verifone

Rulesets

Validation rulesets are conditions you can set to validate transactions to either block or tag them. Rulesets can be used to specify conditions known about the transaction to block them. If a tag is present and made available, similary rulesets can be used to tag a transaction.

Parameters for rulesets

Rulesets can either **block** or **tag**. Rulesets can be built up out of multiple sub rules. The parameters used in the rulesets are:

UI Parameter	API `key`	Description	Requirements
`Account`	`account`	The `account._id`	
`Card`	`card`	The `card._id`	
`Card Issuer's Country`	`issuer_country`	The `card.issuer_country` parameter	A 2-letter ISO3166 alpha-2 country code
`Currency`	`currency_code`	The `account.currency` parameter	ISO 4217 currency code
`Customer`	`customer`	The `customer._id`	
`Customer IP's Country`	`customer_ip_country`	The `transaction.customer_ip` field is used to determine the customer IP's country. This is not a field in the customer model.	A 2-letter ISO3166 alpha-2 country code
`Customer's Country`	`customer_country_code`	The `customer.country_code` parameter	A 2-letter ISO3166 alpha-2 country code
`Customer's IP`	`customer_ip`	The `transaction.customer_ip` parameter. IPv4 and IPv6 can be used.	IPv4 or IPv6
`Organisation`	`organisation`	The `organisation._id`	
`Organisation's Country`	`country_code`	The `organisation.country_code` parameter	A 2-letter ISO3166 alpha-2 country code
`Transaction Amount`	`amount`	The `transaction.amount` parameter	

Operators

Operators can be used to compare the `key` parameter to the `value` parameter. A value can either be compared with any operator or with equal/not equal parameter:

Any operator

Operator	Description
==	Equal to
!=	Not equal to
>	Greater than
>=	Equal to or greater than
<	Less than
<=	Equal to or less than

Equal/not equal

Operator	Description
==	Equal to
!=	Not equal to

Operator per key

UI Parameter	API `key`	Any operator	Equal/not equal
`Account`	`account`		x
`Card`	`card`		x
`Card Issuer's Country`	`issuer_country`		x
`Currency`	`currency_code`		x
`Customer`	`customer`		x
`Customer IP's Country`	`customer_ip_country`		x
`Customer's Country`	`customer_country_code`		x
`Customer's IP`	`customer_ip`		x
`Organisation`	`organisation`		x

UI Parameter	API `key`	Any operator	Equal/not equal
`Organisation's Country`	`country_code`		x
`Transaction Amount`	`amount`	x	

Tagging a transaction using the API

Create the tag

Using a POST request to $BASEURL/v1/tag you can create a tag:

```
{
"organisation":"59de055af3a295574e288fac",
"available":true,
"text":"Suspicious high amount",
"color":"#b95c55"
}
```

- `organisation` - organisations that you wish to create the tag in
- `available` - Indicates wheter the tag is available to be used for rulesets or not
- `text` - The name of the tag
- `color` - HEX color of the tag

Successful response:

```
{
"_id": "1160f232a188fcec3394bb5b",
"organisation": "59de055af3a295574e288fac",
"text": "Suspicious high amount",
"created_at": "2017-10-05T16:35:59.324Z",
"updated_at": "2017-10-05T16:36:23.705Z",
"color": "#b95c55",
"available": true
}
```

- `_id` - ID of the newly created tag
- `organisation` - The organisation that the tag belongs to.

Creating the tagging ruleset

Using a POST request to $BASEURL/v1/validationruleset you can create a ruleset where two rules are combined. This means that a transaction created within the scope of the organisation `59de055af3a295574e288fac` with an amount higher than `551100` and `EUR` currency code will be assigned with the tag created in the previous step:

```
{
"name": "Suspicious high amount",
"organisation": "59de055af3a295574e288fac",
"rules": [
{
"key": "amount",
"operator": ">=",
"value": 551100
},
{
"key": "currency_code",
"operator": "==",
"value": "EUR"
}
],
"action": "tag",
"tag": "1160f232a188fcec3394bb5b"
}
```

- `key` - The key of a rule, for example the amount of the transaction
- `operator` - The operator which operates on the key; for example, an amount "greater than"
- `value` - The value which is used to determine whether the rule takes effect. For example, 551100.
- `action` - The action of the ruleset, and how it affects the transaction. Possible options: tag and block
- `tag` - The Id of the tag used to create a ruleset to tag transactions

If the ruleset is successfully created, the response will be:

```
{
"updated_at": "2017-10-25T09:21:31.822Z",
"created_at": "2017-10-25T09:21:31.822Z",
"name": "Suspicious high amount (Sweden)",
"tag": "1160f232a188fcec3394bb5b",
"organisation": "59de055af3a295574e288fac",
"_id": "59f0579b60efb82d676a2210",
"rules": [
{
"key": "amount",
"operator": ">=",
"value": "551100",
"_id": "59f0579b60efb82d676a2212"
},
{
"key": "currency_code",
"operator": "==",
"value": "EUR",
"_id": "59f0579b60efb82d676a2211"
}
],
"action": "tag"
}
```

