

Тестовое задание для backend разработчика

Необходимо разработать REST API для сайта “Доска объявлений”

На сайте пользователи выкладывают товары, которые хотели бы продать.

Основные возможности:

- авторизация
- регистрация
- получение/редактирование данных текущего пользователя
- смена пароля (при этом необходимо указать текущий пароль)
- поиск и сортировка товаров
- загружать/удалять изображение для товара
- поиск пользователей
- создание/редактирование/удаление товара авторизованным пользователем

Ниже описана документация по API которую необходимо реализовать.

Идентификация текущего пользователя происходит по сгенерированному токenu, который передается в заголовок Authorization.

Ошибки валидации имеют общий вид:

422, Unprocessable Entity

Body:

```
[  
    {"field": "title", "message": "Title should contain at least 3 characters"},  
    ...  
]
```

field - название поля к которому относится ошибка

message - сообщение об ошибке

Правила валидации Вы выбираете сами.

Сайт должен быть разработан на node.js или php.

Предпочтения:

node.js + (mongo/mysql) + express

php + mysql + (Yii1/Yii2)

API

Login user

Request:

POST /api/login

Body:

```
{  
  "email": "email@example.com",  
  "password": "qwerty"  
}
```

Responses:

200, OK

Body:

```
{  
  "token": "3f5uh29fh3kjhp7tyuioiugfvdf9j8wi6onjf8"  
}
```

422, Unprocessable Entity

Body:

```
[  
  {"field": "password", "message": "Wrong email or password"},  
  ...  
]
```

Register

Request:

POST /api/register

Body:

```
{
  "phone": "+380xxxxxxxx", (optional)
  "name": "Alex",
  "email": "alex@mail.com",
  "password": "qwerty",
}
```

Responses:

200, OK

Body:

```
{
  "token": "3f5uh29fh3kjhpx7tyuioiugfvdf9j8wi6onjf8"
}
```

422, Unprocessable Entity

Body:

```
[
  {
    "field": "current_password",
    "message": "Wrong current password"
  },
  ...
]
```

Get current user

Request:

GET /api/me

Headers:

```
{  
  "Authorization": "3f5uh29fh3kjhp7xyuioiugfvdf9j8wi6onjf8"  
}
```

Responses:

200, OK

Body:

```
{  
  "id": 1,  
  "phone": "+380xxxxxxxx",  
  "name": "Alex",  
  "email": "alex@mail.com"  
}
```

401, Unauthorized

Body: empty

Update current user

Request:

PUT /api/me

Headers:

```
{
  "Authorization": "3f5uh29fh3kjhp7tyuioiugfvdf9j8wi6onjf8"
}
```

Body:

```
{
  "phone": "+380xxxxxxxx", (optional)
  "name": "Alex", (optional)
  "email": "alex@mail.com", (optional)
  "current_password": "qwerty", (optional)
  "new_password": "123456" (required if current_password not empty)
}
```

Responses:

200, OK

Body:

```
{
  "id": 1,
  "phone": "+380xxxxxxxx",
  "name": "Alex",
  "email": "alex@mail.com"
}
```

422, Unprocessable Entity

Body:

```
[
  {
    "field": "current_password",
    "message": "Wrong current password"
  },
  ...
]
```

401, Unauthorized

Body: empty

Get user by ID

Request:

GET /api/user/<id>

Headers:

```
{  
  "Authorization": "3f5uh29fh3kjhp7tyuioiugfvdf9j8wi6onjf8"  
}
```

Responses:

200, OK

Body:

```
{  
  "id": 1,  
  "phone": "+380xxxxxxxx",  
  "name": "Alex",  
  "email": "alex@mail.com"  
}
```

404, Not found

Body: empty

Search users

Request:

GET /api/user?name=Alex&email=alex@mail.com

Params:

name - (optional)

email - (optional)

Responses:

200, OK

Body:

```
[
  {
    "id": 1,
    "phone": "+380xxxxxxxx",
    "name": "Alex",
    "email": "alex@mail.com"
  }
]
```

items

Search items

Request:

GET /api/item?title=notebook&user_id=1&order_by=created_at&order_type=desc

Params:

title - (optional)

user_id - (optional)

order_by - [price|created_at] (optional, default=created_at)

order_type - [asc|desc] (optional, default=desc)

Responses:

