素で構成されます。

送料計算方法（**calculationStrategy**）

カート内の複数商品の送料をどのように計算するかを設定します。

値	説明
EACH_PRODUCT	各商品の購入者負担送料を合算します
MOST_HIGH_FEE	最も高い購入者負担送料のみを適用します

割引設定（**discountStrategy**）

設定は任意です。設定しない場合、calculationStrategyのみで送料が決定されます。

- 商品合計金額: 商品本体価格のクーポン値引き済み金額の合計です
- 閾値金額（thresholdPrice）: 商品合計金額がこの金額以上の場合に割引が適用されます（300〜9,999,999円）
- 割引タイプ: 以下のいずれかを選択します
    - 固定額割引（fixedFee）: 指定した金額を送料から割引します（100〜9,999,999円）
    - 割合割引（percentage）: 送料の指定した割合を割引します（1〜100%）。上限金額（maxDiscountAmount）の指定が必要です（100〜9,999,999円）

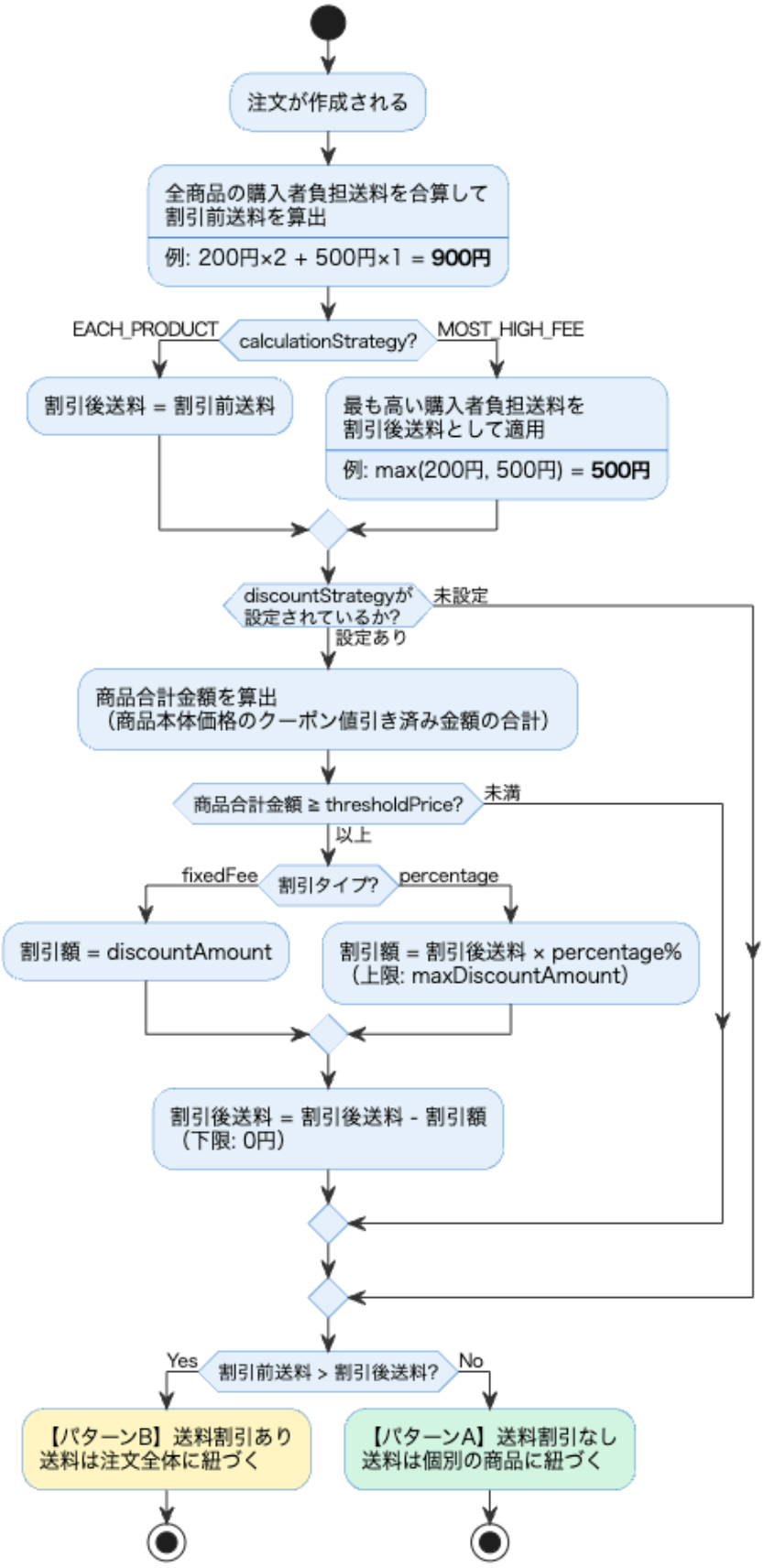
デフォルトの挙動

設定が存在しない場合、以下の設定が適用されます:

- 送料計算方法: EACH_PRODUCT（各商品の送料を合算）
- 割引設定: なし

送料計算フロー

上記の設定に基づき、以下のフローで最終的な送料が決定されます。



送料の記録先と売上計上の違い

送料計算の結果、割引前送料と割引後送料の比較によって以下の2パターンに分かれます。

パターン**A:** 送料が個別の商品に紐づく場合（割引前送料 **=** 割引後送料）

項目	値
orderTransaction.unifiedShippingFee	0
orderTransactionProduct.buyerShippingFee	商品個別の送料が記録される
order.buyerShippingFee	商品個別の送料が記録される
orderShippingProduct.buyerShippingFee	商品個別の送料が記録される
売上計上タイミング	各商品の発送時に、その商品の送料が売上として計上される

