# Задача

* Реализовать методы API указанные в спецификации ниже
* Использовать фреймворк Laravel
* Использовать принципы ООП (объектно ориентированное программирование) и DDD (domain driven design)
* Использовать приложенный дамп таблицы, заполнить ее данными самостоятельно, БД - mysql
* Разместить данные в любом облачном репозитории
* **Цель задачи** - продемонстрировать знание PHP, умение проектировать программные решения и работать с документацией

# Типы данных

В документации используются специальные обозначения для передаваемых и возвращаемых типов данных

### **Число**

Строка числовых символов для urlencoded, int для JSON

Пример

25000

# 

### **Строка**

Строка символов длиной до 512 для urlencoded, string для JSON

Пример

"Какая-то строка"

# 

### **Текст**

Строка символов неограниченной длины для urlencoded, string для JSON

Пример

"В дальнейшем, если возникнет необходимость вывода списка пользователей вместе с их бонусными картами, ресурс Бонусной карты нужно будет сделать вложенным в ресурс Пользователя"

### **Дата**

Строка в формате ‘YYYY-MM-DD’, где ‘DD’, ‘MM’, ‘YYYY’ - соответственно день, месяц и год с лидирующим нулем, ‘-’ - разделитель

Пример

*"*2017-06-23"

# 

### **Массив**

Строка из других типов данных, разделенных символом запятой для urlencoded, array для JSON

Пример

["green", "Red", "yellow"]

[25423,45214,122]

### **Ресурс**

Объект в формате JSON, описывающий некоторую сущность с которой работает сайт

Подробное описание конкретных ресурсов используемых в проекте см. в отдельной главе ниже

Пример

{

"ids": [12355, 15524],

"title": "OLED",

"count": 4,

}

# Запрос к серверному приложению

Запросы к серверному приложению отправляются по HTTP протоколу. Передача параметров осуществляется в формате urlencoded

Параметры, отсутствующие в спецификации API игнорируются приложением

Для всех запросов запрещено передавать в параметрах любые неграфические символы, за исключением знаков переноса строки либо пробелов. Запрещены строки, состоящие только из пробелов и знаков переноса строки

# Ответ серверного приложения

Серверное приложение возвращает данные в формате JSON. Стандартная структура ответа серверного приложения выглядит следующим образом:

{

“status”: “Success”,

“message”: “Успешно”,

“data”: {}

}

где

*status* - строка, определяющая общий результат выполнения конкретного метода

*message* - строка, читаемое сообщение, описывающее результат выполнения метода для пользователя API

*data* - объект, полезная нагрузка ответа серверного приложения

В случаях, когда данные в поле ресурса отсутствуют, такие поля приложению не передаются

# Сообщения об ошибках

Существует базовый набор ошибок, которые передает приложение согласно http-стандарту

### **Ошибка валидации**

В случае, если метод, вызываемый приложением, требует аргумент, который приложение предоставило в неверном формате, будет возвращен ответ с HTTP - кодом **400 (Bad Request)**. При этом в поле field будет возвращено одно из полей и код правила валидации

|  |
| --- |
| {  "status": "FieldInvalid",  "message": "Поле содержит недопустимое значение",  "data": {  "field": "page",  "rule": "Required"  }  } |

где

*field* - название аргумента, значение которого некорректно,

*rule* - название правила, по которому не прошла проверка на валидность

### **Запись не найдена**

В случае, если метод, вызываемый приложением, прекратил свою работу из-за того, что не была найдена запись в базе данных, необходимая для отображения запрашиваемого ресурса, то будет возвращен ответ с HTTP - кодом **404 (Not Found)**

{

“status”: “RecordNotFound”,

“message”: “Запись не найдена”

“data”: {}

}

### **Метод не найден**

В случае, если приложение попробует обратиться по несуществующему URL, то будет возвращен ответ с HTTP - кодом **404 (Not Found)** вида

{

“status”: “UrlNotFound”,

“message”: “URL не найден”,

“data”: {}

}

### **Ошибка сервера**

В случае, если на стороне сервера произошла ошибка, причиной которой является сбой работы серверного приложения, возвращается ответ с HTTP - кодом **500 (Internal Server Error)**

