\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Необходимо разработать REST API для сайта “Доска объявлений”

На сайте пользователи выкладывают товары, которые хотели бы продать. Основные возможности:

Основные возможности:

- авторизация

- получение/редактирование данных текущего пользователя

- поиск и сортировка товаров

- загружать/удалять изображение для товара

- создание/редактирование/удаление товара авторизованным пользователем

Пользователь авторизуется, используя email + password. В случае, если пользователь вводит email, которого нет в системе, новый пользователь с указанными email и password должен быть создан, а пользователь — авторизован. В дальнейшем пользователь может авторизоваться с указанными данными под тем же профилем.

Ниже описана документация по API которую необходимо реализовать. Идентификация текущего пользователя происходит по сгенерированному токену, 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](mailto:email@example.com)”,

“password”: “qwerty”

}

### Responses:

#### 200, OK

Body:

{

“token”: “3f5uh29fh3kjhpx7tyuioiugfvdfr9j8wi6onjf8”

}

#### 422, Unprocessable Entity

Body:

[

{"field":"password","message":"Wrong email or password"},

…

]



## 

## Get current user

### Request:

**GET /api/me**

Headers:

{

“Authorization”: “3f5uh29fh3kjhpx7tyuioiugfvdfr9j8wi6onjf8”

}

### Responses:

#### 200, OK

Body:

{

“id”: 1,

“phone”: “+380xxxxxxxxx”,

“name”: “Alex”,

“email”: “alex@mail.com”

}

#### 401, Unauthorized

Body: empty



## 

## Update current user

### Request:

**PATCH /api/me**

Headers:

{

“Authorization”: “3f5uh29fh3kjhpx7tyuioiugfvdfr9j8wi6onjf8”

}

Body:

{

“phone”: “+380xxxxxxxxx”, (optional)

“name”: “Alex”, (optional)

“email”: “alex@mail.com”, (optional)

}

### Responses:

#### 200, OK

Body:

{

“id”: 1,

“phone”: “+380xxxxxxxxx”,

“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”: “3f5uh29fh3kjhpx7tyuioiugfvdfr9j8wi6onjf8”

}

### Responses:

#### 200, OK

Body:

{

“id”: 1,

“phone”: “+380xxxxxxxxx”,

“name”: “Alex”,

“email”: “alex@mail.com”

}

#### 404, Not found

Body: empty



## 

## Search items

### Request:

**GET /api/item?title=book&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,

"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": 550000,

"image": "http://example.com/images/\*\*/\*.jpg",

"user\_id": 12,

"user": {

"id": 1,

"phone": "+380xxxxxxxxx",

"name": "Alex",

"email": "alex@mail.com"

}

}

#### 404, Not found

Body: empty



## 

## Update item

### Request:

**PATCH/api/item/<id>**

Headers:

{

“Authorization”: “3f5uh29fh3kjhpx7tyuioiugfvdfr9j8wi6onjf8”

}

Body:

{

“title”: “Notebook”, (optional)

“price”: 5500.00, (optional)

}

### Responses:

#### **200, OK**

Body:

{

"id": 1,

"created\_at": "<timestamp in seconds>",

"title": "Notebook",

"price": 5500,

"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 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”: “3f5uh29fh3kjhpx7tyuioiugfvdfr9j8wi6onjf8”

}

### Responses:

#### 204, No content

Body: empty

#### 404, Not found

Body: empty

#### 403, Forbidden

Body: empty

#### 401, Unauthorized

Body: empty



## 

## Create item

### Request:

**POST /api/items**

Headers:

{

“Authorization”: “3f5uh29fh3kjhpx7tyuioiugfvdfr9j8wi6onjf8”

}

Body:

{

“title”: “Notebook”, (required)

“price”: 5500.00, (required)

}

### Responses:

#### 201, Created

Body:

{

“id”: 1,

“created\_at”: <timestamp in seconds>,

“title”: “Notebook”,

“price”: 5500.00,

“image”: [“ht](http://example.com/images/)t[p://example.com/images/](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"},

...

]

#### 404, Not found

Body: empty

#### 403, Forbidden

Body: empty

#### 401, Unauthorized

Body: empty



## 

## Upload item image

### Request:

**PUT /api/item/<id>/image**

Headers:

{

“Authorization”: “3f5uh29fh3kjhpx7tyuioiugfvdfr9j8wi6onjf8”

}

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”: [“ht](http://example.com/images/)t[p://example.com/images/](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. "},

...

]

#### 404, Not found

Body: empty

#### 403, Forbidden

Body: empty

#### 401, Unauthorized

Body: empty



## 

## Remove item image

### Request:

**DELETE /api/item/<id>/image**

Headers:

{

“Authorization”: “3f5uh29fh3kjhpx7tyuioiugfvdfr9j8wi6onjf8”

}

### Responses:

#### 204, No content

Body: empty

#### 404, Not found

Body: empty

#### 403, Forbidden

Body: empty

#### 401, Unauthorized

Body: empty

**Успехов!**