

Техническое задание

Сайт для интернет-сервиса аренды автомобилей

Техническое задание	1
Предмет разработки	3
Шаблон верстки	3
Описание API	3
Подписка на рассылку	4
Авторизация	4
Получение профиля	5
Редактирование профиля	5
Получение списка привязанных карт	5
Привязка / удаление карты	6
История заказов	6
Восстановление пароля - активация	7
Общие требования	7
Стили	7
Валидация форм	7
Программная структура	7
Перезагрузка страниц	7
SEO-оптимизация	8
Мобильная версия сайта	8
Контент пустых страниц	8
Структура интернет-сайта	8
Внешняя страница описания сервиса (Лэндинг)	8
Страница авторизации	8
Личный кабинет авторизованного пользователя	9
Профиль пользователя	10
Форма привязки / удаления банковской карты	11
Тарифы и правила	11
Цепочка страниц восстановления пароля	11
Цепочка страниц регистрации	12
Статические информационные страницы	12
Страницы уведомления об ошибках	12

Предмет разработки

Предметом разработки является создание интернет-сайта сервиса аренды автомобилей.

От Исполнителя требуются выполнение следующих работ:

1. Компоновка HTML-страниц в соответствии с ТЗ на основе предоставленного заказчиком универсального шаблона верстки, удовлетворяющего спецификации Twitter Bootstrap.
2. Создание программного кода на языке JavaScript (или CoffeeScript - по выбору исполнителя), реализующего на подготовленных исполнителем HTML-страницах интерактивные функции по взаимодействию с существующим API сервиса (спецификация API предоставляется Заказчиком).

Перед началом работ Заказчик предоставляет исполнителю:

- Шаблон верстки, удовлетворяющий спецификации Twitter Bootstrap.
- Спецификацию к API сервиса. Все методы API функционируют и отложены на момент начала работ по ТЗ.
- Данное ТЗ

В результате выполнения работ Исполнитель предоставляет Заказчику программный код в виде, пригодном для дальнейшей поддержки и модернизации. Предоставленный программный код является реализацией интернет-сайта, соответствующего требованиям данного ТЗ. Программный код состоит из:

- HTML-страниц сайта
- JavaScript-файлов программной логики сайта
- CSS-файлов стилей. Допускается использование sass / less процессоров.

Шаблон верстки

Для выполнения работ исполнителю передается шаблон верстки - тема INSPINA (<https://wrapbootstrap.com/theme/inspinia-responsive-admin-theme-WB0R5L90S>). Для верстки страниц необходимо использовать элементы предоставленной темы. Допускается изменение HTML-кода страниц и контролов.

Описание API

API расположено по адресу <http://youdrive.today/> - корень путей всех методов.

Для взаимодействия используется протокол JSON.

Любой запрос может завершиться либо успешно, либо с ошибкой. В случае ошибки статус-код ответа будет лежать в диапазоне 400-599 (как правило 403 или 500). Тело ответа будет содержать JSON-объект вида:

```
{
  "success": false,
  "code": 101,
  "message": "User with provided credentials not found",
  "text": "Пользователь с указанным адресом почты или паролем не найден."
}
```

Где

success - признак успешности завершения метода

code - уникальный код ошибки

message - внутреннее описание ошибки (не для пользователя)

text - текст сообщения об ошибке, который может быть использован для отображения пользователю в случае, если на клиентской стороне не предусмотрено какого-то специального обработчика ошибки с указанным кодом.

В случае успешного выполнения метода в ответ вернется JSON-объект с обязательным флагом `success: true` и другими полями, специфичными для конкретного метода. Статус-код ответа в случае успеха равен 200.

API поддерживает кеширование Etag. В методах авторизации / регистрации в ответе на запрос может содержаться заголовок Cookie, устанавливающий руку сессии, которая требуется для выполнения запросов, требующих авторизации.

Подписка на рассылку

Подписка на новости сервиса.

