

LENTA BACKEND

Документация для приложения LENTA BACKEND

Version v1

Security

Basic (HTTP Basic Authentication)

Authenticate

POST /auth/token/login/

auth

Description

Вьюсет получения токена.

Parameters

Name	Located in	Description	Required	Schema
data	body		Yes	<div>↔</div> <div>▼CustomTokenCreate { password: ►Password string email: ►Email string }</div>

Responses

Code	Description	Schema
201		<div>↔</div> <div>▼CustomTokenCreate { password: ►Password string email: ►Email string }</div>

Try this operation

POST /auth/token/logout/

auth

Description

Use this endpoint to logout user (remove user authentication token).

Responses

Code

Description

201

Try this operation

Description

Метод для получения списка товарной иерархии.

Parameters

Name	Located in	Description	Required	Schema
sku	query	<i>sku</i>	No	⇔ string
group	query	<i>group</i>	No	⇔ string
category	query	<i>category</i>	No	⇔ string
subcategory	query	<i>subcategory</i>	No	⇔ string
uom	query	<i>uom</i>	No	⇔ string

Responses

Code	Description	Schema
200		<pre>▼ [⇔ ►Product { }]</pre>

[Try this operation](#)

GET /v1/forecast/

v1

Description

Метод для получения списка прогнозов.

Responses

Code	Description	Schema
200		<div>▼ [</div> <div>↔ ►ForecastGet { }</div> <div>]</div>

Try this operation

Description

Представление для работы с моделью прогноза.

Parameters

Name	Located in	Description	Required	Schema
data	body		Yes	▼ForecastPost {
				store: ▶Store string *
				forecast_date: ▶Дата начала прогноза string *
				forecast: ▶Прогноз на 14 дней { }
}				

Responses

Code	Description	Schema
201		▼ForecastPost {
		store: ▶Store string *
	↔	forecast_date: ▶Дата начала прогноза string *
		forecast: ▶Прогноз на 14 дней { }
		}

Try this operation

GET /v1/forecast/download_file/

v1

Description

Возвращает xls-файл с предсказаниями.

Responses

Code	Description	Schema
200		<div>▼ [</div> <div>↔ ►ForecastGet { }</div> <div>]</div>

Try this operation

Description

Представление модели продаж.

Parameters

Name	Located in	Description	Required	Schema
page	query	A page number within the paginated result set.	No	↔ integer
limit	query	Number of results to return per page.	No	↔ integer

Responses

Code	Description	Schema
200		<div>▼ { count: integer * next: string previous: string results: ► [] }</div>

Try this operation

Description

Представление модели продаж.

Parameters

Name	Located in	Description	Required	Schema
data	body		Yes	<div><div>↔</div><div><div>▼SalesFact {</div><div>store: ▶Store string *</div><div>sku: ▶Sku string *</div><div>fact: ▶[]</div><div>}</div></div></div>

Responses

Code	Description	Schema
201		<div><div>↔</div><div><div>▼SalesFact {</div><div>store: ▶Store string *</div><div>sku: ▶Sku string *</div><div>fact: ▶[]</div><div>}</div></div></div>

Try this operation

Description

Метод для получения списка магазинов.

Parameters

Name	Located in	Description	Required	Schema
store	query	<i>store</i>	No	⇔ string
city	query	<i>city</i>	No	⇔ string
division	query	<i>division</i>	No	⇔ string
type_format	query	<i>type_format</i>	No	⇔ string
loc	query	<i>loc</i>	No	⇔ string
size	query	<i>size</i>	No	⇔ string
is_active	query	<i>is_active</i>	No	⇔ string

Responses

Code	Description	Schema
200		⇔ <div>▼ [► Shops { }]</div>

Try this operation

Description

Вьюсет пользователей.

Parameters

Name	Located in	Description	Required	Schema
limit	query	Number of results to return per page.	No	⇔ integer
offset	query	The initial index from which to return the results.	No	⇔ integer

Responses

Code	Description	Schema
200		<div>▼ { count: integer * next: string previous: string results: ► [] }</div>

Try this operation

Description

Вьюсет пользователей.

Parameters

Name	Located in	Description	Required	Schema
data	body		Yes	↔ ▼ Signup {
				id: ► Id integer username: ► Имя пользователя string * email: ► Почта string * first_name: ► Имя string * last_name: ► Фамилия string * password: ► Password string * }

Responses

Code	Description	Schema
201		▼ Signup {
		id: ► Id integer username: ► Имя пользователя string * email: ► Почта string * first_name: ► Имя string * last_name: ► Фамилия string * password: ► Password string * }

[Try this operation](#)

/v1/users/{id}/

GET /v1/users/{id}/

v1

Description

Вьюсет пользователей.