Updating tags using the API

Using the tag from the example above, you can update the content of a tag with a POST request to $BASEURL/v1/tag/{id}:

```
{
"text":"New Market (updated)",
"color":"#ffffff",
"available": "true"
}
```

- `text` - Update the text to a new one
- `color` - Update the color of the tag
- `available` - Make the tag available or unavailable

In the response, you will find the updated tag:

```
{
"_id": "1160f232a188fcec3394bb5b",
"organisation": "59de055af3a295574e288fac",
"text": "New Market (updated)",
"created_at": "2017-10-05T16:35:59.324Z",
"updated_at": "2017-10-25T09:31:31.522Z",
"color": "#ffffff",
"available": true
}
```

Blocking transactions

Using the actions part of rulesets one can let a set of conditions block transactions. All transactions matching the conditions will be blocked following the activation of the ruleset.

1. In administration, in rulesets, start creating a new ruleset by providing a name.
2. Start specifying the conditions under which a transaction will be blocked: provide in the rules a key, an operator and a value. For example, country + equals + the Netherlands would target all transactions originating from the Netherlands.
3. Provide more rules as needed.
4. Specify whether to block transactions in the actions section of the ruleset.
5. Saving the ruleset activates it.

Using a POST request to $BASEURL/v1/validationruleset you can create a blocking ruleset:

```
{
"name": "Block high amount",
"rules": [
{
"key": "amount",
"operator": ">=",
"value": "551100"
},
{
"key": "currency_code",
"operator": "==",
```

```
"value": "EUR"
}
],
"action": "block"
}
```

Where in the response the ruleset's content is confirmed:

```
{
"updated_at": "2017-10-25T10:01:07.183Z",
"created_at": "2017-10-25T10:01:07.183Z",
"name": "Block high amount",
"organisation": "59de055af3a295574e288fac",
"_id": "59f060e360efb82d676a36cb",
"rules": [
{
"key": "amount",
"operator": ">=",
"value": "551100",
"_id": "59f060e360efb82d676a36cd"
},
{
"key": "currency_code",
"operator": "==",
"value": "EUR",
"_id": "59f060e360efb82d676a36cc"
}
],
"action": "block"
}
```

Changing the content of rulesets

After creating a ruleset you can remove or add rules to it. You can also have a ruleset change its action, to turn a tagging ruleset into a blocking ruleset, and vice versa.

To update the ruleset using the API, send a POST request to $BASEURL/v1/validationruleset/{id}:

```
{
"name": "Change to tag",
"rules": [
{
"key": "amount",
"operator": ">=",
"value": "551100"
},
{
"key": "currency_code",
"operator": "==",
"value": "EUR"
}
],
"action": "tag",
```

```
"tag": "1160f232a188fcec3394bb5b"
}
```

The response shows, having changed the action to tag and adding the applicable tag, the ruleset is now a tagging ruleset:


```
{
"tag": "1160f232a188fcec3394bb5b",
"_id": "59f060e360efb82d676a36cb",
"updated_at": "2017-10-25T10:07:37.597Z",
"created_at": "2017-10-25T10:01:07.183Z",
"name": "test",
"organisation": "59de055af3a295574e288fac",
"rules": [
{
"key": "amount",
"operator": ">=",
"value": "551100",
"_id": "59f0626960efb82d676a36f1"
},
{
"key": "currency_code",
"operator": "==",
"value": "EUR",
"_id": "59f0626960efb82d676a36f0"
}
],
"action": "tag"
}
```


Deleting rulesets

You can delete a ruleset using the UI or directly with the API. Disabling a ruleset temporarily is not possible.

Requesting a DELETE on $BASEURL/v1/validationruleset/{id} gives the following response when successful:


```
{
"message": "Object 59f060e360efb82d676a36cb deleted."
}
```


The following page describes how to create and edit a ruleset. In order to create, edit or view a ruleset you will need the ProviderAdmin or MerchantAdmin role. More information on that can be found here.

Tagging transaction can be a useful way to view trends or tracking the behavior of specific incoming transactions.

UI Guide

Let's take the example of tracking the behaviour or a customer whose payment behavior your consider risky. You want to review their transactions as they come in.

1. Navigate to the Tags tab in the administration section, create a tag. You can provide a name and a color to easily identify the tagged transactions in the reports page. Use a name that allows you to easily identify it, in this example "Review" or similar.
2. Now navigate to the Rulesets tab in the administration section, start creating a new ruleset by providing a name.
3. Start specifying the conditions under which a transaction will be tagged: provide in the rules a key, an operator and a value. For example, country + equals + the Netherlands would target all transactions originating from the Netherlands.
4. Provide more rules as needed.
5. Specify whether to tag transactions in the actions section of the ruleset. You can then select the tag you wish to tag transactions with. Use the tag we created earlier called "Review".
6. Saving the ruleset activates it.