**Интернет-каталог**

Описание

Веб-приложение интернет-каталог, обладающее следующим функционалом: отправка запросов к REST API, вывод ответов на запросы в формате JSON.

Наименование

Интернет-каталог

Предметная область

Разработка прикладного ПО

Данные

В приложении будут использоваться следующие модели данных:

**Products** – модель товаров. Модель включает в себя идентификатор товара, идентификатор категории товара, название (бренд) товара, краткое описание товара.

id = db.Column(db.Integer, primary\_key=True)

category\_id = db.Column(db.Integer, db.ForeignKey('product\_categories.id'), nullable=False)

title = db.Column(db.String(), nullable=False)

description = db.Column(db.String(), nullable=False)

**ProductCategories** – модель категорий товаров. Модель включает в себя идентификатор категории, идентификатор родительской категории, наименование категории.

id = db.Column(db.Integer, primary\_key=True)

parent\_id = db.Column(db.Integer, db.ForeignKey('product\_categories.id'))

title = db.Column(db.String(), nullable=False)

**ProductImages** – модель изображений товаров. Модель включает в себя идентификатор изображения, идентификатор товара, наименование изображения, путь к файлу с изображением.

id = db.Column(db.Integer, primary\_key=True)

product\_id = db.Column(db.Integer, db.ForeignKey('products.id'), nullable=False)

title = db.Column(db.String(), nullable=False)

filename = db.Column(db.String(), nullable=False)

**ProductPrices** – модель цен товаров. Модель включает в себя идентификатор товара, дату создания стоимости, стоимость товара.

product\_id = db.Column(db.Integer, db.ForeignKey('products.id'), nullable=False)

period = db.Column(db.DateTime, nullable=False, default=datetime.utcnow)

price = db.Column(db.Integer, nullable=False)

**ProductCategoryProperties** – модель свойств категорий товаров. Модель включает в себя идентификатор свойства, идентификатор родительского свойства, идентификатор категории товара, название свойства, тип свойства.

id = db.Column(db.Integer, primary\_key=True)

parent\_id = db.Column(db.Integer, db.ForeignKey('product\_category\_properties.id'))

category\_id = db.Column(db.Integer, db.ForeignKey('product\_categories.id'), nullable=False)

title = db.Column(db.String(), nullable=False)

property\_type = db.Column(db.String(), nullable=False)

**PropertyValues** – модель значений свойств категорий товаров. Модель включает в себя идентификатор свойства, идентификатор товара, значение свойства.

property\_id = db.Column(db.Integer, db.ForeignKey('product\_category\_properties.id'), nullable=False)

product\_id = db.Column(db.Integer, db.ForeignKey('products.id'), nullable=False)

value = db.Column(db.String(), nullable=False)

Пользовательские роли

Существует единственнная роль – гостевая.

API приложения

В данном проекте реализован следующий API:

**API метод, возвращающий JSON с полной информацией о товаре по его идентификатору.**

Путь: product-id-route

Метод: GET

Аргументы: 'product\_id' – идентификатор товара

Структура запроса:

{

"product\_id": 16

}

Структура ответа с примером данных:

{

"category": "Jeans",

"description": "bootcut jeans",

"image": [

{

"img\_path": "product\_images/32442432",

"img\_title": "image\_32442432"

}

],

"price": 2294,

"properties": [

{

"property\_title": "size",

"property\_value": "M"

},

{

"property\_title": "country",

"property\_value": "UK"

}

],

"title": "TOM FORD"

}

**API метод, получающий на вход описание товара. Метод создает товар, а затем возвращает JSON с идентификатором товара**.

Путь: product-id-route

Метод: POST

Аргументы: обязательные - "category" (категория товара), "description" (описание товара), "title" (название товара), "price" (цена товара). Необязательные "image" – массив словарей с фотографиями товара, "properties" – массив словарей значений свойств товаров.

Структура запроса с примером данных:

{

"category": "Coats",

"description": "raincoat black",

"image": [

{

"img\_path": "product\_images/324423232",

"img\_title": "image\_3272432"

}

],

"price": 2204,

"properties": [

{

"property\_title": "size",

"property\_value": "M"

},

{

"property\_title": "country",

"property\_value": "UK"

}

],

"title": "TOM FORD"

}

**API метод, возвращающий JSON со списком товаров, удовлетворяющих условиям входных параметров. На вход запроса подается JSON с входными параметрами.**

Путь: products-route

Метод: GET

Аргументы: необязательные - 'product\_category' (категория товаров), 'product\_title' (бренд товаров), 'product\_price\_min' (минимальная цена выборки), 'product\_price\_max' (максимальная цена выборки)

Структура запроса с примером данных:

{

"product\_category": "Jackets",

"product\_title": "Moncler",

"product\_price\_min": 300,

"product\_price\_max": 1000

}

**API метод, возвращающий JSON со списком категорий товаров.**

Путь: product-categories

Метод: GET

Аргументы: нет

Структура ответа с примером данных:

{

"count": 16,

"product\_categories": [

{

"id": 1,

"parent\_id": **null**,

"title": "Clothing"

},

{

"id": 2,

"parent\_id": **null**,

"title": "Shoes"

},

{

"id": 3,

"parent\_id": **null**,

"title": "Bags"

},

{

"id": 4,

"parent\_id": **null**,

"title": "Accessories"

},

{

"id": 5,

"parent\_id": **null**,

"title": "Watches"

},

{

"id": 6,

"parent\_id": 1,

"title": "Jeans"

},

{

"id": 7,

"parent\_id": 1,

"title": "Jackets"

},

{

"id": 8,

"parent\_id": 1,

"title": "Coats"

},

{

"id": 9,

"parent\_id": 1,

"title": "Suits"

},

{

"id": 10,

"parent\_id": 2,

"title": "Boots"

},

{

"id": 11,

"parent\_id": 2,

"title": "Loafers"

},

{

"id": 12,

"parent\_id": 3,

"title": "Backpacks"

},

{

"id": 13,

"parent\_id": 3,

"title": "Tote bags"

},

{

"id": 14,

"parent\_id": 4,

"title": "Belts"

},

{

"id": 15,

"parent\_id": 4,

"title": "Gloves"

},

{

"id": 16,

"parent\_id": 4,

"title": "Hats"

}

]

}

**API метод, создающий новую категорию товаров.**

Путь: product-categories

Метод: POST

Аргументы: обязательные – “title” (название категории), необязательные – “parent\_id” (идентификатор родительской категории).

**API метод, возвращающий JSON со списком изображений.**

Путь: product-images

Метод: GET

Аргументы: нет

Структура ответа с примером данных:

{

"count": 16,

"product\_images": [

{

"filename": "product\_images/product\_1\_image\_1",

"id": 1,

"product\_id": 1,

"title": "product\_1\_image\_1"

},

{

"filename": "product\_images/product\_2\_image\_1",

"id": 2,

"product\_id": 2,

"title": "product\_2\_image\_1"

},

{

"filename": "product\_images/product\_3\_image\_1",

"id": 3,

"product\_id": 3,

"title": "product\_3\_image\_1"

},

{

"filename": "product\_images/product\_4\_image\_1",

"id": 4,

"product\_id": 4,

"title": "product\_4\_image\_1"

},

{

"filename": "product\_images/product\_5\_image\_1",

"id": 5,

"product\_id": 5,

"title": "product\_5\_image\_1"

},

{

"filename": "product\_images/product\_6\_image\_1",

"id": 6,

"product\_id": 6,

"title": "product\_6\_image\_1"

},

{

"filename": "product\_images/product\_7\_image\_1",

"id": 7,

"product\_id": 7,

"title": "product\_7\_image\_1"

},

{

"filename": "product\_images/product\_8\_image\_1",

"id": 8,

"product\_id": 8,

"title": "product\_8\_image\_1"

},

{

"filename": "product\_images/product\_9\_image\_1",

"id": 9,

"product\_id": 9,

"title": "product\_9\_image\_1"

},

{

"filename": "product\_images/product\_10\_image\_1",

"id": 10,

"product\_id": 10,

"title": "product\_10\_image\_1"

},

{

"filename": "product\_images/product\_11\_image\_1",

"id": 11,

"product\_id": 11,

"title": "product\_11\_image\_1"

},

{

"filename": "product\_images/product\_12\_image\_1",

"id": 12,

"product\_id": 12,

"title": "product\_12\_image\_1"

},

{

"filename": "product\_images/product\_13\_image\_1",

"id": 13,

"product\_id": 13,

"title": "product\_13\_image\_1"

},

{

"filename": "product\_images/product\_14\_image\_1",

"id": 14,

"product\_id": 14,

"title": "product\_14\_image\_1"

},

{

"filename": "product\_images/32442432",

"id": 15,

"product\_id": 15,

"title": "image\_32442432"

},

{

"filename": "product\_images/32442432",

"id": 16,

"product\_id": 16,

"title": "image\_32442432"

}

]

}

**API метод для привязки нового изображения к товару**

Путь: product-images

Метод: POST

Аргументы: обязательные – “product\_id” (идентификатор товара), “title” (название изображения), “filename” (путь к файлу с изображением)

Языки программирования и технологии

В проекте использовались следующие языки программирования и технологии:

Python, веб-фреймворк Flask, ORM фреймворк SQLAlchemy, архитектура приложения клиент-сервер.

СУБД

Postgresql.