パターン**B:** 送料が注文全体に紐づく場合（割引前送料 **>** 割引後送料）

項目	値
orderTransaction.unifiedShippingFee	割引後の送料が記録される
orderTransactionProduct.buyerShippingFee	0（送料が個別商品に紐づかないため
order.buyerShippingFee	0（送料が個別商品に紐づかないため
orderShippingProduct.buyerShippingFee	0（送料が個別商品に紐づかないため
売上計上タイミング	OrderTransactionがCOMPLETEDまたはCANCELEDになった時

4.3 具体例

以下の例では、次の商品を使用します:

- 商品A: 商品価格 1,000円、購入者負担送料 200円
- 商品B: 商品価格 2,000円、購入者負担送料 500円

注文内容: 商品A×2個、商品B×1個（商品合計: 4,000円）

例1: MOST_HIGH_FEE、discountStrategy未設定

項目	表
calculationStrategy	MOST_HIGH_FEE
discountStrategy	未設定
送料計算	max(200円, 500円) = 500円
unifiedShippingFee	500（EACH_PRODUCTでの合算900円に対し、実質的に送料が割引されるため）

例2: EACH_PRODUCT、固定額割引（閾値を超える場合）

項目	値
calculationStrategy	EACH_PRODUCT
thresholdPrice	3,000円
fixedFee.discountAmount	300円
商品合計	4,000円 ≧ 3,000円 → 割引適用
送料計算	(200円×2 + 500円×1) - 300円 = 600円
unifiedShippingFee	600

例3: 固定額割引が送料を超える場合（送料無料）

項目	値
calculationStrategy	EACH_PRODUCT
thresholdPrice	3,000円
fixedFee.discountAmount	2,000円
商品合計	4,000円 ≧ 3,000円 → 割引適用
送料計算	(200円×2 + 500円×1) - 2,000円 = -1,100円 → 0円（下限0円）
unifiedShippingFee	0（割引の結果送料が無料になったため）

4.4 インターフェース

query

```
None
query {
  shippingFeeCalculationConfiguration: ShippingFeeCalculationConfiguration
}
```

自ショップの送料割引設定を取得します。設定が存在しない場合、該当フィールドはnullとなりerrorsにNOT_FOUNDエラーが返されます。

ShippingFeeCalculationConfiguration

フィールド名	型	説明
id	ID!	設定のID
calculationStrategy	ShippingFeeCalculationStrategy!	送料の計算方法
discountStrategy	ShippingFeeDiscountStrategy	割引設定。未設定の場合はnull

ShippingFeeCalculationStrategy（Enum）

型	説明
EACH_PRODUCT	各商品の送料を合算
MOST_HIGH_FEE	最も高い送料のみを適用

ShippingFeeDiscountStrategy

フィールド名	型	説明
thresholdPrice	Int!	割引適用の閾値金額（円）。注文金額がこの金額以上で適用
fixedFee	ShippingFeeFixedFeeDiscount	固定額割引。percentageと排他
percentage	ShippingFeePercentageDiscount	割合割引。fixedFeeと排他

ShippingFeeFixedFeeDiscount

フィールド名	型	説明
discountAmount	Int!	割引額（円）

ShippingFeePercentageDiscount

フィールド名	型	説明
percentage	Int!	割引率（%）。1〜100
maxDiscountAmount	Int!	割引上限額（円）

mutation

```
None
mutation {
  setShippingFeeCalculationConfiguration(input:
SetShippingFeeCalculationConfigurationInput!):
SetShippingFeeCalculationConfigurationPayload!
}
```

送料割引設定を設定します。既に設定が存在する場合は上書きされます。

SetShippingFeeCalculationConfigurationInput

フィールド名	型	必須	説明
calculationStrategy	ShippingFeeCalculationStrategy!	Yes	送料の計算方法
discountStrategy	ShippingFeeDiscountStrategyInput	No	割引設定

ShippingFeeDiscountStrategyInput

フィールド名	型	必須	説明
thresholdPrice	Int!	Yes	割引適用の閾値金額（300〜9,999,999円）
fixedFee	ShippingFeeFixedFeeDiscountInput	No	固定額割引。percentageと排他
percentage	ShippingFeePercentageDiscountInput	No	割合割引。fixedFeeと排他

ShippingFeeFixedFeeDiscountInput

フィールド名	型	必須	説明
discountAmount	Int!	Yes	割引額（100〜9,999,999円）

ShippingFeePercentageDiscountInput

フィールド名	型	必須	説明
percentage	Int!	Yes	割引率（1〜100%）
maxDiscountAmount	Int!	Yes	割引上限額（100〜9,999,999円）

SetShippingFeeCalculationConfigurationPayload

フィールド名	型	説明
shippingFeeCalculationConfiguration	ShippingFeeCalculationConfiguration!	更新後の設定

リクエスト例（固定額割引の設定）

```
None
mutation {
  setShippingFeeCalculationConfiguration(
    input: {
      calculationStrategy: MOST_HIGH_FEE
      discountStrategy: { thresholdPrice: 3000, fixedFee: { discountAmount: 500 } }
    }
  ) {
    shippingFeeCalculationConfiguration {
      id
      calculationStrategy
      discountStrategy {
        thresholdPrice
        fixedFee {
          discountAmount
        }
      }
    }
  }
}
```