

# Mirakl Validations

## Intro ↗

The marketplace(s) happysize is integrated with Mirakl. Mirakl provides products, stocks, prices etc. Our camel calls the mass load API from Mirakl and converst the delivered data to hybris impex-format. Before the data will be transforemer a complex validation is done, to keep invalid data away from shop's production database. Mirakl provides two mass storage API's, one for products and for offers, whoch is a price and a stock.

## Validating products ↗

This validation has the most complex rules to validate the data.

First there are a lot of checks for validating mandatory fields.

Field	Rule	Message
mirakl_product_id	is not blank	product has no productId
data (productdata within the product)	is not null	product with productId %s has no data
internal_product_sku	is not blank	product with productId %s has no productSku
ean	is not empty (list) and first value is not empüty (all other entries will be ignored)	product with productSku %s has no ean
validation	is not null and validation.status is not blank	product with productSku %s has no validation status
acceptance	is not null and acceptance.status is not blank	product with productSku %s has no acceptance status
title	is not blank	product with productSku %s has no title
gender	is not blank	product with productSku %s has no gender
seller	is not blank	product with productSku %s has no seller
modelnumber	is not blank	product with productSku %s has no model number
category_code	is not blank	product with productSku %s has no categoryCode
parentID	is not blank	product with productSku %s has no parentID
size_belt or size_men_single_sizes or size_shoes or size_socks or size_us or size_women_double_sizes	is not blank	product with productSku %s has no size

or size_women_single_sizes or size_women_underwear		
---	--	--

Additional there are a lot checks for each product itself, to check allowed content

Field	Rule	Message
seller and internal_produkt_sku	internal_produkt_sku starts with seller	product with productSku %s has a product sku %s without prefix %s
internal_produkt_sku	matches regex d{4}-d{8,}-d{4}	product with productSku %s has a product sku %s with no match pattern d{4}-d{8,}-d{4}
modelnumber and internal_produkt_sku	internal_produkt_sku contains modelnumber	product with productSku %s has a product sku %s with no match to model number %s
validation.status	status is VALID	product with productSku %s has an invalid validation status %s
acceptance.status	status is ACCEPTED	product with productSku %s has an invalid acceptance status %s
gender	is female or is male	product with productSku %s has no valid gender %s
is season is not blank	is a number	product with productSku %s has an invalid season (not a number) %s
is season is not blank	has 4 digits	product with productSku %s has an invalid season (4 digits) %s
sellerId	is known by a config file	product with productSku %s has no configured seller %s
color	has 3 parts splits by " "	product with productSku %s has no valid color format %s
color	every split is not blank	product with productSku %s has not a correct color value %s, at least one element is empty
color split 1 and internal_produkt_sku splits by "-" second split	second split of internal_produkt_sku and with first split of color	product with productSku %s has no match between model number %s and color code %s
color split 2	is hexadecimal	product with productSku %s has not a hexadecimal value %s
color split 2	has 6 digits	product with productSku %s has not 6 digits %s
size_belt or size_men_single_sizes or size_shoes or size_socks or size_us or size_women_double_sizes or size_women_single_sizes or size_women_underwear	more then 1 size field is filled	product with productSku %s has multiple sizes

used size field	size split by " " has 3 parts	product with productSku %s has an invalid size format %s
used size field	all 3 split are not empty	product with productSku %s has an a complete filled size format %s
used size field firstsplit and internal_produkt_sku	first split of internal_produkt_sku is equal to first color split	product with productSku %s has an a size %s, that is not matched to productSkuCode %s
third split of used size	splits by " " and checks allowed combinations	product with productSku %s has an invalid combination of size types %s
material is not blank	split by " " replaced evry split by \\b\\s+\\b with "_" and \\s with empty string and :{2,} with ":" after that every cleand split will be matched to ^[\\p{L}_-]+:(\\d{1,3}[%\\p{L}_-]+)(,\\d{1,3}[%\\p{L}_-]+)*+\$	product with productSku %s has an invalid material %s

After validating each product on its own we also do some checks on groups. For that we collect all products with the same key build with the seller\_modelnumber .

Field	Rule	Message
parentID	parentID must be same on each product within the group	product with productSku %s has different parentID for modelgroup %s
gender	gender must be same on each product within the group	product with productSku %s has different gender for modelgroup %s
category_code	category must be same on each product within the group	product with productSku %s has different category for modelgroup %s
brand_902	brand must be same on each product within the group	product with productSku %s has different brand for modelgroup %s
color and used size field	at least two products with same size and color	product with productSku %s has a duplicate combination of color and size within modelgroup %s
used size field	two products are using different size fields eg one product uses size_belt and another product uses size_shoes	product with productSku %s uses a different size field within modelgroup %s
internal_produkt_sku	internal_produkt_sku is duplicated	product with miraklProductId %s has the same produktSKU as %. Duplicate produktSKU is %s

## Validating offers

Offers has also to be validated, but their used rules a much simpler then their for products.

Field	Rule	Message
<code>offer_id</code>	is not blank	<code>offer has no offerid</code>
<code>product_sku</code>	is not blank	<code>offer with id {} has no productsku</code>
<code>active</code>	is true	<code>offer with id {} is not active</code>
<code>currency_iso_code</code>	is not blank	<code>offer with id {} has no currency</code>
<code>shop_id</code>	is not blank	<code>offer with id {} has no shop id</code>
<code>quantity</code>	is not null	<code>offer with id {} has no quantity</code>
<code>prices</code>	list is empty	<code>offer with id {} has no prices</code>
<code>last_updated</code>	is not null	<code>offer with id {} has no last update time</code>
<code>origin_price</code>	first element of <code>prices</code> <code>origin_price</code> is null	<code>offer with id {} has no origin price</code>
<code>shop_id</code>	is within the list of configured sellers	
<code>active</code>	is not false	
<code>currency_iso_code</code>	equals EUR	
<code>product_sku</code> and <code>shop_id</code>	<code>product_sku</code> starts with <code>shop_id</code>	<code>offer with id {} has a product sku {} without prefix {}</code>
<code>product_sku</code>	matches <code>d{4}-d{8,}-d{4}</code>	<code>offer with id {} has a product sku {} with no match pattern d{4}-d{8,}-d{4}</code>