200, OK

Body:

```
[
  {
    "id": 1,
    "created_at": <timestamp in seconds>,
    "title": "Notebook",
    "price": 5500.00,
    "image": "http://example.com/images/**/*.*.jpg",
    "user_id": 12,
    "user": {
      "id": 1,
      "phone": "+380xxxxxxxxx",
      "name": "Alex",
      "email": "alex@mail.com"
    }
  },
  ...
]
```

Get item by ID

Request:

GET /api/item/<id>

Responses:

200, OK

Body:

```
{
  "id": 1,
  "created_at": <timestamp in seconds>,
  "title": "Notebook",
  "price": 5500.00,
  "image": "http://example.com/images/**/* .jpg",
  "user_id": 12,
  "user": {
    "id": 1,
    "phone": "+380xxxxxxxx",
    "name": "Alex",
    "email": "alex@mail.com"
  }
}
```

404, Not found

Body: empty

Update item

Request:

PUT /api/item/<id>

Headers:

```
{
  "Authorization": "3f5uh29fh3kjhp7xyuioiugfvdf9j8wi6onjf8"
}
```

Body:

```
{
  "title": "Notebook", (optional)
  "price": 5500.00, (optional)
}
```

Responses:

200, OK

Body:

```
{
  "id": 1,
  "created_at": <timestamp in seconds>,
  "title": "Notebook",
  "price": 5500.00,
  "image": "http://example.com/images/**/* .jpg",
  "user_id": 12,
  "user": {
    "id": 1,
    "phone": "+380xxxxxxxx",
    "name": "Alex",
    "email": "alex@mail.com"
  }
}
```

422, Unprocessable Entity

Body:

```
[
  { "field": "title", "message": "Title should contain at least 3 characters" },
  ...
]
```

404, Not found

Body: empty

403, Forbidden

Body: empty

401, Unauthorized

Body: empty

Delete item

Request:

DELETE /api/item/<id>

Headers:

```
{  
    "Authorization": "3f5uh29fh3kjhp7tyuioiugfvdf9j8wi6onjf8"  
}
```

Responses:

200, OK

Body: empty

404, Not found

Body: empty

403, Forbidden

Body: empty

401, Unauthorized

Body: empty

Create item

Request:

POST /api/item

Headers:

```
{
  "Authorization": "3f5uh29fh3kjhp7xyuioiugfvdf9j8wi6onjf8"
}
```

Body:

```
{
  "title": "Notebook", (required)
  "price": 5500.00, (required)
}
```

Responses:

200, OK

Body:

```
{
  "id": 1,
  "created_at": <timestamp in seconds>,
  "title": "Notebook",
  "price": 5500.00,
  "image": "http://example.com/images/**/* .jpg"
  "user_id": 12,
  "user": {
    "id": 1,
    "phone": "+380xxxxxxxxx",
    "name": "Alex",
    "email": "alex@mail.com"
  }
}
```

422, Unprocessable Entity

Body:

```
[
  { "field": "title", "message": "Title is required" },
  ...
]
```

]

403, Forbidden

Body: empty

401, Unauthorized

Body: empty

Upload item image

Request:

POST /api/item/<id>/image

Headers:

```
{  
    "Authorization": "3f5uh29fh3kjhp7tyuioiugfvdf9j8wi6onjf8"  
}
```

Content type: multipart/form-data

Body:

file=<file>

Responses:

200, OK

Body:

```
{  
    "id": 1,  
    "created_at": <timestamp in seconds>,  
    "title": "Notebook",  
    "price": 5500.00,  
    "image": "http://example.com/images/**/* .jpg",  
    "user_id": 12,  
    "user": {  
        "id": 1,  
        "phone": "+380xxxxxxxxx",  
        "name": "Alex",  
        "email": "alex@mail.com"  
    }  
}
```

422, Unprocessable Entity

Body:

```
[  
    {"field": "image", "message": "The file {file} is too big. "},  
    ...  
]
```

403, Forbidden

Body: empty

401, Unauthorized

Body: empty

Remove item image

Request:

DELETE /api/item/<id>/image

Headers:

```
{  
    "Authorization": "3f5uh29fh3kjhp7tyuioiugfvdf9j8wi6onjf8"  
}
```

Responses:

200, OK

Body: empty

404, Not found

Body: empty

403, Forbidden

Body: empty

401, Unauthorized

Body: empty