Parameters

Name	Located in	Description	Required	Schema
id	path	A unique integer value identifying this Пользователь.	Yes	↔ integer

Responses

Code	Description	Schema
200		<div>↔<pre>▼ CustomUser { email: ►Почта string * id: ►Id integer username: ►Имя пользователя string * first_name: ►Имя string * last_name: ►Фамилия string * }</pre></div>

Try this operation

Description

Вьюсет пользователей.

Parameters

Name	Located in	Description	Required	Schema
data	body		Yes	<div>↔<div>▼CustomUser { email: ▶Почта string * id: ▶Id integer username: ▶Имя пользователя string * first_name: ▶Имя string * last_name: ▶Фамилия string * }</div></div>
id	path	A unique integer value identifying this Пользователь.	Yes	↔ integer

Responses

Code	Description	Schema
200		<div>↔<div>▼CustomUser { email: ▶Почта string * id: ▶Id integer username: ▶Имя пользователя string * first_name: ▶Имя string * last_name: ▶Фамилия string * }</div></div>

Try this operation

Description

Вьюсет пользователей.

Parameters

Name	Located in	Description	Required	Schema
data	body		Yes	<div>↔<div>▼CustomUser { email: ▶Почта string * id: ▶Id integer username: ▶Имя пользователя string * first_name: ▶Имя string * last_name: ▶Фамилия string * }</div></div>
id	path	A unique integer value identifying this Пользователь.	Yes	↔ integer

Responses

Code	Description	Schema
200		<div>↔<div>▼CustomUser { email: ▶Почта string * id: ▶Id integer username: ▶Имя пользователя string * first_name: ▶Имя string * last_name: ▶Фамилия string * }</div></div>

Try this operation

DELETE /v1/users/{id}/

v1

Description

Вьюсет пользователей.

Parameters

Name	Located in	Description	Required	Schema
id	path	A unique integer value identifying this Пользователь.	Yes	\Rightarrow integer

Responses

Code	Description
204	

Try this operation

Models

CustomTokenCreate

```
▼ CustomTokenCreate {  
  password: ► Password string  
  email:    ► Email string  
}
```

Product

```
▼ Product {  
  sku:      ► Id товара string *
```

```
    group:      ▶Group string
⇒ category:    ▶Category string
    subcategory: ▶Subcategory string
    uom:        ▶Маркер продажи на вес integer *
  }
```

ForecastGet

```
▼ForecastGet {
  store:      ▶Store string *
  sku:        ▶Sku string *
⇒ forecast_date: ▶Forecast date string *
  forecast:   ▶Forecast { }
}
```

ForecastPost

```
▼ForecastPost {
  store:      ▶Store string *
⇒ forecast_date: ▶Дата начала прогноза string *
  forecast:   ▶Прогноз на 14 дней { }
}
```

SalesGroup

```
▼SalesGroup {
  store: ▶Store string
⇒ sku:   ▶Sku string
  fact:  ▶Fact string
}
```

Fact

```
▼Fact {
```

```
    date:                ▶Date string *
    sales_type:          ▶Sales type integer *
    ↷ sales_units:       ▶Sales units integer *
    sales_units_promo:   ▶Sales units promo integer *
    sales_rub:           ▶Sales rub string *
    sales_run_promo:     ▶Sales run promo string *
  }
```

SalesFact

```
▼SalesFact {
  store: ▶Store string *
  ↷ sku:  ▶Sku string *
  fact:  ▶[]
}
```

Shops

```
▼Shops {
  store:      ▶Id магазина string *
  city:       ▶City string
  division:   ▶Division string
  ↷ type_format: ▶Type format string
  loc:        ▶Loc string
  size:       ▶Size string
  is_active:  ▶Флаг активного магазина integer *
}
```

CustomUser

```
▼CustomUser {
  email:      ▶Почта string *
  id:         ▶Id integer
}
```

```
⇒ username:  ►Имя пользователя string *
  first_name: ►Имя string *
  last_name:  ►Фамилия string *
}
```

Signup

```
▼ Signup {
  id:          ►Id integer
  username:    ►Имя пользователя string *
  email:       ►Почта string *
  ⇒ first_name: ►Имя string *
  last_name:   ►Фамилия string *
  password:    ►Password string *
}
```

Try a sample here

API Design First Workflow

Easily create API specifications

Write API specs in YAML...

...Preview documentation in Swagger