Запрос:

Адрес: `/subscription`

Метод: POST

Тело:

```
{
  "email": "some.new-mail@example.com"
}
```

Ответ:

```
{
  "success": true
}
```

Возможные коды ошибок:

100 - общая ошибка

102 - неверный формат запроса: email отсутствует или не прошел вариацию

117 - Пользователь с таким адресом почты уже подписан

Авторизация

Для осуществления некоторых запросов к API требуется наличие токена сессии. Этот метод позволяет получить токен сессии (`session_id`) по адресу электронной почты и паролю пользователя.

Полученный токен является долгоживущим, однако в любой момент может быть деактивирован сервером. В случае, если токен деактивирован, сервер вернет ошибку со HTTP-статусом 403 и кодом ошибки 101. В этом случае клиентское приложение должно отобразить пользователю экран ввода логина и пароля для получения нового токена сессии.

Запрос:

Адрес: `/session`

Метод: POST

Тело:

```
{
  "email": "some.new-mail@example.com",
  "password": "1234"
}
```

Ответ:

```
{
  "session_id": "e388cb69-45f9-4c6c-845a-fb4cc1ec7ee6",
  "name": "Ivanov Ivan",
  "avatar": null,
  "success": true
}
```

Здесь `session_id` является токеном. Также в ответе будет присутствовать заголовок Set-Cookie со значением `session_id=e388cb69-45f9-4c6c-845a-fb4cc1ec7ee6`

Возможные коды ошибок:

100 - общая ошибка

113 - Не найден пользователь с указанными адресом почты и паролем

114 - Не указан адрес почты или пароль.

Получение профиля

Метод позволяет получить личную информацию пользователя. Для вызова метода требуется передать в куках действующий токен (`session_id=xxx`), установленный в момент авторизации.

Запрос:

Адрес: /user

Метод: GET

Ответ:

```
{
  "first_name": "Ivan",
  "last_name": "Ivanov",
  "email": "some.new-mail@example.com",
  "phone": 79012223344,
  "success": true
}
```

Возможные коды ошибок:

100 - общая ошибка

101 - Передан неверный токен сессии

Редактирование профиля

Метод позволяет обновить личную информацию пользователя. Для вызова метода требуется передать в куках действующий токен (`session_id=xxx`), установленный в момент авторизации.

Любой из параметров метода может быть опущен / не передаваться.

Запрос:

Адрес: /user

Метод: PUT

Тело:

```
{
  "first_name": "Ivan",
  "last_name": "Ivanov",
  "email": "some.new-mail@example.com",
  "phone": 79012223344,
  "new_password": "sdfsdf",
  "confirm_password": "sdfsdf"
}
```

Ответ:

```
{
  "success": true
}
```

Возможные коды ошибок:

100 - общая ошибка

101 - Передан неверный токен сессии

Получение списка привязанных карт

Список привязанных карт может быть получен только при наличии активного токена сессии, передаваемого в куках.

Запрос:

Адрес: /cards

Метод: GET

Ответ:

```
{
  "cards": [
    {"card_id": "419d1161-121f-4859-a97e-1aacabc5dfdc", "type": "visa", "number": "**** * 1234"},
    {"card_id": "318ec277-e368-47f1-a5b8-a40437195d", "type": "mastercard", "number": "**** * 4321"}
  ],
  "success": true
}
```

В случае, когда отсутствуют привязанные к аккаунту карты, в ответе вернется массив cards, не содержащий ни одного элемента.

Возможные коды ошибок:

100 - общая ошибка

101 - Передан неверный токен сессии

Привязка / удаление карты

Привязка карты осуществляется путем перехода на сторонний сервис и лежит за рамками данного ТЗ.

Удаление привязанной ранее карты осуществляется путем вызова следующего метода:

Запрос:

Адрес: /cards/<card_id>

Метод: DELETE

Ответ:

```
{
  "success": true
}
```

Здесь card_id является идентификатором карты пользователя. Идентификатор возвращается в ответе метода получения списка привязанных карт.

