

Тестовое задание на React

Веб сервис для онлайн продажи подарочных сертификатов на React на REST API.

Пример как это может выглядеть:

<https://sycret.ru/service/onsales/?apikey=011ba11bdcad4fa396660c2ec447ef14>

Пример не самый удачный, приведен только для наглядного понимания сути задачи.

Предполагаем, что Вы поймете недостатки и сделаете лучше.

Сценарий работы сервиса:

1. Первая страница: выбор сертификата.

Выводит список сертификатов. Список сертификатов получать при помощи метода OSGetGoodList. Описание методов ниже.

Пользователь просматривает список и выбирает сертификат, по кнопке «Оформить» попадает на страницу сбора контактов.

2. Вторая страница: заполнение контактов.

Поля формы:

«Имя», обязательное

«Телефон», по маске телефона, обязательное

«Почта», обязательное

Сервис проверяет правильность заполнения полей, доступны кнопки «Назад» (обратно на страницу выбора сертификата) и «Оплатить».

Сервис сохраняет собранные данные с помощью метода OSSale.

3. В рамках данного задания кнопка «Оплатить» ведет на страницу-заглушку с текстом «оплата...», страница нужна просто для понимания, что пользователь успешно дошел до оплаты.

Требования:

1. Язык разработки JavaScript, фреймворк React.

2. Разрешено использовать JSX, не влияет на оценку.

3. Адаптивная верстка (результат адекватно выглядит и на мобильном, и на десктопе).

4. Проект нужно выложить на github и прислать ссылку

5. Сборка webpack без глобальных зависимостей (все зависимости должны быть установлены локально в папку сборки, конфиг файл внутри той же папки).

6. Предоставить возможность увидеть проект в работе. На выбор: 1) мы скопировали сборку в папку на нашем сервере (сервер под управлением nginx без дополнительных компонент/библиотек) пусть это будет <https://sycret.ru/папка/>, через браузер открыли URL папки (<https://sycret.ru/папка/>), увидели проект в работе или 2) выложить на свой сервер и прислать URL как посмотреть в работе через браузер).

7. Время, отведенное на выполнение: до 7 дней.

Приложение. Описание используемых в задании методов REST API

<https://sycret.ru/service/api/api>

Реализуется концепция REST API на JSON по протоколу HTTPS.
APIKey для данного задания: 011ba11bdcad4fa396660c2ec447ef14

URL для вызова всех методов один и тот же, название указывается метода как один из входных параметров. Входные параметры всегда содержат ApiKey (указан выше), MethodName (название самого метода), остальные входные параметры в зависимости от метода. Пример:

```
{
  "ApiKey": "011ba11bdcad4fa396660c2ec447ef14",
  "MethodName": "TestMethodName",
  "InputParamName1": "Value1",
  ...
  "InputParamNameN": "ValueN"
}
```

Выходные параметры всегда содержат result (число, «0» если без ошибок), result description (строка, «OK» если без ошибок) и data (сами данные), пример:

```
{
  "data": [
    {
      "OutputParamName1": "OutputParamValue1",
      "OutputParamName2": "OutputParamValue2",
      "OutputParamName3": "OutputParamValue3"
    },
    {
      "OutputParamName1": "OutputParamValue1",
      "OutputParamName2": "OutputParamValue2",
      "OutputParamName3": "OutputParamValue3"
    }
  ],
  "result": 0,
  "resultdescription": "OK"
}
```

Методы API

OSGetGoodList

Получение списка товаров.

В данном задании используются поля Id, , остальные не используются.

Имя параметра	Тип	Nullable	Направление	Описание
ApiKey	Строка	-	in	Ключ авторизации (получаемый у разработчика api)
MethodName	Строка	-	in	Название метода (в данном случае OSGetGoodList)
Data	Список	-	out	Список параметров

Содержание data:

Имя параметра	Тип	Nullable	Направление	Описание
Id	Целое	-	out	Id конкретного товара (курс, товар, сертификат) – используется
TableName	Строка	-	out	Тип товара (курс, товар, сертификат), нужен только для передачи в следующий метод
PrimaryKey	Строка	-	out	Generic_key из Id и TableName, нужен только для передачи в следующий метод
Name	Строка	-	out	Название товара – используется
Description	Строка	+	out	Описание товара – используется
Price	Дробное	-	out	Описание товара – используется
Summa	Дробное	-	out	Сумма к оплате – используется
Discount	Дробное	+	out	Скидка – используется
ImageUrl	Строка	+	out	Ссылка на фото товара (временно не используется)
Rec_SNO	Строка	-	out	Параметры для оплаты через шлюз, в данном задании не используются
Rec_Name	Строка	-	out	
Rec_Sum	Дробное	-	out	
Rec_Quantity	Дробное	-	out	
Rec_Payment_Method	Строка	-	out	

Rec_Payment_Object	Строка	-	out	
Rec_Tax	Строка	-	out	

OSSale

Сохранение данных о выбранном сертификате на сервере

Имя параметра	Тип	Nullable	Направление	Описание
ApiKey	Строка	-	in	Ключ авторизации (получаемый у разработчика api)
MethodName	Строка	-	in	Название метода (в данном случае OSSale)
Id	Целое	-	in	Id конкретного товара полученный из OSGetGoodList, выбранный пользователем ID
TableName	Строка	-	in	Тип товара полученный из OSGetGoodList, просто передать полученное ранее
PrimaryKey	Строка	-	in	Generic_key из Id и TableName полученный из OSGetGoodList, просто передать полученное ранее
Price	Дробное	-	in	Цена без скидки полученная из OSGetGoodList
Summa	Дробное	-	in	Сумма к оплате полученная из OSGetGoodList
ClientName	Строка	-	in	Имя клиента
Phone	Строка	-	in	Телефон клиента в формате «9161237589»
Email	Строка	-	in	Электронная почта клиента
PaymentTypeld	Целое	-	in	Вид оплаты (всегда равен 2)
UseDelivery	Целое	-	in	Пометка для продаж с доставкой, в данном задании всегда 0 – нет
DeliveryAddress	Строка	+	in	Адрес доставки (использовать, если UseDelivery = 1), в данном задании не используется
IsGift	Целое	+	in	Продажа в подарок? 0 – по умолчанию
MsgText	Строка	+	in	Текст сообщения для клиента
PName	Строка	+	in	Имя плательщика (если покупает один клиент для другого)
PPhone	Строка	+	in	Телефон плательщика (если покупает один клиент для другого)
Data	Список	-	out	Список параметров

Содержание data:

Имя параметра	Тип	Nullable	Направление	Описание
CertNumber	Строка	+	out	Id товара (в текущей версии идентичен id)