{

“status”: “GeneralInternalError”,

“message”: “Произошла ошибка”,

“data”: {}

}

# 

# Ресурсы

Ресурс - это некоторый набор данных, который передается для отображения на фронт

Ресурсы делятся на типы в зависимости от набора полей, которые они содержат

Существуют следующие типы ресурсов

* **Полный (full)** - полный набор данных
* **Мини (mini)** - минимальный набор данных

### 

### **Новость - news**

### 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | Формат | Обязательное? | Описание | Пример | Тип ресурса | |
| **slug** | Строка |  | Человекопонятный идентификатор новости | "slug": "sdelayte-svoy-vybor-cash-back-ili-rassrochka" | Полный | Мини |
| **title** | Строка |  | Заголовок новости | "title": "Как оформить онлайн-кредит" | Полный | Мини |
| **shortText** | Текст |  | Короткое описание новости | "shortText": "Делать покупки просто!" | Полный | Мини |
| **text** | Текст |  | Полное содержимое новости | "text": "Делать покупки через интернет-магазине «Домотехника» легко и просто!" | Полный |  |
| **datePublish** | Дата |  | Дата публикации новости | "datePublish": "2051-15-15" | Полный | Мини |
| **image** | Строка | Нет | Обложка новости | "image": "http://cdn.domotekhnika.ru/images/news/  000/000/439/70f03de673a33ba48d8df5-original.png" | Полный | Мини |
| Пример ресурса | | | | | Описание | |
| {  "slug": "sdelayte-svoy-vybor-cash-back-ili-rassrochka",  "title": "Как оформить онлайн-кредит",  "shortText": "Делать покупки через интернет-магазин «Домотехника» легко и просто!",  "datePublish": "2051-15-15",  "image": "http://cdn.domotekhnika.ru/images/news/000/000/439/70f03de673a33ba48d8df5-original.png",  } | | | | | Мини ресурс Новости | |

### 

# Методы

### **Получить список новостей - GET /api/v1/news**

Возвращает список новостей постранично

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **GET** | **/api/v1/news** | | | | |
| *Аутентификация* | *Нет* | | | | |
| *Авторизация* | *Нет* | | | | |
| *Кэширование* | *Нет* | | | | |
| Параметр | Body/Query/Path | Тип | Обязательное? | Описание | Пример |
| **page** | Query | Число |  | Номер страницы  *По умолчанию 1* | page: 10 |
| **limit** | Query | Число |  | Кол-во элементов на одной странице  *По умолчанию 20* | limit: 5 |
| Ответ | | | | Описание | |
| HTTP - код: **200**  {  "status": "Success",  "message": "Успешно",  "data": {  "news”: [...],  "pageCount": 12,  }  } | | | | По ключу *news* возвращается Массив Мини ресурсов Новости, содержащихся на переданной странице  По ключу *pageCount* возвращается общее количество страниц | |
| Пример запроса | | | | Описание | |
| curl http://dt.ru/api/v1/news?page=10&limit=5 | | | |  | |

### **Получить новость - GET /api/v1/news/{slug}**

Возвращает отдельную новость

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **GET** | **/api/v1/news/{slug}** | | | | |
| *Аутентификация* | *Нет* | | | | |
| *Авторизация* | *Нет* | | | | |
| *Кэширование* | *Нет* | | | | |
| Параметр | Body/Query/Path | Тип | Обязательное? | Описание | Пример |
| **slug** | Path | Строка | Да | Человекопонятный идентификатор новости | slug: "vot-eto-cashback-vernem-do-50-s-electrolux" |
| Ответ | | | | Описание | |
| HTTP - код: **200**  {  "status": "Success",  "message": "Успешно",  "data": {  "news”: {...},  }  } | | | | По ключу *news* возвращается Полный ресурс Новости | |
| [Пример запроса](http://team.centerdt.ru:8090/display/EN/API#API-%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5SEO-seo) | | | | [Описание](http://team.centerdt.ru:8090/display/EN/API#API-%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5SEO-seo) | |
| [curl http://dt.ru/api/v1/news/some-slug](http://team.centerdt.ru:8090/display/EN/API#API-%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5SEO-seo) | | | |  | |