Возможные коды ошибок:

100 - общая ошибка

101 - Передан неверный токен сессии

История заказов

Запрос:

Адрес: /orders

Метод: GET

Ответ:

```
{
  "orders": [
    {"order_id": "64a44187-dbe6-46b3-9c11-e1d76701f84f", "created_at":
    "2015-05-03T15:38:45+00:00", "finished_at": "2015-05-03T15:48:45+00:00", "cost": 1231}
  ],
  "success": true
}
```

Даты возвращаются в формате ISO 8601 (http://en.wikipedia.org/wiki/ISO_8601) и пригодны для конструктора moments.js.

Стоимость возвращается в копейках и всегда целочисленная.

Массив отсортирован в правильном порядке (новые сверху) и должен быть выведен на клиентской части без изменения порядка.

Возможные коды ошибок:

100 - общая ошибка

101 - Передан неверный токен сессии

Восстановление пароля - активация

Запрос:

Адрес: /password_reset

Метод: POST

Тело:

```
{  
  "email": "some.new-mail@example.com"  
}
```

Ответ:

```
{  
  "success": true  
}
```

Возможные коды ошибок:

100 - общая ошибка

В целях обеспечения безопасности API не вернет ошибку в случае, если будет указан адрес почты, не зарегистрированный в БД

Общие требования

Стили

В случае необходимости создания собственных стилей необходимо включать соответствующие классы в отдельный файл. Не допускается изменение стилей темы. Не допускается использование inline-стилей.

Стили должны удовлетворять принципам Twitter Bootstrap.

Валидация форм

инпуты - подсветка, валидация данных (события валидации)

сообщения об ошибках

сообщения об успехе

активация кнопки отправки формы

Программная структура

Не допускается дублирование логики!

Все управляющие элементы должны быть объединены в логические модули, инкапсулирующие правила валидации и обработку исключительных ситуаций. Логика обработки введенных данных должна выноситься в родительский модуль.

Желательно: каждый набор логически связанных элементов должен быть выделен в отдельный модуль Angular JS.

Вложенность - должна быть предусмотрена возможность вложенности групп элементов в родительские модули форм таким образом, чтобы была обеспечена независимость логики дочерних элементов от родительского модуля, в которых они используются.

Группы элементов должны быть самостоятельными логическими единицами. Должна быть возможно использовать группы элементов на странице без дополнительных зависимостей.

Перезагрузка страниц

Действия отправки данных форм и вывода результатов операции отправки формы должны происходить без перезагрузки страницы.

Переход между страницами сайта должен выполняться с перезагрузкой страниц. Переходы между страницами внутри личного кабинета допускается делать без перезагрузки страниц.

SEO-оптимизация

Не допускается формирование статического контента страницы программным способом. Все текстовые блоки должны быть расположены непосредственно на странице. Блоки с данными личного кабинета должны, напротив, формироваться программно путем вызова соответствующих методов API.

Мобильная версия сайта

Предоставленная тема адаптирована для работы с устройствами, обладающими малым разрешением экрана, малым физическим размером экрана и обладающими touch-интерфейсом. Все страницы сайта должны корректно работать на подобных устройствах. Под корректной работой подразумевается:

1. Отсутствие явных ошибок верстки (наезжания / перекрытия / отступы / горизонтальные скрыл / анимация - допускается полное отключение анимации при необходимости)
2. Поддержка touch-интерфейса (отсутствие управляющих реакций на наведение)
3. Поддержка адаптации интерфейса под малый размер экрана, заключающаяся в перестроении структуры элементов в вертикальный вид (один под другим) вместо расположения по ширине.

Контент пустых страниц

Возможна ситуация, когда на странице нет данных для отображения: у пользователя нету ни одного заказа, не привязано ни одной карты и т.д. В этом случае на странице должен присутствовать текст, сообщающий о том, что данных для отображения нету. Шаблон сообщения можно взять тут: http://webapplayers.com/inspinia_admin-v2.0/empty_page.html

Структура интернет-сайта

Интернет-сайт состоит из трех основных частей:

1. Внешняя страница описания сервиса, доступная неавторизованными пользователям
2. Внутренние страницы личного кабинета пользователя, доступные после авторизации
3. Общие сервисные страницы процедуры регистрации, авторизации и сопряженных процедур

Внешняя страница описания сервиса (Лэндинг)

Страница доступна всем пользователям без предварительной авторизации. На странице присутствует только один интерактивный элемент - форма email-подписки на рассылку сервиса, расположенная в двух местах - в начале и в конце страницы.

На странице должна присутствовать плавающая шапка (top-bar) с навигационным меню, содержащем ссылку на страницу авторизации и якоря на части текущей страницы.

За основу страницы необходимо взять страницу предоставленной темы:
http://webapplayers.com/inspinia_admin-v2.0/Landing_page/index.html

Предоставленная страница должна быть изменена в соответствии с указаниями, содержащимися в файле landing.zip

Страница авторизации

Адрес страницы: /login.html

Шаблон страницы:

http://webappplayers.com/inspinia_admin-v2.0/login.html

Необходимо разработать контроллер логики авторизации, осуществляющий вызов соответствующего метода API. ОТ момента нажатия кнопки до получения ответа кнопка должна блокироваться (disabled). В случае получения ответа об ошибке авторизации необходимо подсветить поля красным и вывести сообщение о том, что адрес почты или пароль введены неверно. В случае успеха необходимо перевести пользователя на страницу /account.html

Ссылка Forgot password? Должна вести на страницу /restore.html

Ссылка Create an account Должна вести на страницу /register.html

Тексты:

Заменяемый текст	Замена
IN+	YouDrive
Welcome to IN+	Личный кабинет YouDrive
Login in. To see it in action.	–
Username	Email
Password	Пароль
Login	Вход
Forgot password?	Забыли пароль?
Do not have an account?	Нет аккаунта?
Create an account	Создать аккаунт
Inspinia we app framework base on Bootstrap 3 © 2014	–

Личный кабинет авторизованного пользователя

Личный кабинет строится на основе страницы:

http://webappplayers.com/inspinia_admin-v2.0/dashboard_4.html

На каждой странице должно присутствовать верхнее меню (Top bar), содержащее логотип (оставить пока как в теме), кнопку Log out (заменить текст на “Выход”) и следующие пункты меню:

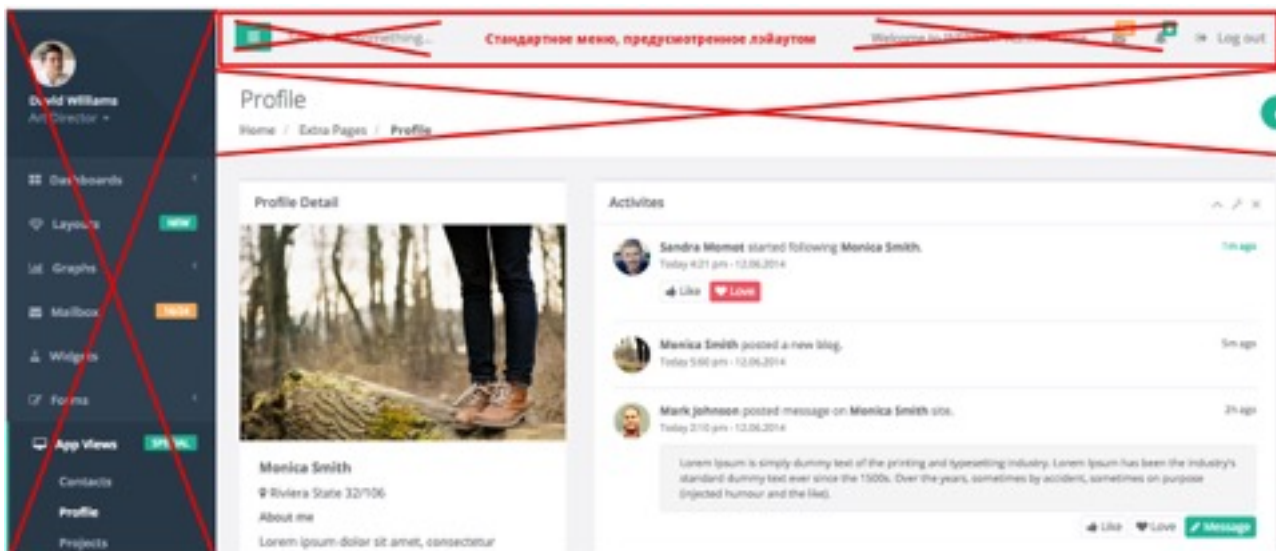
1. “Профиль” (открывается по умолчанию) - /profile.html
2. “Банковская карта” - /card.html
3. “Тарифы и правила” - /rules.html

Меню одноуровневое. При использовании шаблона необходимо помнить про мобильную версию - в ней должно все отображаться корректно.

Можно (но не обязательно) сделать переход между пунктами меню без перезагрузки страницы.

Профиль пользователя

Профиль пользователя предназначен для просмотра и редактирования личных данных, а также для просмотра списка сделанных заказов. Страница готовится на основании



шаблона:

http://webappplayers.com/inspinia_admin-v2.0/profile.html (левое меню и верхние бары отсутствуют).

На странице должно присутствовать два основных контрола:

1. Персональная информация
2. Список заказов

Персональная информация

Блок персональной информации должен иметь два режима работы - режим просмотра и режим редактирования.

В режиме просмотра в блоке выводится:

- Названия блока “Личные данные”
- Имя и фамилия пользователя (bold)
- Телефон (с font-awesome иконкой phone)
- Адрес почты (с font-awesome иконкой envelope)

Под личными данными в блоке выводится кнопка “Редактировать”, нажатие на которую переводит блок в режим редактирования.

В режиме редактирования отображается:

- Названия блока “Личные данные” и красную ссылку “Удалить аккаунт” (<mailto:support@youdrive.today>), прижатую к правому краю.
- Имя (input)
- Фамилия (input)
- Телефон (input с маской +7 (901) 111-22-33)
- Адрес почты (input)
- Новый пароль (type=password)
- Повтор пароля (type=password)

Под личными данными в блоке выводятся кнопки “Отмена” (grey) и “Сохранить” (navy).

В блоке необходимо предусмотреть место для вывода текстового сообщения об ошибке сохранения персональных данных. Инпут адреса почты должен поддерживать валидацию формата введенных данных.

При нажатии на кнопку “Отмена” происходит возврат блока в режим просмотра. При нажатии на кнопку “Сохранить” вызывается метод API редактирования профиля пользователя.

Список заказов

Список заказов строится на основе шаблона “Striped Table”: http://webappplayers.com/inspinia_admin-v2.0/table_basic.html

В таблице выводятся следующие столбцы:

- Номер заказа (строка 6 символов - строится по первым шести символам order_id)
- Дата начала аренды (дата)
- Длительность аренды (интервал времени с помощью moments - вычисляется как длительность между created_at и finished_at)
- Сумма (поле cost)

Необходимо предусмотреть мобильную версию страницы, в которой таблица с заказами будет расположена ниже блока личных данных.

Форма привязки / удаления банковской карты

Страница содержит список привязанных банковских карт. На странице расположена таблица со списком привязанных карт, содержащая следующие столбцы:

- Порядковый номер (№)
- Тип карты - логотип Visa или MasterCard (в аттаче)
- Номер карты (**** * 1234)
- Удалить (font-awesome иконка trash). Клик на кнопке вызывает метод API DELETE /cards

Под таблицей должна быть расположена кнопка “Привязать карту”, прижатая к правому краю. Клик на кнопку открывает внешнюю страницу (выходит за рамки ТЗ).

Тарифы и правила

Страница доступна неавторизованным пользователям и содержит статическую информацию о тарифах и правилах использования сервиса. Используется шаблон http://webappplayers.com/inspinia_admin-v2.0/article.html

Из шаблона надо удалить верхние теги, дату над заголовком, нижние теги, блок Status.

Цепочка страниц восстановления пароля

Процесс восстановления пароля состоит из следующих шагов:

1. Ввод адреса почты, для которого требуется восстановить пароль. Для этого используется страница, аналогичная странице входа, но содержащая только инпут ввода адреса почты и кнопку “Восстановить пароль”, при нажатии на которую

- происходит вызов метода API /reset_password и последующий переход на информационную страницу с сообщением о необходимости проверки почты.
2. В письме пользователю придет ссылка на метод API, который в результате своей работы отобразит страницу установки нового пароля. Для этих целей также используется шаблон, аналогичный странице авторизации, но содержащий два инпута: ввода пароля и повтора ввода пароля. На странице должна присутствовать валидация паролей на совпадение. Также на странице должна быть кнопка “Задать пароль”, которая вызывает метод API /setup_password с передачей введенных паролей и секретного ключа, который будет присутствовать в hidden-поле формы ввода нового пароля.
 3. После успешного выполнения запроса установки нового пароля необходимо показать пользователю стандартную информационную страницу с сообщением об успешном завершении операции смены пароля.

Цепочка страниц регистрации

Процесс регистрации начинается с получения пользователем письма со ссылкой на форму регистрации. Самостоятельная регистрация в сервисе невозможна т.к. регистрации предшествует подписание договора.

Ссылка из письма ведет на страницу ввода персональных данных, построенную на основе этого шаблона:

http://webappplayers.com/inspinia_admin-v2.0/register.html

На странице присутствуют следующие поля:

- Имя
- Фамилия
- Номер телефона
- Адрес электронной почты
- Пароль
- Подтверждение пароля
- Чекбокс “Я согласен с условиями Оферты”, являющийся ссылкой на внешнюю страницу
- Кнопка “Создать аккаунт”

После нажатия кнопки “Создать аккаунт” выполняется запрос на метод редактирования данных профиля. В теле запроса дополнительно передается параметр token из hidden-поля формы, который будет установлен сервером в момент рендеринга страницы.

В случае успешного ответа сервера, необходимо перенаправить пользователя на страницу привязки банковской карты личного кабинета. Токен сессии будет установлен сервером в ответе на запрос регистрации => авторизация не потребует.

Статические информационные страницы

Статические информационные страницы содержат текстовую информацию о выполнении некоторых действий (например, регистрация и смена пароля). Страницы должны быть подготовлены по следующему шаблону:

http://webappplayers.com/inspinia_admin-v2.0/empty_page.html (без шапки и бокового меню).

Текст можно поставить примерный, отражающий суть страницы. Кнопка должна вести на <http://youdrive.today/>

Страницы уведомления об ошибках

В случае, когда при вызове метода API возникает ошибка, для которой не существует обработчика внутри страницы-инициатора запроса, необходимо перевести пользователя на страницу информации об ошибке. Кнопка переводит на <http://youdrive.today/>
Страница должна быть выполнена на основе шаблона:

http://webapplayers.com/inspinia_admin-v2.0/500.html

Замена текста на странице:

Заменяемый текст	Замена
Internal Server Error	Произошла ошибка
The server encountered something unexpected that didn't allow it to complete the request. We apologize. You can go back to main page:	К сожалению, произошло что-то непредвиденное. Технические специалисты уже в курсе. Пожалуйста, попробуйте выполнить операцию позже.
Dashboard	На главную